mathematics

MDPI

Article

A Customized ADMM Approach for Large-Scale Nonconvex Semidefinite Programming

Chuangchuang Sun †

Mississippi State University, Starkville, MS 39762, USA; csun@ae.msstate.edu
† Current address: 75 B. S. Hood Rd, Mississippi State, MS 39762, USA.

Abstract: We investigate a class of challenging general semidefinite programming problems with extra nonconvex constraints such as matrix rank constraints. This problem has extensive applications, including combinatorial graph problems, such as MAX-CUT and community detection, reformulated as quadratic objectives over nonconvex constraints. A customized approach based on the alternating direction method of multipliers (ADMM) is proposed to solve the general large-scale nonconvex semidefinite programming efficiently. We propose two reformulations: one using vector variables and constraints, and the other further reformulating the Burer–Monteiro form. Both formulations admit simple subproblems and can lead to significant improvement in scalability. Despite the nonconvex constraint, we prove that the ADMM iterates converge to a stationary point in both formulations, under mild assumptions. Additionally, recent work suggests that in this matrix form, when the matrix factors are wide enough, the local optimum with high probability is also the global optimum. To demonstrate the scalability of our algorithm, we include results for MAX-CUT, community detection, and image segmentation.

Keywords: semidefinite optimization; symmetric matrix factorization; nonconvex optimization; large-scale graph problems

MSC: 90C22; 90C26

check for updates

Citation: Sun, C. A Customized ADMM Approach for Large-Scale Nonconvex Semidefinite Programming. *Mathematics* **2023**, *11*, 4413. https://doi.org/10.3390/math11214413

Academic Editor: Luca Gemignani

Received: 23 September 2023
Revised: 16 October 2023
Accepted: 16 October 2023
Published: 24 October 2023

Copyright: © 2023 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

We consider rank-constrained semidefinite optimization problems (SDPs) of the type:

$$\begin{aligned} \min_{Z,X} \quad & f(Z), \\ s.t. \quad & \mathcal{A}(Z) = b, Z = XX^T, X \in \mathcal{C}, \end{aligned} \tag{1}$$

where the matrix variable $Z \in \mathbb{S}^n_+$ is an $n \times n$ symmetric semidefinite matrix, and $X \in \mathbb{R}^{n\times r}$ a low-rank symmetric factor. The linear constraints $\mathcal{A}(Z) = b$ constrain either the diagonal or trace of Z, and the set $\mathcal{C}$ controls desirable features of the factor—e.g., nonnegativity, integer, norm-1, etc. ($\mathcal{C}$ may be nonconvex.) The objective function $f(x)$ is convex, differentiable everywhere, with L_f-Lipschitz gradient, but the overall problem (1) is nonconvex.

This problem is equivalent to many important nonconvex SDPs, such as the MAX-CUT problem and its related applications [1–3], rank-constrained nonnegative matrix factorization problem [4,5], and constrained eigenvalue problems [6–8]. It is known that exactly solving (1) globally is in general a very difficult problem, as it includes many NP-hard problems. Methods for heuristically solving (1) fall in three categories: (i) solving the convexified SDP, where (1) does not have the rank-r or $X \in \mathcal{C}$ constraint, using any convex optimization method [9–11], (ii) approximately solving (1) using an alternating minimization method [12,13] and relying on statistical arguments suggesting that the acquired local optimal = the global optimal [13], or (iii) using other application-specific approaches [2,14].

The methods investigated in this paper fall in the second category. Specifically, we investigate solving (1) using ADMM and linearized ADMM on two reformulations. We find that these flexible reformulations allow easy incorporation of low-rank and sparse structures, making the resulting algorithm extremely scalable in both memory and computation, which we demonstrate on a number of popular applications.

However, often nonconvex formulations of SDPs are not favored because the convergence behavior of standard algorithms is not well understood. Specifically, an iterative procedure can do one of four things: diverge, oscillate within a bounded interval, converge to an arbitrary point, or converge to a useful point. We show that linearized ADMM on a nonsymmetric reformulation of (1) can either converge to a stationary point or diverge to $\pm\infty$; it cannot oscillate or converge to a non-stationary point. Additionally, for the case without linear constraints, vanilla ADMM is guaranteed to converge to a stationary point with a monotonically decreasing augmented Lagrangian term, and at a linear rate if the objective is strongly convex.

2. Applications

It is well known that many convex optimization problems can be reformulated as SDPs (e.g., [15]). In nonconvex optimization, SDPs are studied in several key areas, as tight convex relaxations of otherwise NP-hard problems.

2.1. Combinatorial Problems

A simple reparameterization of the constraint $x \in \mathbb{R}^n$, $x_i \in \{-1, 1\}$ is as $X = xx^T$, $\mathbf{diag}(X) = 1$. This property has been heavily exploited for finding lower bounds in combinatorial optimization [9,16,17] and generalized further to polynomial optimization [18,19]. Of high interest is the MAX-CUT problem:

$$\min_{x \in \mathbb{R}^n} \; x^T C x, s.t.\ x_i \in \{-1, 1\}, \quad i = 1, \dots, n, \tag{2}$$

where $C = (A - \mathbf{diag}(A\mathbf{1}))/4$ and $A \in \mathbb{S}^n$ is the symmetric adjacency matrix of an undirected graph. Written in this way, the solution to (1) is exactly the maximum cut of an undirected graph with nonnegative weights A_{ij}.

This seemingly simple framework appears in many other applications, such as community detection [20] and image segmentation [21], and is equivalent to the nonconvex SDP:

$$\min_{Z} \; \mathbf{Tr}(CZ),\ s.t.\ Z_{kk} = 1, Z \succeq 0, \mathbf{rank}(Z) = 1. \tag{3}$$

Lifting $x \in \mathbb{R}^n$ to a skinny matrix $X \in \mathbb{R}^{n \times k}$ generalizes this technique to partitioning [22] and graph coloring problems [23].

2.1.1. Related Works on MAX-CUT

More generally, combinatorial methods can be solved using branch-and-bound schemes, using a linear relaxation of (1) as a bound [24,25], where the binary constraint $x \in \{-1, 1\}$ is relaxed to $0 \leq (x+1)/2 \leq 1$. Historically, these "polyhedral methods" were the main approach to finding exact solutions to the MAX-CUT problem. Though this is an NP(nondeterministic polynomial-time)-hard problem, if the graph is sparse enough, branch-and-bound converges quickly even for very large graphs [25]. However, when the graph is not very sparse, the linear relaxation is loose, and finding efficient branching mechanisms is challenging, causing the algorithm to run slowly. The MAX-CUT problem can also be approximated by one pass of the linear relaxation (with bound $\frac{f_{\text{relax}}}{f_{\text{exact}}} \geq 2 \times$ #edges) [26].

A tighter approximation can be found with the semidefinite relaxation, which is also used for better bounding in branch-and-bound techniques [27–31]. In particular, the rounding algorithm of [9] returns a feasible $\hat{x}$ given optimal Z, and is shown in expectation to satisfy $\frac{x^T C x}{\hat{x}^T C \hat{x}} \geq 0.878$. For this reason, the semidefinite relaxation for problems of type (1) is heavily studied (e.g., [11,32,33]).

2.1.2. Specialization to Community Detection

A small modification of the matrix C generalizes problems of form (2) and (3) to community detection in machine learning. Here, the problem is to identify node clusters in undirected graphs that are more likely to be connected with each other than with nodes outside the cluster. This prediction is useful in many graphical settings, such as interpreting online communities through social networks or linking behavior [34], interpreting biological ecosystems [35], finding disease sources in epidemiology [36], and many more. There are many varieties and methodologies in this field, and it would be impossible to list them all, though many comprehensive overviews exist (e.g., [2]).

The stochastic binary model [37] is one of the simplest generative models for this application. Given a graph with n nodes and parameters $0 < q < p < 1$, the model partitions the nodes into two communities and generates an edge between nodes in a community with probability p and nodes in two different communities with probability q. Following the analysis in [20], we can define $C = \frac{p+q}{2}\mathbf{1}\mathbf{1}^T - A$, where A is the graph adjacency matrix, and the solution to (1) gives a solution to the community detection problem with sharp recovery guarantees.

2.2. Nonnegative Factorization

For a symmetric matrix C, the maximum eigenvalue/eigenvector pair of C is the solution to the nonconvex optimization problem:

$$\max_{x\in\mathbb{R}^n} \; x^T C x, \; s.t. \; \|x\|_2 = 1. \tag{4}$$

By inverting the sign of C, we can transform this into a minimization problem or equivalently acquire the minimum eigenvalue/eigenvector pair. Interestingly, despite the nonconvex nature of (4), we have many efficient globally optimal methods for finding x, e.g., Lanczos, Arnoldi, etc. However, adding any additional constraints, such as nonnegativity of x [38], and simple methods generally do not work without heavy data assumptions [39]. This is of interest in problems such as phase retrieval, recommender systems with positive-only observations, clustering and topic models, etc. Here, we discuss three variations of the nonnegative factorization problem appearing in the literature, all of which are special instances of (1).

2.2.1. Optimization over Spectrahedron

We can frame (4) as a linear objective over the *spectrahedron*:

$$\min_{Z\in\mathbb{S}^n} \; \mathbf{Tr}(CZ), \; s.t. \; \mathbf{Tr}(Z) = 1, Z \succeq 0. \tag{5}$$

If additionally the maximum eigenvalue of C is isolated (corresponding only to one leading eigenvector), then $Z = xx^T$ and $Cx = \lambda_{\max}(C)x$. To see this, by definition,

$$\begin{aligned} \lambda_{\max}(C) \quad &= \max_{x:\|x\|_2=1} \; x^T C x \\ &= \max_{Z:Z=xx^T,\|x\|_2=1} \; \mathbf{Tr}(CZ) \\ &= \max_{Z:\mathbf{Tr}(Z)=1,X\succeq 0} \; \mathbf{Tr}(CZ). \end{aligned} \tag{6}$$

As a consequence, note that though (5) is convex, the solution Z^* will always have rank 1 when $\lambda_{\max}(C)$ has multiplicity 1. A simple extension of (5) often used in nonnegative PCA [40] is:

$$\begin{aligned} &\min_{Z\in\mathbb{S}^n, x\in\mathbb{R}^n} \; \mathbf{Tr}(CZ), \\ &s.t. \; \mathbf{Tr}(Z) = 1, Z \succeq 0, Z = xx^T, \quad x \geq 0, \end{aligned} \tag{7}$$

which is an instance of (1) with $\mathcal{C}$ the nonnegative orthant.

2.2.2. Factorization with Partial Observations

An equivalent way of formulating the top-k nonnegative-eigenvector problem is as the nonnegative minimizer X to $\|XX^T - C\|_2$ where X is $\mathbb{R}^{n\times k}$. However, in many applications, we may not have full view of the matrix C, (e.g., C is a rating matrix). Suppose that an index set Ω defines the observed entries, e.g., $\{i,j\} \in \Omega$ implies that C_{ij} is known. Then, the nonnegative factorization problem can be written as:

$$\begin{aligned} \min_{Z\in\mathbb{S}^n, x\in\mathbb{R}^n} \quad & \sum_{i,j\in\Omega} (Z_{ij} - C_{ij})^2, \\ & s.t.\ Z = xx^T, \quad x \geq 0. \end{aligned} \tag{8}$$

This formulation exists in [41].

2.2.3. Projective Nonnegative Matrix Factorization

A third method toward this goal is to optimize over the low-rank projection matrix itself [42], a variant of nonnegative matrix factorization, solving:

$$\begin{aligned} \min_{Z\in\mathbb{S}^n, X\in\mathbb{R}^{n\times k}} \quad & \|B - ZB\|_2, \\ & s.t.\ Z = XX^T,\ X \geq 0, \end{aligned} \tag{9}$$

Here, the data matrix may not even be symmetric, but $\frac{1}{\mathbf{Tr}(Z)} ZB$ will approximate the projection of B to its top-k singular vectors.

3. Related Work

3.1. Convex Relaxations

If $r = n$ and $\mathcal{C} = \mathbb{S}^n$, then (1) is a convex problem, and can be solved using many conventional methods with strong convergence guarantees. However, even in this case, if n is large, traditional semidefinite solvers are computationally limiting. In the most general case, an interior point method solves at each iteration a KKT system of at least order n^6, and most first-order methods for general SDPs require eigenvalue decompositions, which are of order $O(n^3)$ per iteration.

3.2. Low-Rank Convex Cases

In fact, assuming low-rank solutions often allows for the construction of faster SDP methods. In [43], it is noted that the rank of the primal PSD matrix variable is equal to the multiplicity of the matrix variable arising from the gauge dual formulation, and finding only those r corresponding eigenvectors can recover the primal solution. In [10], a similar observation is made of the Lagrange dual variable and thus the dual problem can be solved via a modified bundle method. More generally, the recently popularized conditional gradient algorithm (also called the Frank–Wolfe algorithm) efficiently solves norm-constrained problems for nonsymmetric matrices [44], exploiting the fact that the dual norm minimizer can be computed efficiently; see also [45–47].

3.3. Nonconvex Cases

In close connection with these observations, [12,48] proposed simply reformulating semidefinite matrix variables $Z = XX^T$, solving the "standard" nonconvex SDP:

$$\min_{X\in\mathbb{R}^{n\times r}} \ \langle C, XX^T\rangle,\ s.t.\ \mathcal{A}(XX^T) = b, \tag{10}$$

by sequentially optimizing the Lagrangian. However, solving (1) is still numerically burdensome; in the augmented Lagrangian term, the objective is quartic in R, and is usually solved using an iterative numerical method, such as L-BFGS.

3.4. Global Optimality of a Nonconvex Problem with Linear Objective

The main motivation behind solving rank-constrained problems using convex optimization methods comes from key results in [49,50] which show that for a linear SDP, when X^* is the optimum and $r = \mathbf{rank}(X^*)$, then $\frac{r(r+1)}{2} \geq m$ where m is the number of linear constraints. Furthermore, a recent work [13] shows that almost all local optima of FSDP are also global optima, suggesting that any stationary point of the FSDP is also a reasonable approximation of (1), if the constraint space of (10) is compact and sufficiently smooth, e.g., $A_i Y$ linearly independent whenever $\langle A_i, YY^T \rangle = b_i$ for all $i = 1, \ldots, m$. The MAX-CUT problem satisfies this constraint; an example of a linear SDP without this condition is the phase retrieval problem [51], when $m > n$.

3.5. Nonconvex Constraint $\mathcal{C}$

Although there are many cases where the linear constraint in (1) serves a distinct purpose, largely it is introduced to tighten the convex relaxation. When working in the nonconvex formulation, for many applications, the linear constraint becomes superfluous, and a more useful reformulation may be:

$$\min_{x,y} \; g(x), \; s.t. \; x = y, y \in \mathcal{C},$$

for some nonconvex set $\mathcal{C}$ (e.g., $\mathcal{C} = \{-1, 1\}^n$). Note that the projection on $\mathcal{C}$ is extremely easy, despite its nonconvexity. Although less explored, this idea is not new; see [52] (chapter 9).

3.6. ADMM for Nonconvex Problems

The alternating direction method of multipliers (ADMM) [53,54] is now a popular method [52] for convex large-scale distributed optimization problems with understood convergence rates [55] and variations [56–58]. It is closely related to dual decomposition methods, but alternates its subproblems, and makes use of augmented Lagrangians, which smooths the subproblems and reduces the influence of the dual ascent step size. Although there are extensions to many variable blocks, most ADMM implementations use two variable block decompositions, solving:

$$\min_{x} \; g(x) + h(y), \; s.t. \; Ax = By,$$

by alternatingly minimizing over each variable in the augmented Lagrangian:

$$\mathcal{L}_\rho(x, y; u) = g(x) + h(y) + u^T(Ax - By) + \frac{\rho}{2}\|Ax - By\|_2^2,$$

and then incrementally updating the dual variable:

$$\begin{aligned} x^+ &= \arg\min_x \mathcal{L}_\rho(x, y; u), \\ y^+ &= \arg\min_x \mathcal{L}_\rho(x^+, y; u), \\ u^+ &= u + \rho(Ax^+ - By^+). \end{aligned}$$

Here, any $\rho > 0$ will achieve convergence.

In general, there is a lack of theoretical justification for ADMM on nonconvex problems despite its good numerical performance. Almost all works concerning ADMM on nonconvex problems investigate when nonconvexity is in the objective functions ([59–63], and also [64,65] for matrix factorization). Under a variety of assumptions (e.g., convergence or boundedness of dual objectives) they are shown to converge to a KKT stationary point.

In comparison, relatively fewer works deal with nonconvex constraints. Ref. [66] tackles polynomial optimization problems by minimizing a general objective over a spherical constraint $\|x\|_2 = 1$, Ref. [67] solves general QCQPs, and Ref. [68] solves the low-rank-plus-sparse matrix separation problem. In all cases, they show that all limit points are also KKT stationary points, but do not show that their algorithms will actually converge to the limit points. In this work, we investigate a class of nonconvex constrained problems and show with much milder assumptions that the sequence always converges to a KKT stationary point.

We now present our main results, the algorithms, and convergence analysis for different formulations.

4. Linearized ADMM on Full SDP

We first investigate a reformulation of (1) as:

$$\begin{aligned} \min_{Z,X,Y} \quad & f(Z) + \delta_{\{0\}}(\mathcal{A}(Z) - b) + \delta_{\mathcal{C}}(Y), \\ s.t. \quad & Z = (XY^T)_\Omega, X = Y, \end{aligned} \tag{11}$$

with variables $Z \in \mathbb{S}^{n\times n}$, $X \in \mathbb{R}^{n\times r}$, and $Y \in \mathbb{R}^{n\times r}$. The affine and $\mathcal{C}$ constraints are lifted to the objective via an indicator function:

$$\delta_{\mathcal{C}}(x) = \begin{cases} 0 & \text{if } x \in \mathcal{C}, \\ \infty & \text{else.} \end{cases}$$

The notation A_Ω for a symmetric matrix A is the projection of A on the sparsity pattern Ω:

$$(A_\Omega)_{ij} = \begin{cases} A_{ij}, & \text{if } \{i,j\} \in \Omega \\ 0, & \text{else,} \end{cases}$$

and we write $A \in \mathbb{S}^n_\Omega$ if $A_\Omega = A$. Specifically, Ω captures the *effective sparsity* of the problem; that is, $f(Z) = f(Z_\Omega)$ and $\mathcal{A}(Z) = \mathcal{A}(Z_\Omega)$. We assume $\{i,i\} \in \Omega$ for all i, so the second is trivially true.

4.1. Duality

As shown in [69], a notion of a dual problem can be established via the augmented Lagrangian of (11):

$$\begin{aligned} &\mathcal{L}_\rho(Z,X,Y;S,U) = \\ &\qquad f(Z) + \delta_{\mathcal{C}}(Y) + \langle U, X-Y\rangle + \langle S, Z - XY^T\rangle \\ &\qquad + \frac{\rho}{2}\|X-Y\|_F^2 + \frac{\rho}{2}\|Z - XY^T\|_F^2, \end{aligned} \tag{12}$$

where the dual problem is:

$$\max_{S,U} \min_{Z,X,Y} \mathcal{L}_\rho(Z,X,Y;S,U).$$

The minimization of $\mathcal{L}_\rho$ over Z and X is the solution to:

$$\begin{aligned} \nabla f(Z) - \mathcal{A}^*(\nu) + S + \rho(Z - XY^T) &= 0 \\ U - SY + \rho(XY^TY - ZY) + \rho(X - Y) &= 0 \\ \mathcal{A}(Z) &= b, \end{aligned} \tag{13}$$

where $\nu > 0$ is a Lagrange dual variable for the local constraint $\mathcal{A}(Z) = b$. The minimization of $\mathcal{L}_\rho$ over Y is the solution to the generalized projection problem:

$$\min_{Y\in\mathcal{C}} \quad \langle Y - \hat{Y}, Y - \hat{Y}\rangle_H = \mathbf{Tr}((Y-\hat{Y})H(Y-\hat{Y})^T), \tag{14}$$

where: $\hat{Y} = U + SX + \rho(X + Z^TX), H = \rho(I + X^TX)$. For general nonconvex problems, it is difficult to guarantee global minimality. Here, we introduce two sought-after properties that are more reasonably attainable.

Definition 1 ([70])**.** *The* tangent cone *of a nonconvex set $\mathcal{C}$ at x is given by $\mathcal{T}_{\mathcal{C}}(x) = \{d :$ for all $t \to 0, \hat{x} \to x, \hat{x} \in \mathcal{C}$, there exists $\hat{d} \to d, \hat{x} + t\hat{d} \in \mathcal{C}\}$. The* normal cone *of $\mathcal{C}$ at x $(:= \mathcal{N}_{\mathcal{C}}(x))$ is the polar of the tangent cone.*

Definition 2. *For a minimization of a smooth constrained function $\min\limits_{x \in \mathcal{C}} f(x)$ we say that x^* is a* KKT-stationary point *if $-\nabla f(x^*) \in \mathcal{N}_{\mathcal{C}}(x^*)$.*

Definition 3. *For a function defined over M variables $\mathcal{L}(X_1, \ldots, X_m)$, we say that $X_1^*, \ldots, X_m^*$ are* (block) coordinatewise minimum points *if for each $k = 1, \ldots, m$,*
$X_k^* = \underset{X}{\operatorname{argmin}}\ \mathcal{L}(X_1^*, \ldots, X_{k-1}^*, X, X_{k+1}^*, \ldots, X_m^*)$.

Note that it is not always the case that stationarity is stronger than coordinatewise minimum. A simple example is $\mathcal{C} = \{-1, 1\}^n$. Then, for all points $x \in \mathcal{C}$, the tangent cone is $\{0\}$ and the normal cone is $\mathbb{R}^n$. Then, every point in $\mathcal{C}$ is stationary, no matter the objective function.

Proposition 1. *If Algorithm 1 converges to coordinatewise minimum points $((X, Z)^*, Y^*, S^*, U^*)$, then the primal points (i) satisfy (13) for some choice of $\nu \geq 0$, (ii) minimize (14), (iii) and are primal-feasible, e.g., $X^* = Y^*$ and $(X^*(Y^T)^*)_\Omega = Z^*$. Furthermore, $(X^*, Y^*, Z^*, S^*, U^*)$ are stationary points of (12).*

Proof. It is clear that the convergent points of Algorithm 1 exactly satisfy the three conditions. To show that these points are stationary, note that the augmented Lagrangian is convex with respect to X, Z jointly, and is a projection on a compact set $\mathcal{C}$ with respect to Y. Therefore:

$$\begin{aligned} \nabla_{X,Z,S,U} \mathcal{L}_\rho(Z^*, X^*, Y^*; S^*, U^*) = 0, \\ -\nabla_Y \bar{\mathcal{L}}_\rho(Z^*, X^*, Y^*; S^*, U^*) \in \mathcal{N}_{\mathcal{C}}(Y^*), \end{aligned} \tag{15}$$

where $\bar{\mathcal{L}}_\rho(Z, X, Y; S, U) = -\langle U, Y\rangle - \langle S, XY^T\rangle + \frac{\rho}{2}\|X - Y\|_F^2 + \frac{\rho}{2}\|Z - XY^T\|_F^2$, with all the differentiable terms of $\mathcal{L}_\rho$ involving Y. □

4.2. Linearized ADMM

We propose to solve (11) via the *linearized* ADMM, e.g., where at each iteration, the objective is replaced by its current linearization:

$$f(Z) \approx \hat{f}^k(Z) := f(Z^{k-1}) + \langle \nabla f(Z^{k-1}), Z - Z^{k-1}\rangle.$$

We then build the linearized augmented Lagrangian function as:

$$\begin{aligned} \hat{\mathcal{L}}^k(Z, X, Y; S, U) = g_k(X, Z) + h(Y) + \langle U, X - Y\rangle \\ + \langle S, Z - XY^T\rangle + \frac{\rho}{2}\|X - Y\|_F^2 + \frac{\rho}{2}\|Z - XY^T\|_F^2 \end{aligned} \tag{16}$$

where $g_k(X, Z) = \hat{f}^k(Z) + \delta_{\{0\}}(\mathcal{A}(Z) - b), \quad h(Y) = \delta_{\mathcal{C}}(Y)$ and $S \in \mathbb{R}^{n \times n}$ and $U \in \mathbb{R}^{n \times r}$ are the dual variables corresponding to the two coupling constraints. The full algorithm is given in Algorithm 1.

Algorithm 1 ADMM for solving (11)

```
Inputs: $\rho_0 > 0$, $\alpha > 1$, tol $\epsilon > 0$
Initialize: $Z^0, X^0; S^0, U^0$ as random matrices
Outputs: $Z$, $X = Y$
for $k = 1 \ldots$ do
    Update $Y^{k+1}$ the solution of:
        $\min_{Y \in \mathbb{R}^{n \times k}} \|Z^k - X^k Y^T + \frac{S^k}{\rho^k}\|_F^2 + \|X^k - Y + \frac{U^k}{\rho^k}\|_F^2, \quad s.t. \quad Y \in \mathcal{C}$    (17)
    Update $(Z, X)^{k+1}$ as the solutions of:
        $\min_{X,Z \in \mathbb{S}^n_\Omega} \mathcal{L}_{k+1}(Z, X, Y^{k+1}; S^k, U^k; \rho^k), \quad s.t. \quad \mathcal{A}(Z) = b$    (18)
    where $\mathcal{L}$ is the linearized augmented Lagrangian as defined in (16).
    Update $S, U$ and $\rho$ via:
        $S^{k+1} = S^k + \rho^k (Z^{k+1} - X^{k+1}(Y^{k+1})^T)_\Omega$
        $U^{k+1} = U^k + \rho^k (X^{k+1} - Y^{k+1})$    (19)
        $\rho^{k+1} = \alpha \rho^k$
    if $\max\{\|X^k - Y^k\|, \|(Z^k - X^k(Y^k)^T)_\Omega\|\} \leq \epsilon$ then
        break
    end if
end for
```

4.2.1. Minimizing over Y

The generalized projection (14) can be solved a number of ways. Note that if $r = 1$, then H is a positive scalar, and the problem reduces to $Y^+ = \mathbf{proj}_{Y \in \mathcal{C}}\left(\frac{1}{H}\hat{Y}\right)$. When $\mathcal{C} = \{-1, 1\}^n$, this process reduces to recovering the signs of $\hat{Y}$ i.e., $Y_i = \mathbf{sign}_{\mathcal{C}}(\hat{Y}_i)$, and when $\mathcal{C} = \{u : \|u\|_2 = 1\}$ the set of unit-norm vectors, Y is just a properly scaled version of $\hat{Y}$: $Y = \frac{1}{\|\hat{Y}\|_2}\hat{Y}$. However, in general, it is difficult to compute the generalized projection over a nonconvex set. When $\mathcal{C}$ is convex, the generalized projection problem (14) can be computed using projected gradient descent. Note that the objective of (14) is 1-strongly convex; thus, we expect fast convergence in this subproblem. In practice, we find that if r is not too large, often a few tens of iterations is enough.

4.2.2. Minimizing over X and Z

Using standard linear algebra techniques, the linear system (13) can be reduced to a few simple instructions. First, we solve for the Lagrange dual variable ν associated with the linear constraints (and localized to the minimization of X and Z):

$$\begin{aligned}&\mathcal{A}(\mathcal{A}^*(\nu)(YY^T + I)) = \\ &\rho(b - \mathcal{A}(DY^T + YY^T)) + \mathcal{A}((G + S)(I + YY^T)),\end{aligned} \tag{20}$$

where $D = \frac{1}{\rho}(SY - U) + Y$ and $G = \nabla f(Z^{k-1})$ the local gradient estimate. When $\mathcal{A} = \mathbf{diag}$, (20) reduces to n scalar element-wise computations $\nu_i = \frac{\rho(b - (DY^T)_{ii}) + ((G+S)(I+YY^T))_{ii}}{(YY^T)_{ii}+1}$. When $\mathcal{A} = \mathbf{Tr}$, $\nu = \frac{\rho(b - \mathbf{Tr}(DY^T) + \mathbf{Tr}((G+S)(I+YY^T))}{\mathbf{Tr}(YY^T)+1}$. Note that in both cases, no $n \times n$ matrix need ever be formed, so the memory requirement remains $O(nr)$. (See Appendix A for elaboration). Then, the primal variables are recovered via $X = BY + D$, and $Z = (XY^T)_\Omega + B$, with $B = -\frac{1}{\rho}(C - \mathcal{A}^*(\nu) + S)$. In these cases, the complexity is dominated by multiplications between $n \times n$ and $n \times r$ matrices. Thus, the method is especially efficient when $r \ll n$.

4.3. Convergence Analysis

Theorem 1. *Assume that $f(Z)$ is L_f-smooth. Assume the dual variables are bounded, e.g., $\max\{\|S^k\|_F, \|U^k\|_F, \|Y^k\|_F\}_k \leq B_P < +\infty$, and $\frac{L_f}{\sigma_{\max}}$ is bounded above, where $\sigma_{\max} = 1 - \frac{\sqrt{\sigma_Y^4 + 4\sigma_Y^2} - \sigma_Y^2}{2}$, $\sigma_Y = \|Y^{k+1}\|_2$. Then, by running Algorithm 1 with $\rho^k = \alpha\rho^{k-1} = \alpha^k\rho_0$, if $\mathcal{L}_k$ is bounded below, then the sequence $\{P^k, D^k\}$ converges to a stationary point of (12).*

Proof. See Appendix B. □

Corollary 1. *If $r \geq \left\lceil \sqrt{2n} \right\rceil$ and the stationary point of Algorithm 1 converges to a second-order critical point of (1), then it is globally optimal for the convex relaxation of (10) [13].*

Unfortunately, the extension of KKT stationary points to global minima is not yet known when $\frac{r(r+1)}{2} < n$ (i.e., $r = 1$). However, our empirical results suggest that even when $r = 1$, often a local solution to (10) well-approximates the global solution to (1).

5. ADMM on Simplified Nonconvex SDP

When the linear constraints are not present, (1) can be reformulated without Z, into:

$$\min_{X,Y} g(X) + \delta_{\mathcal{C}}(Y), s.t. X = Y, \tag{21}$$

with matrix variables $X \in \mathbb{R}^{n\times r}, Y \in \mathbb{R}^{n\times r}$, and where $g(X) = f(XX^T)$ is smooth. We can also define an augmented Lagrangian of (21) as $\mathcal{L}_\rho(X, Y; U) = g(X) + \delta_{\mathcal{C}}(Y) + \langle U, X - Y\rangle + \frac{\rho}{2}\|X - Y\|_F^2$.

Theorem 2. *The coordinatewise minimum points $X^* = Y^*$ satisfying:*

$$\begin{aligned} 0 &= \nabla g(X^*) + U + \rho(X - Y) \\ Y &= \mathbf{proj}_{\mathcal{C}}(X + \frac{1}{\rho}U) \\ X &= Y, \end{aligned} \tag{22}$$

are the stationary points of the problem:

$$\min_{X} g(X), s.t. X \in \mathcal{C}. \tag{23}$$

Proof. The KKT stationary points of (23) can be characterized in terms of the normal cone of $\mathcal{C}$ at X^*; specifically, X^* is stationary if:

$$\langle \nabla g(X^*), X - X^*\rangle \leq 0, \quad \forall X \in \mathcal{C} \cap \mathcal{N}_\epsilon(X^*),$$

where $\mathcal{N}_\epsilon(X^*)$ is some small neighborhood containing X^*. (This is an equivalent definition of the Clarke stationary point [70], since in a close enough neighborhood to X^*, the subdifferential of $\delta_{\mathcal{C}}(x)$ is $\mathcal{N}_{\mathcal{C}}(x)$).

Combining terms in (22) gives $X^* = Y^*$ satisfying $X^* = \mathbf{proj}_{\mathcal{C}}\left(X^* - \frac{1}{\rho}\nabla g(X^*)\right)$. The optimality condition of the projection is $\langle X - (X - \frac{1}{\rho}\nabla g(X^*)), X - X^*\rangle \leq 0, \quad \forall X \in \mathcal{C} \cap \mathcal{N}_\epsilon(X^*)$ which reduces to the desired condition. □

5.1. ADMM

The alternating steps in minimizing the augmented Lagrangian over the primal variables are extremely simple, compared with the previous matrix formulation. In general, we are considering $f(X)$ linear (in which case the update of X involves only addition) or quadratic with strictly positive diagonal Hessian (which adds a small scaling step). $\mathcal{C} = \{-1, 1\}^n, \mathcal{C} = \{x : \|x\|_2 = 1\}$, even when $r > 1$.

5.2. Convergence Analysis

Definition 4. *A differentiable convex function $g(X)$ is L_g-smooth and H_g-strongly convex over $\mathbb{R}^n$ if for any X, Y, $g(X) - g(Y) \geq \langle \nabla f(X), X - Y \rangle - \frac{L_g}{2} \|X - Y\|_F^2$ and $g(X) - g(Y) \leq \langle \nabla f(X), X - Y \rangle - \frac{H_g}{2} \|X - Y\|_F^2$.*

Theorem 3. *Assume $g(X)$ is lower bounded over $\mathcal{C}$, and is L_g-smooth. Given a sequence $\{\rho^k\}$ such that:*

$$\frac{\rho^k - 3L_g}{2} - L_g^2 \frac{\rho^{k+1} + \rho^k}{2(\rho^k)^2} > 0, \qquad \rho^k > L_g$$

for all k, then under Algorithm 2 the augmented Lagrangian $\mathcal{L}(X^k, Y^k; U^k)$ is lower bounded and convergent, with $\{X^k, Y^k, U^k\} \to \{X^*, Y^*, U^*\}$ a stationary and feasible solution of (23).

Proof. See Appendix C. □

Remark 1. *Convergence is guaranteed under a constant penalty coefficient $\rho_k \equiv \rho^0 \geq \frac{3+\sqrt{17}}{2} L_g, \alpha = 1$. However, in implementation, we find empirically that increasing $\{\rho^k\}$ from a relatively small ρ^0 can encourage convergence to more useful global minima.*

Algorithm 2 ADMM for solving (23)

```
Inputs: $\rho_0 > 0, \alpha > 1$, tol $\epsilon > 0$
Initialize: $Z^0, X^0; S^0, U^0$ as random matrices
Outputs: $Z, X = Y$
for $k = 1 \ldots$ do
    Update $Y^{k+1}$ the solution of:
        $\min_{Y \in R^{n \times k}} \|X^k - Y + \frac{U^k}{\rho^k}\|_F^2, s.t. Y \in \mathcal{C}.$    (24)
    Update $X^{k+1}$ as the solution of:
        $0 = \nabla g(X) + U + \rho(X - Y).$    (25)
    Update $U$ and $\rho$ via:
        $U^{k+1} = U^k + \rho^k(X^{k+1} - Y^{k+1}),$    (26)
        $\rho^{k+1} = \alpha \rho^k.$
    if $\|X^k - Y^k\|_F \leq \epsilon$ then
        break
    end if
end for
```

Theorem 4. *If $g(X)$ is H_g-strongly convex and $\rho^k = \rho$ constant, with $\frac{\rho + H_g}{2} \geq \frac{L_g^2}{\rho}, \rho > L_g$ then under Algorithm 2 the augmented Lagrangian $\mathcal{L}(X^k, Y^k; U^k)$ converges to $\mathcal{L}(X^*, Y^*, U^*)$ at a linear rate.*

Proof. See Appendix C.1. □

6. Numerical Experiments

In this section, we give numerical results on the proposed methods for community detection, MAX-CUT, image segmentation, and symmetric matrix factorization. In each application, we evaluate and compare these four methods. (i) SD: the solution to a semidefinite relaxation of (1) (SDR), where $\mathcal{C} = \mathbb{R}^{n,r}$. The binary vector factor x where $xx^T = Z$ is recovered using a Goemans–Williamson style rounding. [9] technique. This is our baseline method and is described in more detail below. (ii) MR1: Algorithm 1 with $r = 1$. (iii) MRR: Algorithm 1 with $r = \left\lceil \sqrt{2n} \right\rceil$, then rounded to a binary vector using a nonsymmetric version of the Goemans–Williamson style rounding [9] technique. Both MR1 and MRR have the following stopping criterion $\max\{P^{(k)}, D^{(k)}\} \leq \epsilon$ for some tolerance parameter $\epsilon > 0$, where: $P^{(k)} := \left\{\frac{\|Z^k - Z^{k-1}\|_2}{\|Z^k\|_2}, \frac{\|X^k - X^{k-1}\|_2}{\|X^k\|_2}, \frac{\|Y^k - Y^{k-1}\|_2}{\|Y^k\|_2}\right\}$, $D^{(k)} := \max\left\{\frac{\|Z^{(k)} - X^{(k)}(Y^{(k)})^T\|_2}{\|Z^k\|_2}, \frac{\|X^{(k)} - Y^{(k)}\|_2}{\|X^{(k)}\|_2}\right\}$. (Here, $D^{(k)}$ is also proportional to the difference in dual iterates, and thus $P^{(k)}$ and $D^{(k)}$ can be interpreted as primal and dual residuals, respectively). (iv) V: Algorithm 2, with stopping criterion $\max\{P^{(k)}, D^{(k)}\} \leq \epsilon$ where $P^{(k)} := \left\{\frac{\|x^k - x^{k-1}\|_2}{\|x^k\|_2}, \frac{\|y^k - y^{k-1}\|_2}{\|y^k\|_2}\right\}$, $D^{(k)} := \frac{\|x^k - y^k\|_2}{\|x^k\|_2}$. The same primal and dual residual interpretation can be used here as well. In all cases, we use the following scheme for ρ: $\rho^k = \min\{\rho_{\max}, \rho^{k-1} * \gamma\}$, where $\rho_{\max} \approx 10{,}000$ and $\gamma \approx 1.05$ (slightly larger than 1).

6.1. Solving the Baseline (SDR)

As a baseline, we compare against the solution of the semidefinite relaxed problem without factor variables X (e.g., $\mathcal{C} = \mathbb{R}^{n,n}$):

$$\min_{Z} f(Z), s.t. \quad \mathcal{A}(Z) = b, Z \succeq 0. \tag{27}$$

For a fair comparison, we use a first-order splitting method very similar to ADMM, which is the Douglas–Rachford Splitting (DRS) method ([71,72], see also [73,74]). We introduce dummy variables and solve the reformulation of (27):

$$\min_{Z_1, Z_2, Z_3} g_1(Z_1) + g_2(Z_2) + g_3(Z_3), s.t. Z_1 + Z_2 + Z_3,$$

where $g_1(Z_1) = \mathbf{Tr}(CZ_1), g_2(Z_2) = \begin{cases} 0, & \mathcal{A}(Z_2) = b \\ +\infty, & \text{else,} \end{cases} \quad g_3(Z_3) = \begin{cases} 0, & Z_3 \succeq 0 \\ +\infty, & \text{else.} \end{cases}$. An application of the DRS on this reformulation (see also Algorithm 3.1 in [75]) is then the following iteration scheme: for $i = 1, 2, 3$,

$$\begin{aligned} X_i^{(k+1)} &= \mathbf{prox}_{tg_i}(Z_i), \hat{Y}_i = 2X_i^{(k+1)} - Z_i^{(k)}, \\ Y^{(k+1)} &= \frac{1}{3}(X_1^{(k+1)} + X_2^{(k+1)} + X_3^{(k+1)}), \\ Z_i^{(k+1)} &= Z_i^{(k)} + \rho(Y^{(k+1)} - X_i^{(k+1)}) \end{aligned}$$

and for a convex function f $z = \mathbf{prox}_{tf}(u) \iff \underset{z}{\operatorname{argmin}} f(z) + \frac{1}{2t}\|z - u\|_2^2$.

6.2. Rounding

Following the technique in [9], we can estimate x from a rank r matrix $X \approx xx^T$ by randomly projecting the main eigenspaces on the unit sphere. The exact procedure is as follows. (i) For the symmetric SDP solution X, we first perform an eigenvalue decomposition $X = Q\Lambda Q^T$ and form a factor $F = Q\Lambda^{1/2}$ where the diagonal elements of Λ are in decreasing magnitude order. Then, we scan $k = 1, \ldots, n$ and find $x_{k,t} = \mathbf{sign}(F_k z_t)$ for trials $t = 1, \ldots 10$. Here, F_k contains the first k columns of F, and each element of $z_t \in \mathbb{R}^k$ is drawn i.i.d from a normal Gaussian distribution. We report the values for

$x_r = \underset{x_{k,t}}{\text{argmin}} \{x_r^T C x_r\}$. (ii) For the MRR method, we repeat the procedure using a factor $F = U\Sigma^{1/2}$ where $X = U\Sigma V^T$ is the SVD of X. (iii) For MR1 and V, we simply take $x_r = \textbf{sign}(x)$ as the binary solution.

6.3. Computer Information

The following simulations are performed on a standard desktop computer with an Intel Xeon processor (3.6 GHz), and 32 GB of RAM. It is running with Matlab R2017a.

6.4. MAX-CUT

Table 1 gives the best MAX-CUT values using best-of-random-guesses and our approaches over four examples from the seventh DIMACS Implementation Challenge in 2002 (see http://dimacs.rutgers.edu/Workshops/7thchallenge/, problems downloaded from http://www.optsicom.es/maxcut/). Often, we find the quality of our recovered solutions close to the best-known solutions and often achieve similar suboptimality as the rounded SDR solutions. However, the runtime comparison (Figure 1) suggests that the ADMM methods (especially MR1 and SDR) are much more computationally efficient and scalable. All experiments are performed with $\epsilon = 1 \times 10^{-3}$.

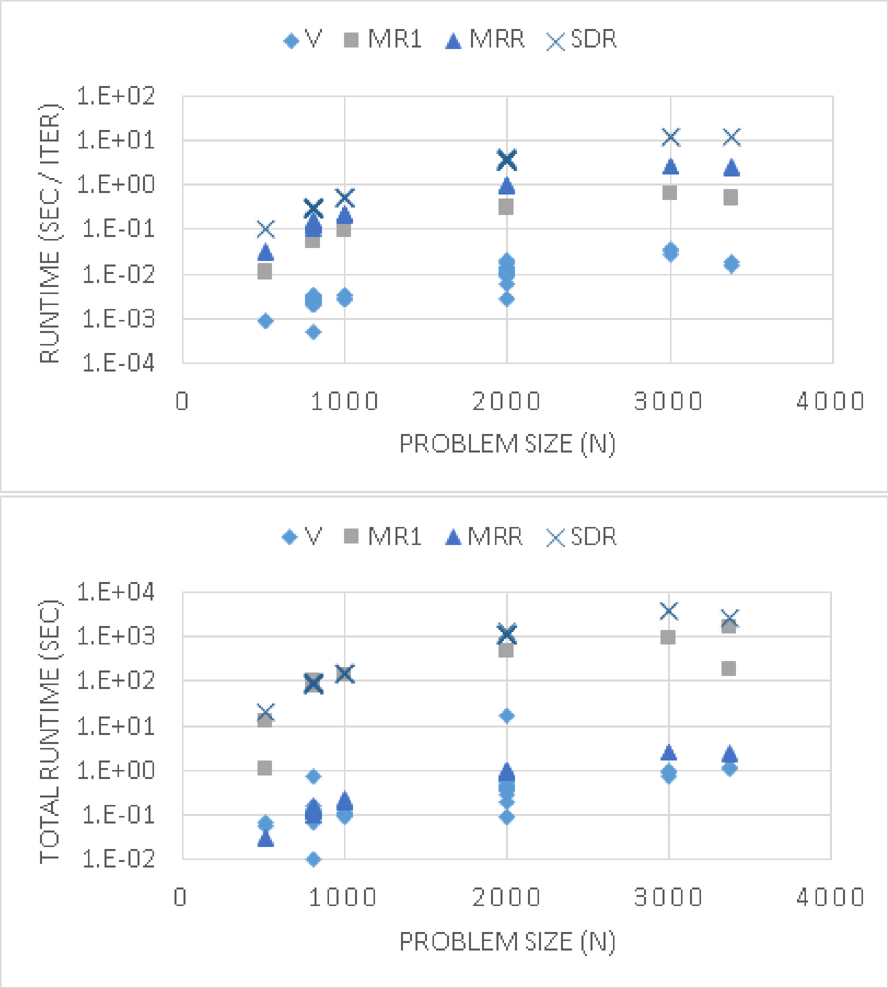

Figure 1. Time comparisons for DIMACS problems. (**top**): average runtime per iteration. (**bottom**): total runtime. We observe that both V and MRR converge in relatively few iterations, with MR1 taking slightly longer. However, as previously observed with splitting methods, the convergence rate is sensitive to the parameter choices $\rho^{(t)}$. For best performance, we start with a relatively small initial penalty coefficient and increase it with the iteration until the upper bound is achieved.

Table 1. MAX-CUT values for graphs from the 7th DIMACS Challenge. MRR = matrix formulation, $r = \left\lceil \sqrt{2n} \right\rceil$. SDR = SDP relaxation + rounding technique.

Database	*n*	Sparsity	BK	V	MR1	MRR	SDR
g3-8	512	0.012	41,684,814	34,105,231	36,780,180	35,943,350	33424095
g3-15	3375	0.018	281,029,888	235,893,612	255,681,256	241,740,931	212,669,181
pm3-8-50	512	0.012	454	394	346	378	416
pm3-15-50	3375	0.018	2964	2594	1966	2140	2616
G1	800	0.0599	11,624	10,938	11,047	11,321	11,360
G2	800	0.0599	11,620	10,834	11,082	11,144	11,343
G3	800	0.0599	11,622	10,858	10,894	11,174	11,367
G4	800	0.0599	11,646	10,849	10,760	11,192	11,429
G5	800	0.0599	11,631	10,796	10,783	11,352	11,394
G6	800	0.0599	2178	1853	1820	1949	1941
G7	800	0.0599	2003	1694	1644	1705	1774
G8	800	0.0599	2003	1688	1641	1728	1766
G9	800	0.0599	2048	1771	1681	1807	1830
G10	800	0.0599	1994	1662	1641	1737	1732
G11	800	0.005	564	496	460	480	506
G12	800	0.005	556	486	448	480	512
G13	800	0.005	580	516	476	498	528
G14	800	0.0147	3060	2715	2768	2861	2901
G15	800	0.0146	3049	2625	2810	2803	2884
G16	800	0.0146	3045	2667	2736	2862	2910
G17	800	0.0146	3043	2638	2789	2840	2920
G18	800	0.0147	988	798	768	841	858
G19	800	0.0146	903	700	641	694	780
G20	800	0.0146	941	723	691	766	788
G21	800	0.0146	931	696	713	810	794
G22	2000	0.01	13,346	12,461	12,548	12,751	12,926
G23	2000	0.01	13,317	12,540	12,528	12,853	12,889
G24	2000	0.01	13,314	12,540	12,447	12,723	12,904
G25	2000	0.01	13,326	12,447	12,558	12,733	12,874
G26	2000	0.01	13,314	12,445	12,475	12,718	12,847
G27	2000	0.01	3318	2824	2508	2807	2909
G28	2000	0.01	3285	2753	2518	2796	2845
G29	2000	0.01	3389	2864	2628	2901	2896
G30	2000	0.01	3403	2887	2639	2937	2971
G31	2000	0.01	3288	2833	2518	2902	2825
G32	2000	0.002	1398	1220	1066	1204	1254
G33	2000	0.002	1376	1202	1054	1166	1250
G34	2000	0.002	1372	1208	1096	1170	1222
G35	2000	0.0059	7670	6605	6914	6764	7209
G36	2000	0.0059	7660	6564	6943	6598	7228
G37	2000	0.0059	7666	6478	6839	6789	7183
G38	2000	0.0059	7681	6486	6759	6768	7212
G39	2000	0.0059	2395	1616	1697	1840	1997
G40	2000	0.0059	2387	1617	1438	1921	1890
G41	2000	0.0059	2398	1606	1656	1778	1899
G42	2000	0.0059	2469	1707	1756	1862	1971
G43	1000	0.02	6659	6222	6236	6398	6475
G44	1000	0.02	6648	6275	6192	6447	6458
G45	1000	0.02	6652	6243	6255	6407	6454
G46	1000	0.02	6645	6217	6233	6398	6407
G47	1000	0.02	6656	6221	6266	6433	6454
G48	3000	0.0013	6000	5882	5006	5402	6000
G49	3000	0.0013	6000	5844	5038	5362	6000
G50	3000	0.0013	5880	5814	4994	5410	5880
G51	1000	0.0118	3846	3317	3446	3524	3642

Table 1. *Cont.*

Database	n	Sparsity	BK	V	MR1	MRR	SDR
G52	1000	0.0118	3849	3360	3471	3499	3662
G53	1000	0.0118	3846	3323	3510	3516	3660
G54	1000	0.0118	3846	3306	3428	3509	3651

6.5. Image Segmentation

Both community detection and MAX-CUT can be used in image segmentation, where each pixel is a node and the similarity between pixels forms the weight of the edges. Generally, solving (1) for this application is not preferred, since the number of pixels in even a moderately sized image is extremely large. However, because of our fast methods, we successfully performed image segmentation on several thumbnail-sized images, as seen in Figure 2.

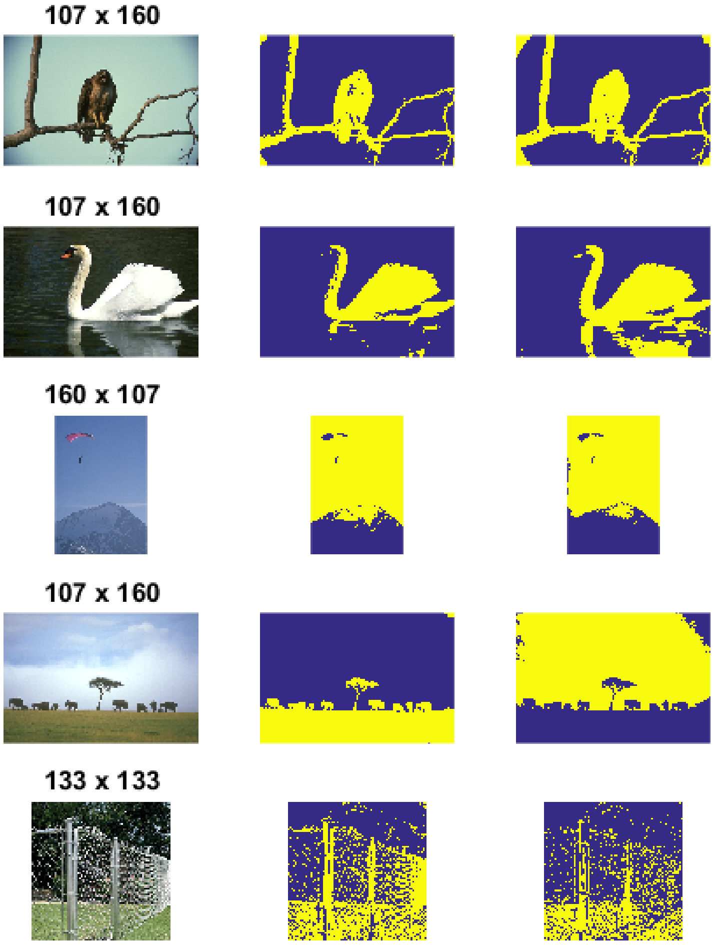

Figure 2. Image segmentation. The center and right columns are the best MAX-CUT and community detection results, respectively.

The C matrix is composed as follows. For each pixel, we compose two feature vectors: f_c^{ij} containing the RGB values and f_p^{ij} containing the pixel location. Scaling f_c^{ij} by some weight c, we form the concatenated feature vector $f^{ij} = [f_c^{ij}, cf_p^{ij}]$, and form the weighted adjacency matrix as the squared distance matrix between each feature vector $A_{(ij),(kl)} = \|f^{ij} - f^{kl}\|_2^2$. For MAX-CUT, we again form $C = A - \mathbf{Diag}(A\mathbf{1})$ as before. For community detection, since we do not have exact p and q values, we use an approximation as $C = a\mathbf{1}\mathbf{1}^T - A$ where $a = \frac{1}{n^2}\mathbf{1}^T A\mathbf{1}$ the mean value of A. Sweeping C and ρ_0, we give the best qualitative result in Figure 2.

6.6. Symmetric Factorization with Partial Observations

Recall the factorization with partial observations formulation as follows:

$$\min_{Z\in\mathbb{S}^n, X\in\mathbb{R}^{n\times r}} \sum_{i,j\in\Omega} (Z_{ij}-C_{ij})^2, s.t. Z = XX^T, \quad X \geq 0. \tag{28}$$

Note that here we generalize the aforementioned formulation with $r = 5$. In this setting, while the strongly convex $Y-$update in the proposed algorithm can no longer be solved in closed form, projected gradient descent is applied to deal with it. The relative error defined as $\|(Z^* - C)_\Omega\| / \|C_\Omega\|$ and CPU time with varying problem size and sparsity are demonstrated in Table 2.

Table 2. Result for nonnegative factorization with partial observations from linearized ADMM (5 trials). STD = standard deviation.

n	1000			3000			5000			8000		
$\lvert\Omega\rvert/n^2$	0.1	0.5	0.8	0.1	0.5	0.8	0.1	0.5	0.8	0.1	0.5	0.8
CPU time/s	9.74	13.53	13.97	61.15	78.99	64.76	117.54	85.24	131.64	212.26	220.42	337.74
$\frac{\|(Z^*-C)_\Omega\|}{\|C_\Omega\|}$	0.86	0.85	0.86	0.89	0.89	0.89	0.89	0.88	0.87	0.88	0.90	0.89
STD	0.043	0.020	0.021	0.010	0.006	0.008	0.008	0.012	0.018	0.004	0.008	0.008

7. Conclusions

We present two methods for solving quadratic combinatorial problems using ADMM on two reformulations. Though the problem has a nonconvex constraint, we give convergence results to KKT solutions under mild conditions. From this, we give empirical solutions to several graph-based combinatorial problems, specifically MAX-CUT and community detection; both can be used in additional downstream applications, like image segmentation.

Funding: This research received no external funding.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: The author declares no conflict of interest.

Appendix A. Derivation of X, Z Update

In a linearized case, consider $G = \nabla f(Z^{k-1}) = G_\Omega$. Then, the optimality conditions are:

$$\begin{aligned} G - \mathcal{A}^*(\nu) + S + \rho(Z - (XY^T)_\Omega) &= 0 \\ U - SY + \rho((XY^T)_\Omega Y - ZY) + \rho(X - Y) &= 0 \\ \mathcal{A}(Z) &= b. \end{aligned}$$

Using $D = \rho^{-1}(SY - U) + Y, B = -\rho^{-1}(G - \mathcal{A}^*(\nu) + S)$, we obtain:

$$\begin{aligned} -B + Z - (XY^T)_\Omega &= 0 \\ -D + (XY^T)_\Omega Y - ZY + X &= 0 \\ \mathcal{A}(Z) &= b. \end{aligned}$$

Substitute for Z: $Z = (XY^T)_\Omega + B \Rightarrow D + ((XY^T)_\Omega + B)Y = (XY^T)_\Omega Y + X \Rightarrow D + BY = X$. Since we assume the diagonal is in Ω, $\mathcal{A}(X_\Omega) = \mathcal{A}(X)$, so to solve for ν:

$$\begin{aligned} &\mathcal{A}((XY^T)_\Omega + B) = \mathcal{A}(XY^T + B) \\ &\quad = \mathcal{A}((D + BY)Y^T + B) = b, \end{aligned}$$

and therefore $\mathcal{A}(B(YY^T+I)) = b - \mathcal{A}(DY^T)$. Insert B and simplify:

$$\begin{aligned} & b - \mathcal{A}(DY^T) \\ = \; & \mathcal{A}((-\rho^{-1}(G - \mathcal{A}^*(\nu) + S))(YY^T + I)) \\ = \; & -\rho^{-1}\mathcal{A}((G - \mathcal{A}^*(\nu) + S)(YY^T + I)), \end{aligned}$$

and thus:

$$\begin{aligned} & b - \mathcal{A}(DY^T) + \rho^{-1}\mathcal{A}((G+S)(YY^T+I)) \\ & = \rho^{-1}\mathcal{A}(\mathcal{A}^*(\nu)(YY^T+I)) = \rho^{-1}H\nu, \end{aligned} \tag{A1}$$

where H is an $m \times m$ matrix with: $H_{ij} = \langle A_i, A_j(YY^T+I)\rangle$. Thus this system reduces to $\nu = H^{-1}\big(b - \mathcal{A}(DY^T) + \rho^{-1}\mathcal{A}((G+S)(YY^T+I))\big)$.

Implicit Inverse of H

When $\mathcal{A} = \textbf{diag}$, (20) reduces to n scalar element-wise computations $\nu_i = \frac{\rho(b-(DY^T)_{ii})+((G+S)(I+YY^T))_{ii}}{(YY^T)_{ii}+1}$. When $\mathcal{A} = \textbf{Tr}$, $\nu = \frac{\rho(b-\textbf{Tr}(DY^T)+\textbf{Tr}((G+S)(I+YY^T))}{\textbf{Tr}(YY^T)+1}$. Note that in both cases, the computation for ν can be done without ever forming an $n \times n$ matrix. For example, for $\mathcal{A} = \textbf{diag}$, $DY^T_{ii} = \rho^{-1}(SYY^T)_{ii} - \rho^{-1}(UY^T)_{ii} + (YY^T)_{ii}$ Recall that for any two matrices $A,B \in \mathbb{R}^{n\times r}$, $(AB^T)_{ii} = A_i^T B_i$ where A_i, B_i are the ith rows of A and B; thus, an efficient way of computing ν is (i) Compute more skinny matrices $F_1 = SY$, $F_2 = GY$. (ii) Compute the element-wise products $G_1 = F_1 \circ Y$, $G_2 = U \circ Y$, $G_3 = F_2 \circ Y$, and $G_4 = Y \circ Y$, where $(A \circ B)_{ij} = A_{ij}B_{ij}$ (element-wise multiplication). (iii) Compute the row sums $g_i = G_i\mathbf{1}$, $i = 1, ..., 4$. (iv) Compute the "numerator vector" $h_1 = \rho(b - (\rho^{-1}(g_1 - g_2) + g_4) + \textbf{diag}(G) + \textbf{diag}(S) + g_3 + g_1$ and "denominator vector" $h_2 = g_4 + 1$. (v) Then, $\nu_i = \frac{(h_1)_i}{(h_2)_i}$.

A similar procedure can be executed for $\mathcal{A} = \textbf{Tr}$ to keep memory requirements low.

Appendix B. Convergence Analysis for Matrix Form

To simplify notation, we first collect the primal and dual variables $P^k = (Z, X, Y)^k$ and $D^k = (\Lambda_1, \Lambda_2)^k$. We define the augmented Lagrangian at iteration k as:

$$\begin{aligned} \mathcal{L}^k : \; = \; & \mathcal{L}(P^k; D^k; \rho^k) = f(Z^k) + \delta_{\mathcal{C}}(Y) \\ + \; & \langle U, X - Y\rangle + \langle S, Z - XY^T\rangle \\ + \; & \frac{\rho}{2}\|X - Y\|_F^2 + \frac{\rho}{2}\|Z - XY^T\|_F^2, \end{aligned} \tag{A2}$$

and its linearization at iteration k as:

$$\begin{aligned} \bar{\mathcal{L}}^k \; := \; & \bar{\mathcal{L}}(P^k; D^k; \rho^k; \bar{f}^k) = \bar{f}^k + \delta_{\mathcal{C}}(Y) \\ + \; & \langle U, X - Y\rangle + \langle S, Z - XY^T\rangle + \frac{\rho}{2}\|X - Y\|_F^2, \\ + \; & \frac{\rho}{2}\|Z - XY^T\|_F^2 \end{aligned} \tag{A3}$$

Here, $\bar{f}^k := f(Z^{k-1}) + \langle G^{k-1}, Z - Z^{k-1}\rangle$ such that f^k is the linearization of f at Z^{k-1}.

Lemma A1. $\nabla^2\mathcal{L}_Y = \nabla^2\bar{\mathcal{L}}_Y \succeq \rho^k I$.

Proof. Given the definition of $\mathcal{L}$, we can see that the Hessian $\nabla^2\mathcal{L}_Y = \rho^k(M + I) \succeq \rho^k I$ where $M = \textbf{blkdiag}(X^TX, X^TX, ...) \succeq 0$. □

Lemma A2. $\nabla^2\bar{\mathcal{L}}_{(X,Z)} \succeq \rho^k\left(1 - \frac{\sqrt{\lambda_N^2 + 4\lambda_N} - \lambda_N}{2}\right)I$.

Proof. For (X, Z), we have $\nabla^2_{(X,Z)}\mathcal{L}_k = \rho^k \begin{bmatrix} I + NN^T & -N \\ -N^T & I \end{bmatrix}$ where $N = \mathbf{blkdiag}(Y^T, \ldots, Y^T) \in \mathbb{R}^{nr \times n^2}$. Note that for block diagonal matrices, $\|N\|_2 = \|Y\|_2$. Note also that the determinant of $\frac{1}{\rho^k}\nabla^2_{(X,Z)}\mathcal{L}_k$ is $\mathbf{det}((I + NN^T) - NN^T) = 1 \geq \mathbf{0}$, so $\nabla^2_{(X,Z)}\tilde{\mathcal{L}}_k \succ \mathbf{0}$ and equivalently $\lambda_{\min}(\nabla^2_{(X,Z)}\mathcal{L}_k) > 0$.

To find the smallest eigenvalue $\lambda_{\min}(\nabla^2_{(X,Z)}\mathcal{L}_k)$, it suffices to find the largest $\sigma > 0$ such that:

$$\begin{aligned} H_2 &= (\rho^k)^{-1}\nabla^2_{(X,Z)}\tilde{\mathcal{L}}_k - \sigma I \qquad \text{(A4)} \\ &= \begin{bmatrix} (1-\sigma)I + NN^T & -N \\ -N^T & (1-\sigma)I \end{bmatrix} \succeq \mathbf{0}. \end{aligned}$$

Equivalently, we want to find the largest $\sigma > 0$ where $(1-\sigma)I \succeq 0$ and the Schur complement of H_2 i.4., $H_3 = (1-\sigma)I + NN^T(1-(1-\sigma)^{-1})) \succeq 0$. Defining $\sigma_Y = \|Y\|_2$ the largest singular vector of Y^{k+1}, and noting that $\lambda_{\min}(\alpha I + A) = \alpha + \lambda_{\min}(A)$ for any positive semidefinite matrix A, we have $\lambda_{\min}(H_3) = (1-\sigma) + (\sigma_Y)^2(1-(1-\sigma)^{-1})$. We can see that $(1-\sigma)\lambda_{\min}(H_3)$ is a convex function in $(1-\sigma)$, with two zeros at $1-\sigma = \frac{\pm\sqrt{\sigma_Y^4 + 4\sigma_Y^2} - (\sigma_Y)^2}{2}$. In between the two roots, $\lambda_{\min}(H_3) < 0$. Since the smaller root cannot satisfy $1-\sigma > 0$, we choose $\sigma_{\max} = 1 - \frac{\sqrt{\sigma_Y^4 + 4\sigma_Y^2} - (\sigma_Y)^2}{2} > 0$ as the largest feasible σ that maintains $\lambda_{\min}(H_3) \geq 0$. As a result, $\lambda_{\min}(\nabla^2_{(X,Z)}\bar{\mathcal{L}}) = \rho^k\sigma_{\max} = \rho^k\left(1 - \frac{\sqrt{\sigma_Y^4 + 4\sigma_Y^2} - \sigma_Y^2}{2}\right)$. Figure A1 shows how this term behaves according to the spectral norm of Y. □

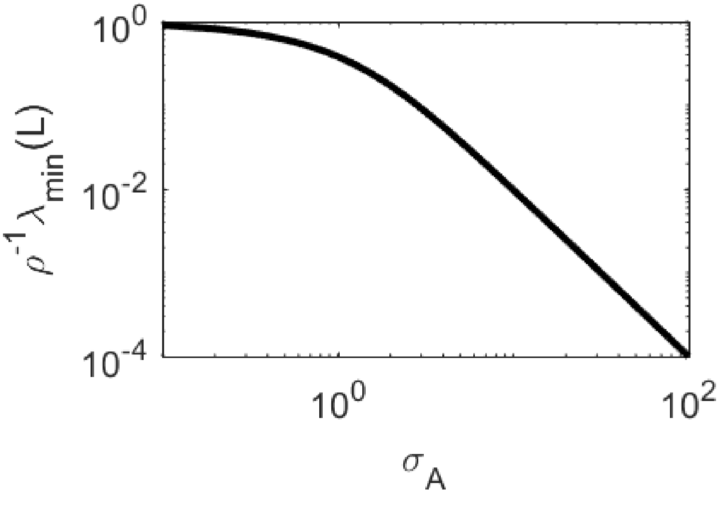

Figure A1. Strong convexity with respect to X, Z. Smallest eigenvalue of $\nabla^2_{X,Z}\mathcal{L}$ as a function of the spectral norm of Y.

We now prove the main theorem.

Lemma A3. *Consider the sequence:*

$$\begin{aligned} \mathcal{L}^k &:= \mathcal{L}(P^k; D^k) \\ &= f(Z^{k-1}) + \langle \nabla f(Z^{k-1}), Z - Z^{k-1} \rangle + \delta_{\mathcal{C}}(Y) \\ &+ \langle U, X - Y \rangle + \langle S, Z - XY^T \rangle + \frac{\rho}{2}\|X - Y\|_F^2 \\ &+ \frac{\rho}{2}\|Z - XY^T\|_F^2. \end{aligned}$$

If $f(Z)$ is L_f-Lipschitz smooth, then sequence $\mathcal{L}^k$ generated from Algorithm 1 satisfies:

$$\begin{aligned} &\mathcal{L}^{k+1} - \mathcal{L}^k \leq \\ &-c_1^k\|X^{k+1} - X^k\|_F^2 - c_2^k\|Z^{k+1} - Z^k\|_F^2 - c_3^k\|Y^{k+1} - Y^k\|_F^2 \\ &\qquad + \frac{\rho^{k+1} + \rho^k}{2(\rho^k)^2}(\|S^{k+1} - S^k\|_F^2 + \|U^{k+1} - U^k\|_F^2). \end{aligned} \tag{A5}$$

with $c_1^k = \frac{\rho^k}{2}\left(1 - \frac{\sqrt{\sigma_Y^4 + 4\sigma_Y^2} - \sigma_Y^2}{2}\right)$, $c_2^k = c_1^k - \frac{L_f}{2}$, *and* $c_3^k = \frac{\rho^k}{2} > 0$.

Proof. The proof outline of Lemma A3 is to show that each update step is a non-ascent step in the linearized augmented Lagrangian, and at least one update step is descent. We can describe the linearized ADMM in terms of four groups of updates: the primal variable Y, the primal variables X and Z, the dual variables U, S, and coefficient ρ. In other words, at iteration k, taking:

- $\Delta X^k = X^{k+1} - X^k$, $\Delta Y^k = Y^{k+1} - Y^k$, and $\Delta Z^k = Z^{k+1} - Z^k$.
- $\mathcal{L}^k = \mathcal{L}(Z^k, X^k, Y^k; D^k; \rho^k; G^k)$,
- $\mathcal{L}^Y = \mathcal{L}(Z^k, X^k, Y^{k+1}; D^k; \rho^k; G^k)$,
- $\mathcal{L}^{XZ} = \mathcal{L}(P^{k+1}; D^k; \rho^k; G^k)$, and
- $\mathcal{L}^{k+1} = \mathcal{L}(P^{k+1}; D^{k+1}; \rho^{k+1}; G^k)$

and $\mathcal{L}^{k+1} - \mathcal{L}^k = (\mathcal{L}^Y - \mathcal{L}^k) + (\mathcal{L}^{XZ} - \mathcal{L}^Y) + (\mathcal{L}^{k+1} - \mathcal{L}^{XZ})$. We now lower bound each term.

- Update Y. For the update of Y in (17), taking $\mathcal{L}^Y = \mathcal{L}(Z^k, X^k, Y^{k+1}; D^k; \rho^k; G^k)$, we have:

$$\begin{aligned} \mathcal{L}^Y - \mathcal{L}^k \quad &\overset{(a)}{\leq} \langle \nabla_Y \mathcal{L}^Y, Y^{k+1} - Y^k \rangle \\ &\quad - \frac{\lambda_{\min}(\nabla^2_{\mathrm{vec}Y} \mathcal{L}^Y)}{2}\|Y^{k+1} - Y^k\|_F^2 \\ &\overset{(b)}{\leq} -\frac{\rho^k}{2}\|Y^{k+1} - Y^k\|_F^2, \end{aligned} \tag{A6}$$

  where (a) follows from the definition of strong convexity, and (b) the optimality of Y^{k+1}.
- Update X, Z. Similarly, the update of (Z, X) in (17), denoting $\mathcal{L}^{XZ} = \mathcal{L}(P^{k+1}; D^k; \rho^k; G^k)$, we have:

$$\begin{aligned} &\bar{\mathcal{L}}^{XZ} - \mathcal{L}^Y \\ &\overset{(a)}{\leq} \langle \nabla_Z \bar{\mathcal{L}}^{XZ}, Z^{k+1} - Z^k \rangle + \langle \nabla_X \bar{\mathcal{L}}^{XZ}, X^{k+1} - X^k \rangle \\ &\quad - \frac{\lambda_{\min}(\nabla^2_{(X,Z)} \mathcal{L}^{XZ})}{2}(\|\Delta Z^k\|_F^2 + \|\Delta X^k\|_F^2) \\ &\overset{(b)}{\leq} -\frac{\lambda_{\min}(\nabla^2_{(X,Z)} \bar{\mathcal{L}}^{XZ})}{2}\left(\|\Delta Z^k\|_F^2 + \|\Delta X^k\|_F^2\right), \end{aligned} \tag{A7}$$

  where (a) follows from the definition of strong convexity, and (b) the optimality of X^{k+1} and Z^{k+1}. To further bound $\mathcal{L}^{XZ} - \bar{\mathcal{L}}^{XZ}$, we use the linearization definitions:

$$\begin{aligned} &\mathcal{L}^{XZ} - \bar{\mathcal{L}}^{XZ} \\ &\quad = \quad f(Z^{k+1}) - f(Z^k) - \langle \nabla f(Z^k), \Delta Z^k \rangle \\ &\quad \overset{(a)}{\leq} \quad \frac{L_f}{2}\|Z^{k+1} - Z^k\|_F^2, \end{aligned} \tag{A8}$$

  where (a) comes from the L_f Lipschitz smooth property of f. For a function f with Lipschitz constant L_f, the following holds $f(y) \leq f(x) + \langle y - x, \nabla f(x) \rangle + \frac{L_f}{2}\|y - x\|_2^2$.

- Update S, U, and ρ. For the update of the dual variables and the penalty coefficient, with $\mathcal{L}^k = \mathcal{L}(P^k; D^k; \rho^k)$, we have:

$$\begin{aligned}
&\mathcal{L}^D - \mathcal{L}^{XZ} \\
&\quad \overset{(a)}{=} \langle S^{k+1} - S^k, Z^{k+1} - X^{k+1}(Y^{k+1})^T \rangle \\
&\qquad + \langle U^{k+1} - U^k, X^{k+1} - Y^{k+1} \rangle \\
&\qquad + \frac{\rho^{k+1} - \rho^k}{2} (\|Z^{k+1} - X^{k+1}(Y^{k+1})^T\|_F^2) \\
&\qquad + \frac{\rho^{k+1} - \rho^k}{2} (\|X^{k+1} - Y^{k+1}\|_F^2) \\
&\quad \overset{(b)}{=} \frac{\rho^{k+1} + \rho^k}{2(\rho^k)^2} (\|S^{k+1} - S^k\|_F^2, + \|U^{k+1} - U^k\|_F^2),
\end{aligned} \tag{A9}$$

where (a) follows the definition of $\mathcal{L}$ and (b) from the dual update procedure.

The lemma statement results by incorporating (A6)–(A9). □

Lemma A4. *If $\mathcal{L}_k$ is unbounded below, then either problem (1) is unbounded below, or the sequence $L_f \|Z_k - Z_{k-1}\|_F$ diverges.*

Proof. First, consider the case that $\mathcal{L}_k$ is unbounded below. First, rewrite $\mathcal{L}^k$ equivalently as:

$$\begin{aligned}
\mathcal{L}^k = {} & f(Z^{k-1}) + \langle \nabla f(Z^{k-1}), Z^k - Z^{k-1} \rangle + \delta_C(Y^k) \\
& + \frac{\rho}{2}\|X^k - Y^k + \frac{1}{\rho^k} U^k\|_F^2 + \frac{\rho}{2}\|Z^k - X^k(Y^k)^T \\
& + \frac{1}{\rho^k} S^k\|_F^2 - \frac{1}{2\rho^k}\|U^k\|_F^2 - \frac{1}{2\rho^k}\|S^k\|_F^2.
\end{aligned}$$

Since $\|U^k\|_F$ and $\|S^k\|_F$ are bounded above, this implies that the linearization $g^k := f(Z^{k-1}) + \langle \nabla f(Z^{k-1}), Z^k - Z^{k-1} \rangle$ is unbounded below.

Note that:

$$\begin{aligned}
& g^k - f(Z^k) \\
& \quad = f(Z^{k-1}) - f(Z^k) - \nabla f(Z^{k-1}), Z^{k-1} - Z^k \rangle \\
& \quad \geq -\frac{L_f}{2}\|Z^k - Z^{k-1}\|_F^2,
\end{aligned}$$

which implies either $f(Z^k) \to -\infty$ or $L_f \|Z^k - Z^{k-1}\|_F^2 \to +\infty$. □

Corollary A1. *If $\mathcal{L}_k$ is unbounded below and the objective $f(Z) = \mathbf{Tr}(CZ)$, then it must be that (1) is unbounded below. This follows immediately since $L_f = 0$.*

Theorem A1. *Assume the dual variables are bounded, e.g., $\max\{\|S^k\|_F, \|U^k\|_F, \|Y^k\|_F\}_k \leq B_P < +\infty$, and $\frac{L_f}{\sigma_{\max}}$ is bounded above, where $\sigma_{\max} = 1 - \frac{\sqrt{\sigma_Y^4 + 4\sigma_Y^2} - \sigma_Y^2}{2}$, $\sigma_Y = \|Y^{k+1}\|_2$. Then, by running Algorithm 1 with $\rho^k = \alpha\rho^{k-1} = \alpha^k \rho_0$, if $\mathcal{L}_k$ is bounded below, then the sequence $\{P^k, D^k\}$ converges to a stationary point of (12).*

Proof. If $f(Z)$ is linear, take $K_0 = 0$. If $f(Z)$ is $L_f >$ smooth, take $\hat{K}$ large enough such that for all $k > K_0$, $\alpha^k \rho \geq L_f \sigma_{\max}$. By assumption, K_0 is always finite.

Taking $\Delta^k_{XYZ} = \left(\|\Delta Z^k\|_F^2 + \|\Delta X^k\|_F^2 + \|\Delta Y^k\|_F^2\right)$ and $c^k = \min\{c_1, c_2, c_3\}$, the summation of (A5) leads to:

$$\begin{aligned}\mathcal{L}^K - \mathcal{L}^{K_0} &= \sum_{k=K_0}^{K-1} \mathcal{L}^{k+1} - \mathcal{L}^k \\ &\leq \sum_{k=K_0}^{K-1} \frac{\rho^{k+1}+\rho^k}{2(\rho^k)^2}\left(\|S^{k+1}-S^k\|_F^2 + \|U^{k+1}-U^k\|_F^2\right) \\ &\qquad - \sum_{k=K_0}^{K-1} c^k \Delta^k_{XYZ} \\ &\overset{(a)}{\leq} 4B_P \sum_{k=K_0}^{K-1} \frac{\rho^{k+1}+\rho^k}{2(\rho^k)^2} - \sum_{k=K_0}^{K-1} c^k \Delta^k_{XYZ} \\ &\overset{(b)}{\leq} 4B_P \sum_{k=K_0}^{K-1} \frac{\rho^{k+1}+\rho^k}{2(\rho^k)^2},\end{aligned} \tag{A10}$$

where (a) follows from the boundedness assumption of the dual variables, and (b) follows from Lemmas A1 and A2, and careful construction of ρ with respect to L_f and $\|Y^{k+1}\|_2$. Further simplifying, we see that L^K is thus bounded above, since:

$$\begin{aligned}\mathcal{L}^K - \mathcal{L}^{K_0} &\leq \lim_{K\to\infty} 4B_P \sum_{k=K_0}^{K-1} \frac{\rho^{k+1}+\rho^k}{2(\rho^k)^2} \\ &= 4B_P \frac{1+\alpha}{2\alpha^{K_0}\rho}\left(1 + \frac{1}{\alpha} + \frac{1}{\alpha^2} + \cdots\right) \\ &= \frac{4B_P}{2\alpha^{K_0}\rho} < +\infty.\end{aligned}$$

If $\mathcal{L}^k$ is not unbounded below, then:

$$0 \leq \sum_{k=K_0}^{K-1} (c_1\|\Delta X^k\|_F^2 + c_2\|\Delta Z^k\|_F^2 + c_3\|\Delta Y^k\|_F^2\Big) \leq +\infty. \tag{A11}$$

Recall $c_3^k = \frac{\rho^k}{2}$, and by boundedness assumption on $\|Y_2^{k+1}$, for $k > K_0$, $c_1^k, c_2^k \propto \rho^k$. Since additionally $\sum_k \rho_k = +\infty$, then this immediately yields $Z^{k+1} - Z^k \to 0, X^{k+1} - X^k \to 0, Y^{k+1} - Y^k \to 0$.

Therefore, since the primal variables are convergent, this implies that:

$$\begin{aligned}Z^{k+1} - (X^{k+1}(Y^{k+1})^T)_\Omega &= \frac{1}{\rho^k}(S^{k+1} - S^k), \\ X^{k+1} - Y^{k+1} &= \frac{1}{\rho^k}(U^{k+1} - U^k),\end{aligned}$$

converges to a constant. But since $\rho^k \to \infty$ and the dual variables are all bounded, then it must be that: $Z^{k+1} - (X^{k+1}(Y^{k+1})^T)_\Omega \to 0, X^{k+1} - Y^{k+1} \to 0$. Therefore, the limit points X^*, Y^*, and Z^* are all feasible, and simply checking the first optimality condition will verify that this accumulation point is a stationary point of (12). □

Appendix C. Convergence Analysis for Vector Form

Lemma A5. *For two adjacent iterations of Algorithm 2, we have:*

$$\|U^{k+1} - U^k\|_2^2 \leq L_g^2\|X^{k+1} - X^k\|_2^2. \tag{A12}$$

Proof. From the first-order optimality conditions for the update of X:

$$\nabla g(X^{k+1}) + U^k + \rho^k(X^{k+1} - Y^{k+1}) = 0. \tag{A13}$$

Combining with the dual update, we obtain $\nabla g(X^{k+1}) + U^{k+1} = 0$. Then, the result follows from the definition of L_g. □

Next, we will show that the augmented Lagrangian is monotonically decreasing and lower bounded.

Lemma A6. *Each step in the augmented Lagrangian update is decreasing, e.g., for:*

$$\begin{aligned}&\mathcal{L}(X, Y; U; \rho) \\ &\quad := g(X) + \delta_{\mathcal{C}}(Y) + \langle U, X - Y\rangle + \frac{\rho}{2}\|X - Y\|_F^2\end{aligned} \tag{A14}$$

we have:

$$\begin{aligned}&\mathcal{L}(Y^{k+1}, X^{k+1}; U^{k+1}; \rho^{k+1}) \le \mathcal{L}(Y^{k+1}, X^{k+1}; U^k; \rho^k) \\ &\quad \le \mathcal{L}(Y^{k+1}, X^k; U^k; \rho^k) \le \mathcal{L}(Y^k, X^k; U^k; \rho^k).\end{aligned} \tag{A15}$$

Furthermore, the amount of decrease is:

$$\begin{aligned}&\mathcal{L}(Y^{k+1}, X^{k+1}; U^{k+1}; \rho^{k+1}) - \mathcal{L}(Y^k, X^k; U^k; \rho^k) \\ &\quad \le -\rho^k\|Y^{k+1} - Y^k\|_F^2 - c^k\|X^{k+1} - X^k\|_F^2.\end{aligned} \tag{A16}$$

Here,

- *if $g(X)$ is H_g-strongly convex (where $H_g = 0$ if g is convex but not strongly convex) then $c^k = \frac{\rho^k + H_g}{2} - L_g^2\frac{\rho^{k+1}+\rho^k}{2(\rho^k)^2}$, and*
- *if $g(X)$ is nonconvex but L_g-smooth, then $c^k = \frac{\rho^k - 3L_g}{2} - L_g^2\frac{\rho^{k+1}+\rho^k}{2(\rho^k)^2}$.*

Proof. Both the updates of Y and X globally minimize $\mathcal{L}$ with respect to those variables. To minimize Y at $(X, U) = (X^k, U^k)$:

$$\begin{aligned}&\mathcal{L}(Y^{k+1}, X; U; \rho) - \mathcal{L}(Y^k, X; U; \rho) && \text{(A17)}\\ \overset{(a)}{\le}\ &\langle \nabla_Y \mathcal{L}(Y^{k+1}, X; U; \rho), \Delta Y^k\rangle - \frac{\rho^k}{2}\|\Delta Y^k\|_2^2 \\ \overset{(b)}{\le}\ &-\frac{\rho^k}{2}\|\Delta Y^k\|_2^2. && \text{(A18)}\end{aligned}$$

To minimize X at $(Y, U) = (Y^{k+1}, U^k)$, we consider two cases. If g is H_g-strongly convex, then:

$$\begin{aligned}&\mathcal{L}(Y, X^{k+1}; U; \rho) - \mathcal{L}(Y, X^k; U; \rho) \\ \overset{(a)}{\le}\ &\langle \nabla_X \mathcal{L}(Y, X^{k+1}; U; \rho), X^{k+1} - X^k\rangle \\ &-\frac{\rho^k + H_g}{2}\|X^{k+1} - X^k\|_2^2 \\ \overset{(b)}{\le}\ &-\frac{\rho^k + H_g}{2}\|X^{k+1} - X^k\|_2^2,\end{aligned} \tag{A19}$$

where (a) follows from the strong convexity of $\mathcal{L}(Y, X; U; \rho)$ with respect to X, and (b) follows from the optimality condition of the update. If g is nonconvex but L_g-Lipschitz, then note that:

$$\begin{aligned}
& g(X^{k+1}) - g(X^k) \\
\leq\ & \langle \nabla g(X^k), X^{k+1} - X^k \rangle + \frac{L_g}{2} \|X^{k+1} - X^k\|_F^2 \\
\overset{(a)}{=}\ & \langle \nabla g(X^k) - \nabla g(X^{k+1}), X^{k+1} - X^k \rangle \\
& \quad + \frac{L_g}{2} \|X^{k+1} - X^k\|_F^2 + \langle \nabla g(X^{k+1}), X^{k+1} - X^k \rangle \\
\overset{(b)}{\leq}\ & \|\nabla g(X^k) - \nabla g(X^{k+1})\|_F \|X^{k+1} - X^k\|_F \\
& \quad + \frac{L_g}{2} \|X^{k+1} - X^k\|_F^2 + \langle \nabla g(X^{k+1}), X^{k+1} - X^k \rangle \\
\overset{(c)}{\leq}\ & \frac{3L_g}{2} \|X^{k+1} - X^k\|_F^2 + \langle \nabla g(X^{k+1}), X^{k+1} - X^k \rangle
\end{aligned}$$

where (a) follows from adding and subtracting a term, (b) from Cauchy–Schwartz, and (c) from the Lipschitz gradient condition on g. Therefore:

$$\begin{aligned}
& \mathcal{L}(Y, X^{k+1}; U; \rho) - \mathcal{L}(Y, X^k; U; \rho) \\
\overset{(a)}{\leq}\ & \langle \nabla_X \mathcal{L}(Y, X^{k+1}; U; \rho), X^{k+1} - X^k \rangle \\
& \quad - \frac{\rho^k - 3L_g}{2} \|X^{k+1} - X^k\|_2^2 \\
\overset{(b)}{\leq}\ & -\frac{\rho^k - 3L_g}{2} \|X^{k+1} - X^k\|_2^2.
\end{aligned}$$

In the dual variables, using $\{X, Y\} = \{X^{k+1}, Y^{k+1}\}$ we have:

$$\begin{aligned}
& \mathcal{L}(Y, X; U^{k+1}; \rho^{k+1}) - \mathcal{L}(Y, X; \mu^k; \rho^k) \\
\overset{(a)}{\leq}\ & \langle U^{k+1} - U^k, X - Y \rangle + \frac{\rho^{k+1} - \rho^k}{2} \|X - Y\|_F^2 \\
\overset{(b)}{\leq}\ & \frac{\rho^{k+1} + \rho^k}{2(\rho^k)^2} \|U^{k+1} - U^k\|_2^2 \\
\overset{(c)}{\leq}\ & L_g^2 \frac{\rho^{k+1} + \rho^k}{2(\rho^k)^2} \|X^{k+1} - X^k\|_2^2,
\end{aligned}$$

where (a) follows the definition of $\mathcal{L}$, (b) follows from the update of U, and (c) follows from Lemma (A5) since $\rho^k > 0$ for al k. Incorporating these observations completes the proof. □

Lemma A7. *If $\rho^k \geq L_g$ and the objective $g(X)$ is lower-bounded over $\mathcal{C}$, then the augmented Lagrangian (A14) is lower bounded.*

Proof. From the L_g-Lipschitz continuity of $\nabla g(X)$, it follows that:

$$g(X) \geq g(Y) + \langle \nabla g(X), X - Y \rangle - \frac{L_g}{2} \|X - Y\|_F^2 \tag{A20}$$

for any X and Y. By definition:

$$\begin{aligned} & \mathcal{L}(Y^k, X^k; U^k; \rho^k) \\ = \quad & g(X^k) + \langle U^k, X^k - Y^k \rangle + \frac{\rho^k}{2} \|X^k - Y^k\|_F^2 \\ \overset{(a)}{=} \quad & g(X^k) - \langle \nabla g(X^k), X^k - Y^k \rangle + \frac{\rho^k}{2} \|X^k - Y^k\|_F^2 \\ \overset{(b)}{\geq} \quad & g(Y^k) + \frac{\rho^k - L_g}{2} \|X^k - Y^k\|_F^2, \end{aligned} \tag{A21}$$

where (a) follows from the optimality in updating X and (b) follows from (A20). Since $\mathcal{L}^k$ is unbounded below, then $g(Y^k)$ is unbounded below. Since $Y^k \in \mathcal{C}$ for all k, this implies that g is unbounded below over $\mathcal{C}$. □

Thus, if $g(X)$ is lower-bounded over $\mathcal{C}$, since the sequence $\{\mathcal{L}(X^k, Y^k; U^k)\}$ is monotonically decreasing and lower bounded, the sequence $\{\mathcal{L}(X^k, Y^k; U^k)\}$ converges. Given the monotonic descent of each subproblem (Lemma A6) and strong convexity of $\mathcal{L}^k$ with respect to X and Y, it is clear that $X^k \to X^*$, $Y^k \to Y^*$ fixed points. Combining with Lemma A5 gives also $U^k \to U^*$.

The proof of Theorem 3 easily follows from Lemma A7.

Appendix C.1. Linear Rate of Convergence when g Is Strongly Convex

Lemma A8. *Consider Algorithm 2 with ρ^k constant. Then, collecting the variables all vectorized $x = (X, Y, Y)$,*

$$\mathcal{L}^{k+1} - \mathcal{L}^k \leq -c_3 \|x^{k+1} - x^k\|^2,$$

where g is H_g strongly convex and:

$$c_3 = \max_{\theta \in (0,1)} \min \left\{ \theta \left(\frac{\rho + H_g}{2} - \frac{L_g^2}{\rho} \right), (1-\theta) \left(\frac{\rho + H_g}{2H_g} - \frac{L_g^2}{\rho H_g} \right), -\rho \right\}.$$

Proof. From Lemma A6, we already have that:

$$\mathcal{L}^{k+1} - \mathcal{L}^k \leq -\rho \|Y^{k+1} - Y^k\|^2 - c\|X^{k+1} - X^k\|^2,$$

where for constant ρ, $c = \frac{\rho + H_g}{2} - \frac{L_g^2}{\rho}$. Moreover, when $g(X)$ is H_g-strongly convex,

$$\begin{aligned} \|U^{k+1} - U^k\|_2 &= \|\nabla g(X^{k+1}) - \nabla g(X^k)\|_2 \\ &\geq H_g \|X^{k+1} - X^k\|_2. \end{aligned}$$

Therefore:

$$\mathcal{L}^{k+1} - \mathcal{L}^k \leq -\theta \frac{c}{H_g} \|U^{k+1} - U^k\|_2^2 - (1-\theta) c \|X^{k+1} - X^k\|^2.$$

for any $\theta \in (0,1)$, We thus have:

$$\begin{aligned} & \mathcal{L}^{k+1} - \mathcal{L}^k \\ \leq \quad & -\theta c \|\Delta X^k\|_F^2 - (1-\theta) \frac{c}{H_g} \|\Delta U\|_F^2 - \rho \|\Delta Y^k\|_F^2 \\ \leq \quad & -\min\left\{ \theta c, (1-\theta) \frac{c}{H_g}, -\rho \right\} \begin{bmatrix} X^{k+1} - X^k \\ Y^{k+1} - Y^k \\ U^{k+1} - U^k \end{bmatrix}, \end{aligned}$$

with $\Delta U = U^{k+1} - U^k$. Note that this does not mean $\mathcal{L}$ is strong convex with respect to the *collected* variables $x = (X, Y, Z)$ ($\mathcal{L}$ is not even convex). But with respect to each variable X, Y, and Z, it is strongly convex. □

Lemma A9. *Again with $\rho^k > 1$ constant and collecting $x = (X, Y, Z)$, we have:*

$$\mathcal{L}^{k+1} - \mathcal{L}^* \le c_4\|x^{k+1} - x^*\|^2, \quad c_4 = \min\{L_g + \rho + 2, 2\rho, 1\}$$

whenever Y^{k+1} and Y^* are both in $\mathcal{C}$.

Proof. Over the domain $\mathcal{C}$, the augmented Lagrangian can be written as:

$$\mathcal{L}(x) = g(X) + \langle U, X - Y\rangle + \frac{\rho}{2}\|X - Y\|_F^2,$$

with gradient $\nabla\mathcal{L}(x) = \begin{bmatrix} \nabla g(X) + U + \rho(X - Y) \\ -Y + \rho(Y - X) \\ X - Y \end{bmatrix}$ and thus:

$$\begin{aligned} & \|\nabla\mathcal{L}(x_1) - \nabla\mathcal{L}(x_2)\|_F^2 \\ = \; & \|\nabla_X\mathcal{L}(x_1) - \nabla_X\mathcal{L}(x_2)\|_F^2 \\ & \quad + \|\nabla_Y\mathcal{L}(x_1) - \nabla_Y\mathcal{L}(x_2)\|_F^2 \\ & \quad + \|\nabla_U\mathcal{L}(x_1) - \nabla_U\mathcal{L}(x_2)\|_F^2 \\ \le \; & (L_g + \rho + 2)\|X_1 - X_2\|_F^2 + (2\rho)\|Y_1 - Y_2\|_F^2 \\ & \quad + \|U_1 - U_2\|_F^2 \\ \le \; & \min\{L_g + \rho + 2, 2\rho, 1\}\|x_2 - x_1\|_2^2, \end{aligned}$$

which reveals the Lipschitz smoothness constraint for $\mathcal{L}$ as $c_4 = \min\{L_g + \rho + 2, 2\rho, 1\}$. Then, using first-order optimality conditions,

$$\begin{aligned} \mathcal{L}^{k+1} & \le \mathcal{L}^* + \langle\nabla\mathcal{L}(x^*), x^{k+1} - x^*\rangle + c_4\|x^{k+1} - x^*\|_2^2 \\ & \overset{(a)}{\le} \mathcal{L}^* + c_4\|x^{k+1} - x^*\|_2^2, \end{aligned}$$

where (a) follows from the optimality of $\mathcal{L}^*$. □

Lemma A10. *Consider $g(x)$ H_g-strongly convex in x, and ρ large enough so that $c_3 > 0$. Then, the number of steps for $|\mathcal{L}^k - \mathcal{L}^0| \le \epsilon$ is $O(\log(1/\epsilon))$.*

This proof is standard in the linear convergence of block coordinate descent when the objective is strongly convex. Note that $\mathcal{L}$ is not strongly convex or even convex, but still all the steps hold.

Proof. Take $x^k = \{X^k, Y^k, U^k\}$ and $x^* = \{X^*, Y^*, U^*\}$. Then:

$$\begin{aligned} \mathcal{L}(x^k) - \mathcal{L}(x^*) & = \mathcal{L}(x^k) - \mathcal{L}(x^{k+1}) + \mathcal{L}(x^{k+1}) - \mathcal{L}(x^*) \\ & \ge c_3\|x^{k+1} - x^k\|^2 + \mathcal{L}(x^{k+1}) - \mathcal{L}(x^*) \\ & \ge \left(\frac{c_3}{c_4} + 1\right)(\mathcal{L}(x^{k+1}) - \mathcal{L}(x^*)) \end{aligned}$$

Therefore:

$$\frac{\mathcal{L}(x^k) - \mathcal{L}(x^*)}{\mathcal{L}(x^0) - \mathcal{L}(x^*)} \le \left(\frac{c_4}{c_4 + c_3}\right)^k$$

and so:

$$\mathcal{L}(x^k) - \mathcal{L}(x^*) \le \epsilon$$

if:

$$k \geq D_1 \log(1/\epsilon) + D_2$$

where:

$$D_1 = \log^{-1}\left(\frac{c_4 + c_3}{c_4}\right), \qquad D_2 = \frac{\log(\mathcal{L}(x^0) - \mathcal{L}(x^*))}{\log\left(\frac{c_4+c_3}{c_4}\right)}.$$

□

References

1. Bandeira, A.S.; Boumal, N.; Voroninski, V. On the low-rank approach for semidefinite programs arising in synchronization and community detection. In Proceedings of the Conference on Learning Theory, New York, NY, USA, 23–26 June 2016; pp. 361–382.
2. Fortunato, S.; Hric, D. Community detection in networks: A user guide. *Phys. Rep.* **2016**, *659*, 1–44. [CrossRef]
3. Javanmard, A.; Montanari, A.; Ricci-Tersenghi, F. Phase transitions in semidefinite relaxations. *Proc. Natl. Acad. Sci. USA* **2016**, *113*, E2218–E2223. [CrossRef] [PubMed]
4. Gillis, N. Nonnegative Matrix Factorization: Complexity, Algorithms and Applications. Doctoral Dissertation, Université Catholique de Louvain (CORE), Louvain-La-Neuve, France, 2011.
5. Ding, C.; He, X.; Simon, H.D. On the equivalence of nonnegative matrix factorization and spectral clustering. In Proceedings of the 2005 SIAM International Conference on Data Mining (SIAM), New Orleans, LA, USA, 11–15 July 2005; pp. 606–610.
6. Da Costa, A.P.; Seeger, A. Cone-constrained eigenvalue problems: Theory and algorithms. *Comput. Optim. Appl.* **2010**, *45*, 25–57. [CrossRef]
7. Gander, W.; Golub, G.H.; von Matt, U. A constrained eigenvalue problem. In *Numerical Linear Algebra, Digital Signal Processing and Parallel Algorithms*; Springer: Berlin/Heidelberg, Germany, 1991; pp. 677–686.
8. Júdice, J.J.; Sherali, H.D.; Ribeiro, I.M. The eigenvalue complementarity problem. *Comput. Optim. Appl.* **2007**, *37*, 139–156. [CrossRef]
9. Goemans, M.X.; Williamson, D.P. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM* **1995**, *42*, 1115–1145. [CrossRef]
10. Helmberg, C.; Rendl, F. A spectral bundle method for semidefinite programming. *SIAM J. Optim.* **2000**, *10*, 673–696. [CrossRef]
11. Fujie, T.; Kojima, M. Semidefinite programming relaxation for nonconvex quadratic programs. *J. Glob. Optim.* **1997**, *10*, 367–380. [CrossRef]
12. Burer, S.; Monteiro, R.D. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Math. Program.* **2003**, *95*, 329–357. [CrossRef]
13. Boumal, N.; Voroninski, V.; Bandeira, A. The non-convex Burer-Monteiro approach works on smooth semidefinite programs. In Proceedings of the 30th Annual Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, 5–10 December 2016; pp. 2757–2765.
14. Lee, D.D.; Seung, H.S. Algorithms for non-negative matrix factorization. In Proceedings of the Advances in Neural Information Processing Systems Conference, Denver, CO, USA, 4 October 2001; pp. 556–562.
15. Wolkowicz, H.; Saigal, R.; Vandenberghe, L. *Handbook of Semidefinite Programming: Theory, Algorithms, and Applications*; Springer Science & Business Media: New York, NY, USA, 2012; Volume 27.
16. Laurent, M. Sums of squares, moment matrices and optimization over polynomials. In *Emerging Applications of Algebraic Geometry*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 157–270.
17. Rendl, F. Semidefinite relaxations for partitioning, assignment and ordering problems. *4OR* **2012**, *10*, 321–346. [CrossRef]
18. Blekherman, G.; Parrilo, P.A.; Thomas, R.R. Semidefinite optimization and convex algebraic geometry. In Proceedings of the 2012 Annual Meeting of the Society for Industrial and Applied Mathematics (SIAM), Minneapolis, MN, USA, 9–13 July 2012.
19. Anjos, M.F.; Lasserre, J.B. Introduction to semidefinite, conic and polynomial optimization. In *Handbook on Semidefinite, Conic and Polynomial Optimization*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 1–22.
20. Abbe, E.; Bandeira, A.S.; Hall, G. Exact recovery in the stochastic block model. *IEEE Trans. Inf. Theory* **2016**, *62*, 471–487. [CrossRef]
21. Shi, J.; Malik, J. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2000**, *22*, 888–905.
22. Karisch, S.E.; Rendl, F. Semidefinite programming and graph equipartition. *Top. Semidefin. Inter. Point Methods* **1998**, *18*, 77–95.
23. Karger, D.; Motwani, R.; Sudan, M. Approximate graph coloring by semidefinite programming. *J. ACM* **1998**, *45*, 246–265. [CrossRef]
24. Barahona, F.; Grötschel, M.; Jünger, M.; Reinelt, G. An application of combinatorial optimization to statistical physics and circuit layout design. *Oper. Res.* **1988**, *36*, 493–513. [CrossRef]
25. De Simone, C.; Diehl, M.; Jünger, M.; Mutzel, P.; Reinelt, G.; Rinaldi, G. Exact ground states of Ising spin glasses: New experimental results with a branch-and-cut algorithm. *J. Stat. Phys.* **1995**, *80*, 487–496. [CrossRef]
26. Poljak, S.; Tuza, Z. The expected relative error of the polyhedral approximation of the MAX-CUT problem. *Oper. Res. Lett.* **1994**, *16*, 191–198. [CrossRef]

27. Helmberg, C.; Rendl, F. Solving quadratic (0,1)-problems by semidefinite programs and cutting planes. *Math. Program.* **1998**, *82*, 291–315. [CrossRef]
28. Rendl, F.; Rinaldi, G.; Wiegele, A. A branch and bound algorithm for MAX-CUT based on combining semidefinite and polyhedral relaxations. In Proceedings of the 12th International IPCO Conference, Ithaca, NY, USA, 25–27 June 2007; Volume 4513, pp. 295–309.
29. Burer, S.; Vandenbussche, D. A finite branch-and-bound algorithm for nonconvex quadratic programming via semidefinite relaxations. *Math. Program.* **2008**, *113*, 259–282. [CrossRef]
30. Bao, X.; Sahinidis, N.V.; Tawarmalani, M. Semidefinite relaxations for quadratically constrained quadratic programming: A review and comparisons. *Math. Program.* **2011**, *129*, 129–157. [CrossRef]
31. Krislock, N.; Malick, J.; Roupin, F. Improved semidefinite branch-and-bound algorithm for k-cluster. *HAL Open Sci. Prepr.* **2012**, hal-00717212. Available online: https://inria.hal.science/file/index/docid/717823/filename/krislock-malick-roupin-2012a.pdf (accessed on 22 September 2023).
32. Poljak, S.; Rendl, F.; Wolkowicz, H. A recipe for semidefinite relaxation for (0, 1)-quadratic programming. *J. Glob. Optim.* **1995**, *7*, 51–73. [CrossRef]
33. Helmberg, C. *Semidefinite Programming for Combinatorial Optimization*; Konrad-Zuse-Zentrum für Informationstechnik: Berlin, Germany, 2000.
34. Papadopoulos, S.; Kompatsiaris, Y.; Vakali, A.; Spyridonos, P. Community detection in social media. *Data Min. Knowl. Discov.* **2012**, *24*, 515–554. [CrossRef]
35. Girvan, M.; Newman, M.E. Community structure in social and biological networks. *Proc. Natl. Acad. Sci. USA* **2002**, *99*, 7821–7826. [CrossRef]
36. Keeling, M. The implications of network structure for epidemic dynamics. *Theor. Popul. Biol.* **2005**, *67*, 1–8. [CrossRef]
37. Holland, P.W.; Laskey, K.B.; Leinhardt, S. Stochastic blockmodels: First steps. *Soc. Netw.* **1983**, *5*, 109–137. [CrossRef]
38. Queiroz, M.; Judice, J.; Humes, C., Jr. The symmetric eigenvalue complementarity problem. *Math. Comput.* **2004**, *73*, 1849–1863. [CrossRef]
39. Deshpande, Y.; Montanari, A.; Richard, E. Cone-constrained principal component analysis. In Proceedings of the 28th Conference on Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014; pp. 2717–2725.
40. Zass, R.; Shashua, A. Nonnegative sparse PCA. In Proceedings of the 20th Annual Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 8–9 December 2007; pp. 1561–1568.
41. Lee, D.D.; Seung, H.S. Learning the parts of objects by non-negative matrix factorization. *Nature* **1999**, *401*, 788. [CrossRef] [PubMed]
42. Yuan, Z.; Oja, E. Projective nonnegative matrix factorization for image compression and feature extraction. In Proceedings of the Scandinavian Conference on Image Analysis, Joensuu, Finland, 19–22 June 2005; pp. 333–342.
43. Friedlander, M.P.; Macedo, I. Low-rank spectral optimization via gauge duality. *SIAM J. Sci. Comput.* **2016**, *38*, A1616–A1638. [CrossRef]
44. Jaggi, M.; Sulovsk, M. A simple algorithm for nuclear norm regularized problems. In Proceedings of the 27th International Conference on Machine Learning (ICML-10), Haifa, Israel, 21–24 June 2010; pp. 471–478.
45. Candès, E.J.; Recht, B. Exact matrix completion via convex optimization. *Found. Comput. Math.* **2009**, *9*, 717. [CrossRef]
46. Recht, B.; Fazel, M.; Parrilo, P.A. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Rev.* **2010**, *52*, 471–501. [CrossRef]
47. Udell, M.; Horn, C.; Zadeh, R.; Boyd, S. Generalized low rank models. *Found. Trends Mach. Learn.* **2016**, *9*, 1–118. [CrossRef]
48. Burer, S.; Monteiro, R.D. Local minima and convergence in low-rank semidefinite programming. *Math. Program.* **2005**, *103*, 427–444. [CrossRef]
49. Pataki, G. On the rank of extreme matrices in semidefinite programs and the multiplicity of optimal eigenvalues. *Math. Oper. Res.* **1998**, *23*, 339–358. [CrossRef]
50. Barvinok, A.I. Problems of distance geometry and convex properties of quadratic maps. *Discret. Comput. Geom.* **1995**, *13*, 189–202. [CrossRef]
51. Candes, E.J.; Eldar, Y.C.; Strohmer, T.; Voroninski, V. Phase retrieval via matrix completion. *SIAM Rev.* **2015**, *57*, 225–251. [CrossRef]
52. Boyd, S.; Parikh, N.; Chu, E.; Peleato, B.; Eckstein, J. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.* **2011**, *3*, 1–122.
53. Glowinski, R.; Marroco, A. Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de Dirichlet non linéaires. *Rev. Fr. Autom. Inform. Rech. Oper. Anal. Numer.* **1975**, *9*, 41–76. [CrossRef]
54. Gabay, D.; Mercier, B. *A Dual Algorithm for the Solution of Non Linear Variational Problems via Finite Element Approximation*; Institut de Recherche d'Informatique et d'Automatique: Rocquencourt, France, 1975.
55. Eckstein, J.; Yao, W. Understanding the convergence of the alternating direction method of multipliers: Theoretical and computational perspectives. *Pac. J. Optim.* **2015**, *11*, 619–644.
56. Sun, R.; Luo, Z.Q.; Ye, Y. On the expected convergence of randomly permuted ADMM. *arXiv* **2015**, arXiv:1503.06387.
57. Yin, W. Three-Operator Splitting and its Optimization Applications. *Set-Valued Var. Anal.* **2017**, *25*, 829–858.

58. Goldstein, T.; O'Donoghue, B.; Setzer, S.; Baraniuk, R. Fast alternating direction optimization methods. *SIAM J. Imaging Sci.* **2014**, *7*, 1588–1623. [CrossRef]
59. Hong, M.; Luo, Z.Q.; Razaviyayn, M. Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems. *SIAM J. Optim.* **2016**, *26*, 337–364. [CrossRef]
60. Wang, Y.; Yin, W.; Zeng, J. Global convergence of ADMM in nonconvex nonsmooth optimization. *arXiv* **2015**, arXiv:1511.06324.
61. Li, G.; Pong, T.K. Global convergence of splitting methods for nonconvex composite optimization. *SIAM J. Optim.* **2015**, *25*, 2434–2460. [CrossRef]
62. Magnússon, S.; Weeraddana, P.C.; Rabbat, M.G.; Fischione, C. On the convergence of alternating direction Lagrangian methods for nonconvex structured optimization problems. *IEEE Trans. Control. Netw. Syst.* **2016**, *3*, 296–309. [CrossRef]
63. Liu, Q.; Shen, X.; Gu, Y. Linearized admm for non-convex non-smooth optimization with convergence analysis. *arXiv* **2017**, arXiv:1705.02502.
64. Lu, S.; Hong, M.; Wang, Z. A nonconvex splitting method for symmetric nonnegative matrix factorization: Convergence analysis and optimality. In Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA, USA, 5–9 March 2017.
65. Xu, Y.; Yin, W.; Wen, Z.; Zhang, Y. An alternating direction algorithm for matrix completion with nonnegative factors. *Front. Math. China* **2012**, *7*, 365–384. [CrossRef]
66. Jiang, B.; Ma, S.; Zhang, S. Alternating direction method of multipliers for real and complex polynomial optimization models. *Optimization* **2014**, *63*, 883–898. [CrossRef]
67. Huang, K.; Sidiropoulos, N.D. Consensus-ADMM for general quadratically constrained quadratic programming. *IEEE Trans. Signal Process.* **2016**, *64*, 5297–5310. [CrossRef]
68. Shen, Y.; Wen, Z.; Zhang, Y. Augmented Lagrangian alternating direction method for matrix separation based on low-rank factorization. *Optim. Methods Softw.* **2014**, *29*, 239–263. [CrossRef]
69. Rockafellar, R.T. Augmented Lagrange multiplier functions and duality in nonconvex programming. *SIAM J. Control* **1974**, *12*, 268–285. [CrossRef]
70. Clarke, F.H. *Optimization and Nonsmooth Analysis*; SIAM: Philadelphia, PA, USA, 1990; Volume 5.
71. Lions, P.L.; Mercier, B. Splitting algorithms for the sum of two nonlinear operators. *SIAM J. Numer. Anal.* **1979**, *16*, 964–979. [CrossRef]
72. Douglas, J.; Rachford, H.H. On the numerical solution of heat conduction problems in two and three space variables. *Trans. Am. Math. Soc.* **1956**, *82*, 421–439. [CrossRef]
73. Spingarn, J.E. Applications of the method of partial inverses to convex programming: decomposition. *Math. Program.* **1985**, *32*, 199–223. [CrossRef]
74. Eckstein, J.; Bertsekas, D.P. On the Douglas Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Math. Program.* **1992**, *55*, 293–318. [CrossRef]
75. Combettes, P.L.; Pesquet, J.C. A proximal decomposition method for solving convex variational inverse problems. *Inverse Probl.* **2008**, *24*, 065014. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.