

Hardness Results for Structured Linear Systems

Rasmus Kyng*
Yale University
rasmus.kyng@yale.edu

Peng Zhang†
Georgia Institute of Technology
pzhang60@gatech.edu

September 26, 2017

Abstract

We show that if the nearly-linear time solvers for Laplacian matrices and their generalizations can be extended to solve just slightly larger families of linear systems, then they can be used to quickly solve all systems of linear equations over the reals. This result can be viewed either positively or negatively: either we will develop nearly-linear time algorithms for solving all systems of linear equations over the reals, or progress on the families we can solve in nearly-linear time will soon halt.

*Supported by ONR Award N00014-16-1-2374.

†Partially supported by the NSF under Grant No. 1637566.

Contents

1	Introduction	2
1.1	Our Results	3
1.2	Approximately Solving Linear Systems and Normal Equations	4
1.3	Graph-Structured Block Matrices	5
1.4	Our Reduction: Discussion and an Example	7
2	Preliminaries	9
2.1	Approximately Solving A Linear System	9
2.2	Measuring the Difficulty of Solving a Linear System	11
2.3	Matrix Classes and Reductions Between Them	11
3	Main Results	12
3.1	Outline of Remaining Sections	13
4	Reducing Zero-Sum Power Two Linear Systems to Two-Commodity Linear Systems	13
4.1	Reduction Between Exact Solvers	17
4.2	Relationship Between Schur Complements	21
4.3	Approximate solvers	22
4.4	Bounding Condition Number of the New Matrix	24
4.5	Putting it All Together	30
5	$\mathcal{MC}_2$ Efficiently Reducible to $\mathcal{MC}_2^{>0}$	31
5.1	Construction	31
5.2	Bounding Condition Number of the New System in $\mathcal{MC}_2^{>0}$	36
5.3	Putting it All Together	37
6	Rounding and Scaling Weights to Integers	37
7	$\mathcal{G}$ Efficiently Reducible to $\mathcal{G}_{z,2}$	41
7.1	$\mathcal{G} \leq_f \mathcal{G}_z$	41
7.2	$\mathcal{G}_z \leq_f \mathcal{G}_{z,2}$	46
8	2D Trusses	52
9	Connections with Interior Point Methods	56
9.1	Brief Overview of Interior Point Methods	56
9.2	2-Commodity Flow	56
9.3	Isotropic Total Variation Minimization	63
10	A Natural Decision Problem: Vector in the Image of a Matrix	65
10.1	Why Not Solve the Exact Decision Problem?	65
A	Linear Algebra Background	67
B	Using Complexity Parameters	67

1 Introduction

We establish a dichotomy result for the families of linear equations that can be solved in nearly-linear time. If nearly-linear time solvers exist for a slight generalization of the families for which they are currently known, then nearly-linear time solvers exist for all linear systems over the reals.

This type of reduction is related to the successful research program of fine-grained complexity, such as the result [WW10] which showed that the existence of a “truly subcubic” time algorithm for All-Pairs Shortest Paths Problem is equivalent to the existence of “truly subcubic” time algorithm for a wide range of other problems. For any constant $a \geq 1$, our result establishes for 2-commodity matrices, and several other classes of graph structured linear systems, that we can solve a linear system in a matrix of this type with s nonzeros in time $\tilde{O}(s^a)$ if and only if we can solve linear systems in all matrices with polynomially bounded integer entries in time $\tilde{O}(s^a)$.

In the RealRAM model, given a matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ and a vector $\mathbf{c} \in \mathbb{R}^n$, we can solve the linear system $\mathbf{Ax} = \mathbf{c}$ in $O(n^\omega)$ time, where ω is the matrix multiplication constant, for which the best currently known bound is $\omega < 2.3727$ [Str69, Wil12]. Such a running time bound is cost prohibitive for the large sparse matrices often encountered in practice. Iterative methods [Saa03], first order methods [BV04], and matrix sketches [Woo14] can all be viewed as ways of obtaining significantly better performance in cases where the matrices have additional structure.

In contrast, when $\mathbf{A}$ is an $n \times n$ Laplacian matrix with m non-zeros, and polynomially bounded entries, the linear system $\mathbf{Ax} = \mathbf{c}$ can be solved approximately to ϵ -accuracy in $O((m+n) \log^{1/2+o(1)} n \log(1/\epsilon))$ time [ST14, CKM⁺14]. This result spurred a series of major developments in fast graph algorithms, sometimes referred to as “the Laplacian Paradigm” of designing graph algorithms [Ten10]. The asymptotically fastest known algorithms for Maximum Flow in directed unweighted graphs [Mad13, Mad16], Negative Weight Shortest Paths and Maximum Weight Matchings [CMSV17], Minimum Cost Flows and Lossy Generalized Flows [LS14, DS08] all rely on fast Laplacian linear system solvers.

The core idea of the Laplacian paradigm can be viewed as showing that the linear systems that arise from interior point algorithms, or second-order optimization methods, have graph structure, and can be preconditioned and solved using graph theoretic techniques. These techniques could potentially be extended to a range of other problems, provided fast solvers can be found for the corresponding linear systems. Here a natural generalization is in terms of the number of labels per vertex: graph Laplacians correspond to graph labeling problems where each vertex has one label, and these labels interact pairwise via edges. Multi-label variants of these exist in Markov random fields [SZS⁺08], image processing [WYYZ08], Euclidean embedding of graphs [CLM⁺16], data processing for cryo-electron microscopy [SS11, SS12, ZS14], phase retrieval [ABFM14, MTW14], and many image processing problems (e.g. [OSB15, ANKKS⁺12]). Furthermore, linear systems with multiple labels per vertex arise when solving multi-commodity flow problems using primal-dual methods. Linear systems related to multi-variate labelings of graphs have been formulated as the quadratically-coupled flow problem [KMP12] and Graph-Structured Block Matrices [Spi16]. They also occur naturally in linear elasticity problems for simulating the effect of forces on truss systems [DS07].

Due to these connections, a central question in the Laplacian paradigm of designing graph algorithms is whether all Graph-Structured Block Matrices can be solved in (approximately) nearly-linear time. Even obtaining subquadratic running time would be constitute significant progress. There has been some optimism in this direction due to the existence of faster solvers for special cases: nearly-linear time solvers for Connection Laplacians [KLP⁺16], 1-Laplacians of collapsible 3-D simplicial complexes [CFM⁺14], and algorithms with runtime about $n^{5/4}$ for 2D planar truss stiffness matrices [DS07]. Furthermore, there exists a variety of faster algorithms for approximating multi-

commodity flows to $(1+\epsilon)$ accuracy in time that scales as $\text{poly}(\epsilon^{-1})$ [GK98, Mad10, Fle00, LMP⁺91], even obtaining nearly-linear running times when the graph is undirected [She13, KLOS14, Pen16].

The subquadratic variants of these routines also interact naturally with tools that in turn utilize Laplacian solvers [KMP12, LMP⁺91]. These existing tight algorithmic connections, as well as the solver for Connection Laplacians, and the fact that combinatorial preconditioners partly originated from speeding up interior point methods through preconditioning Hessians [Vai89], together provide ample reason to hope that one could develop nearly-linear time solvers for linear systems related to multicommodity flows. Any algorithm that solves such systems to high accuracy in $m^{1+\alpha}$ time would in turn imply multicommodity flow algorithms that run in about $n^{1/2}m^{1+\alpha}$ time [LS14], while the current best running times are about $n^{2.5}$ [LS15].

Unfortunately, we show that if linear systems in general 2D truss stiffness matrices or 2-commodity Laplacians can be solved approximately in nearly-linear time, then all linear systems in matrices with polynomially bounded integer entries can be solved in nearly-linear time. In fact, we show in a strong sense that any progress made in developing solvers for these classes of matrices will translate directly into similarly fast solvers for all matrices with polynomially bounded integer entries. Thus developing faster algorithms for these systems will be as difficult as solving *all* linear systems faster.

Since linear system solvers used inside Interior Point Methods play a central role in the Laplacian paradigm for designing high-accuracy algorithms, this may suggest that in the high-accuracy regime the paradigm will not extend to most problems that require multiple labels/variables per edge or vertex. Alternatively, an algorithmic optimist might view our result as a road-map for solving all linear systems via reductions to fast linear system solvers for Graph-Structured Block Matrices.

1.1 Our Results

Fast linear system solvers for Laplacians, Connection Laplacians, Directed Laplacians, and 2D Planar Truss Stiffness matrices are all based on iterative methods and only produce approximate solutions. The running time for these solvers scales logarithmically with the error parameter ϵ , i.e. as $\log(1/\epsilon)$. Similarly, the running time for iterative methods usually depends on the condition number of the matrix, but for state-of-the-art solvers for Laplacians, Connection Laplacians, and Directed Laplacians, the dependence is logarithmic. Consequently, a central open question is whether fast approximate solvers exist for other structured linear systems, with running times that depend logarithmically on the condition number and the accuracy parameter.

Integral linear systems are reducible to Graph-Structured Block Matrices. Our reductions show that if fast approximate linear system solvers exist for multi-commodity Laplacians, 2D Truss Stiffness, or Total Variation (TV) Matrices, then fast approximate linear system solvers exist for any matrix, in the very general sense of minimizing $\min_{\mathbf{x}} \|\mathbf{Ax} - \mathbf{c}\|_2^2$. Thus our result also applies to singular matrices, where we solve the pseudo-inverse problem to high accuracy. Theorem 1.1 gives an informal statement of our main result. The result is stated formally in Section 3 as Theorem 3.2.

Theorem 1.1 (Hardness for Graph-Structured Linear Systems (Informal)). *We consider three types of Graph-Structured Block Matrices: Multi-commodity Laplacians, Truss Stiffness Matrices, and Total Variation Matrices. Suppose that for one or more of these classes, the linear system $\mathbf{Ax} = \mathbf{c}$ in a matrix $\mathbf{A}$ with s non-zeros can be solved in time $\tilde{O}(s^a)$, for some constant $a \geq 1$, with the running time having logarithmic dependence on condition number and accuracy¹. Then*

¹This is the kind of running time guarantee established for Laplacians, Directed Laplacians, Connection Laplacians, and bounded-weight planar 2D Truss Stiffness matrices.

linear systems in all matrices with polynomially bounded integer entries and condition number can be solved to high accuracy in time $\tilde{O}(s^a)$, where again s is the number of non-zero entries of the matrix.

Our results can easily be adapted to show that if fast exact linear system solvers exist for multi-commodity Laplacians, then exact solvers exist for all non-singular integer matrices. However, this is of less interest since there is less evidence that would suggest we should expect fast exact solvers to exist.

The notion of approximation used throughout this paper is the same as that used in the Laplacian solver literature (see Section 2.1). To further justify the notion of approximate solutions to linear systems, we show that it lets us solve a natural decision problem for linear systems:

We show that deciding if a vector is approximately in the image of a matrix can be reduced to approximately solving linear systems. We show this in Section 10. We also show that the exact image decision problem requires working with exponentially small numbers, even when the input has polynomially bounded integral entries and condition number. This means that in fixed-point arithmetic, we can only hope to solve an approximate version of the problem. The problem of approximately solving general linear systems can be reduced the problem of approximately solving Graph-Structured Block Matrix linear systems. Together, these statements imply that we can also reduce the problem of deciding whether a vector is approximately in the image of a general matrix to the problem of approximately solving Graph-Structured Block Matrix linear systems.

We establish surprising separations between many problems known to have fast solutions and problems that are as hard solving general linear systems. Our results trace out several interesting dichotomies: restricted cases of 2D truss stiffness matrices have fast solvers, but fast solvers for all 2D truss stiffness matrices would imply equally fast solvers for all linear systems. TV matrices can be solved quickly in the anisotropic case, but in the isotropic case imply solvers for all linear systems. Fast algorithms exist for multi-commodity problems in the low accuracy regime, but existing approaches for the high accuracy regime seem to require fast solvers for multi-commodity linear systems, which again would imply fast solvers for all linear systems.

Our reductions only require the simplest cases of the classes we consider: 2-Commodity Laplacians are sufficient, as are (non-planar) 2D Truss Stiffness matrices, and Total Variation Matrices with 2-by-2 interactions. Linear systems of these three classes have many applications, and faster solvers for these would be useful in all applications. Trusses have been studied as the canonical multi-variate problem, involving definitions such as Fretsaw extensions [ST08] and factor widths [BCPT05], and fast linear system solvers exist for the planar 2D case with bounded weights [DS07]. Total Variation Matrices are widely used in image denoising [CS05]. The anisotropic version can be solved using nearly-linear time linear system solvers [KMT11], while the isotropic version has often been studied using linear systems for which fast solvers are not known [GY04, WR07, CMMP13a]. Multi-commodity flow problems have been the subject of extensive study, with significant progress on algorithms with low accuracy [GK98, Mad10, Fle00, LMP⁺91, She13, KLOS14, Pen16], while high accuracy approaches use slow general linear system solvers.

1.2 Approximately Solving Linear Systems and Normal Equations

The simplest notion of solving a linear system $\mathbf{A}\mathbf{x} = \mathbf{c}$, is to seek an $\mathbf{x}$ s.t. the equations are exactly satisfied. More generally, if the system is not guaranteed to have a solution, we can ask for an $\mathbf{x}$ which minimizes $\|\mathbf{A}\mathbf{x} - \mathbf{c}\|_2^2$. An $\mathbf{x}$ which minimizes this always exists. In general, it may

not be unique. Finding an $\mathbf{x}$ which minimizes $\|\mathbf{Ax} - \mathbf{c}\|_2^2$ is equivalent to solving the linear system $\mathbf{A}^\top \mathbf{Ax} = \mathbf{A}^\top \mathbf{c}$, which is referred to as the normal equation for the linear system $\mathbf{Ax} = \mathbf{c}$ (see [TBI97]). The problem of solving the normal equations (or equivalently, minimizing $\|\mathbf{Ax} - \mathbf{c}\|_2^2$), is a generalization of the problem of linear system solving, since the approach works when $\mathbf{A}$ is non-singular, while also giving meaningful results when $\mathbf{A}$ is singular. The normal equation problem can also be understood in terms of the Moore-Penrose pseudo-inverse of a matrix $\mathbf{M}$, which is denoted $\mathbf{M}^\dagger$ as $\mathbf{x} = (\mathbf{A}^\top \mathbf{A})^\dagger \mathbf{A}^\top \mathbf{c}$ is a solution to the normal equations. Taking the view of linear system solving as minimizing $\|\mathbf{Ax} - \mathbf{c}\|_2^2$ also gives sensible ways to define an approximate solution to a linear system: It is an $\mathbf{x}$ that ensures $\|\mathbf{Ax} - \mathbf{c}\|_2^2$ is close to $\min_{\mathbf{x}} \|\mathbf{Ax} - \mathbf{c}\|_2^2$. In Section 2, we formally define several notions of approximate solutions to linear systems that we will use throughout the paper.

An important special case of linear systems is when the matrix of coefficients of the system is positive semi-definite. Since $\mathbf{A}^\top \mathbf{A}$ is always positive semi-definite, solving the normal equations for a linear system falls into this case. Linear systems over positive semi-definite matrices can be solved (approximately) by approaches known as iterative methods, which often lead to much faster algorithms than the approaches used for general linear systems. Iterative methods inherently produce approximate solutions².

1.3 Graph-Structured Block Matrices

Graph-Structured Block Matrices are a type of linear system that arise in many applications. Laplacian matrices and Connection Laplacians both fall in this category.

Suppose we have a collection of n disjoint sets of variables $X_1, \dots, X_n$, with each set having the same size, $|X_i| = d$. Let $\mathbf{x}^i$ denote the vector³ of variables in X_i , and consider an equation of the form $\mathbf{S}\mathbf{x}^i - \mathbf{T}\mathbf{x}^j = \mathbf{0}$, where $\mathbf{S}$ and $\mathbf{T}$ are both $r \times d$ matrices. Now we form a linear system $\mathbf{B}\mathbf{x} = \mathbf{0}$ by stacking m equations of the form given above as the rows of the system. Note that, very importantly, we allow a different choice of $\mathbf{S}$ and $\mathbf{T}$ for every pair of i and j . This matrix $\mathbf{B} \in \mathbb{R}^{mr \times nd}$ we refer to as a Incidence-Structured Block Matrix (ISBM), while we refer to $\mathbf{B}^\top \mathbf{B}$ as a Graph-Structured Block Matrix (GSBM). Note that $\mathbf{B}$ is not usually PSD, but $\mathbf{B}^\top \mathbf{B}$ is. The number of non-zeros in $\mathbf{B}^\top \mathbf{B}$ is $O(md^2)$. GSBMs come up in many applications, where we typically want to solve a linear system in the normal equations of $\mathbf{B}$.

Laplacian matrices are GSBMs where $d = 1$ and $\mathbf{S} = \mathbf{T} = w$, where w is a real number, and we allow different w for each pair of i and j . The corresponding ISBM for Laplacians is called an edge-vertex incidence matrix. Connection Laplacians are GSBMs where $d = O(1)$ and $\mathbf{S} = \mathbf{T}^\top = w\mathbf{Q}$, for some rotation matrix $\mathbf{Q}$ and a real number w . Again, we allow a different rotation matrix and scaling for every edge. For both Laplacians and Connection Laplacians, there exist linear system solvers that run in time $O(m \text{polylog}(n, \epsilon^{-1}))$ and produce ϵ approximate solutions to the corresponding normal equations.

We now introduce several classes of ISBMs and their associated GSBMs. Our Theorem 3.2 shows that fast linear system solvers for any of these classes would imply fast linear system solvers for all matrices with polynomially bounded entries and condition number.

Definition 1.2 (2-commodity incidence matrix). A *2-commodity incidence matrix* is an ISBM where $d = 2$ and $r = 1$, and $\mathbf{S} = \mathbf{T}$, and we allow three types of $\mathbf{S}$: $\mathbf{S} = w \begin{pmatrix} 1 & 0 \end{pmatrix}$, $\mathbf{S} = w \begin{pmatrix} 0 & 1 \end{pmatrix}$

²A seeming counterexample to this is Conjugate Gradient which is an iterative method that produces exact solutions in the RealRAM model. But it requires extremely high precision calculations to exhibit this behaviour in finite precision arithmetic, and so Conjugate Gradient is also best understood as an approximate method.

³We use superscripts to index a sequence of vectors or matrices, and we use subscripts to denote entries of a vector or matrix, see Section 2.

and $\mathbf{S} = w \begin{pmatrix} 1 & -1 \\ 0 & 0 \end{pmatrix}$, where in each case w is a real number which may depend on the pair i and j . We denote the set of all 2-commodity incidence matrices by $\mathcal{MC}_2$. The corresponding GSBM is called a 2-commodity Laplacian. The ISBM definition is equivalent to requiring the GSBM to have the form

$$\mathbf{L}^1 \otimes \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} + \mathbf{L}^2 \otimes \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} + \mathbf{L}^{1+2} \otimes \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$$

where $\otimes$ is the tensor product and $\mathbf{L}^1$, $\mathbf{L}^2$, and $\mathbf{L}^{1+2}$ are all Laplacian matrices.

We adopt a convention that the first variable in a set X_i is labelled $\mathbf{u}_i$ and the second is labelled $\mathbf{v}_i$. Using this convention, given a 2-commodity incidence matrix $\mathbf{B}$, the equation $\mathbf{B}\mathbf{x} = \mathbf{0}$ must consist of scalings of the following three types of equations: $\mathbf{u}_i - \mathbf{u}_j = 0$, $\mathbf{v}_i - \mathbf{v}_j = 0$, and $\mathbf{u}_i - \mathbf{v}_i - (\mathbf{u}_j - \mathbf{v}_j) = 0$.

Definition 1.3 (Strict 2-commodity incidence matrix). A *strict 2-commodity incidence matrix* is a 2-commodity incidence matrix where the corresponding 2-commodity Laplacian has the property that $\mathbf{L}^1$, $\mathbf{L}^2$, and $\mathbf{L}^{1+2}$ all have the same non-zero pattern. We denote the set of all strict 2-commodity incidence matrices by $\mathcal{MC}_2^{>0}$. We denote the set of all strict 2-commodity incidence matrices with *integer entries* by $\mathcal{MC}_{2,\mathbb{Z}}^{>0}$.

Linear systems in $\mathcal{MC}_2^{>0}$ are exactly the systems that one has to solve to when solving 2-commodity problems using Interior Point Methods (IPMs). For readers unfamiliar with 2-commodity problems or IPMs, we provide a brief explanation of why this is the case in Section 9. The $\mathcal{MC}_2^{>0}$ is more restrictive than $\mathcal{MC}_2$, and $\mathcal{MC}_{2,\mathbb{Z}}^{>0}$ in turn is even more restrictive. One could hope that fast linear system solvers exist for $\mathcal{MC}_2^{>0}$ or $\mathcal{MC}_{2,\mathbb{Z}}^{>0}$, even if they do not exist for $\mathcal{MC}_2$. However, our reductions show that even getting a fast approximate solver for $\mathcal{MC}_{2,\mathbb{Z}}^{>0}$ with polynomially bounded entries and condition number will lead to a fast solver for all matrices with polynomially bounded entries and condition number.

The next class we consider is 2D Truss Stiffness Matrices. They have been studied extensively in the numerical linear algebra community [ST08, BCPT05]. For *Planar* 2D Trusses with some bounds on ranges of edges, Daitch and Spielman obtained linear system solvers that run in time $\tilde{O}(n^{5/4} \log(1/\epsilon))$.

Definition 1.4 (2D Truss Incidence Matrices). Let $G = (V, E)$ be a graph whose vertices are n points in 2-dimension: $\mathbf{s}^1, \dots, \mathbf{s}^n \in \mathbb{R}^2$. Consider $X_1, \dots, X_n$ where $d = 2$. A *2D Truss Incidence Matrix* is an ISBM where $d = 2$ and $r = 1$, and for each i and j , we have $\mathbf{S} = \mathbf{T}$ and $\mathbf{S} = w(\mathbf{s}^i - \mathbf{s}^j)^\top$, and w is a real number that may depend on the pair i and j , but $\mathbf{s}^i$ depends only on i and vice versa for $\mathbf{s}^j$. We denote the class of all 2D Truss Incidence Matrices by $\mathcal{T}_2$.

Another important class of matrices is Total Variation Matrices (TV matrices). TV matrices come from Interior Point Methods for solving total variation minimization problem in image, see for example [GY05] and [CMMP13b]. Not all TV matrices are GSBMs, but many GSBMs can be expressed as TV matrices.

Definition 1.5 (TV matrix and 2-TV Incidence Matrices). Let $E_1 \cup \dots \cup E_s$ be a partition of the edge set of a graph. For each $1 \leq i \leq s$, let $\mathbf{B}^i$ be the edge-vertex incidence matrix of E_i , $\mathbf{W}^i$ be a diagonal matrix of edge weights, and $\mathbf{r}^i$ be a vector satisfying $\mathbf{W}^i \succcurlyeq \mathbf{r}^i(\mathbf{r}^i)^\top$. Given these objects, the associated *total variation matrix* (TV matrix) is a matrix $\mathbf{M}$ defined as

$$\mathbf{M} = \sum_{1 \leq i \leq s} (\mathbf{B}^i)^\top \left(\mathbf{W}^i - \mathbf{r}^i(\mathbf{r}^i)^\top \right) \mathbf{B}^i.$$

A *2-TV Incidence Matrix* is defined as any ISBM whose corresponding GSBM is a TV matrix with $\mathbf{W}^i \in \mathbb{R}^{2 \times 2}$ and $\mathbf{r}^i \in \mathbb{R}^2$. We denote the class of all 2-TV incidence matrices by $\mathcal{V}_2$.

1.4 Our Reduction: Discussion and an Example

In this section we give a brief sketch of the ideas behind our reduction from general linear systems, over matrices in $\mathcal{G}$, to multi-commodity linear systems, over matrices in $\mathcal{MC}_2$, and we demonstrate the most important transformation through an example.

The starting point for our approach is the folklore idea that any linear system can be written as a factor-width 3 system by introducing a small number of extra variables. Using a set of multi-commodity constraints, we are able to express one particular factor-width 3 equation, namely $2x'' = x + x'$. After a sequence of preprocessing steps, we are then able to efficiently express arbitrary linear systems over integer matrices using constraints of this form. A number of further issues arise when the initial matrix does not have full column rank, requiring careful weighting of the constraints we introduce.

Given a matrix $\mathbf{A}$ with polynomially bounded integer entries and condition number, we reduce the linear system $\mathbf{A}\mathbf{x} = \mathbf{c}$ to a linear system $\mathbf{B}\mathbf{y} = \mathbf{d}$, where $\mathbf{B}$ is a strict multi-commodity edge-vertex incidence matrix with integer entries (i.e. in $\mathcal{MC}_{2,\mathbb{Z}}^{>0}$), with polynomially bounded entries and condition number. More precisely, we reduce $\mathbf{A}^\top \mathbf{A}\mathbf{x} = \mathbf{A}^\top \mathbf{c}$ to $\mathbf{B}^\top \mathbf{B}\mathbf{y} = \mathbf{B}^\top \mathbf{d}$. These systems always have a solution. We show that we can find an ϵ -approximate solution to the linear system $\mathbf{A}^\top \mathbf{A}\mathbf{x} = \mathbf{A}^\top \mathbf{c}$ by a simple mapping on any $\mathbf{y}$ that ϵ' -approximately solves the linear system $\mathbf{B}^\top \mathbf{B}\mathbf{y} = \mathbf{B}^\top \mathbf{d}$, where ϵ' is only polynomially smaller than ϵ . If $\mathbf{A}$ has s non-zero entries and the maximum absolute value of an entry in $\mathbf{A}$ is U , then $\mathbf{B}$ will have $O(s \log(sU))$ non-zero entries and our algorithm computes the reduction in time $O(s \log(sU))$. Note that $\mathbf{B}^\top \mathbf{B}$ has $r = O(s \log(sU))$ non-zeros, because every row of $\mathbf{B}$ has $O(1)$ entries. All together, this means that getting a solver for $\mathbf{B}^\top \mathbf{B}\mathbf{x} = \mathbf{B}^\top \mathbf{d}$ with running time $\tilde{O}(r^a \log(1/\epsilon))$ will give a solver for $\mathbf{A}$ with $\tilde{O}(s^a \log(1/\epsilon))$ running time.

We achieve this through a chain of reductions. Each reduction produces a new matrix and vector, as well as a new error parameter giving the accuracy required in the new system to achieve the accuracy desired in the original system.

1. We get a new linear system $\mathbf{A}^{\mathbb{Z},2} \mathbf{x}^{\mathbb{Z},2} = \mathbf{c}^{\mathbb{Z},2}$ where $\mathbf{A}^{\mathbb{Z},2}$ has integer entries, and the entries of each row of $\mathbf{A}^{\mathbb{Z},2}$ sum to zero, i.e. $\mathbf{A}^{\mathbb{Z},2} \mathbf{1} = \mathbf{0}$, and finally in every row the sum of the positive coefficients is a power of two.
2. $\mathbf{A}^{\mathbb{Z},2} \mathbf{x}^{\mathbb{Z},2} = \mathbf{c}^{\mathbb{Z},2}$ is then transformed to $\mathbf{B}\mathbf{y} = \mathbf{d}$, where $\mathbf{B}$ is a 2-commodity edge-vertex incidence matrix.
3. $\mathbf{B}\mathbf{y} = \mathbf{d}$ is then transformed to $\mathbf{B}^{>0} \mathbf{y} = \mathbf{d}^{>0}$, where $\mathbf{B}^{>0}$ is a strict 2-commodity edge-vertex incidence matrix.
4. $\mathbf{B}^{>0} \mathbf{y} = \mathbf{d}^{>0}$ is then transformed to $\mathbf{B}^{>0,\mathbb{Z}} \mathbf{y} = \mathbf{d}^{>0,\mathbb{Z}}$, where $\mathbf{B}^{>0,\mathbb{Z}}$ is a 2-commodity edge-vertex incidence matrix with integer entries.

We will demonstrate step 2, the main transformation, by example. When the context is clear, we drop the superscripts of matrices for simplicity. The reduction handles each row (i.e. equation) of the linear system independently, so we focus on the reduction for a single row.

Consider a linear system $\mathbf{A}\mathbf{x} = \mathbf{c}$, and let us pick a single row (i.e. equation) $\mathbf{A}_i \mathbf{x} = \mathbf{c}_i$ ⁴. We will repeatedly pick pairs of existing variables of $\mathbf{x}$, say x and x' , based on their current coefficients in $\mathbf{A}_i \mathbf{x} = \mathbf{c}_i$, and modify the row by adding $C(2x'' - (x + x'))$ to the left hand side where x'' is a new variable and C is a real number we pick. As we will see in a moment, we can use this pair-and-replace operation to simplify the row until it eventually becomes a 2-commodity equation. At

⁴We use $\mathbf{A}_i$ to denote the i th row of $\mathbf{A}$, and $\mathbf{c}_i$ to denote the i th entry of $\mathbf{c}$, see Section 2.

the same time as we modify $\mathbf{A}_i$, we also store an auxiliary equation $C(x + x' - 2x'') = 0$. Suppose initially that $\mathbf{A}_i \mathbf{x} = \mathbf{c}_i$ is satisfied. After this modification of $\mathbf{A}_i \mathbf{x} = \mathbf{c}_i$, if the auxiliary equation is satisfied, $\mathbf{A}_i \mathbf{x} = \mathbf{c}_i$ is still satisfied by the same values of x and x' . Crucially, we can express the auxiliary equation $C(x + x' - 2x'') = 0$ by a set of ten 2-commodity equations, i.e. a “2-commodity gadget” for this equation. Our final output matrix *will not* contain the equation $C(x + x' - 2x'') = 0$ as a row, but will instead contain 10 rows of 2-commodity equations from our gadget construction. Eventually, our pair-and-replace scheme will also transform the row $\mathbf{A}_i \mathbf{x} = \mathbf{c}_i$ into a 2-commodity equation on just two variables.

Next, we need to understand how the pair-and-replace scheme makes progress. The pairing handles the positive and the negative coefficients of $\mathbf{A}_i$ separately, and eventually ensures that $\mathbf{A}_i \mathbf{x} = \mathbf{c}_i$ has only a single positive and a single negative coefficient in the modified row $\mathbf{A}_i \mathbf{x} = \mathbf{c}_i$, in particular it is of the form $ax - ax' = \mathbf{c}_i$ for two variables x and x' that appear in the modified vector of variables $\mathbf{x}$, i.e. it is a 2-commodity equation.

To understand the pairing scheme, it is helpful to think about the entries of $\mathbf{A}$ written using binary (ignoring the sign of the entry). The pairing scheme proceeds in a sequence of rounds: In the first round we pair variables whose 1st (smallest) bit is 1. There must be an even number of variables with smallest bit 1, as the sum of the positive (and respectively negative) coefficients is a power of 2. We then replace the terms corresponding to the 1st bit of the pair with a new single variable with a coefficient of 2. After the first round, every coefficient has zero in the 1st bit. In the next round, we pair variables whose 2nd bit is 1, and replace the terms corresponding to the 2nd bit of the pair with a new single variable with a coefficient of 4, and so on. Because the positive coefficients sum to a power of two, we are able to guarantee that pairing is always possible. It is not too hard to show that we do not create a large number of new variables or equations using this scheme.

For example, let us consider an equation

$$\begin{array}{ll}
3\mathbf{x}_1 + 5\mathbf{x}_2 + 4\mathbf{x}_3 + 4\mathbf{x}_4 - 16\mathbf{x}_5 = 1 & \\
\downarrow & \text{Replace } \mathbf{x}_1 + \mathbf{x}_2 \text{ by } 2\mathbf{x}_6. \\
& \text{Add auxiliary equation } \mathbf{x}_1 + \mathbf{x}_2 - 2\mathbf{x}_6 = 0. \\
2(\mathbf{x}_1 + 2\mathbf{x}_2 + \mathbf{x}_6 + 2\mathbf{x}_3 + 2\mathbf{x}_4) - 16\mathbf{x}_5 = 1 & \\
\downarrow & \text{Replace } 2(\mathbf{x}_1 + \mathbf{x}_6) \text{ by } 4\mathbf{x}_7. \\
& \text{Add auxiliary equation } 2(\mathbf{x}_1 + \mathbf{x}_6 - 2\mathbf{x}_7) = 0. \\
4(\mathbf{x}_2 + \mathbf{x}_7 + \mathbf{x}_3 + \mathbf{x}_4) - 16\mathbf{x}_5 = 1 & \\
\downarrow & \text{Replace } 4(\mathbf{x}_2 + \mathbf{x}_7) \text{ by } 8\mathbf{x}_8, \\
& \text{and } 4(\mathbf{x}_3 + \mathbf{x}_4) \text{ by } 8\mathbf{x}_9. \\
& \text{Add auxiliary equations } 4(\mathbf{x}_2 + \mathbf{x}_7 - 2\mathbf{x}_8) = 0, \\
& \text{and } 4(\mathbf{x}_3 + \mathbf{x}_4 - 2\mathbf{x}_9) = 0. \\
8(\mathbf{x}_8 + \mathbf{x}_9) - 16\mathbf{x}_5 = 1 & \\
\downarrow & \text{Replace } 8(\mathbf{x}_8 + \mathbf{x}_9) \text{ by } 16\mathbf{x}_{10}. \\
& \text{Add auxiliary equation } 8(\mathbf{x}_8 + \mathbf{x}_9 - 2\mathbf{x}_{10}) = 0. \\
16\mathbf{x}_{10} - 16\mathbf{x}_5 = 1 &
\end{array}$$

In this way, we process $\mathbf{A}\mathbf{x} = \mathbf{c}$ to produce a new set of equations $\mathbf{B}\mathbf{y} = \mathbf{d}$ where $\mathbf{B}$ is a

2-commodity matrix. If $\mathbf{A}\mathbf{x} = \mathbf{c}$ has an exact solution, this solution can be obtained directly from an exact solution to $\mathbf{B}\mathbf{y} = \mathbf{d}$. We also show that an approximate solution to $\mathbf{B}\mathbf{y} = \mathbf{d}$ leads to an approximate solution for $\mathbf{A}\mathbf{x} = \mathbf{c}$, and we show that $\mathbf{B}$ does not have much larger entries or condition number than $\mathbf{A}$.

The situation is more difficult when $\mathbf{A}\mathbf{x} = \mathbf{c}$ does not have a solution and we want to obtain an approximate minimizer $\arg\min_{\mathbf{x} \in \mathbb{R}^n} \|\mathbf{A}\mathbf{x} - \mathbf{c}\|_2^2$ from an approximate solution to $\arg\min_{\mathbf{y} \in \mathbb{R}^{n'}} \|\mathbf{B}\mathbf{y} - \mathbf{d}\|_2^2$. This corresponds to approximately applying the Moore-Penrose pseudo-inverse of $\mathbf{A}$ to $\mathbf{c}$. We deal with the issues that arise here using a carefully chosen scaling of each auxiliary constraint to ensure a strong relationship between different solutions.

In order to switch from a linear system in a general 2-commodity matrix to a linear system in a *strict* 2-commodity matrix, we need to reason very carefully about the changes to the null space that this transformation inherently produces. By choosing sufficiently small weights, we are nonetheless able to establish a strong relationship between the normal equation solutions despite the change to the null space.

2 Preliminaries

We use subscripts to denote entries of a matrix or a vector: let $\mathbf{A}_i$ denote the i th row of matrix $\mathbf{A}$ and $\mathbf{A}_{ij}$ denote the (i, j) th entry of $\mathbf{A}$; let $\mathbf{x}_i$ denote the i th entry of vector $\mathbf{x}$ and $\mathbf{x}_{i:j}$ ($i < j$) denote the vector of entries $\mathbf{x}_i, \mathbf{x}_{i+1}, \dots, \mathbf{x}_j$. We use superscripts to index a sequence of matrices or vectors, e.g., $\mathbf{A}^1, \mathbf{A}^2, \dots$, and $\mathbf{x}^1, \mathbf{x}^2, \dots$, except when some other meaning is clearly stated.

We use $\mathbf{A}^\dagger$ to denote the Moore-Penrose pseudo-inverse of a matrix $\mathbf{A}$. We use $\text{im}(\mathbf{A})$ to denote the image of a matrix $\mathbf{A}$. We use $\|\cdot\|_2$ to denote the Euclidean norm on vectors and the spectral norm on matrices. When $\mathbf{M}$ is an $n \times n$ positive semidefinite matrix, we define a norm on vectors $\mathbf{x} \in \mathbb{R}^n$ by $\|\mathbf{x}\|_{\mathbf{M}} \stackrel{\text{def}}{=} \sqrt{\mathbf{x}^\top \mathbf{M} \mathbf{x}}$. We let $\text{nnz}(\mathbf{A})$ denote the number of non-zero entries in a matrix $\mathbf{A}$. We define $\|\mathbf{A}\|_{\max} = \max_{i,j} |\mathbf{A}_{ij}|$, $\|\mathbf{A}\|_1 = \max_j \sum_i |\mathbf{A}_{ij}|$ and $\|\mathbf{A}\|_\infty = \max_i \sum_j |\mathbf{A}_{ij}|$. We let $\min_+(\mathbf{A}) = \min_{i,j \text{ s.t. } \mathbf{A}_{ij} \neq 0} |\mathbf{A}_{ij}|$. Given a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ and a vector $\mathbf{c} \in \mathbb{R}^m$ for some m, n , we call the tuple $(\mathbf{A}, \mathbf{c})$ a linear system. Given matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, let $\Pi_{\mathbf{A}} \stackrel{\text{def}}{=} \mathbf{A}(\mathbf{A}\mathbf{A}^\top)^\dagger \mathbf{A}^\top$, i.e. the orthogonal projection onto $\text{im}(\mathbf{A})$. Note that $\Pi_{\mathbf{A}} = \Pi_{\mathbf{A}}^\top$ and $\Pi_{\mathbf{A}} = \Pi_{\mathbf{A}}^2$.

2.1 Approximately Solving A Linear System

In this section we formally define the notions of approximate solutions to linear systems that we work with throughout this paper.

Definition 2.1 (Linear System Approximation Problem, LSA). Given linear system $(\mathbf{A}, \mathbf{c})$, where $\mathbf{A} \in \mathbb{R}^{m \times n}$, and $\mathbf{c} \in \mathbb{R}^m$, and given a scalar $0 \leq \epsilon \leq 1$, we refer to the LSA problem for the triple $(\mathbf{A}, \mathbf{c}, \epsilon)$ as the problem of finding $\mathbf{x} \in \mathbb{R}^n$ s.t.

$$\|\mathbf{A}\mathbf{x} - \Pi_{\mathbf{A}}\mathbf{c}\|_2 \leq \epsilon \|\Pi_{\mathbf{A}}\mathbf{c}\|_2,$$

and we say that such an $\mathbf{x}$ is a solution to the LSA instance $(\mathbf{A}, \mathbf{c}, \epsilon)$.

This definition of a LSA instance and solution has several advantages: when $\text{im}(\mathbf{A}) = \mathbb{R}^m$, we get $\Pi_{\mathbf{A}} = \mathbf{I}$, and it reduces to the natural condition $\|\mathbf{A}\mathbf{x} - \mathbf{c}\|_2 \leq \epsilon \|\mathbf{c}\|_2$, which because $\text{im}(\mathbf{A}) = \mathbb{R}^m$, can be satisfied for any ϵ , and for $\epsilon = 0$ tells us that $\mathbf{A}\mathbf{x} = \mathbf{c}$.

When $\text{im}(\mathbf{A})$ does not include all of $\mathbb{R}^m$, the vector $\Pi_{\mathbf{A}}\mathbf{c}$ is exactly the projection of $\mathbf{c}$ onto $\text{im}(\mathbf{A})$, and so a solution can still be obtained for any ϵ . Further, as $(\mathbf{I} - \Pi_{\mathbf{A}})\mathbf{c}$ is orthogonal to

$\Pi_A \mathbf{c}$ and $\mathbf{Ax}$, it follows that

$$\|\mathbf{Ax} - \mathbf{c}\|_2^2 = \|(I - \Pi_A)\mathbf{c}\|_2^2 + \|\mathbf{Ax} - \Pi_A \mathbf{c}\|_2^2.$$

Thus, when $\mathbf{x}$ is a solution to the LSA instance $(\mathbf{A}, \mathbf{c}, \epsilon)$, then $\mathbf{x}$ also gives an $\epsilon^2 \|\Pi_A \mathbf{c}\|_2^2$ additive approximation to

$$\min_{\mathbf{x} \in \mathbb{R}^n} \|\mathbf{Ax} - \mathbf{c}\|_2^2 = \|(I - \Pi_A)\mathbf{c}\|_2^2. \quad (1)$$

Similarly, an $\mathbf{x}$ which gives an additive $\epsilon^2 \|\Pi_A \mathbf{c}\|_2^2$ approximation to Problem (1) is always a solution to the LSA instance $(\mathbf{A}, \mathbf{c}, \epsilon)$. These observations prove the following (well-known) fact:

Fact 2.2. *Let $\mathbf{x}^* \in \arg \min_{\mathbf{x} \in \mathbb{R}^n} \|\mathbf{Ax} - \mathbf{c}\|_2^2$, then for every $\mathbf{x}$,*

$$\|\mathbf{Ax} - \mathbf{c}\|_2^2 \leq \|\mathbf{Ax}^* - \mathbf{c}\|_2^2 + \epsilon^2 \|\Pi_A \mathbf{c}\|_2^2$$

if and only if $\mathbf{x}$ is a solution to the LSA instance $(\mathbf{A}, \mathbf{c}, \epsilon)$.

When the linear system $\mathbf{Ax} = \mathbf{c}$ does not have a solution, a natural notion of solution is any minimizer of Problem (1). A simple calculation shows that this is equivalent to requiring that $\mathbf{x}$ is a solution to the linear system $\mathbf{A}^\top \mathbf{Ax} = \mathbf{A}^\top \mathbf{c}$, which always has a solution even when $\mathbf{Ax} = \mathbf{c}$ does not. The system $\mathbf{A}^\top \mathbf{Ax} = \mathbf{A}^\top \mathbf{c}$ is referred to as the *normal equation* associated with $\mathbf{Ax} = \mathbf{c}$ (see [TBI97]).

Fact 2.3. *$\mathbf{x}^* \in \arg \min_{\mathbf{x} \in \mathbb{R}^n} \|\mathbf{Ax} - \mathbf{c}\|_2^2$, if and only if $\mathbf{A}^\top \mathbf{Ax}^* = \mathbf{A}^\top \mathbf{c}$, and this linear system always has a solution.*

This leads to a natural question: Suppose we want to approximately solve the linear system $\mathbf{A}^\top \mathbf{Ax} = \mathbf{A}^\top \mathbf{c}$. Can we choose our notion of approximation to be equivalent to that of a solution to the LSA instance $(\mathbf{A}, \mathbf{c}, \epsilon)$?

A second natural question is whether we can choose a notion of distance between a proposed solution $\mathbf{x}$ and an optimal solution $\mathbf{x}^* \in \arg \min_{\mathbf{x} \in \mathbb{R}^n} \|\mathbf{Ax} - \mathbf{c}\|_2^2$ s.t. this distance being small is equivalent to $\mathbf{x}$ being a solution to the LSA instance $(\mathbf{A}, \mathbf{c}, \epsilon)$? The answer to both questions is yes, as demonstrated by the following facts:

Fact 2.4. *Suppose $\mathbf{x}^* \in \arg \min_{\mathbf{x} \in \mathbb{R}^n} \|\mathbf{Ax} - \mathbf{c}\|_2^2$ then*

1. $\|\mathbf{A}^\top \mathbf{Ax} - \mathbf{A}^\top \mathbf{c}\|_{(\mathbf{A}^\top \mathbf{A})^\dagger} = \|\mathbf{Ax} - \Pi_A \mathbf{c}\|_2 = \|\mathbf{x} - \mathbf{x}^*\|_{\mathbf{A}^\top \mathbf{A}}.$
2. *The following statements are each equivalent to $\mathbf{x}$ being a solution to the LSA instance $(\mathbf{A}, \mathbf{c}, \epsilon)$:*
 - (a) $\|\mathbf{A}^\top \mathbf{Ax} - \mathbf{A}^\top \mathbf{c}\|_{(\mathbf{A}^\top \mathbf{A})^\dagger} \leq \epsilon \|\mathbf{A}^\top \mathbf{c}\|_{(\mathbf{A}^\top \mathbf{A})^\dagger}$ *if and only if $\mathbf{x}$ is a solution to the LSA instance $(\mathbf{A}, \mathbf{c}, \epsilon)$.*
 - (b) $\|\mathbf{x} - \mathbf{x}^*\|_{\mathbf{A}^\top \mathbf{A}} \leq \epsilon \|\mathbf{x}^*\|_{\mathbf{A}^\top \mathbf{A}}$ *if and only if $\mathbf{x}$ is a solution to the LSA instance $(\mathbf{A}, \mathbf{c}, \epsilon)$.*

For completeness, we prove Fact 2.4 in Appendix A. Fact 2.4 explains connection between our Definition 2.1, and the usual convention for measuring error in the Laplacian solver literature [ST14]. In this setting, we consider a Laplacian matrix $\mathbf{L}$, which can be written as $\mathbf{L} = \mathbf{A}^\top \mathbf{A} \in \mathbb{R}^{n \times n}$, and a vector $\mathbf{b}$ s.t. $\Pi_{\mathbf{A}^\top \mathbf{A}} \mathbf{b} = \mathbf{b}$. This condition on $\mathbf{b}$ is easy to verify in the case of Laplacians, since for the Laplacian of a connected graph, $\Pi_{\mathbf{A}^\top \mathbf{A}} = \mathbf{I} - \frac{1}{n} \mathbf{1}\mathbf{1}^\top$. Additionally, it is also equivalent to the condition that there exists $\mathbf{c}$ s.t. $\mathbf{b} = \mathbf{A}^\top \mathbf{c}$. For Laplacians it is possible to compute both $\mathbf{A}$ and a vector $\mathbf{c}$ s.t. $\mathbf{b} = \mathbf{A}^\top \mathbf{c}$ in time linear in $\text{nnz}(\mathbf{L})$. For Laplacian solvers, the approximation error of an approximate solution $\mathbf{x}$ is measured by the ϵ s.t. $\|\mathbf{A}^\top \mathbf{Ax} - \mathbf{b}\|_{(\mathbf{A}^\top \mathbf{A})^\dagger} \leq \epsilon \|\mathbf{b}\|_{(\mathbf{A}^\top \mathbf{A})^\dagger}$. By Fact 2.4, we see that this is exactly equivalent to $\mathbf{x}$ being a solution to the LSA instance $(\mathbf{A}, \mathbf{c}, \epsilon)$.

2.2 Measuring the Difficulty of Solving a Linear System

Running times for iterative linear system solvers generally depend on the number of non-zeros in the input matrix, the condition number of the input matrix, the accuracy, and the bit complexity.

In this section, we formally define several measures of complexity of the linear systems we use. This is crucial, because we want to make sure that our reductions do not rely on mapping into extremely ill-conditioned matrices, and so we use these measures to show that this is in fact not the case.

Definition 2.5.

1. Given a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, we define the maximum singular value $\sigma_{\max}(\mathbf{A})$ in the usual way as $\sigma_{\max}(\mathbf{A}) = \max_{\mathbf{x} \in \mathbb{R}^n, \mathbf{x} \neq \mathbf{0}} \sqrt{\frac{\mathbf{x}^\top \mathbf{A}^\top \mathbf{A} \mathbf{x}}{\mathbf{x}^\top \mathbf{x}}}$.
2. Given a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ which is not all zeros, we define the minimum non-zero singular value $\sigma_{\min}(\mathbf{A})$ as $\sigma_{\min}(\mathbf{A}) = \min_{\mathbf{x} \in \mathbb{R}^n, \mathbf{x} \perp \text{null}(\mathbf{A})} \sqrt{\frac{\mathbf{x}^\top \mathbf{A}^\top \mathbf{A} \mathbf{x}}{\mathbf{x}^\top \mathbf{x}}}$.
3. Given a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ which is not all zeros, we define the non-zero condition number of $\mathbf{A}$ as $\kappa(\mathbf{A}) = \frac{\sigma_{\max}(\mathbf{A})}{\sigma_{\min}(\mathbf{A})}$.

Definition 2.6. The sparse parameter complexity of an LSA instance $(\mathbf{A}, \mathbf{c}, \epsilon)$ where $\mathbf{A} \in \mathbb{Z}^{m \times n}$ and $\text{nnz}(\mathbf{A}) \geq \max(m, n)$, and $\epsilon > 0$, is

$$\mathcal{S}(\mathbf{A}, \mathbf{c}, \epsilon) \stackrel{\text{def}}{=} \left(\text{nnz}(\mathbf{A}), \max \left(\|\mathbf{A}\|_{\max}, \|\mathbf{c}\|_{\max}, \frac{1}{\min_+(\mathbf{A})}, \frac{1}{\min_+(\mathbf{c})} \right), \kappa(\mathbf{A}), \epsilon^{-1} \right).$$

Note in the definition above that when $\mathbf{A} \neq \mathbf{0}$ and $\mathbf{c} \neq \mathbf{0}$ have only integer entries, we trivially have $\min_+(\mathbf{A}) \geq 1$ and $\min_+(\mathbf{c}) \geq 1$. However, including $\frac{1}{\min_+(\mathbf{A})}$, and $\frac{1}{\min_+(\mathbf{c})}$ in the definition stated above is useful when working with intermediate matrices whose entries are not integer valued.

2.3 Matrix Classes and Reductions Between Them

We use the term *matrix class* to refer to an infinite set of matrices $\mathcal{M}$. In this section, we formally define a notion of efficient reduction between linear systems in different classes of matrices.

Definition 2.7 (Efficient f -reducibility). Suppose we have two matrix classes $\mathcal{M}_1$ and $\mathcal{M}_2$, and there exist two algorithms $\mathcal{A}_{1 \rightarrow 2}$ and $\mathcal{A}_{1 \leftarrow 2}$ s.t. given an LSA instance $(\mathbf{M}^1, \mathbf{c}^1, \epsilon)$, where $\mathbf{M}^1 \in \mathcal{M}_1$, the call $\mathcal{A}_{1 \rightarrow 2}(\mathbf{M}^1, \mathbf{c}^1, \epsilon)$ returns an LSA instance $(\mathbf{M}^2, \mathbf{c}^2, \epsilon_2)$ s.t. if $\mathbf{x}^2$ is a solution to LSA instance $(\mathbf{M}^2, \mathbf{c}^2, \epsilon_2)$ then $\mathbf{x}^1 = \mathcal{A}_{1 \leftarrow 2}(\mathbf{M}^1, \mathbf{M}^2, \mathbf{x}^2)$ is a solution to LSA instance $(\mathbf{M}^1, \mathbf{c}^1, \epsilon)$.

Consider a function of $f : \mathbb{R}_+^4 \rightarrow \mathbb{R}_+^4$ s.t. every output coordinate is an increasing function of every input coordinate. Suppose that we always have

$$\mathcal{S}(\mathbf{M}^2, \mathbf{c}^2, \epsilon_2) \leq f(\mathcal{S}(\mathbf{M}^1, \mathbf{c}^1, \epsilon_1)),$$

and the running times of $\mathcal{A}_{1 \rightarrow 2}(\mathbf{M}^1, \mathbf{c}^1, \epsilon_1)$ and $\mathcal{A}_{1 \leftarrow 2}(\mathbf{M}^1, \mathbf{M}^2, \mathbf{x}^2)$ are both bounded by $O(\text{nnz}(\mathbf{M}^1))$.

Then we say that $\mathcal{M}_1$ is efficiently f -reducible to $\mathcal{M}_2$, which we also write as

$$\mathcal{M}_1 \leq_f \mathcal{M}_2.$$

Lemma 2.8. Suppose $\mathcal{M}_1 \leq_f \mathcal{M}_2$ and $\mathcal{M}_2 \leq_g \mathcal{M}_3$. Then $\mathcal{M}_1 \leq_{g \circ f} \mathcal{M}_3$.

Proof. The proof is simply by the trivial composition of the two reductions. $\square$

Definition 2.9. We let $\mathcal{G}$ denote the class of all matrices with integer valued entries s.t. there is at least one non-zero entry in every row and column⁵.

3 Main Results

In this section, we use the notions of sparse parameter complexity and matrix class reductions to prove our main technical result, Theorem 3.1, which shows that linear systems in general matrices with integer entries can be efficiently reduced to linear systems in several different classes of Incidence Structured Block Matrices. From this result, we derive as corollary our main result, Theorem 3.2, which states that fast high accuracy solvers for several types of ISBMs imply fast high accuracy solvers for all linear systems in general matrices with integer entries.

Theorem 3.1. *Let $f(s, U, K, \epsilon) = (O(s \log(sU)), \text{poly}(UK\epsilon^{-1}s), \text{poly}(UK\epsilon^{-1}s), \text{poly}(UK\epsilon^{-1}s))$, then*

1. $\mathcal{G} \leq_f \mathcal{MC}_{2, \mathbb{Z}}^{>0}$.
2. $\mathcal{G} \leq_f \mathcal{T}_2$.
3. $\mathcal{G} \leq_f \mathcal{V}_2$.

Theorem 3.2. *Suppose we have an algorithm which solves every Linear System Approximation Problem $(\mathbf{A}, \mathbf{c}, \epsilon)$ with sparse parameter complexity $\mathcal{S}(\mathbf{A}, \mathbf{c}, \epsilon) \leq (s, U, K, \epsilon^{-1})$ in time $O(s^a \text{polylog}(s, U, K, \epsilon^{-1}))$ for some $a \geq 1$, whenever $\mathbf{A} \in \mathcal{R}$ for at least one of $\mathcal{R} \in \{\mathcal{MC}_{2, \mathbb{Z}}^{>0}, \mathcal{T}_2, \mathcal{V}_2\}$. I.e. we have a “fast” solver⁶ for one of the matrix classes $\mathcal{MC}_{2, \mathbb{Z}}^{>0}, \mathcal{T}_2$, or $\mathcal{V}_2$. Then every Linear System Approximation Problem $(\mathbf{A}, \mathbf{c}, \epsilon)$ where $\mathbf{A} \in \mathcal{G}$ with sparse parameter complexity $\mathcal{S}(\mathbf{A}, \mathbf{c}, \epsilon) \leq (s, U, K, \epsilon^{-1})$ can be solved in time $O(s^a \text{polylog}(s, U, K, \epsilon^{-1}))$.*

Proof. The theorem is a immediate corollary of Theorem 3.1. $\square$

Definition 3.3. We let $\mathcal{G}_{z,2}$ denote the class of all matrices with integer valued entries s.t. there is at least one non-zero entry in every row and column, and every row has zero row sum, and for each row, the sum of the positive coefficients is a power of 2.

Lemma 3.4. *Let $f(s, U, K, \epsilon) = (O(s), O(\epsilon^{-1}s^{9/2}U^3), O(\epsilon^{-1}s^8U^3K), O(s^{5/2}U^2\epsilon^{-1}))$, then*

$$\mathcal{G} \leq_f \mathcal{G}_{z,2}.$$

Lemma 3.5. *Let $f(s, U, K, \epsilon) = (O(s \log(sU)), O(s^{3/2}U \log^{1/2}(sU)), O(Ks^4U^2 \log^2(sU)), O(sU^2\epsilon^{-1}))$, then*

$$\mathcal{G}_{z,2} \leq_f \mathcal{MC}_2.$$

Lemma 3.6. *Let $f(s, U, K, \epsilon) = (O(s), O(\epsilon^{-1}U^2K), O(\epsilon^{-1}s^2U^2K), O(\epsilon^{-1}))$, then*

$$\mathcal{MC}_2 \leq_f \mathcal{MC}_2^{>0}.$$

⁵If there is a row or column with only zeros, then it can always be handled trivially in the context of solving linear systems

⁶The reduction requires only a single linear system solve, and uses the solution in a black-box way. So the reduction also applies if the solver for the class $\mathcal{R}$ only works with high probability or only has running time guarantees in expectation.

Lemma 3.7. *Let $f(s, U, K, \epsilon) = (s, \epsilon^{-1}sU, 2K, O(\epsilon^{-1}))$, then*

$$\mathcal{MC}_2^{>0} \leq_f \mathcal{MC}_{2,\mathbb{Z}}^{>0}.$$

Lemma 3.8. *Let $f(s, U, K, \epsilon)$ be as defined in Lemma 3.5 then*

$$\mathcal{G}_{z,2} \leq_f \mathcal{T}_2.$$

Lemma 3.9. *Let $f(s, U, K, \epsilon) = (s, U, K, \epsilon^{-1})$, then*

$$\mathcal{MC}_2 \leq_f \mathcal{V}_2.$$

Proof of Theorem 3.1. Follows by appropriate composition (Lemma 2.8) applied to the the Lemmas above, i.e. 3.4, 3.5, 3.6, 3.7, 3.8 and 3.9. □

3.1 Outline of Remaining Sections

In Section 4 is presents the proof of Lemma 3.5, i.e. $\mathcal{G}_{z,2} \leq_f \mathcal{MC}_2$. This statement is our most important reduction. In Section 5, we prove Lemma 3.5, the surprising reduction $\mathcal{MC}_2 \leq_h \mathcal{MC}_2^{>0}$, which shows that we can solve *normal equations* even while changing null-spaces involved substantially. In Section 5, we show how to round weights in a 2-commodity problem to integers, proving Lemma 3.7. In Section 7 is presents the proof of Lemma 3.4. This is a simpler reduction that we use to establish the properties that our Lemma 3.5 relies on. In Section 8, we present the proof of Lemma 3.8. Section 9 describes how Interior Point Methods give rise to multi-commodity and Total Variation linear systems. Section 9 also contains a proof of Lemma 3.9.

4 Reducing Zero-Sum Power Two Linear Systems to Two-Commodity Linear Systems

To prove Lemma 3.5, we need to provide mapping algorithms $\mathcal{A}_{\mathcal{G}_{z,2} \rightarrow \mathcal{MC}_2}$ for mapping linear system approximation (LSA) instances over matrices in $\mathcal{G}_{z,2}$ to LSA problem instances over matrices in $\mathcal{MC}_2$, as well as $\mathcal{A}_{\mathcal{MC}_2 \leftarrow \mathcal{G}_{z,2}}$ for mapping the resulting solutions back. These leads to the following main components:

- Algorithm 1 states the pseudo-code for the algorithm $\text{REDUCE } \mathcal{G}_{z,2} \text{ TO } \mathcal{MC}_2$, which implements the desired mapping of problem instances. Given LSA problem instance $(\mathbf{A}, \mathbf{c}^A, \epsilon^A)$ where $\mathbf{A} \in \mathcal{G}_{z,2}$, the call $\text{REDUCE } \mathcal{G}_{z,2} \text{ TO } \mathcal{MC}_2(\mathbf{A}, \mathbf{c}^A, \epsilon^A)$ returns an LSA problem instance $(\mathbf{B}, \mathbf{c}^B, \epsilon^B)$ where $\mathbf{B} \in \mathcal{MC}_2$. (Strictly speaking, $\text{REDUCE } \mathcal{G}_{z,2} \text{ TO } \mathcal{MC}_2$ also has a parameter α which we will set before using the algorithm.)
- Algorithm 2 provides the pseudo-code for $\mathcal{MC}_2\text{Gadget}$, a short subroutine used in $\text{REDUCE } \mathcal{G}_{z,2} \text{ TO } \mathcal{MC}_2$ to represent the equation $a + b = 2c$ using two-commodity constraints.
- Algorithm 3 provides the (trivial) pseudo-code for $\text{MAPSOLN } \mathcal{MC}_2 \text{ TO } \mathcal{G}_{z,2}$ used to map solutions to LSA problems over $\mathcal{MC}_2$ back to solutions over $\mathcal{G}_{z,2}$ by restricting onto the original variables.

Pseudocode of the key reduction routine that creates the new linear system, REDUCE $\mathcal{G}_{z,2}$ TO $\mathcal{MC}_2$, is shown in Algorithm 1. Note that given two singleton multi-sets each containing a single equation, e.g. $\{\mathbf{a}_1^\top \mathbf{x} = c_1\}$ and $\{\mathbf{a}_2^\top \mathbf{x} = c_2\}$ where $\mathbf{a}_1, \mathbf{a}_2$ are vectors, we define $\{\mathbf{a}_1^\top \mathbf{x} = c_1\} + \{\mathbf{a}_2^\top \mathbf{x} = c_2\} = \{\mathbf{a}_1^\top \mathbf{x} + \mathbf{a}_2^\top \mathbf{x} = c_1 + c_2\}$ and we define $\{\mathbf{a}_1^\top \mathbf{x} = c_1\} \cup \{\mathbf{a}_2^\top \mathbf{x} = c_2\} = \{\mathbf{a}_1^\top \mathbf{x} = c_1, \mathbf{a}_2^\top \mathbf{x} = c_2\}$.

Algorithm 1 REDUCE $\mathcal{G}_{z,2}$ TO $\mathcal{MC}_2$

Input: $(\mathbf{A}, \mathbf{c}^A, \epsilon_1)$ where $\mathbf{A} \in \mathcal{G}_{z,2}$ is an $m \times n$ matrix, $\mathbf{c}^A \in \mathbb{R}^n$, $0 < \epsilon_1 < 1$, and $\alpha > 0$.

Output: $(\mathbf{B}, \mathbf{c}^B, \epsilon_2)$ where $\mathbf{B} \in \mathcal{MC}_2$ is an $m \times n$ matrix, $\mathbf{c}^B \in \mathbb{R}^n$, and $0 < \epsilon_2 < 1$.

```

1:  $\epsilon_2 \leftarrow \epsilon_1 \left(1 + \frac{1}{\alpha}\right)^{-1/2} \left(1 + \frac{\|\mathbf{c}^A\|_2^2 \sigma_{\max}^2(\mathbf{A})}{\alpha + 1}\right)^{-1/2}$ 
2:  $\mathcal{X} \leftarrow \{\mathbf{u}_1, \dots, \mathbf{u}_n, \mathbf{v}_1, \dots, \mathbf{v}_n\}$  { $\mathcal{MC}_2$  variables and index of new variables}
3: Let  $\mathbf{x}$  be the vector of variables corresponding to the set of variables  $\mathcal{X}$ 
4:  $t \leftarrow n + 1$ 
5: Initialize  $\mathcal{A} \leftarrow \emptyset, \mathcal{B} \leftarrow \emptyset$  {Multisets of main and  $\mathcal{MC}_2$  auxiliary equations respectively}
6: for each equation  $1 \leq i \leq m$  in  $\mathbf{A}$  do
7:   Let  $\mathcal{I}^{+1} \leftarrow \{j : \mathbf{A}_{ij} > 0\}, \mathcal{I}^{-1} \leftarrow \{j : \mathbf{A}_{ij} < 0\}$ 
8:   if  $|\mathcal{I}^{+1}| = 1$  and  $|\mathcal{I}^{-1}| = 1$  then
9:     Let the only elements in  $\mathcal{I}^{+1}$  and  $\mathcal{I}^{-1}$  be  $j_+$  and  $j_-$  respectively. {Note  $\mathbf{A}_{ij_+} = -\mathbf{A}_{ij_-}$ }
10:     $\mathcal{A} \leftarrow \mathcal{A} \cup \{\mathbf{A}_{ij_+} \mathbf{u}_t - \mathbf{A}_{ij_-} \mathbf{u}_{j_-} = \mathbf{c}_i\}$ 
11:     $w_i \leftarrow \alpha$ 
12:     $\mathcal{B} \leftarrow \mathcal{B} \cup w_i^{1/2} \cdot \{\mathbf{A}_{ij_+} \mathbf{u}_{j_+} - \mathbf{A}_{ij_+} \mathbf{u}_t = 0\}$ 
13:     $\mathcal{X} \leftarrow \mathcal{X} \cup \{\mathbf{u}_t, \mathbf{v}_t\}$ , update  $\mathbf{x}$  accordingly
14:     $t \leftarrow t + 1$ 
15:   else
16:     $\mathcal{A}_i \leftarrow \{\mathbf{A}_i \mathbf{u} = \mathbf{c}_i\}$ 
17:     $\mathcal{B}_i \leftarrow \emptyset$ 
18:    for  $s = -1, +1$  do
19:       $r \leftarrow 0$ 
20:      while  $\mathcal{A}_i$  has strictly more than 1 coefficient with sign  $s$  do
21:        For each  $j$ , let  $\hat{\mathbf{A}}_{ij}$  be the coefficient of  $\mathbf{u}_j$  in  $\mathcal{A}_i$ .
22:         $\mathcal{I}_{\text{odd}}^s \leftarrow \{j \in \mathcal{I}^s : \lfloor \hat{\mathbf{A}}_{ij} / 2^r \rfloor \text{ is odd}\}$ 
23:        Pair the indices of  $\mathcal{I}_{\text{odd}}^s$  into  $k$  disjoint pairs  $(j_k, l_k)$ 
24:        for each pair of indices  $(j_k, l_k)$  do
25:           $\mathcal{A}_i \leftarrow \mathcal{A}_i + s \cdot 2^r \{(2\mathbf{u}_t - (\mathbf{u}_{j_k} + \mathbf{u}_{l_k})) = 0\}$ 
26:           $\mathcal{B}_i \leftarrow \mathcal{B}_i \cup -s \cdot 2^r \cdot \mathcal{MC}_2\text{GADGET}(\mathbf{u}_{j_k}, \mathbf{u}_{l_k}, t)$ 
27:           $\mathcal{X} \leftarrow \mathcal{X} \cup \{\mathbf{u}_t, \dots, \mathbf{u}_{t+6}, \mathbf{v}_t, \dots, \mathbf{v}_{t+6}\}$ , update  $\mathbf{x}$  accordingly
28:           $t \leftarrow t + 7$ 
29:           $r \leftarrow r + 1$ 
30:        end for
31:      end while
32:    end for
33:     $w_i \leftarrow \alpha |\mathcal{B}_i|$ 
34:     $\mathcal{B} \leftarrow \mathcal{B} \cup w_i^{1/2} \cdot \mathcal{B}_i$ 
35:     $\mathcal{A} \leftarrow \mathcal{A} \cup \mathcal{A}_i$ .
36:  end if
37: end for
38: return  $\epsilon_2$  and  $\mathbf{B}, \mathbf{c}$  s.t.  $\mathbf{B}\mathbf{x} = \mathbf{c}$  corresponds to the equations in  $\mathcal{A} \cup \mathcal{B}$ , on the variable set  $\mathcal{X}$ .
```

The central object created by Algorithm 1 is the matrix $\mathbf{B}$, which contains both new equations and new variables. We will superscript the variables with A to distinguish variables appear in the original equation $\mathbf{A}\mathbf{x}^A = \mathbf{c}^A$ from new variables. We will term the new variables as $\mathbf{x}^{\text{aux}}$, and write a vector solution for the new problem, $\mathbf{x}^B$, as:

$$\mathbf{x}^B = \begin{pmatrix} \mathbf{x}^A \\ \mathbf{x}^{\text{aux}} \end{pmatrix}. \quad (2)$$

Let n_A be the dimension of $\mathbf{x}^A$, and n_B be the dimension of $\mathbf{x}^B$, respectively. We order the variables so that for an appropriately chosen index h we have

1. $\mathbf{x}_{1:h}^{\text{aux}}$ corresponds to the $\mathbf{u}$ -coordinates of the auxiliary variables created in $\mathcal{MC}_2$ -gadgets.
2. $\mathbf{x}_{h+1:n_B-n_A}^{\text{aux}}$ corresponds to the $\mathbf{v}$ -coordinates of the auxiliary variables created in $\mathcal{MC}_2$ -gadgets.

With this ordering $\mathbf{x}_t^B = \mathbf{u}_t$ for $t \leq n_A + h$.

Furthermore, we will distinguish the equations in $\mathbf{B}$ into ones formed from manipulating $\mathbf{A}$, i.e. the equations added to the set $\mathcal{A}$, from the auxiliary equations, i.e. the equations added to the set $\mathcal{B}$. We use $\mathbf{W}^{1/2} = \text{DIAG}_i(w_i^{1/2})$ to refer to the diagonal matrix of weights w_i applied to the auxiliary equations $\mathcal{B}$ in Algorithm 1. In Algorithm 1, a real value $\alpha > 0$ is set initially and used when computing the weights $w_i^{1/2}$. For convenience, throughout most of this section, we will treat α as an arbitrary constant, and only eventually substitute in its value to complete our main proof.

This leads to the following representation of $\mathbf{B}$ and $\mathbf{c}^B$ which we will use throughout our analysis of the algorithm:

$$\mathbf{B} = \begin{pmatrix} \hat{\mathbf{A}} \\ \mathbf{W}^{1/2} \hat{\mathbf{B}} \end{pmatrix}. \quad (3)$$

Here the equations of $\hat{\mathbf{A}}$ corresponds to $\mathcal{A}$ in $\text{REDUCE } \mathcal{G}_{z,2} \text{ TO } \mathcal{MC}_2$, and $\hat{\mathbf{B}}$ corresponds to the auxiliary constraints, i.e. equations of $\mathcal{B}$ in $\text{REDUCE } \mathcal{G}_{z,2} \text{ TO } \mathcal{MC}_2$. Also, the vector $\mathbf{c}^B$ created is simply an extension of $\mathbf{c}^A$:

$$\mathbf{c}^B = \begin{pmatrix} \mathbf{c}^A \\ \mathbf{0} \end{pmatrix}. \quad (4)$$

Finally, as Algorithm 1 creates new equations for each row of $\mathbf{A}$ independently, we will use S_i to denote the subset of indices of the rows of $\hat{\mathbf{B}}$ that's created from $\mathbf{A}_{i,:}$, aka. the the auxiliary constraints generated from the call to $\mathcal{MC}_2\text{GADGET}$ upon processing $\mathbf{A}_{i,:}$. We will also denote the size of these using

$$m_i \stackrel{\text{def}}{=} |S_i|,$$

and use $\hat{\mathbf{B}}_i$ to denote these part of $\hat{\mathbf{B}}$ that corresponds to these rows. The Gadget routine used in the reduction, and the (trivial) solution mapper are stated below.

Algorithm 2 $\mathcal{MC}_2\text{GADGET}$

Input: Scalar variables $\mathbf{u}_{j_k}, \mathbf{u}_{l_k}$, integer t .

Output: Set of equations for a multi-commodity gadget that computes the average of $\mathbf{u}_{j_k}$ and $\mathbf{u}_{l_k}$.

1: **return**

$$\begin{aligned} \{ & \mathbf{u}_{t+3} - \mathbf{v}_{t+3} - (\mathbf{u}_{t+4} - \mathbf{v}_{t+4}) = 0, \\ & \mathbf{u}_t - \mathbf{u}_{t+3} = 0, \\ & \mathbf{u}_{t+4} - \mathbf{u}_{j_k} = 0, \\ & \mathbf{v}_{t+3} - \mathbf{v}_{t+1} = 0, \\ & \mathbf{v}_{t+2} - \mathbf{v}_{t+4} = 0, \\ & \mathbf{u}_{t+5} - \mathbf{v}_{t+5} - (\mathbf{u}_{t+6} - \mathbf{v}_{t+6}) = 0, \\ & \mathbf{u}_t - \mathbf{u}_{t+5} = 0, \\ & \mathbf{u}_{t+6} - \mathbf{u}_{l_k} = 0, \\ & \mathbf{v}_{t+5} - \mathbf{v}_{t+2} = 0, \\ & \mathbf{v}_{t+1} - \mathbf{v}_{t+6} = 0 \} \end{aligned}$$

Algorithm 3 $\text{MAPSOLN } \mathcal{MC}_2\text{TO}\mathcal{G}_{z,2}$

Input: $m \times n$ matrix $\mathbf{A} \in \mathcal{G}_{z,2}$, $m' \times n'$ matrix $\mathbf{B} \in \mathcal{MC}_2$, vector $\mathbf{c}^A \in \mathbb{R}^m$, vector $\mathbf{x}^B \in \mathbb{R}^{n'}$.

Output: Vector $\mathbf{x}^A \in \mathbb{R}^n$.

1: **if** $\mathbf{A}^\top \mathbf{c}^A = \mathbf{0}$ **then**
2: **return** $\mathbf{x}^A \leftarrow \mathbf{0}$
3: **else**
4: **return** $\mathbf{x}^A \leftarrow \mathbf{x}_{1:n}^B$
5: **end if**

We upper bound the number of nonzero entries of $\mathbf{B}$ by the number of nonzero entries of $\mathbf{A}$.

Lemma 4.1.

- We have $\text{nnz}(\mathbf{B}) = O(\text{nnz}(\mathbf{A}) \log \|\mathbf{A}\|_\infty)$.
- Both dimensions of $\mathbf{B}$ are $O(\text{nnz}(\mathbf{A}) \log \|\mathbf{A}\|_\infty)$.
- The number of iterations required by Algorithm 1 to process row $\mathbf{A}_i$ is at most $\log \|\mathbf{A}_i\|_1$.
- When processing row $\mathbf{A}_i$, every new variable that appears in the $\mathcal{A}_i$ equation is paired in the following iteration of the while-loop in Line 20 (unless it is the final iteration), and then disappears from $\mathcal{A}_i$.

Proof. Since Algorithm 1 constructs new equations for each row of $\mathbf{A}$ independently, we bound the number of new variables and new equations (that is, the size of the submatrix $\begin{pmatrix} \hat{\mathbf{A}}_i \\ \hat{\mathbf{B}}_i \end{pmatrix}$) for each row i of $\mathbf{A}$ separately.

Let n_i be the number of nonzero entries of the i th row of $\mathbf{A}$. We count the number of variables in the equation $\mathcal{A}_i$, at each iteration of the while-loop in line 20 of Algorithm 1. Let $X^{(r)}$ be the

subset of variables with nonzero coefficients in $\mathcal{A}_i$, at the end of iteration r . Let $X_{aux}^{(r)}$ be the subset of $X^{(r)}$ containing all auxiliary variables created in iteration r .

Note in each iteration, Algorithm 1 replaces two variables by a new auxiliary variable. It gives that

$$\left| X_{aux}^{(1)} \right| \leq \frac{n_i}{2}, \text{ and } \left| X^{(1)} \setminus X_{aux}^{(1)} \right| \leq n_i.$$

Since each auxiliary variable in $X_{aux}^{(1)}$ has coefficient 2, in the 2nd iteration, together with another variable of coefficient 2, it must be replaced by a new auxiliary variable. Thus, at the end of the 2nd iteration, all auxiliary variables $X_{aux}^{(1)}$ will not appear in the equation $\mathcal{A}_i$. This implies that

$$\left| X_{aux}^{(2)} \right| \leq \frac{n_i}{2} + \frac{n_i}{4}, \text{ and } \left| X^{(2)} \setminus X_{aux}^{(2)} \right| \leq n_i.$$

Similarly, at the end of the t^{th} iteration, we have

$$\left| X_{aux}^{(r)} \right| \leq \sum_{1 \leq s \leq r} \frac{n_i}{2^s} \leq n_i,$$

and

$$\left| X^{(r)} \setminus X_{aux}^{(r)} \right| \leq n_i.$$

Since Algorithm 1 pulls out a factor 2 in each iteration, the total number of iterations is at most $\log \|\mathbf{A}_i\|_1$. Since each auxiliary variable in $\mathcal{A}_i$ during the construction corresponds to $O(1)$ auxiliary variables and $O(1)$ equations in $\hat{\mathbf{B}}_i$, the total number of auxiliary variables and equations for row i of $\mathbf{A}$ is

$$O(n_i \log \|\mathbf{A}_i\|_1).$$

Therefore, the number of variables and the number of equations in $\mathbf{B}$ (that is, the both dimensions of $\mathbf{B}$) are

$$O(\text{nnz}(\mathbf{A}) \log \|\mathbf{A}\|_\infty).$$

Since each row of $\mathbf{B}$ has $O(1)$ nonzero coefficients, we have

$$\text{nnz}(\mathbf{B}) = O(\text{nnz}(\mathbf{A}) \log \|\mathbf{A}\|_\infty).$$

This completes the proof. □

4.1 Reduction Between Exact Solvers

The most important relation between $\mathbf{A}$ and $\mathbf{B}$ is given by the following claim.

Claim 4.2 (Reduction between exact solvers). *Fix any $\mathbf{x}^A \in \mathbb{R}^n$. Then*

$$\left\| \mathbf{A} \mathbf{x}^A - \mathbf{c}^A \right\|_2^2 = \frac{\alpha + 1}{\alpha} \min_{\mathbf{x}^{aux}} \left\| \mathbf{B} \begin{pmatrix} \mathbf{x}^A \\ \mathbf{x}^{aux} \end{pmatrix} - \mathbf{c}^B \right\|_2^2.$$

As a Corollary of Claim 4.2, we observe the following:

Lemma 4.3. *Given LSA problem instance $(\mathbf{A}, \mathbf{c}^A, \epsilon^A)$ where $\mathbf{A} \in \mathcal{G}_{z,2}$, suppose*

$$(\mathbf{B}, \mathbf{c}^B, \epsilon^B) = \text{REDUCE } \mathcal{G}_{z,2} \text{ TO } \mathcal{MC}_2(\mathbf{A}, \mathbf{c}^A, \epsilon^A).$$

Then $\mathbf{B} \in \mathcal{MC}_2$ and if $\mathbf{x}^B = \begin{pmatrix} \mathbf{x}^A \\ \mathbf{x}^{aux} \end{pmatrix}$ is a solution to the exact LSA problem $(\mathbf{B}, \mathbf{c}^B, 0)$, then $\mathbf{x}^A$ is a solution to the exact LSA problem $(\mathbf{A}, \mathbf{c}^A, 0)$.

Proof. This follows immediately from minimizing over $\mathbf{x}^A$ on both sides of the equation established by Claim 4.2 and then applying Fact 2.2. $\square$

Before proving Claim 4.2 we first note a basic guarantee obtained by $\text{REDUCE } \mathcal{G}_{z,2} \text{ TO } \mathcal{MC}_2$:

$$\mathbf{1}^\top \left(\begin{pmatrix} \hat{\mathbf{A}}_i \\ \hat{\mathbf{B}}_i \end{pmatrix} \mathbf{x}^B - \begin{pmatrix} \mathbf{c}_i^A \\ \mathbf{0} \end{pmatrix} \right) = \mathbf{A}_i \mathbf{x}^A - \mathbf{c}_i^A. \quad (5)$$

To verify this guarantee, we consider two cases separately. The first case is when the condition $|\mathcal{I}^{+1}| = 1$ AND $|\mathcal{I}^{-1}| = 1$ is true (see Algorithm 1, Line 7). In this case, the main constraint in the output corresponding to row i is $\{\mathbf{A}_{ij_+} \mathbf{x}_t^B - \mathbf{A}_{ij_-} \mathbf{x}_{j_-}^B = \mathbf{c}_i^A\}$, while the auxiliary constraints contain only a single row $\{\mathbf{A}_{ij_+} \mathbf{x}_{j_+}^B - \mathbf{A}_{ij_+} \mathbf{x}_t^B = 0\}$, and adding these proves the guarantee for this case. The second case is when the condition in Algorithm 1, Line 7) is false. We consider the case $s = +1$. The case $s = -1$ is proved similarly. Note that we will refer to variables j_k and l_k only in the context of a fixed value of t , which always ensures that they are unambiguously defined. In the case $s = +1$, each time we modify $\mathcal{A}_i$ by adding $2^r \left(2\mathbf{x}_{t+1}^B - (\mathbf{x}_{j_k}^B + \mathbf{x}_{l_k}^B) \right) = 0$, we also use $\mathcal{MC}_2\text{Gadget}$ to create auxiliary constraints that sum up to exactly $-2^r \left(2\mathbf{x}_{t+1}^B - (\mathbf{x}_{j_k}^B + \mathbf{x}_{l_k}^B) \right)$, so adding these together will cancel out the changes.

Proof of Claim 4.2. For convenience, we also write

1. $\delta_j \stackrel{\text{def}}{=} \hat{\mathbf{B}}_j \mathbf{x}^B,$
2. $\hat{\delta}_i \stackrel{\text{def}}{=} \hat{\mathbf{A}}_i \mathbf{x}^B - \mathbf{c}_i^A,$
3. $\epsilon_i \stackrel{\text{def}}{=} \mathbf{A}_i \mathbf{x}^A - \mathbf{c}_i^A.$

Thus

$$\left\| \begin{pmatrix} \hat{\mathbf{A}}_i \\ \mathbf{W}_i^{1/2} \hat{\mathbf{B}}_i \end{pmatrix} \mathbf{x}^B - \begin{pmatrix} \mathbf{c}_i^A \\ \mathbf{0} \end{pmatrix} \right\|_2^2 = \hat{\delta}_i^2 + w_i \sum_{j \in S_i} \delta_j^2.$$

Summing over all rows, we get

$$\left\| \mathbf{B} \begin{pmatrix} \mathbf{x}^A \\ \mathbf{x}^{\text{aux}} \end{pmatrix} - \mathbf{c}^B \right\|_2^2 = \sum_i \left(\hat{\delta}_i^2 + w_i \sum_{j \in S_i} \delta_j^2 \right).$$

Similarly, the square of row i of $\mathbf{A} \mathbf{x}^A - \mathbf{c}^A$ is ϵ_i^2 and summing over all rows we get

$$\|\mathbf{A} \mathbf{x}^A - \mathbf{c}^A\|_2^2 = \sum_i \epsilon_i^2.$$

Recalling Equation (5), we have

$$\mathbf{1}^\top \left(\begin{pmatrix} \hat{\mathbf{A}}_i \\ \hat{\mathbf{B}}_i \end{pmatrix} \mathbf{x}^B - \begin{pmatrix} \mathbf{c}_i^A \\ \mathbf{0} \end{pmatrix} \right) = \mathbf{A}_i \mathbf{x}^A - \mathbf{c}_i^A.$$

Thus

$$\hat{\delta}_i + \sum_{j \in S_i} \delta_j = \epsilon_i.$$

By the Cauchy-Schwarz inequality (applied to two vectors $\mathbf{a}$ and $\mathbf{b}$ given by $\mathbf{a}_1 = \hat{\delta}_i, \mathbf{a}_j = \sqrt{w_i}\delta_j, \mathbf{b}_1 = 1, \mathbf{b}_j = 1/\sqrt{w_i}$),

$$\epsilon_i^2 = \left(\hat{\delta}_i + \sum_{j \in S_i} \delta_j \right)^2 \leq \left(\hat{\delta}_i^2 + w_i \sum_{j \in S_i} \delta_j^2 \right) \left(1 + \frac{m_i}{w_i} \right).$$

the equality holds if and only if

$$\hat{\delta}_i = w_i \delta_j \quad \forall j \in S_i. \quad (6)$$

Note that we have ensured that for all i , $w_i = \alpha m_i$.

By summing over rows we conclude that for every $\mathbf{x}^A$ and every $\mathbf{x}^{\text{aux}}$, we have

$$\|\mathbf{A}\mathbf{x}^A - \mathbf{c}^A\|_2^2 \leq \frac{\alpha + 1}{\alpha} \left\| \mathbf{B} \begin{pmatrix} \mathbf{x}^A \\ \mathbf{x}^{\text{aux}} \end{pmatrix} - \mathbf{c}^B \right\|_2^2. \quad (7)$$

The inequality above will be an equality if Equations (6) are satisfied. We now show that for every fixed $\mathbf{x}^A$, minimizing over $\mathbf{x}^{\text{aux}}$ ensures that (4.1) holds with equality.

In particular, we will momentarily prove the following Claim.

Claim 4.4. *For any fixed $\mathbf{x}^A$ and its associated ϵ_i values, for each row i of $\mathbf{A}$, the linear system*

$$\hat{\mathbf{A}}_i \begin{pmatrix} \mathbf{x}^A \\ \mathbf{x}^{\text{aux}} \end{pmatrix} = \mathbf{c}_i^A + \frac{\alpha}{\alpha + 1} \epsilon_i, \quad (8)$$

$$\hat{\mathbf{B}}_j \begin{pmatrix} \mathbf{x}^A \\ \mathbf{x}^{\text{aux}} \end{pmatrix} = \frac{1}{(\alpha + 1)m_i} \epsilon_i, \forall j \in S_i. \quad (9)$$

has a solution (which may not be unique).

Since every auxiliary variable is associated with only one row i of $\mathbf{A}$, Claim 4.4 implies that we can choose $\mathbf{x}^{\text{aux}}$ s.t. all these linear systems are satisfied at once.

Given such a choice of $\mathbf{x}^{\text{aux}}$, we get that Equations (6) are satisfied so

$$\|\mathbf{A}\mathbf{x}^A - \mathbf{c}^A\|_2^2 = \frac{\alpha + 1}{\alpha} \left\| \mathbf{B} \begin{pmatrix} \mathbf{x}^A \\ \mathbf{x}^{\text{aux}} \end{pmatrix} - \mathbf{c}^B \right\|_2^2.$$

Given Claim 4.4, this completes the proof of Claim 4.2. $\square$

Proof of Claim 4.4. We focus on the case of the condition $|\mathcal{I}^{+1}| = 1$ AND $|\mathcal{I}^{-1}| = 1$ in Algorithm 1, Line 7 being false. The case when then condition is true is very similar, but easier as it deals with a set of just two equations.

We will construct an assignment to all the variables of $\mathbf{x}^{\text{aux}}$ s.t. Equations (8) and (9) are satisfied. We start with an assignment $\mathbf{x}^A$ to the main variables, and we then assign values to auxiliary variables in the order they are created by the algorithm $\text{REDUCE}_{\mathcal{G}_{z,2}\text{ToMC}_2}$. Note that we will refer to variables j_k and l_k only in the context of a fixed value of t , which always ensures that they are unambiguously defined. When the algorithm processes pair $\mathbf{x}_{j_k}^B, \mathbf{x}_{l_k}^B = \mathbf{u}_{j_k}, \mathbf{u}_{l_k}$, the value of these variables will have been set already, while $\mathbf{x}_t^B = \mathbf{u}_t$ and the other newly created auxiliary variables have not. Every auxiliary variable is associated with only one row, so we never get multiple assignments to a variable using this procedure.

Recall the constraints created by the $\mathcal{MC}_2\text{Gadget}$ call are

$$\begin{aligned} \{ & \mathbf{u}_{t+3} - \mathbf{v}_{t+3} - (\mathbf{u}_{t+4} - \mathbf{v}_{t+4}) = 0, \\ & \mathbf{u}_t - \mathbf{u}_{t+3} = 0, \\ & \mathbf{u}_{t+4} - \mathbf{u}_{j_k} = 0, \\ & \mathbf{v}_{t+3} - \mathbf{v}_{t+1} = 0, \\ & \mathbf{v}_{t+2} - \mathbf{v}_{t+4} = 0, \\ & \mathbf{u}_{t+5} - \mathbf{v}_{t+5} - (\mathbf{u}_{t+6} - \mathbf{v}_{t+6}) = 0, \\ & \mathbf{u}_t - \mathbf{u}_{t+5} = 0, \\ & \mathbf{u}_{t+6} - \mathbf{u}_{l_k} = 0, \\ & \mathbf{v}_{t+5} - \mathbf{v}_{t+2} = 0, \\ & \mathbf{v}_{t+1} - \mathbf{v}_{t+6} = 0 \} \end{aligned}$$

Let $\mathbf{z} = (\mathbf{u}_{j_k}, \mathbf{u}_{l_k}, \mathbf{u}_t, \mathbf{u}_{t+1}, \mathbf{u}_{t+2}, \mathbf{u}_{t+3}, \dots, \mathbf{u}_{t+6}, \mathbf{v}_t, \mathbf{v}_{t+1}, \dots, \mathbf{v}_{t+6}) \in \mathbb{R}^{16}$. Let $\mathbf{G} \in \mathbb{R}^{10 \times 16}$ be the matrix s.t. $\mathbf{G}\mathbf{z} = \mathbf{0}$ corresponds to the constraints listed above. Note that all coefficients of $\mathbf{u}_{t+1}$, $\mathbf{u}_{t+2}$ and $\mathbf{v}_t$ are zero. We set these three variables to zero.

For some ϵ , which we will fix later, we choose $\mathbf{u}_t$ such that

$$\mathbf{u}_t = \frac{1}{2}(\mathbf{u}_{j_k} + \mathbf{u}_{l_k}) + 5\epsilon.$$

Again, for the same ϵ , we fix the following values for $\mathbf{u}_{t+i}, \mathbf{v}_{t+j}, 3 \leq i \leq 6, 1 \leq j \leq 6$

$$\begin{aligned} \mathbf{u}_{t+3} &= \mathbf{u}_t - \epsilon, \\ \mathbf{u}_{t+4} &= \mathbf{u}_{j_k} + \epsilon, \\ \mathbf{u}_{t+5} &= \mathbf{u}_t - \epsilon, \\ \mathbf{u}_{t+6} &= \mathbf{u}_{l_k} + \epsilon, \\ \mathbf{v}_{t+1} &= 0, \\ \mathbf{v}_{t+2} &= 5\epsilon - (\mathbf{u}_t - \mathbf{u}_{j_k}), \\ \mathbf{v}_{t+3} &= \epsilon, \\ \mathbf{v}_{t+4} &= 4\epsilon - (\mathbf{u}_t - \mathbf{u}_{j_k}), \\ \mathbf{v}_{t+5} &= 6\epsilon - (\mathbf{u}_t - \mathbf{u}_{j_k}), \\ \mathbf{v}_{t+6} &= -\epsilon. \end{aligned}$$

This ensures $\mathbf{G}\mathbf{z} = \epsilon\mathbf{1}$. Note that for some r , $\widehat{\mathbf{B}}_i \mathbf{x}^B = 2^r \mathbf{G}\mathbf{z} = 2^r \epsilon \mathbf{1}$, so by choosing $\epsilon = 2^{-r} \frac{1}{(\alpha+1)m_i} \epsilon_i$, we can ensure Equations (9) are satisfied.

Also

$$\mathbf{c}_i^A + \epsilon_i = \mathbf{1}^\top \begin{pmatrix} \widehat{\mathbf{A}}_i \\ \widehat{\mathbf{B}}_i \end{pmatrix} \mathbf{x}^B = \widehat{\mathbf{A}}_i \mathbf{x}^B + \mathbf{1}^\top \widehat{\mathbf{B}}_i \mathbf{x}^B = \widehat{\mathbf{A}}_i \mathbf{x}^B + \frac{1}{\alpha+1} \epsilon_i.$$

Thus, we have

$$\widehat{\mathbf{A}}_i \mathbf{x}^B = \mathbf{c}_i^A + \frac{\alpha}{\alpha+1} \epsilon_i,$$

which is Equation 8.

This completes the proof of the claim. □

Remark. The optimal solutions for $\min_{\mathbf{x}^B} \|\mathbf{B}\mathbf{x}^B - \mathbf{c}^B\|_2$ and $\min_{\mathbf{x}^A} \|\mathbf{A}\mathbf{x}^A - \mathbf{c}^A\|_2$ have a one-to-one map, however, the optimal values are different:

From the proof of Claim 4.2 and Equation (1)

$$\|\mathbf{c}^A - \mathbf{\Pi}_A \mathbf{c}^A\|_2^2 = \left(1 + \frac{1}{\alpha}\right) \|\mathbf{c}^B - \mathbf{\Pi}_B \mathbf{c}^B\|_2^2,$$

where we set the weight $w_i = \alpha m_i, \forall i$. Note when $\alpha \rightarrow \infty$, the two optimal values approach the same value.

4.2 Relationship Between Schur Complements

Definition 4.5 (Schur Complement). Let $\mathbf{C} \in \mathbb{R}^{n \times n}$ be a symmetric PSD matrix. Write $\mathbf{C} = \begin{pmatrix} \mathbf{C}_{11} & \mathbf{C}_{12} \\ \mathbf{C}_{12}^\top & \mathbf{C}_{22} \end{pmatrix}$ where $\mathbf{C}_{11}, \mathbf{C}_{12}, \mathbf{C}_{22}$ are block matrices, the *Schur complement* of $\mathbf{C}$ is

$$\mathbf{C}_S \stackrel{\text{def}}{=} \mathbf{C}_{11} - \mathbf{C}_{12} \mathbf{C}_{22}^{-1} \mathbf{C}_{12}^\top.$$

If $\mathbf{C}_{22}$ is not invertible, then we use pseudo-inverse, that is, $\mathbf{C}_S \stackrel{\text{def}}{=} \mathbf{C}_{11} - \mathbf{C}_{12} \mathbf{C}_{22}^\dagger \mathbf{C}_{12}^\top$.

The following fact is important for Schur complement.

Fact 4.6 (Schur complement). *For any fixed vector $\mathbf{x}$,*

$$\min_{\mathbf{y}} \begin{pmatrix} \mathbf{x}^\top & \mathbf{y}^\top \end{pmatrix} \mathbf{C} \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} = \mathbf{x}^\top \mathbf{C}_S \mathbf{x}.$$

Proof. We expand the left hand side,

$$\begin{pmatrix} \mathbf{x}^\top & \mathbf{y}^\top \end{pmatrix} \mathbf{C} \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} = \mathbf{x}^\top \mathbf{C}_{11} \mathbf{x} + 2\mathbf{x}^\top \mathbf{C}_{12} \mathbf{y} + \mathbf{y}^\top \mathbf{C}_{22} \mathbf{y}. \quad (10)$$

Taking derivative w.r.t. $\mathbf{y}$ and setting it to be 0 give that

$$2\mathbf{C}_{22} \mathbf{y} + 2\mathbf{C}_{12}^\top \mathbf{x} = \mathbf{0}.$$

Plugging $\mathbf{y} = -\mathbf{C}_{22}^\dagger \mathbf{C}_{12}^\top \mathbf{x}$ into (10),

$$\min_{\mathbf{y}} \begin{pmatrix} \mathbf{x}^\top & \mathbf{y}^\top \end{pmatrix} \mathbf{C} \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} = \mathbf{x}^\top \mathbf{C}_{11} \mathbf{x} - \mathbf{x}^\top \mathbf{C}_{12} \mathbf{C}_{22}^\dagger \mathbf{C}_{12}^\top \mathbf{x} = \mathbf{x}^\top \mathbf{C}_S \mathbf{x}.$$

This completes the proof. $\square$

Recall that $\mathbf{B}$ is the output coefficient matrix of Algorithm 1. Write $\mathbf{B} = \begin{pmatrix} \mathbf{B}_1 & \mathbf{B}_2 \end{pmatrix}$, where $\mathbf{B}_1$ is the submatrix corresponding to $\mathbf{x}^A$ and $\mathbf{B}_2$ is the submatrix corresponding to $\mathbf{x}^{\text{aux}}$. Then,

$$\mathbf{B}^\top \mathbf{B} = \begin{pmatrix} \mathbf{B}_1^\top \mathbf{B}_1 & \mathbf{B}_1^\top \mathbf{B}_2 \\ \mathbf{B}_2^\top \mathbf{B}_1 & \mathbf{B}_2^\top \mathbf{B}_2 \end{pmatrix}.$$

Claim 4.7. $\frac{\alpha}{\alpha+1} \mathbf{A}^\top \mathbf{A}$ is the Schur complement of $\mathbf{B}^\top \mathbf{B}$.

Proof. By (4.2), for any fixed $\mathbf{x}^A$,

$$\min_{\mathbf{x}^{\text{aux}}} \mathbf{x}^\top \mathbf{B}^\top \mathbf{B} \mathbf{x} = \frac{\alpha}{\alpha+1} (\mathbf{x}^A)^\top \mathbf{A}^\top \mathbf{A} \mathbf{x}^A.$$

By Claim 4.6, $\frac{\alpha}{\alpha+1} \mathbf{A}^\top \mathbf{A}$ is the Schur complement of $\mathbf{B}^\top \mathbf{B}$. $\square$

4.3 Approximate solvers

We now show that approximate solvers for $\mathbf{B}$ also translate to approximate solvers for $\mathbf{A}$. The following Lemma about the length of projections involving integral matrices is crucial for our bounds.

Lemma 4.8. *Let $\mathbf{A}$ and $\mathbf{c}$ a matrix and a vector such that $\|\mathbf{Ac}\|_2^2$ is integral. If $\mathbf{A}^\top \mathbf{c} \neq \mathbf{0}$, then*

$$\|\Pi_{\mathbf{A}} \mathbf{c}\|_2^2 \geq \frac{1}{\sigma_{\max}^2(\mathbf{A})} = \frac{1}{\lambda_{\max}(\mathbf{A}^\top \mathbf{A})}.$$

Proof. As $\|\mathbf{Ac}\|_2^2$ is integral, the condition of $\mathbf{A}^\top \mathbf{c} \neq \mathbf{0}$ implies $\|\mathbf{A}^\top \mathbf{c}\|_2 \geq 1$. Consider the SVD of $\mathbf{A}$:

$$\mathbf{A} = \sum_i \sigma_i \mathbf{r}^i (\mathbf{s}^i)^\top.$$

Substituting it in for $\mathbf{A}^\top$ gives

$$\mathbf{A}^\top \mathbf{c} = \sum_i \left(\sigma_i (\mathbf{r}^i)^\top \mathbf{c} \right) \mathbf{s}^i,$$

and we will use $\alpha_i = \sigma_i (\mathbf{r}^i)^\top \mathbf{c}$ to denote the coefficients of $\mathbf{c}$ against the singular vectors of $\mathbf{A}$. Note that $\sigma_i = 0$ implies $\alpha_i = 0$. The norm condition on $\mathbf{A}^\top \mathbf{c}$ also means $\sum_i \alpha_i^2 = \sum_{i:\sigma_i \neq 0} \alpha_i^2 \geq 1$, which then gives

$$\|\Pi_{\mathbf{A}} \mathbf{c}\|_2^2 = \mathbf{c}^\top \mathbf{A} \left(\mathbf{A}^\top \mathbf{A} \right)^\dagger \mathbf{A}^\top \mathbf{c} = \left(\sum_i \alpha_i (\mathbf{s}^i)^\top \right) \left(\sum_{i:\sigma_i \neq 0} \frac{1}{\sigma_i^2} \mathbf{s}^i (\mathbf{s}^i)^\top \right) \left(\sum_i \alpha_i \mathbf{s}^i \right) = \sum_{i:\sigma_i \neq 0} \frac{\alpha_i^2}{\sigma_i^2} \geq \frac{1}{\sigma_{\max}^2(\mathbf{A})}.$$

This completes the proof. $\square$

Lemma 4.9. *Let $\mathbf{Bx}^B = \mathbf{c}^B$ be the linear system returned by a call to $\text{REDUCE } \mathcal{G}_{z,2} \text{ TO } \mathcal{MC}_2(\mathbf{A}, \mathbf{c}^A, \epsilon^A)$, and let ϵ^B be the error parameter returned by this call. Let $\mathbf{x}^B$ be a vector such that*

$$\|\mathbf{Bx}^B - \Pi_{\mathbf{B}} \mathbf{c}^B\|_2 \leq \epsilon^B \|\Pi_{\mathbf{B}} \mathbf{c}^B\|_2. \quad (11)$$

Let $\tilde{\mathbf{x}}^A$ be the vector returned by a call to $\text{MAPSOLN } \mathcal{MC}_2 \text{ TO } \mathcal{G}_{z,2}(\mathbf{A}, \mathbf{B}, \mathbf{c}^A, \mathbf{x}^B)$. Then,

$$\|\mathbf{A}\tilde{\mathbf{x}}^A - \Pi_{\mathbf{A}} \mathbf{c}^A\|_2 \leq \epsilon^A \|\Pi_{\mathbf{A}} \mathbf{c}^A\|_2.$$

Proof. We first consider the case $\mathbf{A}^\top \mathbf{c}^A = \mathbf{0}$, then Algorithm 3 returns $\tilde{\mathbf{x}}^A = \mathbf{0}$, which gives

$$\|\mathbf{A}\tilde{\mathbf{x}}^A - \Pi_{\mathbf{A}} \mathbf{c}^A\|_2 = 0,$$

and completes the proof for this case. When $\mathbf{A}^\top \mathbf{c}^A \neq \mathbf{0}$, the Algorithm 3 returns $\tilde{\mathbf{x}}^A = \mathbf{x}^A$, where $\mathbf{x}^B = \begin{pmatrix} \mathbf{x}^A \\ \mathbf{x}^{\text{aux}} \end{pmatrix}$. Let

$$\mathbf{x}^{B*} \stackrel{\text{def}}{=} \begin{pmatrix} \mathbf{x}^{A*} \\ \mathbf{x}^{\text{aux}*} \end{pmatrix} \in \arg \min_{\mathbf{x}} \|\mathbf{Bx} - \mathbf{c}^B\|_2.$$

By Fact 2.2, the condition (11) is equivalent to

$$\|\mathbf{x}^B - \mathbf{x}^{B*}\|_{\mathbf{B}^\top \mathbf{B}} \leq \epsilon^B \|\Pi_{\mathbf{B}} \mathbf{c}^B\|_2.$$

By Claim (4.2), $\mathbf{x}^{A*} \in \arg \min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{c}^A\|_2$.

We now upper bound the error of $\|\mathbf{x}^A - \mathbf{x}^{A*}\|_{\mathbf{A}^\top \mathbf{A}}$. Note

$$\mathbf{x}^B - \mathbf{x}^{B*} = \begin{pmatrix} \mathbf{x}^A - \mathbf{x}^{A*} \\ \mathbf{x}^{\text{aux}} - \mathbf{x}^{\text{aux}*} \end{pmatrix}.$$

By Claim 4.6 and Claim 4.7,

$$\|\mathbf{x}^A - \mathbf{x}^{A*}\|_{\mathbf{A}^\top \mathbf{A}}^2 \leq \left(1 + \frac{1}{\alpha}\right) \|\mathbf{x}^B - \mathbf{x}^{B*}\|_{\mathbf{B}^\top \mathbf{B}}^2.$$

Then, we lower bound $\|\mathbf{\Pi}_A \mathbf{c}^A\|_2$. By Claim 4.2,

$$\|\mathbf{c}^A - \mathbf{\Pi}_A \mathbf{c}^A\|_2^2 = \left(1 + \frac{1}{\alpha}\right) \|\mathbf{c}^B - \mathbf{\Pi}_B \mathbf{c}^B\|_2^2.$$

Since $\mathbf{\Pi}_A \mathbf{c}^A \perp \mathbf{c}^A - \mathbf{\Pi}_A \mathbf{c}^A$, and $\mathbf{\Pi}_B \mathbf{c}^B \perp \mathbf{c}^B - \mathbf{\Pi}_B \mathbf{c}^B$, we have

$$\begin{aligned} \|\mathbf{\Pi}_A \mathbf{c}^A\|_2^2 &= \|\mathbf{c}^A\|_2^2 - \|\mathbf{c}^A - \mathbf{\Pi}_A \mathbf{c}^A\|_2^2 \\ &= \|\mathbf{c}^B\|_2^2 - \|\mathbf{c}^B - \mathbf{\Pi}_B \mathbf{c}^B\|_2^2 - \frac{1}{\alpha + 1} \|\mathbf{c}^A - \mathbf{\Pi}_A \mathbf{c}^A\|_2^2 \\ &= \|\mathbf{\Pi}_B \mathbf{c}^B\|_2^2 - \frac{1}{\alpha + 1} \|\mathbf{c}^A - \mathbf{\Pi}_A \mathbf{c}^A\|_2^2. \end{aligned}$$

Thus,

$$\|\mathbf{\Pi}_B \mathbf{c}^B\|_2^2 = \|\mathbf{\Pi}_A \mathbf{c}^A\|_2^2 + \frac{1}{\alpha + 1} \|\mathbf{c}^A - \mathbf{\Pi}_A \mathbf{c}^A\|_2^2.$$

Since $\mathbf{A}^\top \mathbf{c}^A \neq \mathbf{0}$, and because $\mathbf{A}^\top \mathbf{c}^A$ is integral, by Lemma 4.8, we have

$$\|\mathbf{\Pi}_A \mathbf{c}^A\|_2^2 \geq \frac{1}{\sigma_{\max}^2(\mathbf{A})}.$$

Thus,

$$\|\mathbf{\Pi}_B \mathbf{c}^B\|_2^2 \leq \|\mathbf{\Pi}_A \mathbf{c}^A\|_2^2 + \frac{1}{\alpha + 1} \|\mathbf{c}^A\|_2^2 \leq \left(1 + \frac{\|\mathbf{c}^A\|_2^2 \sigma_{\max}^2(\mathbf{A})}{\alpha + 1}\right) \|\mathbf{\Pi}_A \mathbf{c}^A\|_2^2.$$

Therefore,

$$\|\mathbf{A}\mathbf{x}^A - \mathbf{\Pi}_A \mathbf{c}^A\|_2 \leq \left(1 + \frac{1}{\alpha}\right)^{1/2} \left(1 + \frac{\|\mathbf{c}^A\|_2^2 \sigma_{\max}^2(\mathbf{A})}{\alpha + 1}\right)^{1/2} \epsilon^B \|\mathbf{\Pi}_A \mathbf{c}^A\|_2.$$

Finally, by the setting

$$\epsilon^B = \frac{\epsilon^A}{\left(1 + \frac{1}{\alpha}\right)^{1/2} \left(1 + \frac{\|\mathbf{c}^A\|_2^2 \sigma_{\max}^2(\mathbf{A})}{\alpha + 1}\right)^{1/2}},$$

we have

$$\|\mathbf{A}\mathbf{x}^A - \mathbf{\Pi}_A \mathbf{c}^A\|_2 \leq \epsilon^A \|\mathbf{\Pi}_A \mathbf{c}^A\|_2.$$

This completes the proof. $\square$

4.4 Bounding Condition Number of the New Matrix

In this section, we show that the condition number of $\mathbf{B}$ is upper bounded by the condition number of $\mathbf{A}$ with a $\text{poly}(n)$ multiplicative factor.

We first characterize the null space of $\mathbf{B}$. Recall that in Equation (2), we write $\mathbf{x}^B = \begin{pmatrix} \mathbf{x}^A \\ \mathbf{x}^{\text{aux}} \end{pmatrix}$. In the following, we will employ a *different indexing* of $\mathbf{x}^{\text{aux}}$ than the one defined at the beginning of Section 4. We also reorder the columns of $\mathbf{B}$ so that $\mathbf{B}\mathbf{x}^B = \mathbf{0}$ represents the same equations as before. For appropriately chosen indices g_1 and g_2 , we define

1. $\mathbf{x}_{1:g_1}^{\text{aux}}$ corresponds to the $\mathbf{u}$ -coordinates of the auxiliary variables created in $\mathcal{MC}_2$ -gadgets whose coefficients are nonzero.
2. $\mathbf{x}_{g_1+1:g_2}^{\text{aux}}$ corresponds to the $\mathbf{v}$ -coordinates of the auxiliary variables created in $\mathcal{MC}_2$ -gadgets whose coefficients are nonzero.
3. $\mathbf{x}_{g_2+1:n_B-n_A}^{\text{aux}}$ corresponds to the coordinates of the auxiliary variables created in $\mathcal{MC}_2$ -gadgets whose coefficients are zero.

Using this ordering, for $0 \leq i \leq (g_2 - g_1)/6$, $\mathbf{x}_{g_1+6i+1:g_1+6i+6}^{\text{aux}}$ corresponds to the $\mathbf{v}$ -coordinates of the auxiliary variables with non-zero coefficients in a single $\mathcal{MC}_2$ -gadget.

Given any fixed $\mathbf{x}^A \in \text{null}(\mathbf{A})$, we extend $\mathbf{x}^A$ to a vector in dimension n_B :

$$\mathbf{p}(\mathbf{x}^A) \stackrel{\text{def}}{=} \begin{pmatrix} \mathbf{x}^A \\ \mathbf{x}^{\text{aux}} \end{pmatrix}. \quad (12)$$

We assign the values of the auxiliary variables of $\mathbf{p}(\mathbf{x}^A)$ in the order that they are created in Algorithm 1. In a $\mathcal{MC}_2$ -gadget, suppose the values of variables $\mathbf{u}_{j_k}$ and $\mathbf{u}_{l_k}$ have already been assigned. Let $\mathbf{u}_t, \mathbf{u}_{t+i}, \mathbf{v}_{t+j}, 3 \leq i \leq 6, 1 \leq j \leq 6$ be the auxiliary variables created in this gadget. We assign values as follows,

$$\begin{aligned} \mathbf{u}_t &= (\mathbf{u}_{j_k} + \mathbf{u}_{l_k})/2, \\ \mathbf{u}_{t+3} &= \mathbf{u}_t, \\ \mathbf{u}_{t+4} &= \mathbf{u}_{j_k}, \\ \mathbf{u}_{t+5} &= \mathbf{u}_t, \\ \mathbf{u}_{t+6} &= \mathbf{u}_{l_k}, \\ \mathbf{v}_{t+1} &= 0, \\ \mathbf{v}_{t+2} &= \mathbf{u}_t - \mathbf{u}_{j_k}, \\ \mathbf{v}_{t+3} &= 0, \\ \mathbf{v}_{t+4} &= \mathbf{u}_t - \mathbf{u}_{j_k}, \\ \mathbf{v}_{t+5} &= \mathbf{u}_t - \mathbf{u}_{j_k}, \\ \mathbf{v}_{t+6} &= 0. \end{aligned}$$

This gives the first g_2 entries of $\mathbf{x}^{\text{aux}}$, and we set all the rest of the entries to be 0.

Let $\mathbf{e}_i \in \mathbb{R}^{(g_2-g_1)/6}$ be the i th standard basis vector and $\mathbf{1}$ be the all-one vector in 6 dimensions. We define $\mathbf{p}^i \in \mathbb{R}^{n_B}$ to be a vector whose nonzero entries are given by

$$\mathbf{p}_{n_A+g_1+1:n_A+g_2}^i = \mathbf{e}_i \otimes \mathbf{1}. \quad (13)$$

Let $\mathbf{e}_j \in \mathbb{R}^{n_B - n_A - g_2}$ is the j th standard basis vector. We define $\mathbf{q}^j \in \mathbb{R}^{n_B}$ to be a vector whose nonzero entries are given by

$$\mathbf{q}_{n_A + g_2 + 1:n_B}^j = \mathbf{e}_j. \quad (14)$$

Lemma 4.10. $\text{null}(\mathbf{B}) = \text{Span}(\mathbf{p}(\mathbf{x}^A), \mathbf{p}^i, \mathbf{q}^j : \mathbf{x}^A \in \text{null}(\mathbf{A}), 1 \leq i \leq \frac{g_2 - g_1}{6}, 1 \leq j \leq n_B - g_2).$

Proof. Let $\mathbf{S}$ be the subspace of $\text{Span}(\mathbf{p}(\mathbf{x}^A), \mathbf{p}^i, \mathbf{q}^j : \mathbf{x}^A \in \text{null}(\mathbf{A}), 1 \leq i \leq \frac{g_2 - g_1}{6}, 1 \leq j \leq n_B - g_2).$. According to definitions of the vectors, we can check that $\mathbf{S} \subseteq \text{null}(\mathbf{B})$.

It remains to show that $\mathbf{S} \supseteq \text{null}(\mathbf{B})$. Let $\mathbf{x} \in \text{null}(\mathbf{B})$. By Claim 4.2 with $\mathbf{c}^A = \mathbf{0}$, we have

$$\mathbf{A}\mathbf{x}^A = \mathbf{0},$$

that is, $\mathbf{x}^A \in \text{null}(\mathbf{A})$. According to the $\mathcal{MC}_2$ -gadget constraints in Algorithm 2, we have

$$\mathbf{v}_{t+1} = \mathbf{v}_{t+3} = \mathbf{v}_{t+6} = \gamma, \text{ and } \mathbf{v}_{t+2} = \mathbf{v}_{t+4} = \mathbf{v}_{t+5} = \theta,$$

where $\gamma - \theta = (\mathbf{u}_{j_k} - \mathbf{u}_{j_l})/2$. Besides, since all entries in $\mathbf{x}_{n_A + g_2 + 1:n_B}$ have zero coefficients, they are free to choose. Thus, $\mathbf{x} \in \mathbf{S}$, that is, $\mathbf{S} \supseteq \text{null}(\mathbf{B})$.

This completes the proof. $\square$

We upper bound the largest singular value of $\mathbf{B}$ in the following claim.

Lemma 4.11. $\lambda_{\max}(\mathbf{B}^\top \mathbf{B}) = O\left(\alpha \text{nnz}(\mathbf{A})^2 \|\mathbf{A}\|_\infty^2 \log^2 \|\mathbf{A}\|_\infty\right).$

Proof. By the Courant-Fischer Theorem,

$$\lambda_{\max}(\mathbf{B}^\top \mathbf{B}) = \max_{\mathbf{x}} \frac{\mathbf{x}^\top \mathbf{B}^\top \mathbf{B} \mathbf{x}}{\mathbf{x}^\top \mathbf{x}}.$$

We write $\mathbf{x}$ as $(\mathbf{u}_1, \mathbf{v}_2, \dots, \mathbf{u}_{n_B}, \mathbf{v}_{n_B})$, so that each $(\mathbf{u}_i, \mathbf{v}_i)$ corresponds to the two coordinates of vertex i in the graph. Expanding the right hand side, we get

$$\frac{\sum_{(i,j) \in E_1} w_{(i,j)} (\mathbf{u}_i - \mathbf{u}_j)^2 + \sum_{(i,j) \in E_2} w_{(i,j)} (\mathbf{v}_i - \mathbf{v}_j)^2 + \sum_{(i,j) \in E_{1+2}} w_{(i,j)} (\mathbf{u}_i - \mathbf{v}_i - (\mathbf{u}_j - \mathbf{v}_j))^2}{\sum_i \mathbf{u}_i^2 + \mathbf{v}_i^2}$$

where $w_{(i,j)}$'s are the edge weights. Let

$$w_{\max} = \max_{(i,j) \in E_1 \cup E_2 \cup E_{1+2}} w_{(i,j)}.$$

According to our construction of $\mathbf{B}$ in Algorithm 1,

$$w_{\max} \leq \alpha \text{nnz}(\mathbf{B}) \|\mathbf{A}\|_\infty^2. \quad (15)$$

Thus, $\lambda_{\max}(\mathbf{B}^\top \mathbf{B})$ is at most

$$w_{\max} \max_{\mathbf{u}} \frac{\sum_{(i,j) \in E_1} (\mathbf{u}_i - \mathbf{u}_j)^2 + \sum_{(i,j) \in E_2} (\mathbf{v}_i - \mathbf{v}_j)^2 + \sum_{(i,j) \in E_{1+2}} (\mathbf{u}_i - \mathbf{v}_i - (\mathbf{u}_j - \mathbf{v}_j))^2}{\sum_i \mathbf{u}_i^2 + \mathbf{v}_i^2}.$$

We upper bound each term of the numerator. Let d_1, d_2, d_{1+2} be the maximum vertex degree of the graphs G_1, G_2, G_{1+2} , respectively. By Cauchy-Schwarz inequality,

$$\sum_{(i,j) \in E_1} (\mathbf{u}_i - \mathbf{u}_j)^2 \leq 2 \sum_{(i,j) \in E_1} \mathbf{u}_i^2 + \mathbf{u}_j^2 \leq 2d_1 \left(\sum_i \mathbf{u}_i^2 + \mathbf{v}_i^2 \right),$$

$$\sum_{(i,j) \in E_2} (\mathbf{v}_i - \mathbf{v}_j)^2 \leq 2d_2 \left(\sum_i \mathbf{u}_i^2 + \mathbf{v}_i^2 \right),$$

and

$$\begin{aligned} \sum_{(i,j) \in E_{1+2}} (\mathbf{u}_i - \mathbf{v}_i - (\mathbf{u}_j - \mathbf{v}_j))^2 &\leq 4 \sum_{(i,j) \in E_{1+2}} \mathbf{u}_i^2 + \mathbf{v}_i^2 + \mathbf{u}_j^2 + \mathbf{v}_j^2 \\ &\leq 4d_{1+2} \left(\sum_i \mathbf{u}_i^2 + \mathbf{v}_i^2 \right). \end{aligned}$$

Plugging the above inequalities to the expansion, we have

$$\begin{aligned} \lambda_{\max}(\mathbf{B}^\top \mathbf{B}) &\leq w_{\max} \cdot 8 \max\{d_1, d_2, d_{1+2}\} \\ &\leq 8w_{\max} \text{nnz}(\mathbf{B}). \end{aligned}$$

By the upper bound of $\text{nnz}(\mathbf{B})$ in Lemma 4.1 and the upper bound of $w_{\max}$ in Equation (15), we have

$$\lambda_{\max}(\mathbf{B}^\top \mathbf{B}) \leq O\left(\alpha \text{nnz}(\mathbf{A})^2 \|\mathbf{A}\|_\infty^2 \log^2 \|\mathbf{A}\|_\infty\right).$$

This completes the proof. $\square$

Recall that, for a vector $\mathbf{x} \in \mathbb{R}^{n_B}$, $\mathbf{x}_{n_A+1:n_A+g_1}$ corresponds to the auxiliary $\mathbf{u}$ -variables with non-zero coefficients in $\mathcal{MC}_2$ -gadgets, $\mathbf{x}_{n_A+1+g_1+6i:n_A+6+g_1+6i}$ for $0 \leq i \leq (g_2 - g_1)/6$ corresponds to the auxiliary $\mathbf{v}$ -variables with non-zero coefficients in a single $\mathcal{MC}_2$ -gadget, and $\mathbf{x}_{n+g_2+1:n_B}$ corresponds to the variables with zero coefficient.

Lemma 4.12. *Let $0 \leq \epsilon \leq (100n_B \log \|\mathbf{A}\|_\infty)^{-1}$. Let $\mathbf{x}^B \in \mathbb{R}^{n_B}$ satisfying*

1. $\mathbf{x}_{n_A+g_2+1:n_B}^B = \mathbf{0}$,
2. $\mathbf{x}_{n_A+g_1+6i+1}^B = 0, \forall 0 \leq i \leq (g_2 - g_1)/6$, and
3. $\|\mathbf{B}\mathbf{x}^B\|_2 \leq \epsilon \|\mathbf{x}^B\|_2$.

Then,

$$\|\mathbf{A}\mathbf{x}^A\|_2 \leq 4 \left(1 + \frac{1}{\alpha}\right)^{1/2} n_B^{1/2} \epsilon \|\mathbf{x}^A\|_2.$$

Proof. We first show that under the conditions of the Lemma,

$$\|\mathbf{x}^A\|_\infty \geq \frac{1}{4} \|\mathbf{x}^B\|_\infty. \tag{16}$$

Let $\delta \stackrel{\text{def}}{=} \epsilon \|\mathbf{x}^B\|_2$. By the 3rd condition, each entry of $\mathbf{B}\mathbf{x}^B$ has absolute value at most δ .

We first show that all $\mathbf{u}$ -variables in $\mathbf{x}^{\text{aux}}$ cannot be large. According to Algorithm 1, for each row of $\mathbf{B}$, all nonzero coefficients have same absolute value, which is at least 1. Based on this fact

and the $\mathcal{MC}_2$ -gadget constructed in Algorithm 2, we have

$$\begin{aligned}
\frac{\mathbf{u}_{j_k} + \mathbf{u}_{l_k}}{2} - 5\delta &\leq \mathbf{u}_t \leq \frac{\mathbf{u}_{j_k} + \mathbf{u}_{l_k}}{2} + 5\delta \\
\frac{\mathbf{u}_{j_k} + \mathbf{u}_{l_k}}{2} - 6\delta &\leq \mathbf{u}_{t+3} \leq \frac{\mathbf{u}_{j_k} + \mathbf{u}_{l_k}}{2} + 6\delta \\
\mathbf{u}_{j_k} - \delta &\leq \mathbf{u}_{t+4} \leq \mathbf{u}_{j_k} + \delta \\
\frac{\mathbf{u}_{j_k} + \mathbf{u}_{l_k}}{2} - 6\delta &\leq \mathbf{u}_{t+5} \leq \frac{\mathbf{u}_{j_k} + \mathbf{u}_{l_k}}{2} + 6\delta \\
\mathbf{u}_{l_k} - \delta &\leq \mathbf{u}_{t+6} \leq \mathbf{u}_{l_k} + \delta
\end{aligned}$$

where $\mathbf{u}_{j_k}, \mathbf{u}_{l_k}$ being paired-and-replaced, and all others are entries of $\mathbf{x}^{\text{aux}}$. By the triangle inequality,

$$|\mathbf{u}_t|, |\mathbf{u}_{t+k}| \leq \frac{1}{2}(|\mathbf{u}_{j_k}| + |\mathbf{u}_{l_k}|) + 6\delta, \quad \forall 3 \leq k \leq 6.$$

Note that the sum of coefficients on both sides are equal. We can repeat this type of substitution on the right hand side until $\mathbf{u}_{j_k}$ and $\mathbf{u}_{l_k}$ are variables of $\mathbf{x}^A$. At the i th iteration of Algorithm 1 line 20, we have

$$|\mathbf{u}_t|, |\mathbf{u}_{t+k}| \leq \sum_j \alpha_j |\mathbf{x}_j^A| + 6r\delta, \quad \forall 3 \leq k \leq 6,$$

where $\alpha_j \geq 0$ and $\sum_j \alpha_j = 1$. By the Hölder inequality,

$$|\mathbf{u}_t|, |\mathbf{u}_{t+k}| \leq \|\mathbf{x}^A\|_\infty + 6r\delta, \quad \forall 3 \leq k \leq 6. \quad (17)$$

We then argue that all $\mathbf{v}$ -variables in $\mathbf{x}^{\text{aux}}$ cannot be large. Note that by the 2nd condition, in the $\mathcal{MC}_2$ -gadget, we have $\mathbf{v}_{t+1} = 0$. According to the equations in Algorithm 2, we have

$$\begin{aligned}
\mathbf{u}_{t+3} - \mathbf{u}_{t+4} - 3\delta &\leq \mathbf{v}_{t+2} \leq \mathbf{u}_{t+3} - \mathbf{u}_{t+4} + 3\delta \\
-\delta &\leq \mathbf{v}_{t+3} \leq \delta \\
\mathbf{u}_{t+3} - \mathbf{u}_{t+4} - 2\delta &\leq \mathbf{v}_{t+4} \leq \mathbf{u}_{t+3} - \mathbf{u}_{t+4} + 2\delta \\
\mathbf{u}_{t+3} - \mathbf{u}_{t+4} - 4\delta &\leq \mathbf{v}_{t+5} \leq \mathbf{u}_{t+3} - \mathbf{u}_{t+4} + 4\delta \\
-\delta &\leq \mathbf{v}_{t+6} \leq \delta
\end{aligned}$$

By the triangle inequality,

$$|\mathbf{v}_{t+k}| \leq |\mathbf{u}_{t+3}| + |\mathbf{u}_{t+4}| + 4\delta, \quad \forall 2 \leq k \leq 6.$$

By (17), at the r th iteration of Algorithm 1 line 20,

$$|\mathbf{v}_{t+k}| \leq 2\|\mathbf{x}^A\|_\infty + 12r\delta, \quad \forall 2 \leq k \leq 6.$$

Since there are at most $\log \|\mathbf{A}\|_\infty$ iterations, the above inequality together with Equation (17) implies

$$\|\mathbf{x}^{\text{aux}}\|_\infty \leq 2\|\mathbf{x}^A\|_\infty + 12\delta \log \|\mathbf{A}\|_\infty.$$

Adding $\|\mathbf{x}^A\|_\infty$ on both sides and substituting $\delta = \epsilon \|\mathbf{x}^B\|_2$ gives

$$\begin{aligned}
\|\mathbf{x}^B\|_\infty &\leq 3\|\mathbf{x}^A\|_\infty + 12\epsilon \|\mathbf{x}^B\|_2 \log \|\mathbf{A}\|_\infty \\
&\leq 3\|\mathbf{x}^A\|_\infty + 12\epsilon \sqrt{n_B} \|\mathbf{x}^B\|_\infty \log \|\mathbf{A}\|_\infty.
\end{aligned}$$

Given $\epsilon \leq (100\sqrt{n_B} \log \|\mathbf{A}\|_\infty)^{-1}$, we have Equation (16). This further implies

$$\|\mathbf{x}^B\|_2 \leq 4\sqrt{n_B} \|\mathbf{x}^A\|_2. \quad (18)$$

By Claim 4.2 with $\mathbf{c}^B = \mathbf{0}$, we have

$$\|\mathbf{A}\mathbf{x}^A\|_2 \leq \sqrt{\frac{\alpha+1}{\alpha}} \|\mathbf{B}\mathbf{x}^B\|_2.$$

By the 3rd condition and the bound (18), we have

$$\|\mathbf{A}\mathbf{x}^A\|_2 \leq 4 \left(1 + \frac{1}{\alpha}\right)^{1/2} n_B^{1/2} \epsilon \|\mathbf{x}^A\|_2.$$

This completes the proof. $\square$

We use the above lemma to lower bound the smallest nonzero eigenvalue of $\mathbf{B}^\top \mathbf{B}$.

Lemma 4.13. $\lambda_{\min}(\mathbf{B}^\top \mathbf{B}) = \Omega\left(\frac{\min\{1, \lambda_{\min}(\mathbf{A}^\top \mathbf{A})\}}{n_B^2}\right).$

Proof. Let

$$\delta \stackrel{\text{def}}{=} \min\{1, \lambda_{\min}(\mathbf{A}^\top \mathbf{A})\}, \quad (19)$$

and

$$\epsilon \stackrel{\text{def}}{=} \frac{\delta}{1616(1+\alpha^{-1})n_B^2}. \quad (20)$$

The goal is to prove that $\lambda_{\min}(\mathbf{B}^\top \mathbf{B}) \geq \epsilon$. Assume by contradiction, there exists an $\mathbf{x} \perp \text{null}(\mathbf{B})$ such that

$$(\mathbf{x}^B)^\top \mathbf{B}^\top \mathbf{B} \mathbf{x}^B < \epsilon (\mathbf{x}^B)^\top \mathbf{x}^B. \quad (21)$$

We show a contradiction by case analysis according to whether $\mathbf{x}^A$ is orthogonal to the null space of $\mathbf{A}$.

Case 1: Suppose $\mathbf{x}^A \perp \text{null}(\mathbf{A})$.

Recall that $\mathbf{x}_{n_A+1:n_A+g_1}^B$ corresponds to the auxiliary $\mathbf{u}$ -variables in the constraints, $\mathbf{x}_{n_A+g_1+1:n_A+g_2}^B$ corresponds to the auxiliary $\mathbf{v}$ -variables in the constraints, and $\mathbf{x}_{n_A+g_2+1:n_B}^B$ corresponds to the variables with zero coefficient. By Lemma 4.10, we have $\mathbf{x}^B \perp \mathbf{q}^j, \forall 1 \leq j \leq n_A - g_2$, where $\mathbf{q}^j$ is defined in (14). Thus, $\mathbf{x}_{n_A+g_2+1:n_B}^B = \mathbf{0}$.

Let $\mathbf{r} \stackrel{\text{def}}{=} \sum_{0 \leq i \leq (g_2-g_1)/6} \mathbf{x}_{n_A+g_1+6i+1}^B \mathbf{p}^i$, where $\mathbf{p}^i$ is defined in (13). By Lemma 4.10, $\mathbf{r} \in \text{null}(\mathbf{B})$. Let $\mathbf{y}^B \stackrel{\text{def}}{=} \mathbf{x}^B - \mathbf{r}$. Similarly, we write

$$\mathbf{y}^B = \begin{pmatrix} \mathbf{y}^A \\ \mathbf{y}^{\text{aux}} \end{pmatrix},$$

where $\mathbf{y}^A$ corresponds to original variables and $\mathbf{y}^{\text{aux}}$ corresponds to auxiliary variables. Note $\mathbf{y}^A = \mathbf{x}^A$. Since $\mathbf{r} \in \text{null}(\mathbf{B})$, we have

$$(\mathbf{y}^B)^\top \mathbf{B}^\top \mathbf{B} \mathbf{y}^B = (\mathbf{x}^B)^\top \mathbf{B}^\top \mathbf{B} \mathbf{x}^B.$$

Since $\mathbf{x}^B \perp \mathbf{r}$, we have

$$(\mathbf{y}^B)^\top \mathbf{y}^B = (\mathbf{x}^B)^\top \mathbf{x}^B + \mathbf{r}^\top \mathbf{r} \geq (\mathbf{x}^B)^\top \mathbf{x}^B.$$

By assumption (21), we have

$$(\mathbf{y}^B)^\top \mathbf{B}^\top \mathbf{B} \mathbf{y}^B < \epsilon (\mathbf{y}^B)^\top \mathbf{y}^B.$$

By our definition of $\mathbf{y}^B$, $\mathbf{y}^B$ satisfies the conditions of Lemma 4.12. By Lemma 4.12 and the definition of ϵ in Equation (20), we have

$$(\mathbf{x}^A)^\top \mathbf{A}^\top \mathbf{A} \mathbf{x}^A < \delta (\mathbf{x}^A)^\top \mathbf{x}^A.$$

Since $\mathbf{x}^A \perp \text{null}(\mathbf{A})$ and the definition of δ in Equation 19, we have

$$\lambda_{\min}(\mathbf{A}^\top \mathbf{A}) < \delta = \min \left\{ 1, \lambda_{\min}(\mathbf{A}^\top \mathbf{A}) \right\},$$

which is a contradiction.

Case 2: Suppose $\mathbf{x}^A \in \text{null}(\mathbf{A})$.

Recall that $\mathbf{p}(\mathbf{x}^A)$ is the extension of vector $\mathbf{x}^A$ defined in Equation (12). Let $\mathbf{r} = \sum_{0 \leq i \leq (g_2 - g_1)/6} \alpha_i \mathbf{p}^i$ be a vector such that

$$\mathbf{y}^B \stackrel{\text{def}}{=} \mathbf{x}^B - \mathbf{p}(\mathbf{x}^A) - \mathbf{r}$$

satisfying $\mathbf{y}_{n_A + g_1 + 6i + 1}^B = \mathbf{0}, \forall 0 \leq i < (g_2 - g_1)/6$. By this definition, we have

$$\mathbf{y}^A = \mathbf{0}. \tag{22}$$

Since $\mathbf{p}(\mathbf{x}^A) + \mathbf{r}$ is in the null space of $\mathbf{B}$, by assumption (21),

$$(\mathbf{y}^B)^\top \mathbf{B}^\top \mathbf{B} \mathbf{y}^B < \epsilon (\mathbf{y}^B)^\top \mathbf{y}^B.$$

Note $\mathbf{y}^B$ satisfies the conditions of Lemma 4.12. By Equations (16) and (22), we have

$$\|\mathbf{y}^B\|_\infty = 0.$$

This implies that $\mathbf{x}^B = \mathbf{p}(\mathbf{x}^A) + \mathbf{r}$ is in the null space of $\mathbf{B}$, which contradicts that $\mathbf{x}^B \perp \text{null}(\mathbf{B})$.

Case 3: Suppose $\mathbf{x}^A = \tilde{\mathbf{z}} + \hat{\mathbf{z}}$, where $\tilde{\mathbf{z}} \neq \mathbf{0}$ is in $\text{null}(\mathbf{A})$ and $\hat{\mathbf{z}} \neq \mathbf{0}$ is orthogonal to $\text{null}(\mathbf{A})$.

Similarly, let $\mathbf{r}_1 = \sum_i \alpha_i \mathbf{p}^i$ such that $\mathbf{y}^B \stackrel{\text{def}}{=} \mathbf{x}^B - \mathbf{r}_1$ satisfying $\mathbf{y}_{n_A + g_1 + 6i + 1}^B = \mathbf{0}, \forall i$. By this definition, we have

$$\mathbf{y}^A = \mathbf{x}^A.$$

Then by assumption (21),

$$(\mathbf{y}^B)^\top \mathbf{B}^\top \mathbf{B} \mathbf{y}^B < \epsilon (\mathbf{y}^B)^\top \mathbf{y}^B.$$

$\mathbf{y}^B$ satisfies the conditions of Lemma 4.12. By Lemma 4.12, we have

$$(\mathbf{y}^A)^\top \mathbf{A}^\top \mathbf{A} \mathbf{y}^A \leq 16 \left(1 + \frac{1}{\alpha} \right) n_B \epsilon (\mathbf{y}^A)^\top \mathbf{y}^A.$$

Since $\mathbf{y}^A = \tilde{\mathbf{z}} + \hat{\mathbf{z}}$ with $\tilde{\mathbf{z}} \in \text{null}(\mathbf{A})$ and $\hat{\mathbf{z}} \perp \text{null}(\mathbf{A})$, we have

$$\hat{\mathbf{z}}^\top \mathbf{A}^\top \mathbf{A} \hat{\mathbf{z}} \leq 16 \left(1 + \frac{1}{\alpha} \right) n_B \epsilon \left(\tilde{\mathbf{z}}^\top \tilde{\mathbf{z}} + \hat{\mathbf{z}}^\top \hat{\mathbf{z}} \right).$$

If $\hat{\mathbf{z}}^\top \hat{\mathbf{z}} > \tilde{\mathbf{z}}^\top \tilde{\mathbf{z}} / (100n_B)$, then we have a contradiction with (19) and we have done. Otherwise, we have

$$\|\hat{\mathbf{z}}\|_2^2 \leq \frac{\|\tilde{\mathbf{z}}\|_2^2}{100n_B}.$$

Let $\mathbf{r}_2 = \sum_i \beta_i \mathbf{p}^i$ such that

$$\mathbf{z}^B \stackrel{\text{def}}{=} \mathbf{y}^B - \mathbf{p}(\tilde{\mathbf{z}}) - \mathbf{r}_2$$

satisfies $\mathbf{z}_{n_A+g_1+6i+1}^B = 0, \forall i$. By assumption (21),

$$(\mathbf{z}^B)^\top \mathbf{B}^\top \mathbf{B} \mathbf{z}^B < \epsilon (\mathbf{z}^B)^\top \mathbf{z}^B.$$

Vector $\mathbf{z}^B$ satisfies the conditions of Lemma 4.12. By Equation (16), we have

$$\|\mathbf{z}^B\|_2 \leq 4\sqrt{n_B} \|\hat{\mathbf{z}}\|_2.$$

Since $\hat{\mathbf{z}}^\top \hat{\mathbf{z}} \leq \tilde{\mathbf{z}}^\top \tilde{\mathbf{z}} / (100n_B)$, we have

$$\|\mathbf{z}^B\|_2 \leq \|\tilde{\mathbf{z}}\|_2.$$

On the other hand,

$$\begin{aligned} \|\mathbf{z}^B\|_2^2 &= \|\mathbf{x}^B\|_2^2 + \|\mathbf{p}(\tilde{\mathbf{z}}) + \mathbf{r}_1 + \mathbf{r}_2\|_2^2 \\ &> \|\mathbf{p}(\tilde{\mathbf{z}}) + \mathbf{r}_1 + \mathbf{r}_2\|_2^2 \\ &\geq \|\mathbf{x}^A\|_2^2 \\ &= \|\tilde{\mathbf{z}}\|_2^2 + \|\hat{\mathbf{z}}\|_2^2 \\ &> \|\tilde{\mathbf{z}}\|_2^2. \end{aligned}$$

The first equality is due to $\mathbf{x}^B \perp (\mathbf{p}(\tilde{\mathbf{z}}) + \mathbf{r}_1 + \mathbf{r}_2)$, the third inequality is due to the $\mathbf{x}^A$ part of $\mathbf{r}_1 + \mathbf{r}_2$ is $\mathbf{0}$, and the fourth equality is due to that $\tilde{\mathbf{z}} \perp \hat{\mathbf{z}}$. Thus, we get a contradiction.

This completes the proof. $\square$

Lemma 4.11 and Lemma 4.13 immediately imply the following lemma.

Lemma 4.14. $\kappa(\mathbf{B}) = O\left(\frac{\text{nnz}(\mathbf{A})^2 \|\mathbf{A}\|_\infty \log^2 \|\mathbf{A}\|_\infty}{\min\{1, \sigma_{\min}(\mathbf{A})\}}\right).$

4.5 Putting it All Together

Proof of Lemma 3.5. We set $\alpha = 1$ in Algorithm 1.

By Lemma 4.1 and Claim B.1, we have

$$\text{nnz}(\mathbf{B}) = O(s \log(sU)).$$

According to our reduction in Algorithm 1, the largest entry and the smallest nonzero entry of the right hand side vector does not change. Besides, all nonzero entries of $\mathbf{B}$ have absolute value at least 1. The largest entry of $\mathbf{B}$ is upper bounded by Equation (15)

$$\sqrt{w_{\max}} = O\left(\sqrt{\text{nnz}(\mathbf{B})} \|\mathbf{A}\|_\infty\right).$$

By Claim B.1, we have

$$\sqrt{w_{\max}} = O\left(s^{3/2} U \log^{1/2}(sU)\right).$$

By Lemma 4.14 and Lemma B.1, B.3, we have

$$\kappa(\mathbf{B}) = O\left(\frac{\text{nnz}(\mathbf{A})^2 \|\mathbf{A}\|_\infty \log^2 \|\mathbf{A}\|_\infty}{\min\{1, \sigma_{\min}(\mathbf{A})\}}\right) = O(s^4 U^2 K \log^2(sU)).$$

By Lemma 4.9, we have

$$(\epsilon^{\mathbf{B}})^{-1} = (\epsilon^{\mathbf{A}})^{-1} O(1 + \|\mathbf{c}^{\mathbf{A}}\|_2 \sigma_{\max}(\mathbf{A})) = (\epsilon^{\mathbf{A}})^{-1} O(sU^2).$$

This completes the proof. $\square$

5 $\mathcal{MC}_2$ Efficiently Reducible to $\mathcal{MC}_2^{>0}$

The matrices generated by interior point methods (see Section 9 for details) are more restrictive than $\mathcal{MC}_2$: for every edge present, the weights of all three types of matrices are non-zero. Formally, the class of matrices $\mathcal{MC}_2^{>0}$ consists of matrices of the form

$$\mathbf{L}^1 \otimes \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} + \mathbf{L}^2 \otimes \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} + \mathbf{L}^{1+2} \otimes \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix},$$

where the non-zero support of all three matrices, $\mathbf{L}^1$, $\mathbf{L}^2$, and $\mathbf{L}^{1+2}$ are the same (but the weights may vary greatly). On the other hand, the matrices that we generate in Section 4 can be transformed into such a matrix by adding a small value, δ to the weight of all edges where one of the types of edges have non-zero support. We will describe this construction in Section 5.1, and bound its condition number in Section 5.2.

5.1 Construction

In this section, we show the construction from an instance of $\mathcal{MC}_2$ to an instance of $\mathcal{MC}_2^{>0}$. The strategy is to add extra edges with a sufficiently small weight, such that $\mathbf{L}^1$, $\mathbf{L}^2$, $\mathbf{L}^{1+2}$ have identical nonzero structure and the solution of the linear system does not change much.

The reduction from $\mathcal{MC}_2$ to $\mathcal{MC}_2^{>0}$, with pseudo-code in Algorithm 4, simply adds edges with weight δ to all the missing edges. The transformation of solutions of the corresponding instance of $\mathcal{MC}_2^{>0}$ back to a solution of the original $\mathcal{MC}_2$ instance in Algorithm 5 simply returns the same vector $\mathbf{x}$.

Algorithm 4 REDUCE $\mathcal{MC}_2$ TO $\mathcal{MC}_2^{>0}$

Input: $(\mathbf{B}, \mathbf{c}^B, \epsilon_1)$ where $\mathbf{B} \in \mathcal{MC}_2$ is an $m \times n$ matrix, $\mathbf{c}^B \in \mathbb{R}^m$, and $0 < \epsilon_1 < 1$.

Output: $(\mathbf{B}^{>0}, \mathbf{c}^{B^{>0}}, \epsilon_2)$ where $\mathbf{B}^{>0} \in \mathcal{MC}_2^{>0}$ is an $m' \times n'$ matrix, $\mathbf{c}^{B^{>0}} \in \mathbb{R}^{m'}$, and $0 < \epsilon_2 < 1$.

```

1:  $\delta \leftarrow \frac{\epsilon_1}{100\kappa(\mathbf{B})\sigma_{\max}(\mathbf{B})\|\mathbf{c}^B\|_2}$ 
2:  $\widehat{\mathcal{B}} \leftarrow \emptyset$  {New equations for  $\mathcal{MC}_2^{>0}$ }
3: for each pair of vertices  $i, j$  whose blocks are involved in some equation in  $\mathbf{B}$  do
4:   if  $\mathbf{B}$  does not contain a type 1 equation between  $i$  and  $j$  then
5:      $\widehat{\mathcal{B}} \leftarrow \widehat{\mathcal{B}} \cup \{\mathbf{u}_i - \mathbf{u}_j = 0\}$ .
6:   end if
7:   if  $\mathbf{B}$  does not contain a type 2 equation between  $i$  and  $j$  then
8:      $\widehat{\mathcal{B}} \leftarrow \widehat{\mathcal{B}} \cup \{\mathbf{v}_i - \mathbf{v}_j = 0\}$ .
9:   end if
10:  if  $\mathbf{B}$  does not contain a type 1 + 2 equation between  $i$  and  $j$  then
11:     $\widehat{\mathcal{B}} \leftarrow \widehat{\mathcal{B}} \cup \{\mathbf{u}_i + \mathbf{v}_i - (\mathbf{u}_j + \mathbf{v}_j) = 0\}$ .
12:  end if
13: end for
14: Let  $\widetilde{\mathbf{B}}$  be the coefficient matrix of equations in  $\widehat{\mathcal{B}}$ .
15: return
```

$$\mathbf{B}^{>0} = \begin{pmatrix} \mathbf{B} \\ \delta \widetilde{\mathbf{B}} \end{pmatrix}, \quad \mathbf{c}^{B^{>0}} = \begin{pmatrix} \mathbf{c}^B \\ \mathbf{0} \end{pmatrix}, \quad \epsilon_2 = \frac{\epsilon_1}{100}.$$

Algorithm 5 MAPSOLN $\mathcal{MC}_2^{>0}$ TO $\mathcal{MC}_2$

Input: $m \times n$ matrix $\mathbf{B} \in \mathcal{MC}_2$, $m' \times n'$ matrix $\mathbf{B}^{>0} \in \mathcal{MC}_2^{>0}$, vector $\mathbf{c}^B \in \mathbb{R}^m$ vector $\mathbf{x} \in \mathbb{R}^{n'}$.

Output: Vector $\mathbf{y} \in \mathbb{R}^n$.

```

1: if  $\mathbf{B}^\top \mathbf{c}^B = \mathbf{0}$  then
2:   return  $\mathbf{y} \leftarrow \mathbf{0}$ 
3: else
4:   return  $\mathbf{y} \leftarrow \mathbf{x}$ 
5: end if
```

Let $\mathbf{B}^{>0}\mathbf{x} = \mathbf{c}^{B^{>0}}$ be the linear system returned by a call to REDUCE $\mathcal{MC}_2$ TO $\mathcal{MC}_2^{>0}(\mathbf{B}, \mathbf{c}^B, \epsilon_1)$. As we only add up to two edges per original edge in $\mathbf{B}$, and do not introduce any new variables, the size of $\mathbf{B}^{>0}$ is immediate from this routine. According to the algorithms, $\mathbf{c}^{B^{>0}}$ is simply

$$\mathbf{c}^{B^{>0}} = \begin{pmatrix} \mathbf{c}^B \\ \mathbf{0} \end{pmatrix} = \begin{pmatrix} \mathbf{c}^A \\ \mathbf{0} \end{pmatrix}.$$

Note the two zero vectors in the above equation have different dimensions.

For a given matrix $\mathbf{B}$, we will choose the additive term to ensure non-zeros to be:

$$\delta \stackrel{\text{def}}{=} \frac{\epsilon_1}{100\kappa(\mathbf{B})\sigma_{\max}(\mathbf{B})\|\mathbf{c}^B\|_2}. \quad (23)$$

Note that we can bound $\kappa(\mathbf{B}), \sigma_{\max}(\mathbf{B})$ by condition number of $\mathbf{A}$ (i.e, the linear system instance of $\mathcal{G}$).

Lemma 5.1. $\text{nnz}(\mathbf{B}^{>0}) = O(\text{nnz}(\mathbf{B}))$. By our setting of δ in (23), the largest entry of $\mathbf{B}^{>0}$ does not change, the smallest entry of $\mathbf{B}^{>0}$ is at least δ .

Note that the addition of all three types of edges means that the null space of $\mathbf{B}^{>0}$ is now given by the connected components in its graph theoretic structure, and is likely significantly different from the null space of $\mathbf{B}$. In order to solve the Linear System Approximation Problem (LSA) problems

$$\begin{aligned} \min_{\mathbf{x}} \|\mathbf{B}\mathbf{x} - \mathbf{c}^{\mathbf{B}}\|_2, \\ \min_{\mathbf{x}} \|\mathbf{B}^{>0}\mathbf{x} - \mathbf{c}^{\mathbf{B}^{>0}}\|_2, \end{aligned}$$

we need to solve the two linear systems:

$$\mathbf{B}^\top \mathbf{B}\mathbf{x} = \mathbf{B}^\top \mathbf{c}^{\mathbf{B}}, \quad (24)$$

$$(\mathbf{B}^{>0})^\top \mathbf{B}^{>0}\mathbf{x} = (\mathbf{B}^{>0})^\top \mathbf{c}^{\mathbf{B}^{>0}}. \quad (25)$$

First note that the RHS the two equations are the same because:

$$(\mathbf{B}^{>0})^\top \mathbf{c}^{\mathbf{B}^{>0}} = \begin{pmatrix} \mathbf{B}^\top & \delta \tilde{\mathbf{B}}^\top \end{pmatrix} \begin{pmatrix} \mathbf{c}^{\mathbf{B}} \\ \mathbf{0} \end{pmatrix} = \mathbf{B}^\top \mathbf{c}^{\mathbf{B}}.$$

This means that for a sufficiently small choice of δ , the differences between the solutions of these two linear systems is small.

Lemma 5.2. *Let $\mathbf{B}^{>0}\mathbf{x} = \mathbf{c}^{\mathbf{B}^{>0}}$ be the linear system returned by a call to $\text{REDUCE}\mathcal{MC}_2\text{To}\mathcal{MC}_2^{>0}(\mathbf{B}, \mathbf{c}^{\mathbf{B}}, \epsilon_1)$. Let $\mathbf{x}^{\mathbf{B}^*} \in \arg \min_{\mathbf{x}} \|\mathbf{B}\mathbf{x} - \mathbf{c}^{\mathbf{B}}\|_2$ and $\mathbf{x}^{\mathbf{B}^*} \perp \text{null}(\mathbf{B})$. Let $\mathbf{x}^{\mathbf{B}^{>0}*} \in \arg \min_{\mathbf{x}} \|\mathbf{B}^{>0}\mathbf{x} - \mathbf{c}^{\mathbf{B}^{>0}}\|_2$. Then we have:*

$$\|\mathbf{x}^{\mathbf{B}^*} - \mathbf{x}^{\mathbf{B}^{>0}*}\|_{\mathbf{B}^\top \mathbf{B}} \leq O(\delta(1+\delta)\kappa(\mathbf{B})\|\mathbf{c}^{\mathbf{B}}\|_2).$$

Proof. The desired distance can be written as

$$\|\mathbf{x}^{\mathbf{B}^*} - \mathbf{x}^{\mathbf{B}^{>0}*}\|_{\mathbf{B}^\top \mathbf{B}} = \|\mathbf{B}(\mathbf{x}^{\mathbf{B}^*} - \mathbf{x}^{\mathbf{B}^{>0}*})\|_2.$$

Also, the optimality of $\mathbf{x}^{\mathbf{B}^*}$ means that $\mathbf{B}\mathbf{x}^{\mathbf{B}^*} - \mathbf{c}^{\mathbf{B}}$ is perpendicular to anything in the rank space of $\mathbf{B}$, in particular,

$$\mathbf{B}(\mathbf{x}^{\mathbf{B}^*} - \mathbf{x}^{\mathbf{B}^{>0}*}) \perp \mathbf{B}\mathbf{x}^{\mathbf{B}^*} - \mathbf{c}^{\mathbf{B}},$$

which in turn gives:

$$\|\mathbf{B}(\mathbf{x}^{\mathbf{B}^*} - \mathbf{x}^{\mathbf{B}^{>0}*})\|_2^2 = \|\mathbf{B}\mathbf{x}^{\mathbf{B}^{>0}*} - \mathbf{c}^{\mathbf{B}}\|_2^2 - \|\mathbf{B}\mathbf{x}^{\mathbf{B}^*} - \mathbf{c}^{\mathbf{B}}\|_2^2. \quad (26)$$

We now bound the right hand side. Since $\mathbf{x}^{\mathbf{B}^{>0}*}$ is a minimizer of $\|\mathbf{B}^{>0}\mathbf{x} - \mathbf{c}^{\mathbf{B}^{>0}}\|_2$, we have

$$\|\mathbf{B}^{>0}\mathbf{x}^{\mathbf{B}^{>0}*} - \mathbf{c}^{\mathbf{B}^{>0}}\|_2^2 \leq \|\mathbf{B}^{>0}\mathbf{x}^{\mathbf{B}^*} - \mathbf{c}^{\mathbf{B}^{>0}}\|_2^2,$$

where we can extract out the $\delta \tilde{\mathbf{B}}$ term in $\mathbf{B}^{>0}$ separately to get:

$$\|\mathbf{B}\mathbf{x}^{\mathbf{B}^{>0}*} - \mathbf{c}^{\mathbf{B}}\|_2^2 + \delta^2 \|\tilde{\mathbf{B}}\mathbf{x}^{\mathbf{B}^{>0}*}\|_2^2 \leq \|\mathbf{B}\mathbf{x}^{\mathbf{B}^*} - \mathbf{c}^{\mathbf{B}}\|_2^2 + \delta^2 \|\tilde{\mathbf{B}}\mathbf{x}^{\mathbf{B}^*}\|_2^2.$$

Together with Equation (26), this then gives

$$\left\| \mathbf{x}^{\mathbf{B}^*} - \mathbf{x}^{\mathbf{B}^{>0*}} \right\|_{\mathbf{B}^\top \mathbf{B}} \leq \delta \left\| \tilde{\mathbf{B}} \mathbf{x}^{\mathbf{B}^*} \right\|_2.$$

It remains to upper bound $\left\| \tilde{\mathbf{B}} \mathbf{x}^{\mathbf{B}^{>0*}} \right\|_2$. By the assumption of $\mathbf{x}^{\mathbf{B}^*} \perp \text{null}(\mathbf{B})$, we get:

$$\frac{\left\| \tilde{\mathbf{B}} \mathbf{x}^{\mathbf{B}^*} \right\|_2^2}{\left\| \mathbf{B} \mathbf{x}^{\mathbf{B}^*} \right\|_2^2} \leq \frac{\lambda_{\max}(\tilde{\mathbf{B}}^\top \tilde{\mathbf{B}})}{\lambda_{\min}(\mathbf{B}^\top \mathbf{B})}.$$

By a proof similar to Lemma 4.11, which upper bounds $\lambda_{\max}(\mathbf{B}^\top \mathbf{B})$, we have:

$$\sigma_{\max}(\tilde{\mathbf{B}}) = O(1 + \delta) \sigma_{\max}(\mathbf{B}),$$

which implies

$$\left\| \tilde{\mathbf{B}} \mathbf{x}^{\mathbf{B}^*} \right\|_2 \leq O(1 + \delta) \kappa(\mathbf{B}) \left\| \mathbf{B} \mathbf{x}^{\mathbf{B}^*} \right\|_2.$$

Since $\mathbf{x}^{\mathbf{B}^*}$ is a minimizer of $\min_{\mathbf{x}} \left\| \mathbf{B} \mathbf{x} - \mathbf{c}^{\mathbf{B}} \right\|_2$, we have

$$\left\| \mathbf{B} \mathbf{x}^{\mathbf{B}^*} \right\|_2 = \left\| \mathbf{\Pi}_{\mathbf{B}} \mathbf{c}^{\mathbf{B}} \right\|_2 \leq \left\| \mathbf{c}^{\mathbf{B}} \right\|_2.$$

Therefore,

$$\left\| \mathbf{x}^{\mathbf{B}^*} - \mathbf{x}^{\mathbf{B}^{>0*}} \right\|_{\mathbf{B}^\top \mathbf{B}} \leq O(\delta(1 + \delta) \kappa(\mathbf{B})) \left\| \mathbf{c} \right\|_2,$$

which completes the proof. $\square$

We now check that the approximate solutions of the two linear systems (24) and (25) are also close to each other.

Lemma 5.3. *Let $\mathbf{B}^{>0} \mathbf{x} = \mathbf{c}^{\mathbf{B}^{>0}}$ be the linear system returned by a call to $\text{REDUCEMC}_2\text{ToMC}_2^{>0}(\mathbf{B}, \mathbf{c}^{\mathbf{B}}, \epsilon_1)$, and let ϵ_2 be the error parameter returned by this call. Let $\mathbf{x}$ be a vector such that*

$$\left\| \mathbf{B}^{>0} \mathbf{x} - \mathbf{\Pi}_{\mathbf{B}^{>0}} \mathbf{c}^{\mathbf{B}^{>0}} \right\|_2 \leq \epsilon_2 \left\| \mathbf{\Pi}_{\mathbf{B}^{>0}} \mathbf{c}^{\mathbf{B}^{>0}} \right\|_2.$$

Let $\mathbf{y}$ be the vector returned by a call to $\text{MAPSOLNMC}_2^{>0}\text{ToMC}_2(\mathbf{B}, \mathbf{B}^{>0}, \mathbf{c}^{\mathbf{B}}, \mathbf{x})$. Then

$$\left\| \mathbf{B} \mathbf{y} - \mathbf{\Pi}_{\mathbf{B}} \mathbf{c}^{\mathbf{B}} \right\|_2 \leq \epsilon_1 \left\| \mathbf{\Pi}_{\mathbf{B}} \mathbf{c}^{\mathbf{B}} \right\|_2.$$

Proof. In Algorithm 5, if the condition $\mathbf{B}^\top \mathbf{c}^{\mathbf{B}} = \mathbf{0}$ is true, then $\mathbf{y} = \mathbf{0}$. This implies

$$\left\| \mathbf{B} \mathbf{y} - \mathbf{\Pi}_{\mathbf{B}} \mathbf{c}^{\mathbf{B}} \right\|_2 = \mathbf{0}.$$

In the following, we assume that $\mathbf{B}^\top \mathbf{c}^{\mathbf{B}} \neq \mathbf{0}$, in which case $\mathbf{y} = \mathbf{x}$. We first show that the construction implies $\left\| \mathbf{B} \mathbf{x} - \mathbf{\Pi}_{\mathbf{B}} \mathbf{c}^{\mathbf{B}} \right\|_2 \leq \left\| \mathbf{B}^{>0} \mathbf{x} - \mathbf{\Pi}_{\mathbf{B}^{>0}} \mathbf{c}^{\mathbf{B}^{>0}} \right\|_2$. Once again, let $\mathbf{x}^{\mathbf{B}^*}$ and $\mathbf{x}^{\mathbf{B}^{>0*}}$ be the vectors that minimize $\left\| \mathbf{B} \mathbf{x} - \mathbf{c}^{\mathbf{B}} \right\|_2$ and $\left\| \mathbf{B}^{>0} \mathbf{x} - \mathbf{c}^{\mathbf{B}^{>0}} \right\|_2$ respectively. This choice gives:

$$\left\| \mathbf{B} \mathbf{x} - \mathbf{\Pi}_{\mathbf{B}} \mathbf{c}^{\mathbf{B}} \right\|_2 = \left\| \mathbf{x} - \mathbf{x}^{\mathbf{B}^*} \right\|_{\mathbf{B}^\top \mathbf{B}}.$$

As $\|\cdot\|_{\mathbf{B}^\top \mathbf{B}}$ is a norm, by triangle inequality we have:

$$\|\mathbf{x} - \mathbf{x}^{\mathbf{B}^*}\|_{\mathbf{B}^\top \mathbf{B}} \leq \|\mathbf{x} - \mathbf{x}^{\mathbf{B}^{>0*}}\|_{\mathbf{B}^\top \mathbf{B}} + \|\mathbf{x}^{\mathbf{B}^{>0*}} - \mathbf{x}^{\mathbf{B}^*}\|_{\mathbf{B}^\top \mathbf{B}}.$$

We will bound these two terms separately.

Since $\mathbf{B}^\top \mathbf{B} \preceq \mathbf{B}^\top \mathbf{B} + \delta^2 \tilde{\mathbf{B}}^\top \tilde{\mathbf{B}} = (\mathbf{B}^{>0})^\top \mathbf{B}^{>0}$, the first term is less than its norm in the $(\mathbf{B}^{>0})^\top \mathbf{B}^{>0}$ norm:

$$\|\mathbf{x} - \mathbf{x}^{\mathbf{B}^{>0*}}\|_{\mathbf{B}^\top \mathbf{B}} \leq \|\mathbf{x} - \mathbf{x}^{\mathbf{B}^{>0*}}\|_{(\mathbf{B}^\top \mathbf{B} + \delta^2 \tilde{\mathbf{B}}^\top \tilde{\mathbf{B}})} = \|\mathbf{B}^{>0} \mathbf{x} - \mathbf{\Pi}_{\mathbf{B}^{>0}} \mathbf{c}^{\mathbf{B}^{>0}}\|_2,$$

while the second term is precisely the distances between the two optimums, which we just bounded in Lemma 5.2. Combining these bounds then gives:

$$\|\mathbf{B} \mathbf{x} - \mathbf{\Pi}_{\mathbf{B}} \mathbf{c}^{\mathbf{B}}\|_2 \leq \|\mathbf{B}^{>0} \mathbf{x} - \mathbf{\Pi}_{\mathbf{B}^{>0}} \mathbf{c}^{\mathbf{B}^{>0}}\|_2 + O(\delta(1+\delta)\kappa(\mathbf{B}) \|\mathbf{c}^{\mathbf{B}}\|_2). \quad (27)$$

As the equations in $\mathbf{B}^{>0}$ is a superset of the ones in $\mathbf{B}$, we have

$$\|\mathbf{B}^{>0} \mathbf{x}^{\mathbf{B}^{>0*}} - \mathbf{c}^{\mathbf{B}^{>0}}\|_2 \geq \|\mathbf{B} \mathbf{x}^{\mathbf{B}^{>0*}} - \mathbf{c}^{\mathbf{B}}\|_2 \geq \|\mathbf{B} \mathbf{x}^{\mathbf{B}^*} - \mathbf{c}^{\mathbf{B}}\|_2.$$

Substituting $\|\mathbf{B} \mathbf{x}^{\mathbf{B}^*} - \mathbf{c}^{\mathbf{B}}\|_2^2 = \|\mathbf{c}^{\mathbf{A}}\|_2^2 - \|\mathbf{\Pi}_{\mathbf{B}} \mathbf{c}^{\mathbf{B}}\|_2^2$ and its equivalent in $\mathbf{B}^{>0}$ gives

$$\|\mathbf{\Pi}_{\mathbf{B}^{>0}} \mathbf{c}^{\mathbf{B}^{>0}}\|_2 \leq \|\mathbf{\Pi}_{\mathbf{B}} \mathbf{c}^{\mathbf{B}}\|_2.$$

Together with the condition of the lemma,

$$\|\mathbf{B}^{>0} \mathbf{x} - \mathbf{\Pi}_{\mathbf{B}^{>0}} \mathbf{c}^{\mathbf{B}^{>0}}\|_2 \leq \epsilon_2 \|\mathbf{\Pi}_{\mathbf{B}} \mathbf{c}^{\mathbf{B}}\|_2.$$

Plugging this into Equation (27), we have

$$\|\mathbf{B} \mathbf{x} - \mathbf{\Pi}_{\mathbf{B}} \mathbf{c}^{\mathbf{B}}\|_2 = O(\epsilon_2 \|\mathbf{\Pi}_{\mathbf{B}} \mathbf{c}^{\mathbf{B}}\|_2 + \delta \kappa(\mathbf{B}) \|\mathbf{c}^{\mathbf{B}}\|_2).$$

It remains to upper bound $\|\mathbf{c}^{\mathbf{B}}\|_2$ by a function of $\|\mathbf{\Pi}_{\mathbf{B}} \mathbf{c}^{\mathbf{B}}\|_2$. By our construction in Algorithm 1 and assumption, $\|\mathbf{B} \mathbf{c}^{\mathbf{B}}\|_2^2$ is a positive integer. By Lemma 4.8,

$$\|\mathbf{\Pi}_{\mathbf{B}} \mathbf{c}^{\mathbf{B}}\|_2 \geq \frac{1}{\sigma_{\max}(\mathbf{B})}.$$

Thus,

$$\|\mathbf{B} \mathbf{x} - \mathbf{\Pi}_{\mathbf{B}} \mathbf{c}^{\mathbf{B}}\|_2 = O(\epsilon_2 + \delta \kappa(\mathbf{B}) \sigma_{\max}(\mathbf{B}) \|\mathbf{c}^{\mathbf{B}}\|_2) \|\mathbf{\Pi}_{\mathbf{B}} \mathbf{c}^{\mathbf{B}}\|_2.$$

By our setting of δ in Equation (23), we have

$$\|\mathbf{B} \mathbf{x} - \mathbf{\Pi}_{\mathbf{B}} \mathbf{c}^{\mathbf{B}}\|_2 \leq \epsilon_1 \|\mathbf{\Pi}_{\mathbf{B}} \mathbf{c}^{\mathbf{B}}\|_2.$$

This completes the proof. $\square$

5.2 Bounding Condition Number of the New System in $\mathcal{MC}_2^{>0}$

We now establish bounds on the numerical quantities related to $\mathbf{B}^{>0}$. By a proof similar to the upper bound on $\lambda_{\max}(\mathbf{B}^\top \mathbf{B})$ in Lemma 4.11, we have:

$$\lambda_{\max}((\mathbf{B}^{>0})^\top \mathbf{B}^{>0}) = O\left(\lambda_{\max}(\mathbf{B}^\top \mathbf{B})\right).$$

As a result, we focus on the lower bound here:

Lemma 5.4. *The matrix $\mathbf{B}^{>0}$ from the linear system returned by a call to $\text{REDUCE}\mathcal{MC}_2\text{To}\mathcal{MC}_2^{>0}(\mathbf{B}, \mathbf{c}^B, \epsilon_1, \delta)$, where δ is set according to Equation 23, satisfies*

$$\sigma_{\min}(\mathbf{B}^{>0}) = \Omega\left(\frac{\epsilon_1}{\kappa(\mathbf{B})\sigma_{\max}(\mathbf{B})n_B}\right).$$

Proof. Let G be a unit-edge weight graph whose vertex set and edge set are same as the underlying graph of $\mathbf{B}^{>0}$. Let $\mathbf{L}_G$ be the associated Laplacian matrix of G , and let $\mathbf{M} := \mathbf{L}_G \otimes \mathbf{C}$, where

$$\mathbf{C} = \begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix}$$

is a symmetric PSD matrix. Note

$$\mathbf{M} = \mathbf{L}_G \otimes \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} + \mathbf{L}_G \otimes \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} + \mathbf{L}_G \otimes \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix},$$

which means $\mathbf{M}$ has the same null space as $(\mathbf{B}^{>0})^\top \mathbf{B}^{>0}$.

$$\lambda_{\min}((\mathbf{B}^{>0})^\top \mathbf{B}^{>0}) = \min_{\mathbf{x} \perp \text{null}(\mathbf{B}^{>0})} \frac{\mathbf{x}^\top (\mathbf{B}^{>0})^\top \mathbf{B}^{>0} \mathbf{x}}{\mathbf{x}^\top \mathbf{x}} \geq \delta^2 \min_{\mathbf{x} \perp \text{null}(\mathbf{M})} \frac{\mathbf{x}^\top \mathbf{M} \mathbf{x}}{\mathbf{x}^\top \mathbf{x}},$$

the last inequality is due to that δ is the minimum edge weight.

Note that the sum of the 3 types of blocks is positive definite and has eigenvalue at least 1: the type 1 and 2 blocks already sum to $\mathbf{I}$. Formally:

$$\lambda_{\min}(\mathbf{M}) = \lambda_{\min}(\mathbf{L}_G) \cdot \lambda_{\min}(\mathbf{C}) = \lambda_{\min}(\mathbf{L}_G).$$

The result then follows from the folklore bound that the minimum non-zero eigenvalue of a unit weighted graph is at least $\frac{1}{n^2}$. One way to see this is via Cheeger's inequality (see e.g. [Spi07]) applied to each block: this decomposition is equivalent to separating the matrix into its diagonal blocks based on the connected components, and then invoking the fact that the minimum weight of a cut is at least 1, and there are at most n vertices. Together these tools imply $\lambda_{\min}(\mathbf{L}_G) = \Omega\left(\frac{1}{n_B^2}\right)$, and in turn:

$$\sigma_{\min}(\mathbf{B}^{>0}) \geq \Omega\left(\frac{\delta}{n}\right) = \Omega\left(\frac{\epsilon_1}{\kappa(\mathbf{B})\sigma_{\max}(\mathbf{B})n_B}\right).$$

□

This also implies a bound on the condition number of $\mathbf{B}^{>0}$:

Lemma 5.5. $\kappa(\mathbf{B}^{>0}) = O\left(\epsilon_1^{-1} \sigma_{\max}^2(\mathbf{B}) \kappa(\mathbf{B}) n_B\right)$.

5.3 Putting it All Together

Proof of Lemma 3.6. By Lemma 5.1, we have

$$\text{nnz}(\mathbf{B}^{>0}) = O(s),$$

The largest entry is U , and the smallest nonzero entry is δ . By Equation (23) and Lemma B.2 and B.3,

$$\delta = \Omega\left(\frac{\epsilon}{KU^2}\right).$$

By Lemma 5.5 and Lemma B.2 and B.3, the condition number is

$$\kappa(\mathbf{B}^{>0}) = O\left(\frac{s^2 U^2 K}{\epsilon}\right).$$

By Lemma 5.3, the accuracy error for $\mathbf{B}^{>0}$ is $O(\epsilon)$. □

6 Rounding and Scaling Weights to Integers

In this section, we show a reduction from the linear system with strict 2-commodity matrix to the linear system with integral strict 2-commodity matrix.

Algorithm 6 presents the pseudo-code for the algorithm $\text{REDUCE}\mathcal{MC}_2^{>0}\text{To}\mathcal{MC}_{2,\mathbb{Z}}^{>0}$. Given an instance $(\mathbf{B}^{>0}, \mathbf{c}^{\mathbf{B}^{>0}}, \epsilon_1)$ where $\mathbf{B}^{>0} \in \mathcal{MC}_2^{>0}$, the call $\text{REDUCE}\mathcal{MC}_2^{>0}\text{To}\mathcal{MC}_{2,\mathbb{Z}}^{>0}(\mathbf{B}^{>0}, \mathbf{c}^{\mathbf{B}^{>0}}, \epsilon_1)$ returns an instance $(\mathbf{B}^{>0,\mathbb{Z}}, \mathbf{c}^{\mathbf{B}^{>0,\mathbb{Z}}}, \epsilon_2)$ where $\mathbf{B}^{>0,\mathbb{Z}} \in \mathcal{MC}_{2,\mathbb{Z}}^{>0}$. Algorithm 7 provides the (trivial) pseudo-code for $\text{MAPSOLN}\mathcal{MC}_{2,\mathbb{Z}}^{>0}\text{To}\mathcal{MC}_2^{>0}$ which maps a solution of an instance over $\mathcal{MC}_{2,\mathbb{Z}}^{>0}$ to a solution of an instance over $\mathcal{MC}_2^{>0}$.

Algorithm 6 $\text{REDUCE}\mathcal{MC}_2^{>0}\text{To}\mathcal{MC}_{2,\mathbb{Z}}^{>0}$

Input: $(\mathbf{B}^{>0}, \mathbf{c}^{\mathbf{B}^{>0}}, \epsilon_1)$ where $\mathbf{B}^{>0} \in \mathcal{MC}_2^{>0}$ is an $m \times n$ matrix, $\mathbf{c}^{\mathbf{B}^{>0}} \in \mathbb{R}^m$, and $0 < \epsilon_1 < 1$

Output: $(\mathbf{B}^{>0,\mathbb{Z}}, \mathbf{c}^{\mathbf{B}^{>0,\mathbb{Z}}}, \epsilon_2)$ where $\mathbf{B}^{>0,\mathbb{Z}} \in \mathcal{MC}_{2,\mathbb{Z}}^{>0}$ is an $m' \times n'$ matrix, $\mathbf{c}^{\mathbf{B}^{>0,\mathbb{Z}}} \in \mathbb{R}^{m'}$, and $0 <$

$\epsilon_2 < 1$.
1: $k \leftarrow \left\lceil \log_2 \frac{\text{nnz}(\mathbf{B}^{>0})}{\epsilon_1} \right\rceil + 2$
2: $\mathbf{B}^{>0,\mathbb{Z}} \leftarrow \mathbf{B}^{>0}$
3: $\mathbf{c}^{\mathbf{B}^{>0,\mathbb{Z}}} \leftarrow \mathbf{c}^{\mathbf{B}^{>0}}$
4: **for** each entry $(\mathbf{B}^{>0,\mathbb{Z}})_{ij}$ **do**
5: $(\mathbf{B}^{>0,\mathbb{Z}})_{ij} \leftarrow \lceil (\mathbf{B}^{>0,\mathbb{Z}})_{ij} \cdot 2^k \rceil$
6: **end for**
7: **for** each entry $\mathbf{c}_i^{\mathbf{B}^{>0,\mathbb{Z}}}$ **do**
8: $\mathbf{c}_i^{\mathbf{B}^{>0,\mathbb{Z}}} \leftarrow \mathbf{c}_i^{\mathbf{B}^{>0,\mathbb{Z}}} \cdot 2^k$
9: **end for**
10: **return** $(\mathbf{B}^{>0,\mathbb{Z}}, \mathbf{c}^{\mathbf{B}^{>0,\mathbb{Z}}}, \epsilon_1/3)$.

For simplicity, we analyze an intermediate linear system $\mathbf{B}^{int} \mathbf{y} = \mathbf{c}^{\mathbf{B}^{>0}}$, where

$$(\mathbf{B}^{int})_{ij} \stackrel{\text{def}}{=} 2^{-k} \cdot (\mathbf{B}^{>0,\mathbb{Z}})_{ij}.$$

Recall the definition of $\mathbf{B}^{>0,\mathbb{Z}}$ in Algorithm 6 line 5, we have

$$(\mathbf{B}^{int})_{ij} = 2^{-k} \cdot \left\lceil (\mathbf{B}^{>0})_{ij} \cdot 2^k \right\rceil. \tag{28}$$

Algorithm 7 MAPSOLN $\mathcal{MC}_{2,\mathbb{Z}}^{>0}$ TO $\mathcal{MC}_2^{>0}$

Input: $m \times n$ matrix $\mathbf{B}^{>0} \in \mathcal{MC}_2$, $m' \times n'$ matrix $\mathbf{B}^{>0,\mathbb{Z}} \in \mathcal{MC}_2^{>0}$, vector $\mathbf{x} \in \mathbb{R}^{n'}$.

Output: Vector $\mathbf{y} \in \mathbb{R}^n$.

1: **return** $\mathbf{x}$

The linear system $\mathbf{B}^{int}\mathbf{y} = \mathbf{c}^{B^{>0}}$ is exactly the linear system $\mathbf{B}^{>0,\mathbb{Z}}\mathbf{y} = \mathbf{c}^{B^{>0,\mathbb{Z}}}$ multiplying a factor 2^{-k} on both sides.

Note the condition number and the eigen-space of a matrix, the optimal solutions of the projection problems are invariant under scaling. Let $\mathbf{y}^* \in \arg \min_{\mathbf{y}} \left\| \mathbf{B}^{>0,\mathbb{Z}}\mathbf{y} - \mathbf{c}^{B^{>0,\mathbb{Z}}} \right\|_2 = \arg \min_{\mathbf{y}} \left\| \mathbf{B}^{int}\mathbf{y} - \mathbf{c}^{B^{>0}} \right\|_2$. The following two inequalities:

$$\|\mathbf{y} - \mathbf{y}^*\|_{(\mathbf{B}^{>0,\mathbb{Z}})^\top \mathbf{B}^{>0,\mathbb{Z}}} \leq \epsilon \left\| \mathbf{\Pi}_{\mathbf{B}^{>0,\mathbb{Z}}} \mathbf{c}^{B^{>0,\mathbb{Z}}} \right\|_2$$

and

$$\|\mathbf{y} - \mathbf{y}^*\|_{(\mathbf{B}^{int})^\top \mathbf{B}^{int}} \leq \epsilon \left\| \mathbf{\Pi}_{\mathbf{B}^{int}} \mathbf{c}^{B^{>0}} \right\|_2$$

are equivalent. Thus, it suffices to analyze the linear system $\mathbf{B}^{int}\mathbf{y} = \mathbf{c}^{B^{>0}}$.

Solving the two projection problems

$$\begin{aligned} \min_{\mathbf{x}} \left\| \mathbf{B}^{>0}\mathbf{x} - \mathbf{c}^{B^{>0}} \right\|_2, \\ \min_{\mathbf{y}} \left\| \mathbf{B}^{int}\mathbf{y} - \mathbf{c}^{B^{>0}} \right\|_2, \end{aligned}$$

is equivalent to solving the following two linear systems:

$$\begin{aligned} (\mathbf{B}^{>0})^\top \mathbf{B}^{>0}\mathbf{x} &= (\mathbf{B}^{>0})^\top \mathbf{c}^{B^{>0}}, \\ (\mathbf{B}^{int})^\top \mathbf{B}^{int}\mathbf{y} &= (\mathbf{B}^{int})^\top \mathbf{c}^{B^{>0}}. \end{aligned} \tag{29}$$

Note $\mathbf{c}^{B^{>0}} = (\mathbf{c}^A; \mathbf{0})$, and all entries of the rows of $\mathbf{B}^{>0}$ corresponding to the original rows of $\mathbf{A}$ are integers. Thus,

$$(\mathbf{B}^{int})^\top \mathbf{c}^{B^{>0}} = (\mathbf{B}^{>0})^\top \mathbf{c}^{B^{>0}}.$$

That is, the 2nd linear system is equivalent to

$$(\mathbf{B}^{int})^\top \mathbf{B}^{int}\mathbf{y} = (\mathbf{B}^{>0})^\top \mathbf{c}^{B^{>0}}. \tag{30}$$

Let

$$\mathbf{M} \stackrel{\text{def}}{=} (\mathbf{B}^{>0})^\top \mathbf{B}^{>0} \text{ and } \widehat{\mathbf{M}} \stackrel{\text{def}}{=} (\mathbf{B}^{int})^\top \mathbf{B}^{int}.$$

We bound the eigenvalues of $\widehat{\mathbf{M}}$ by the following lemma.

Lemma 6.1. $\lambda_{\max}(\widehat{\mathbf{M}}) \leq (1 + 2^{-k+2})\lambda_{\max}(\mathbf{M})$ and $\lambda_{\min}(\widehat{\mathbf{M}}) \geq \lambda_{\min}(\mathbf{M})$. Furthermore, $\kappa(\widehat{\mathbf{M}}) \leq (1 + 2^{-k+2})\kappa(\mathbf{M})$.

Proof. Note $\mathbf{M}$ can be written as $\mathbf{B}^\top \mathbf{W} \mathbf{B}$, where $\mathbf{B}$ is the incidence-structured block matrix with unit nonzero entries and $\mathbf{W}$ is the diagonal matrix with edge weights. Let $\mathbf{x}$ be an arbitrary vector, and let $\mathbf{y} \stackrel{\text{def}}{=} \mathbf{B}\mathbf{x}$.

$$\mathbf{x}^\top \mathbf{M} \mathbf{x} = \mathbf{y}^\top \mathbf{W} \mathbf{y} = \sum_i \mathbf{W}_{ii} \mathbf{y}_i^2,$$

and

$$\mathbf{x}^\top \widehat{\mathbf{M}} \mathbf{x} = \sum_i \widehat{\mathbf{W}}_i \mathbf{y}_i^2.$$

Let

$$\alpha \stackrel{\text{def}}{=} \min_i \frac{\widehat{\mathbf{W}}_{ii}}{\mathbf{W}_{ii}} \text{ and } \beta \stackrel{\text{def}}{=} \max_i \frac{\widehat{\mathbf{W}}_{ii}}{\mathbf{W}_{ii}}.$$

We have

$$\alpha \sum_i \mathbf{W}_i \mathbf{y}_i^2 \leq \sum_i \widehat{\mathbf{W}}_i \mathbf{y}_i^2 \leq \beta \sum_i \mathbf{W}_i \mathbf{y}_i^2.$$

This implies that

$$\alpha \mathbf{M} \preceq \widehat{\mathbf{M}} \preceq \beta \mathbf{M}. \quad (31)$$

Now we bound the values of α, β . According to Equation (28),

$$\alpha = \min_i \frac{\widehat{\mathbf{W}}_{ii}}{\mathbf{W}_{ii}} \geq \frac{\left(2^{-k} \cdot \mathbf{W}_{ii}^{1/2} \cdot 2^k\right)^2}{\mathbf{W}_{ii}} = 1. \quad (32)$$

Similarly,

$$\beta = \max_i \frac{\widehat{\mathbf{W}}_{ii}}{\mathbf{W}_{ii}} \leq \frac{\left(2^{-k} \left(\mathbf{W}_{ii}^{1/2} \cdot 2^k + 1\right)\right)^2}{\mathbf{W}_{ii}} \leq 1 + 2^{-k+2}. \quad (33)$$

The last inequality is due to $\mathbf{W}_{ii} \geq 1$. Thus,

$$\lambda_{\max}(\widehat{\mathbf{M}}) \leq (1 + 2^{-k+2}) \lambda_{\max}(\mathbf{M}) \text{ and } \lambda_{\min}(\widehat{\mathbf{M}}) \geq \lambda_{\min}(\mathbf{M}).$$

The bound on the condition number $\kappa(\widehat{\mathbf{M}}) \leq (1 + 2^{-k+2}) \kappa(\mathbf{M})$ immediately follows the above two inequalities. $\square$

We show the exact solutions of the two linear systems in Equation (29) and (30) are close, by the following lemma.

Lemma 6.2. *Let*

$$\mathbf{x}^* \stackrel{\text{def}}{=} \mathbf{M}^\dagger (\mathbf{B}^{>0})^\top \mathbf{c}^{\mathbf{B}^{>0}} \text{ and } \mathbf{y}^* \stackrel{\text{def}}{=} \widehat{\mathbf{M}}^\dagger (\mathbf{B}^{>0})^\top \mathbf{c}^{\mathbf{B}^{>0}}.$$

$\mathbf{x}^*$ and $\mathbf{y}^*$ are exact solutions of linear system (29) and (30) respectively.

$$\|\mathbf{y}^* - \mathbf{x}^*\|_{\mathbf{M}} \leq 2^{-k+2} \left\| (\mathbf{B}^{>0})^\top \mathbf{c}^{\mathbf{B}^{>0}} \right\|_{\mathbf{M}^\dagger}.$$

Proof. Note $\mathbf{M}, \widehat{\mathbf{M}}$ are both symmetric. Expanding the left hand side,

$$\begin{aligned} \|\mathbf{y}^* - \mathbf{x}^*\|_2^2 &= (\mathbf{c}^{\mathbf{B}^{>0}})^\top \mathbf{B}^{>0} \left(\widehat{\mathbf{M}}^\dagger - \mathbf{M}^\dagger \right) \mathbf{M} \left(\widehat{\mathbf{M}}^\dagger - \mathbf{M}^\dagger \right) (\mathbf{B}^{>0})^\top \mathbf{c}^{\mathbf{B}^{>0}} \\ &= (\mathbf{c}^{\mathbf{B}^{>0}})^\top \mathbf{B}^{>0} \mathbf{M}^{\dagger 1/2} \left(\mathbf{M}^{1/2} \widehat{\mathbf{M}}^\dagger \mathbf{M}^{1/2} - \mathbf{I} \right)^2 \mathbf{M}^{\dagger 1/2} (\mathbf{B}^{>0})^\top \mathbf{c}^{\mathbf{B}^{>0}} \\ &\leq \left\| \mathbf{M}^{1/2} \widehat{\mathbf{M}}^\dagger \mathbf{M}^{1/2} - \mathbf{I} \right\|_2^2 \left\| \mathbf{M}^{\dagger 1/2} (\mathbf{B}^{>0})^\top \mathbf{c}^{\mathbf{B}^{>0}} \right\|_2^2. \end{aligned}$$

The last inequality is by the Courant-Fischer theorem.

By Equation (31) and the fact that $\mathbf{M}, \widehat{\mathbf{M}}$ have same null space,

$$(\beta^{-1} - 1) \mathbf{I} \preceq \mathbf{M}^{1/2} \widehat{\mathbf{M}}^\dagger \mathbf{M}^{1/2} - \mathbf{I} \preceq (\alpha^{-1} - 1) \mathbf{I}.$$

By Equation (32) and (33),

$$\left\| \mathbf{M}^{1/2} \widehat{\mathbf{M}}^\dagger \mathbf{M}^{1/2} - \mathbf{I} \right\|_2 \leq 2^{-k+2}.$$

Therefore,

$$\|\mathbf{y}^* - \mathbf{x}^*\|_{\mathbf{M}} \leq 2^{-k+2} \left\| (\mathbf{B}^{>0})^\top \mathbf{c}^{\mathbf{B}^{>0}} \right\|_{\mathbf{M}^\dagger}.$$

□

We then show the approximate solutions of the two linear systems in Equation (29) and (30) are close.

Lemma 6.3. *Let $\mathbf{x}$ be a vector such that*

$$\left\| \mathbf{B}^{int} \mathbf{x} - \mathbf{\Pi}_{\mathbf{B}^{int}} \mathbf{c}^{\mathbf{B}^{>0}} \right\|_2 \leq \epsilon_2 \left\| \mathbf{\Pi}_{\mathbf{B}^{int}} \mathbf{c}^{\mathbf{B}^{>0}} \right\|_2.$$

Then,

$$\left\| \mathbf{B}^{>0} \mathbf{x} - \mathbf{\Pi}_{\mathbf{B}^{>0}} \mathbf{c}^{\mathbf{B}^{>0}} \right\|_2 \leq \left(\epsilon_2 + 2^{-k+2}(1 + \epsilon_2) \right) \left\| \mathbf{\Pi}_{\mathbf{B}^{>0}} \mathbf{c}^{\mathbf{B}^{>0}} \right\|_2.$$

Proof. Let

$$\mathbf{x}^* \stackrel{\text{def}}{=} \mathbf{M}^\dagger (\mathbf{B}^{>0})^\top \mathbf{c}^{\mathbf{B}^{>0}} \text{ and } \mathbf{y}^* \stackrel{\text{def}}{=} \widehat{\mathbf{M}}^\dagger (\mathbf{B}^{>0})^\top \mathbf{c}^{\mathbf{B}^{>0}}.$$

Expanding the left hand side,

$$\begin{aligned} \left\| \mathbf{B}^{>0} \mathbf{x} - \mathbf{\Pi}_{\mathbf{B}^{>0}} \mathbf{c}^{\mathbf{B}^{>0}} \right\|_2 &= \|\mathbf{x} - \mathbf{x}^*\|_{\mathbf{M}} \\ &\leq \|\mathbf{x} - \mathbf{y}^*\|_{\mathbf{M}} + \|\mathbf{y}^* - \mathbf{x}^*\|_{\mathbf{M}}. \end{aligned}$$

The last inequality is due to the triangle inequality. By Equation (31), the first term can be upper bounded by

$$\|\mathbf{x} - \mathbf{y}^*\|_{\mathbf{M}} \leq \alpha^{-1/2} \|\mathbf{x} - \mathbf{y}^*\|_{\widehat{\mathbf{M}}}.$$

The second term is upper bounded by Lemma 6.2. Thus, we have

$$\left\| \mathbf{B}^{>0} \mathbf{x} - \mathbf{\Pi}_{\mathbf{B}^{>0}} \mathbf{c}^{\mathbf{B}^{>0}} \right\|_2 \leq \alpha^{-1/2} \epsilon_2 \left\| \mathbf{\Pi}_{\mathbf{B}^{int}} \mathbf{c}^{\mathbf{B}^{>0}} \right\|_2 + 2^{-k+2} \left\| \mathbf{\Pi}_{\mathbf{B}^{>0}} \mathbf{c}^{\mathbf{B}^{>0}} \right\|_2.$$

Note that

$$\left\| \mathbf{\Pi}_{\mathbf{B}^{int}} \mathbf{c}^{\mathbf{B}^{>0}} \right\|_2 = \left\| \mathbf{B}^{>0} \mathbf{c}^{\mathbf{B}^{>0}} \right\|_{\widehat{\mathbf{M}}^\dagger} \text{ and } \left\| \mathbf{\Pi}_{\mathbf{B}^{>0}} \mathbf{c}^{\mathbf{B}^{>0}} \right\|_2 = \left\| \mathbf{B}^{>0} \mathbf{c}^{\mathbf{B}^{>0}} \right\|_{\mathbf{M}^\dagger}.$$

Again, by Equation (31),

$$\left\| \mathbf{B}^{>0} \mathbf{c}^{\mathbf{B}^{>0}} \right\|_{\widehat{\mathbf{M}}^\dagger} \leq \alpha^{-1/2} \left\| \mathbf{B}^{>0} \mathbf{c}^{\mathbf{B}^{>0}} \right\|_{\mathbf{M}^\dagger}.$$

By Equation (32) and (33),

$$\left\| \mathbf{B}^{>0} \mathbf{x} - \mathbf{\Pi}_{\mathbf{B}^{>0}} \mathbf{c}^{\mathbf{B}^{>0}} \right\|_2 \leq \left(\epsilon_2 + 2^{-k+2}(1 + \epsilon_2) \right) \left\| \mathbf{\Pi}_{\mathbf{B}^{>0}} \mathbf{c}^{\mathbf{B}^{>0}} \right\|_2.$$

This completes the proof. □

Proof of Lemma 3.7. We set

$$k \stackrel{\text{def}}{=} \left\lceil \log_2 \frac{s}{\epsilon} \right\rceil + 2.$$

After rounding and scaling, the number of nonzero entries does not change. The value of the smallest nonzero entry does not decrease. The value of the largest entry is multiplied by $2^k = s\epsilon^{-1}$, which is upper bounded by $s\epsilon^{-1}U$.

By Lemma 6.1,

$$\kappa(\mathbf{B}^{>0, \mathbb{Z}}) \leq (1 + s^{-1/2}\epsilon^{1/2})\kappa(\mathbf{B}^{>0}).$$

By Lemma 6.3, the accuracy is bounded by $\epsilon/3$. $\square$

7 $\mathcal{G}$ Efficiently Reducible to $\mathcal{G}_z$

In this section, we prove Lemma 3.4.

Definition 7.1. We let $\mathcal{G}_z$ denote the class of all matrices with integer valued entries s.t. there is at least one non-zero entry in every row and column, and every row has zero row sum.

7.1 $\mathcal{G} \leq_f \mathcal{G}_z$

In this section, we show the reduction from $\mathcal{G}$ to $\mathcal{G}_z$. Given an instance of linear system $(\mathbf{A}, \mathbf{c}^A, \epsilon_1)$ with $\mathbf{A} \in \mathcal{G}$, the goal is to construct an instance of linear system $(\mathbf{A}^Z, \mathbf{c}^Z, \epsilon_2)$ with $\mathbf{A}^Z \in \mathcal{G}_z$ such that, there exists a map between the solutions of the two linear systems.

Algorithm 8 shows the construction of $(\mathbf{A}^Z, \mathbf{c}^Z, \epsilon_2)$, and Algorithm 9 shows the transform from a solution of $(\mathbf{A}^Z, \mathbf{c}^Z, \epsilon_2)$ to a solution of $(\mathbf{A}, \mathbf{c}^A, \epsilon_1)$. Recall that $\mathbf{A}$ has n_A columns. According to the algorithm, $\mathbf{A}^Z$ has $(n_A + 1)$ columns, and $\mathbf{c}^Z = \mathbf{c}$.

Algorithm 8 REDUCE $\mathcal{G}$ TO $\mathcal{G}_z$

Input: $(\mathbf{A}, \mathbf{c}^A, \epsilon_1)$ where $\mathbf{A} \in \mathcal{G}$ is an $m \times n$ matrix, $\mathbf{c}^A \in \mathbb{R}^m$, and $0 < \epsilon_1 < 1$.

Output: $(\mathbf{A}^Z, \mathbf{c}^Z, \epsilon_2)$ where $\mathbf{A}^Z \in \mathcal{G}_z$ is an $m' \times n'$ matrix, $\mathbf{c}^Z \in \mathbb{R}^{m'}$, and $0 < \epsilon_2 < 1$.

1: **return** $((\mathbf{A} \quad -\mathbf{A}\mathbf{1}), \mathbf{c}^A, O(\frac{\epsilon_1}{n}))$.

Algorithm 9 MAPSOLN $\mathcal{G}_z$ TO $\mathcal{G}$

Input: $m \times n$ matrix $\mathbf{A} \in \mathcal{G}$, $m' \times n'$ matrix $\mathbf{A}^Z \in \mathcal{G}_z$, vector $\mathbf{x}^Z \in \mathbb{R}^{n'}$.

Output: Vector $\mathbf{x} \in \mathbb{R}^n$.

1: **return** $\mathbf{x} \leftarrow \mathbf{x}_{1:n}^Z - \mathbf{x}_{n+1}^Z \mathbf{1}$

Lemma 7.2. Let $\mathbf{A}^Z \mathbf{x}^Z = \mathbf{c}^Z$ be the linear system returned by a call to REDUCE $\mathcal{G}$ TO $\mathcal{G}_z(\mathbf{A}, \mathbf{c}^A, \epsilon_1)$. Then,

$$\text{nnz}(\mathbf{A}^Z) = O(\text{nnz}(\mathbf{A})),$$

and the largest entry of $\mathbf{A}^Z$ is at most $\|\mathbf{A}\|_\infty$.

We show the relation between the exact solvers of the two linear systems.

Claim 7.3. Let $\mathbf{x}^{Z*} \in \arg \min_{\mathbf{x}} \|\mathbf{A}^Z \mathbf{x} - \mathbf{c}^Z\|_2$. Let $\mathbf{x}^*$ be the output vector of Algorithm 9, that is,

$$\mathbf{x}^* = \mathbf{x}_{1:n_A}^{Z*} - \mathbf{x}_{n_A+1}^{Z*} \mathbf{1}.$$

Then,

$$\mathbf{x}^* \in \arg \min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{c}^A\|_2.$$

Proof. Note $\mathbf{1} \in \text{Null}(\mathbf{A}^Z)$, thus,

$$\mathbf{A}^Z \mathbf{x}^{Z*} = \mathbf{A}^Z (\mathbf{x}^{Z*} - \mathbf{x}_{n_A+1}^{Z*} \mathbf{1}).$$

Expanding it gives

$$\mathbf{A}^Z \mathbf{x}^{Z*} = \begin{pmatrix} \mathbf{A} & -\mathbf{A}\mathbf{1} \end{pmatrix} \begin{pmatrix} \mathbf{x}^* \\ \mathbf{0} \end{pmatrix} = \mathbf{A}\mathbf{x}^*.$$

Note $\mathbf{c}^Z = \mathbf{c}$. Therefore, $\mathbf{x}^* \in \arg \min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{c}^A\|_2$. $\square$

We then show the relation between the approximate solvers.

Lemma 7.4. *Let $\mathbf{x}^Z$ be a vector such that $\|\mathbf{A}^Z \mathbf{x}^Z - \Pi_{\mathbf{A}^Z} \mathbf{c}^A\|_2 \leq \epsilon_2 \|\Pi_{\mathbf{A}^Z} \mathbf{c}^A\|_2$. Let $\mathbf{x}$ be the output vector of Algorithm 9, that is, $\mathbf{x} = \mathbf{x}_{1:n_A}^Z - \mathbf{x}_{n_A+1}^Z \mathbf{1}$. Then,*

$$\|\mathbf{A}\mathbf{x} - \Pi_{\mathbf{A}} \mathbf{c}^A\|_2 \leq O\left(\frac{\sqrt{n_A} \sigma_{\min}(\mathbf{A})}{\sigma_{\min}(\mathbf{A}^Z)}\right) \epsilon_2 \|\Pi_{\mathbf{A}} \mathbf{c}^A\|_2.$$

Proof. Let $\mathbf{x}^{Z*} \in \arg \min_{\mathbf{x}} \|\mathbf{A}^Z \mathbf{x} - \mathbf{c}^Z\|$ whose last entry is 0. Expanding the left hand side norm,

$$\|\mathbf{A}^Z \mathbf{x}^Z - \Pi_{\mathbf{A}^Z} \mathbf{c}^A\|_2 = \|\mathbf{A}^Z (\mathbf{x}^Z - \mathbf{x}^{Z*})\|_2.$$

By Claim 7.3, $\mathbf{x}^* \in \arg \min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{c}^A\|_2$. Thus,

$$\|\mathbf{A}(\mathbf{x} - \mathbf{x}^*)\|_2 = \|\mathbf{A}^Z(\mathbf{x}^Z - \mathbf{x}^{Z*})\|_2.$$

Note that

$$\mathbf{A}\mathbf{x} = \mathbf{A}^Z \mathbf{x}^Z \text{ and } \mathbf{A}\mathbf{x}^* = \mathbf{A}^Z \mathbf{x}^{Z*}.$$

Together with the Lemma condition, this gives

$$\|\mathbf{A}\mathbf{x} - \Pi_{\mathbf{A}} \mathbf{c}^A\|_2 = \|\mathbf{A}^Z \mathbf{x}^Z - \Pi_{\mathbf{A}^Z} \mathbf{c}^Z\|_2 \leq \epsilon_2 \|\Pi_{\mathbf{A}^Z} \mathbf{c}^A\|_2. \quad (34)$$

It remains to upper bound $\|\Pi_{\mathbf{A}^Z} \mathbf{c}^A\|_2$ by a function of $\|\Pi_{\mathbf{A}} \mathbf{c}^A\|_2$. Note

$$\|\Pi_{\mathbf{A}} \mathbf{c}^A\|_2 = \|\mathbf{A}^\top \mathbf{c}^A\|_{(\mathbf{A}^\top \mathbf{A})^\dagger} \text{ and } \|\Pi_{\mathbf{A}^Z} \mathbf{c}^A\|_2 = \|(\mathbf{A}^Z)^\top \mathbf{c}^A\|_{((\mathbf{A}^Z)^\top \mathbf{A}^Z)^\dagger}. \quad (35)$$

It suffices to work on $\|\mathbf{A}^\top \mathbf{c}^A\|_{(\mathbf{A}^\top \mathbf{A})^\dagger}$ and $\|(\mathbf{A}^Z)^\top \mathbf{c}^A\|_{((\mathbf{A}^Z)^\top \mathbf{A}^Z)^\dagger}$. Since $\mathbf{A}^Z = \begin{pmatrix} \mathbf{A} & -\mathbf{A}\mathbf{1} \end{pmatrix}$, we have

$$\|(\mathbf{A}^Z)^\top \mathbf{c}^A\|_2^2 = \|\mathbf{A}^\top \mathbf{c}^A\|_2^2 + \|\mathbf{1}^\top \mathbf{A}^\top \mathbf{c}^A\|_2^2.$$

By Cauchy-Schwarz inequality,

$$\|\mathbf{1}^\top \mathbf{A}^\top \mathbf{c}^A\|_2 \leq \|\mathbf{1}\|_2 \|\mathbf{A}^\top \mathbf{c}^A\|_2 = \sqrt{n_A} \|\mathbf{A}^\top \mathbf{c}^A\|_2.$$

Plugging this gives

$$\left\| (\mathbf{A}^Z)^\top \mathbf{c}^A \right\|_2^2 \leq (n_A + 1) \left\| \mathbf{A}^\top \mathbf{c}^A \right\|_2^2.$$

By Courant-Fischer theorem,

$$\left\| (\mathbf{A}^Z)^\top \mathbf{c}^A \right\|_{((\mathbf{A}^Z)^\top \mathbf{A}^Z)^\dagger}^2 \leq \lambda_{\max} \left(((\mathbf{A}^Z)^\top \mathbf{A}^Z)^\dagger \right) \left\| (\mathbf{A}^Z)^\top \mathbf{c}^A \right\|_2^2 \leq \lambda_{\min}^{-1} \left((\mathbf{A}^Z)^\top \mathbf{A}^Z \right) \left\| (\mathbf{A}^Z)^\top \mathbf{c}^A \right\|_2^2$$

and since $\mathbf{A}^\top \mathbf{c}^A$ is in the eigenspace of $\mathbf{A}^\top \mathbf{A}$,

$$\left\| \mathbf{A}^\top \mathbf{c}^A \right\|_{(\mathbf{A}^\top \mathbf{A})^\dagger}^2 \geq \lambda_{\min} \left((\mathbf{A}^\top \mathbf{A})^\dagger \right) \left\| \mathbf{A}^\top \mathbf{c}^A \right\|_2^2 = \lambda_{\max}^{-1} \left(\mathbf{A}^\top \mathbf{A} \right) \left\| \mathbf{A}^\top \mathbf{c}^A \right\|_2^2.$$

Combining the above two inequalities,

$$\left\| (\mathbf{A}^Z)^\top \mathbf{c}^A \right\|_{((\mathbf{A}^Z)^\top \mathbf{A}^Z)^\dagger}^2 \leq \frac{\lambda_{\max}(\mathbf{A}^\top \mathbf{A})}{\lambda_{\min}((\mathbf{A}^Z)^\top \mathbf{A}^Z)} (n_A + 1) \left\| \mathbf{A}^\top \mathbf{c}^A \right\|_{(\mathbf{A}^\top \mathbf{A})^\dagger}^2.$$

Given equations (34) and (35), and condition $\left\| \mathbf{A}^Z \mathbf{x}^Z - \mathbf{c}^A \right\|_2 \leq \epsilon \left\| \mathbf{\Pi}_{\mathbf{A}^Z} \mathbf{c}^A \right\|_2$, we have

$$\left\| \mathbf{A} \mathbf{x} - \mathbf{\Pi}_A \mathbf{c}^A \right\|_2 \leq \epsilon_2 \frac{\sigma_{\max}(\mathbf{A})}{\sigma_{\min}(\mathbf{A}^Z)} \sqrt{n_A + 1} \left\| \mathbf{\Pi}_A \mathbf{c}^A \right\|_2.$$

This completes the proof. $\square$

We compute the nonzero condition number of $\mathbf{A}^Z$. Note

$$(\mathbf{A}^Z)^\top \mathbf{A}^Z = \begin{pmatrix} \mathbf{M} & -\mathbf{M} \mathbf{1} \\ -\mathbf{1}^\top \mathbf{M} & \mathbf{1}^\top \mathbf{M} \mathbf{1} \end{pmatrix}, \quad (36)$$

where $\mathbf{M} = \mathbf{A}^\top \mathbf{A}$.

Claim 7.5. $\lambda_{\max}((\mathbf{A}^Z)^\top \mathbf{A}^Z) \leq 2n_A \lambda_{\max}(\mathbf{A}^\top \mathbf{A})$.

Proof. Let λ_1 be the largest eigenvalue of $(\mathbf{A}^Z)^\top \mathbf{A}^Z$, and $\mathbf{y} = (\tilde{\mathbf{y}}; \alpha)$ be the associated eigenvector of unit length. Let μ_1 be the largest eigenvalue of $\mathbf{A}^\top \mathbf{A} = \mathbf{M}$.

$$\begin{aligned} \lambda_1 &= \mathbf{y}^\top (\mathbf{A}^Z)^\top \mathbf{A}^Z \mathbf{y} = \begin{pmatrix} \tilde{\mathbf{y}}^\top & \alpha \end{pmatrix} \begin{pmatrix} \mathbf{M} & -\mathbf{M} \mathbf{1} \\ -\mathbf{1}^\top \mathbf{M} & \mathbf{1}^\top \mathbf{M} \mathbf{1} \end{pmatrix} \begin{pmatrix} \tilde{\mathbf{y}} \\ \alpha \end{pmatrix} \\ &= \tilde{\mathbf{y}}^\top \mathbf{M} \tilde{\mathbf{y}} - 2\alpha \mathbf{1}^\top \mathbf{M} \tilde{\mathbf{y}} + \alpha^2 \mathbf{1}^\top \mathbf{M} \mathbf{1}. \end{aligned}$$

By Courant-Fischer Theorem,

$$\tilde{\mathbf{y}}^\top \mathbf{M} \tilde{\mathbf{y}} \leq \mu_1 \tilde{\mathbf{y}}^\top \tilde{\mathbf{y}}, \quad \text{and} \quad \mathbf{1}^\top \mathbf{M} \mathbf{1} \leq \mu_1 n_A.$$

By Cauchy-Schwarz inequality,

$$\left| \mathbf{1}^\top \mathbf{M} \tilde{\mathbf{y}} \right| \leq \|\mathbf{1}\|_2 \|\mathbf{M} \tilde{\mathbf{y}}\|_2 \leq \sqrt{n_A} \mu_1 \|\tilde{\mathbf{y}}\|_2.$$

Putting all together,

$$\begin{aligned} \lambda_1 &\leq \mu_1 \left(\|\tilde{\mathbf{y}}\|_2^2 + 2|\alpha| \sqrt{n_A} \|\tilde{\mathbf{y}}\|_2 + \alpha^2 n_A \right) \\ &= \mu_1 \left(\|\tilde{\mathbf{y}}\|_2 + |\alpha| \sqrt{n_A} \right)^2. \end{aligned}$$

Since $\|\tilde{\mathbf{y}}\|_2, |\alpha| \leq 1$, we have

$$\lambda_1 \leq 2\mu_1 n_A.$$

This completes the proof. $\square$

Before lower bounding the smallest nonzero singular value of $\mathbf{A}^Z$, we characterize the null space of $\mathbf{A}^Z$ by the following Claim.

Claim 7.6.

$$\text{null}(\mathbf{A}^Z) = \text{Span} \left(\mathbf{1}, \begin{pmatrix} \mathbf{z} \\ 0 \end{pmatrix} : \mathbf{z} \in \text{null}(\mathbf{A}) \right).$$

Proof. Let $\mathcal{S}$ be the subspace of $\text{Span} \left(\mathbf{1}, \begin{pmatrix} \mathbf{z} \\ 0 \end{pmatrix} : \mathbf{z} \in \text{null}(\mathbf{A}) \right)$.

We first show that $\mathcal{S} \subseteq \text{null}(\mathbf{A}^Z)$. Clearly,

$$\mathbf{A}^Z \mathbf{1} = \begin{pmatrix} \mathbf{A} & -\mathbf{A}\mathbf{1} \end{pmatrix} \mathbf{1} = \mathbf{0}.$$

For each $\mathbf{z} \in \text{null}(\mathbf{A})$, we have

$$\mathbf{A}^Z \begin{pmatrix} \mathbf{z} \\ 0 \end{pmatrix} = \begin{pmatrix} \mathbf{A} & -\mathbf{A}\mathbf{1} \end{pmatrix} \begin{pmatrix} \mathbf{z} \\ 0 \end{pmatrix} = \mathbf{A}\mathbf{z} = \mathbf{0}.$$

Thus, $\mathcal{S} \subseteq \text{null}(\mathbf{A}^Z)$.

We then show that $\mathcal{S} \supseteq \text{null}(\mathbf{A}^Z)$. For any $\mathbf{z}' \in \text{null}(\mathbf{A}^Z)$, $\mathbf{z}' - \mathbf{z}'_{n_A+1} \mathbf{1} \in \text{null}(\mathbf{A}^Z)$. Let $\mathbf{z} \in \mathbb{R}^n$ such that the i th entry of $\mathbf{z}$ is $\mathbf{z}'_i - \mathbf{z}'_{n_A+1}$. Thus,

$$\mathbf{0} = \mathbf{A}^Z (\mathbf{z}' - \mathbf{z}'_{n_A+1} \mathbf{1}) = \begin{pmatrix} \mathbf{A} & -\mathbf{A}\mathbf{1} \end{pmatrix} \begin{pmatrix} \mathbf{z} \\ 0 \end{pmatrix} = \mathbf{A}\mathbf{z}.$$

That is, any vector in $\mathcal{S}$ can be written as a linear combination of $\mathbf{1}$ and $\begin{pmatrix} \mathbf{z} \\ 0 \end{pmatrix}$ where $\mathbf{z} \in \text{null}(\mathbf{A})$.

Thus, $\mathcal{S} \supseteq \text{null}(\mathbf{A}^Z)$.

Therefore, $\text{null}(\mathbf{A}^Z) = \mathcal{S}$. □

By the above claim, we know that $\mathbf{A}$ and $\mathbf{A}^Z$ have same rank. We bound the smallest nonzero singular value of $\mathbf{A}^Z$ by the following Claim.

Claim 7.7. $\lambda_{\min}((\mathbf{A}^Z)^\top \mathbf{A}^Z) \geq \lambda_{\min}(\mathbf{A}^\top \mathbf{A}) / (n_A + 1)$.

Proof. Let λ_k be the smallest nonzero eigenvalue of $(\mathbf{A}^Z)^\top \mathbf{A}^Z$, and $\mathbf{y} = (\tilde{\mathbf{y}}; \alpha)$ be the associated eigenvector of unit length. Let μ_k be the smallest nonzero eigenvalue of $\mathbf{A}^\top \mathbf{A} = \mathbf{M}$, and $\mathbf{x}$ be the associated eigenvector of unit length.

$$\lambda_k = \tilde{\mathbf{y}}^\top \mathbf{M} \tilde{\mathbf{y}} - 2\alpha \mathbf{1}^\top \mathbf{M} \tilde{\mathbf{y}} + \alpha^2 \mathbf{1}^\top \mathbf{M} \mathbf{1}.$$

Since $\mathbf{M}$ is symmetric PSD, it can be written as $\mathbf{M} = \mathbf{M}^{1/2} \mathbf{M}^{1/2}$.

$$\lambda_k = \left\| \mathbf{M}^{1/2} (\tilde{\mathbf{y}} - \alpha \mathbf{1}) \right\|_2^2.$$

We decompose $\tilde{\mathbf{y}} - \alpha \mathbf{1} := \mathbf{z}_1 + \mathbf{z}_2$, where $\mathbf{z}_1 \in \text{null}(\mathbf{M})$ and $\mathbf{z}_2 \perp \text{null}(\mathbf{M})$. Then, $\lambda_k = \left\| \mathbf{M}^{1/2} \mathbf{z}_2 \right\|_2^2$.

By Courant-Fischer Theorem

$$\lambda_k \geq \mu_k \left\| \mathbf{z}_2 \right\|_2^2.$$

To lower bound λ_k , it suffices to lower bound $\left\| \mathbf{z}_2 \right\|_2$.

Since $\mathbf{y} = (\tilde{\mathbf{y}}; \alpha)$,

$$\|\tilde{\mathbf{y}} - \alpha \mathbf{1}\|_2^2 = \left\| \mathbf{y} - \alpha \begin{pmatrix} \mathbf{1} \\ 1 \end{pmatrix} \right\|_2^2.$$

Since $\mathbf{y} \perp \mathbf{1}$, which is in $\text{null}(\mathbf{A}^Z)$, we have

$$\|\tilde{\mathbf{y}} - \alpha \mathbf{1}\|_2^2 = \|\mathbf{y}\|_2^2 + \alpha^2(1 + n_A) = \|\tilde{\mathbf{y}}\|_2^2 + \alpha^2(2 + n_A).$$

Vectors $\tilde{\mathbf{y}}$, $-\alpha \mathbf{1}$, $\tilde{\mathbf{y}} - \alpha \mathbf{1}$ form a triangle. Let θ be the angle between $\tilde{\mathbf{y}}$ and $\alpha \mathbf{1}$. See Figure 1.

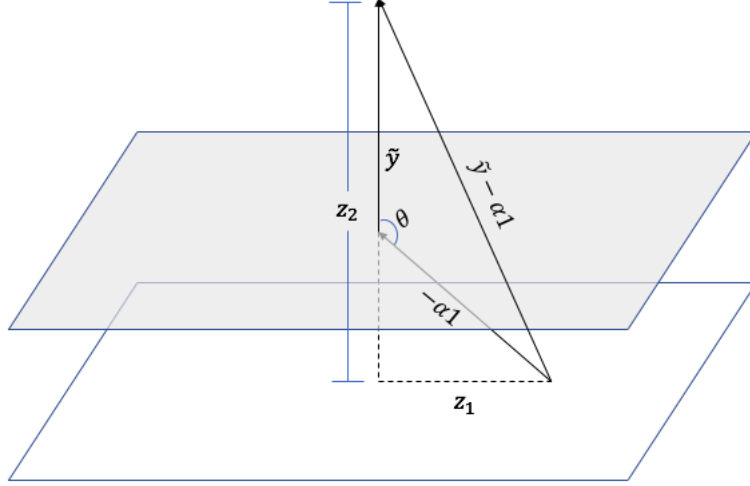


Figure 1: The relation between vectors $\tilde{\mathbf{y}}$, $-\alpha \mathbf{1}$ and $\tilde{\mathbf{y}} - \alpha \mathbf{1}$.

Then,

$$\cos(\theta) = \frac{\|\tilde{\mathbf{y}}\|_2^2 + \|\alpha \mathbf{1}\|_2^2 - \|\tilde{\mathbf{y}} - \alpha \mathbf{1}\|_2^2}{2 \|\tilde{\mathbf{y}}\|_2 \|\alpha \mathbf{1}\|_2} = -\frac{|\alpha|}{\|\tilde{\mathbf{y}}\|_2 \sqrt{n_A}}.$$

Thus,

$$\|z_2\|_2 = \|\tilde{\mathbf{y}}\|_2 + \|\alpha \mathbf{1}\|_2 \cos(\pi - \theta) = \|\tilde{\mathbf{y}}\|_2 + \frac{\alpha^2}{\|\tilde{\mathbf{y}}\|_2} \geq \|\tilde{\mathbf{y}}\|_2.$$

On the other hand,

$$1 = \alpha^2 + \|\tilde{\mathbf{y}}\|_2^2 = \left(\sum_i \tilde{y}_i \right)^2 + \sum_i \tilde{y}_i^2 \leq (n_A + 1) \sum_i \tilde{y}_i^2 = (n_A + 1) \|\tilde{\mathbf{y}}\|_2^2.$$

Therefore,

$$\lambda_k \geq \mu_k \|\tilde{\mathbf{y}}\|_2^2 \geq \frac{\mu_k}{n_A + 1}.$$

This completes the proof. □

The above lemmas give an upper bound on the condition number of $\mathbf{A}^Z$.

Lemma 7.8. $\kappa(\mathbf{A}^Z) \leq O(n_A^{3/2})\kappa(\mathbf{A})$.

7.2 $\mathcal{G}_Z \leq_f \mathcal{G}_{Z,2}$

In this section, we show the reduction from $\mathcal{G}_Z$ to $\mathcal{G}_{Z,2}$. Given an instance of linear system $(\mathbf{A}^Z, \mathbf{c}^Z, \epsilon_1)$ with $\mathbf{A}^Z \in \mathcal{G}_Z$, the goal is to construct an instance of linear system $(\mathbf{A}^{Z,2}, \mathbf{c}^{Z,2}, \epsilon_2)$ with $\mathbf{A}^{Z,2} \in \mathcal{G}_{Z,2}$ such that, there is a map between the solutions of these two linear systems.

Algorithm 10 shows the construction of $(\mathbf{A}^{Z,2}, \mathbf{c}^{Z,2}, \epsilon_2)$, and Algorithm 11 shows the transform from a solution of $(\mathbf{A}^{Z,2}, \mathbf{c}^{Z,2}, \epsilon_2)$ to a solution of $(\mathbf{A}^Z, \mathbf{c}^Z, \epsilon_1)$. Note $\mathbf{c}^{Z,2} = (\mathbf{c}^Z; 0)$.

Algorithm 10 REDUCE $\mathcal{G}_Z$ TO $\mathcal{G}_{Z,2}$

Input: $(\mathbf{A}^Z, \mathbf{c}^Z, \epsilon_1)$ where $\mathbf{A}^Z \in \mathcal{G}_Z$ is an $m \times n$ matrix, $\mathbf{c}^Z \in \mathbb{R}^m$, and $0 < \epsilon_1 < 1$.

Output: $(\mathbf{A}^{Z,2}, \mathbf{c}^{Z,2}, \epsilon_2)$ where $\mathbf{A}^{Z,2} \in \mathcal{G}_{Z,2}$ is an $m' \times n'$ matrix, $\mathbf{c}^{Z,2} \in \mathbb{R}^{m'}$, and $0 < \epsilon_2 < 1$.

- 1: $w \leftarrow 3\epsilon_1^{-1}\sqrt{m}\sigma_{\max}(\mathbf{A}^Z) \|\mathbf{A}^Z\|_{\infty} \|\mathbf{c}^Z\|_2$
 - 2: $k^* \leftarrow \min \{k \in \mathbb{Z} : 2^k \geq \|\mathbf{A}^Z\|_{\infty}\}$
 - 3: Let $\mathbf{a} \in \mathbb{R}^n$
 - 4: **for** $i \leftarrow 1$ to n **do**
 - 5: $\mathbf{a}_i \leftarrow 2^{k^*} - \|\mathbf{A}_i^Z\|_1 / 2$
 - 6: **end for**
 - 7: $\mathbf{A}^{Z,2} = \begin{pmatrix} \mathbf{A}^Z & \mathbf{a} & -\mathbf{a} \\ \mathbf{0} & w & -w \end{pmatrix}$
 - 8: $\mathbf{c}^{Z,2} \leftarrow \begin{pmatrix} \mathbf{c}^Z \\ 0 \end{pmatrix}$
 - 9: **return** $(\mathbf{A}^{Z,2}, \mathbf{c}^{Z,2}, O(\frac{\epsilon_1}{\sigma_{\max}(\mathbf{A}^Z)\|\mathbf{c}^Z\|_2}))$.
-

Algorithm 11 MAPSOLN $\mathcal{G}_{Z,2}$ TO $\mathcal{G}_Z$

Input: $m \times n$ matrix $\mathbf{A}^Z \in \mathcal{G}_Z$, $m' \times n'$ matrix $\mathbf{A}^{Z,2} \in \mathcal{G}_{Z,2}$, vector $\mathbf{c}^Z \in \mathbb{R}^m$, vector $\mathbf{x} \in \mathbb{R}^{n'}$.

Output: Vector $\mathbf{y} \in \mathbb{R}^n$.

- 1: **if** $(\mathbf{A}^Z)^\top \mathbf{c}^Z = \mathbf{0}$ **then**
 - 2: **return** $\mathbf{y} \leftarrow \mathbf{0}$
 - 3: **else**
 - 4: **return** $\mathbf{y} \leftarrow \mathbf{x}_{1:n}$
 - 5: **end if**
-

Lemma 7.9. Let $\mathbf{A}^{Z,2} \mathbf{x}^{Z,2} = \mathbf{c}^{Z,2}$ be the linear system returned by a call to REDUCE $\mathcal{G}_Z$ TO $\mathcal{G}_{Z,2}(\mathbf{A}^Z, \mathbf{c}^Z, \epsilon_1)$. Then,

$$\text{nnz}(\mathbf{A}^{Z,2}) = O(\text{nnz}(\mathbf{A}^Z)).$$

and the largest entry of $\mathbf{A}^{Z,2}$ is at most $\epsilon_1^{-1}\sqrt{m}\sigma_{\max}(\mathbf{A}^Z) \|\mathbf{A}^Z\|_{\infty} \|\mathbf{c}^Z\|_2$.

Claim 7.10. $\text{null}(\mathbf{A}^{Z,2}) = \text{Span} \left(\begin{pmatrix} \mathbf{0} \\ 1 \\ 1 \end{pmatrix}, \begin{pmatrix} \mathbf{x} \\ 0 \\ 0 \end{pmatrix} : \mathbf{x} \in \text{null}(\mathbf{A}^Z) \right)$.

Proof. Let $\mathbf{y} = (\mathbf{x}; \alpha; \beta) \in \mathbb{R}^{n_A+2}$ such that $\mathbf{y} \in \text{null}(\mathbf{A}^{Z,2})$, that is,

$$\mathbf{A}^{Z,2} \mathbf{y} = \begin{pmatrix} \mathbf{A} & \mathbf{a} & -\mathbf{a} \\ \mathbf{0} & w & -w \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \mathbf{A}^Z \mathbf{x} + (\alpha - \beta) \mathbf{a} \\ (\alpha - \beta) w \end{pmatrix} = \mathbf{0}.$$

Since $w > 0$, we have

$$\alpha = \beta.$$

and thus

$$\mathbf{A}^Z \mathbf{x} = \mathbf{0}.$$

Therefore,

$$\text{null}(\mathbf{A}^{Z,2}) = \text{Span} \left(\begin{pmatrix} \mathbf{0} \\ 1 \\ 1 \end{pmatrix}, \begin{pmatrix} \mathbf{x} \\ 0 \\ 0 \end{pmatrix} : \mathbf{x} \in \text{null}(\mathbf{A}^Z) \right).$$

This completes the proof. $\square$

We write $\mathbf{x}^{Z,2}$ as $(\mathbf{x}; \alpha, \beta)$. By Claim 7.10, the vector $(\mathbf{0}; 1; 1)$ is in the null space of $\mathbf{A}^{Z,2}$. Thus,

$$\mathbf{A}^{Z,2} \mathbf{x}^{Z,2} = \mathbf{A}^{Z,2} \left(\mathbf{x}^{Z,2} - \frac{\alpha + \beta}{2} \begin{pmatrix} \mathbf{0} \\ 1 \\ 1 \end{pmatrix} \right) = \mathbf{A}^{Z,2} \begin{pmatrix} \mathbf{x} \\ \frac{\alpha - \beta}{2} \\ \frac{\beta - \alpha}{2} \end{pmatrix}.$$

Without loss of generality, we assume $\alpha = \beta$, that is, $\mathbf{x}^{Z,2} = (\mathbf{x}; \alpha; -\alpha)$.

We first show that the exact solutions of the two linear systems are close.

Lemma 7.11. *Let $\mathbf{x}^{Z,2*} \in \arg \min_{\mathbf{x}} \|\mathbf{A}^{Z,2} \mathbf{x} - \mathbf{c}^{Z,2}\|_2$. Write $\mathbf{x}^{Z,2*} = (\mathbf{s}^*; \alpha; -\alpha)$. Then,*

$$\mathbf{s}^* \in \arg \min_{\mathbf{x}} \|\mathbf{A}^Z \mathbf{x} - \mathbf{c}^Z\|_{\mathbf{P}},$$

where $\mathbf{P} = \mathbf{I} - \frac{\mathbf{a}\mathbf{a}^\top}{w^2 + \|\mathbf{a}\|_2^2}$.

Proof. Without loss of generality, we write $\mathbf{x}^{Z,2} = (\mathbf{x}; \alpha; -\alpha)$. We expand $\mathbf{A}^{Z,2} \mathbf{x}^{Z,2} - \mathbf{c}^{Z,2}$,

$$\begin{aligned} & \min_{\mathbf{x}^{Z,2}} \|\mathbf{A}^{Z,2} \mathbf{x}^{Z,2} - \mathbf{c}^{Z,2}\|_2^2 \\ &= \min_{\mathbf{x}, \alpha} \left\| \begin{pmatrix} \mathbf{A}^Z & \mathbf{a} & -\mathbf{a} \\ \mathbf{0} & w & -w \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \alpha \\ -\alpha \end{pmatrix} - \begin{pmatrix} \mathbf{c}^Z \\ 0 \end{pmatrix} \right\|_2^2 \\ &= \min_{\mathbf{x}, \alpha} \|\mathbf{A}^Z \mathbf{x} - \mathbf{c}^Z + 2\alpha \mathbf{a}\|_2^2 + 4w^2 \alpha^2 \\ &= \min_{\mathbf{x}} \min_{\alpha} 4 \left(w^2 + \|\mathbf{a}\|_2^2 \right) \left(\alpha + \frac{\mathbf{a}^\top (\mathbf{A}^Z \mathbf{x} - \mathbf{c}^Z)}{2(w^2 + \|\mathbf{a}\|_2^2)} \right)^2 + \|\mathbf{A}^Z \mathbf{x} - \mathbf{c}^Z\|_2^2 - \frac{(\mathbf{a}^\top (\mathbf{A}^Z \mathbf{x} - \mathbf{c}^Z))^2}{w^2 + \|\mathbf{a}\|_2^2}. \end{aligned}$$

Since the minimization over $\mathbf{x}$ and α is independent of each other, for any fixed $\mathbf{x}$, the above value is minimized when

$$\alpha = -\frac{\mathbf{a}^\top (\mathbf{A}^Z \mathbf{x} - \mathbf{c}^Z)}{2(w^2 + \|\mathbf{a}\|_2^2)}.$$

Plugging this value of α gives

$$\begin{aligned} \min_{\mathbf{x}^{Z,2}} \|\mathbf{A}^{Z,2} \mathbf{x}^{Z,2} - \mathbf{c}^{Z,2}\|_2^2 &= \min_{\mathbf{x}} \|\mathbf{A}^Z \mathbf{x} - \mathbf{c}^Z\|_2^2 - \frac{(\mathbf{a}^\top (\mathbf{A}^Z \mathbf{x} - \mathbf{c}^Z))^2}{w^2 + \|\mathbf{a}\|_2^2} \\ &= \min_{\mathbf{x}} (\mathbf{A}^Z \mathbf{x} - \mathbf{c}^Z)^\top \left(\mathbf{I} - \frac{\mathbf{a}\mathbf{a}^\top}{w^2 + \|\mathbf{a}\|_2^2} \right) (\mathbf{A}^Z \mathbf{x} - \mathbf{c}^Z) \\ &= \min_{\mathbf{x}} \|\mathbf{A}^Z \mathbf{x} - \mathbf{c}^Z\|_{\mathbf{P}}^2, \end{aligned}$$

where $\mathbf{P} = \mathbf{I} - \frac{\mathbf{a}\mathbf{a}^\top}{w^2 + \|\mathbf{a}\|_2^2}$. The fact $\mathbf{x}^{Z,2*} \in \arg \min_{\mathbf{x}} \|\mathbf{A}^{Z,2}\mathbf{x} - \mathbf{c}^{Z,2}\|_2$ implies that

$$\mathbf{s}^* \in \arg \min_{\mathbf{x}} \|\mathbf{A}^Z \mathbf{x} - \mathbf{c}^Z\|_{\mathbf{P}},$$

which completes the proof. $\square$

Lemma 7.12. *Let $\mathbf{x}^{Z,2*} \in \arg \min_{\mathbf{x}} \|\mathbf{A}^{Z,2}\mathbf{x} - \mathbf{c}^{Z,2}\|_2$ such that $\mathbf{x}^{Z,2*}$ has the form $(\mathbf{s}^*; \alpha; -\alpha)$ and $\|\mathbf{x}^{Z,2*}\|_2$ is minimized. Let $\mathbf{x}^* \in \arg \min_{\mathbf{x}} \|\mathbf{A}^Z \mathbf{x} - \mathbf{c}^Z\|_2$ such that $\|\mathbf{x}^*\|_2$ is minimized. Then,*

$$\|\mathbf{A}^Z(\mathbf{s}^* - \mathbf{x}^*)\|_2 \leq \frac{\|\mathbf{a}\|_2}{w} \sigma_{\max}(\mathbf{A}^Z) \|\mathbf{c}^Z\|_2 \|\Pi_{\mathbf{A}^Z} \mathbf{c}^Z\|_2.$$

Proof. Let $\gamma \stackrel{\text{def}}{=} \frac{\|\mathbf{a}\|_2^2}{w^2 + \|\mathbf{a}\|_2^2}$. By the definition of $\mathbf{P}$ in Lemma 7.11,

$$(1 - \gamma) \mathbf{I} \preceq \mathbf{P} \preceq \mathbf{I}.$$

Thus,

$$(1 - \gamma) \|\mathbf{A}^Z \mathbf{s}^* - \mathbf{c}^Z\|_2^2 \leq \|\mathbf{A}^Z \mathbf{s}^* - \mathbf{c}^Z\|_{\mathbf{P}}^2 \leq \|\mathbf{A}^Z \mathbf{x}^* - \mathbf{c}^Z\|_{\mathbf{P}}^2 \leq \|\mathbf{A}^Z \mathbf{x}^* - \mathbf{c}^Z\|_2^2. \quad (37)$$

The second inequality is due to $\mathbf{s}^* \in \arg \min_{\mathbf{x}} \|\mathbf{A}^Z \mathbf{x} - \mathbf{c}^Z\|_{\mathbf{P}}$. Note that $\mathbf{A}^Z \mathbf{x}^* - \mathbf{c}^Z$ is orthogonal to the column space of $\mathbf{A}^Z$, in particular, it is orthogonal to $\mathbf{A}^Z(\mathbf{s}^* - \mathbf{x}^*)$.

$$\|\mathbf{A}^Z(\mathbf{s}^* - \mathbf{x}^*)\|_2^2 = \|\mathbf{A}^Z \mathbf{s}^* - \mathbf{c}^Z\|_2^2 - \|\mathbf{A}^Z \mathbf{x}^* - \mathbf{c}^Z\|_2^2.$$

By Equation (37),

$$\|\mathbf{A}^Z(\mathbf{s}^* - \mathbf{x}^*)\|_2^2 \leq \frac{\gamma}{1 - \gamma} \|\mathbf{A}^Z \mathbf{x}^* - \mathbf{c}^Z\|_2^2 \leq \frac{\gamma}{1 - \gamma} \|\mathbf{c}^Z\|_2^2.$$

If $\Pi_{\mathbf{A}^Z} \mathbf{c}^Z = \mathbf{0}$, then $\mathbf{s}^* = \mathbf{x}^* = \mathbf{0}$ and the claim holds. Otherwise, by Lemma 4.8,

$$\|\mathbf{A}^Z(\mathbf{s}^* - \mathbf{x}^*)\|_2 \leq \sqrt{\frac{\gamma}{1 - \gamma}} \sigma_{\max}(\mathbf{A}^Z) \|\mathbf{c}^Z\|_2 \|\Pi_{\mathbf{A}^Z} \mathbf{c}^Z\|_2.$$

Plugging the value of γ completes the proof. $\square$

We then show the approximate solvers of the two linear systems are close.

Lemma 7.13. *Let ϵ_2 be the error parameter returned by a call to $\text{REDUCE } \mathcal{G}_z \text{ TO } \mathcal{G}_{z,2}(\mathbf{A}^Z, \mathbf{c}^Z, \epsilon_1)$ (Algorithm 10). Let $\mathbf{x}^{Z,2}$ be a vector such that $\|\mathbf{A}^{Z,2} \mathbf{x}^{Z,2} - \Pi_{\mathbf{A}^{Z,2}} \mathbf{c}^{Z,2}\|_2 \leq \epsilon_2 \|\Pi_{\mathbf{A}^{Z,2}} \mathbf{c}^{Z,2}\|_2$. Let $\mathbf{x}$ be the vector returned by Algorithm 11. Then,*

$$\|\mathbf{A}^Z \mathbf{x} - \Pi_{\mathbf{A}^Z} \mathbf{c}^Z\|_2 \leq \epsilon_1 \|\Pi_{\mathbf{A}^Z} \mathbf{c}^Z\|_2.$$

Proof. If $\mathbf{\Pi}_{\mathbf{A}^Z} \mathbf{c}^Z = \mathbf{0}$, then $\mathbf{x} = \mathbf{0}$ and the statement holds. If $\mathbf{\Pi}_{\mathbf{A}^Z} \mathbf{c}^Z \neq \mathbf{0}$, then $\mathbf{x} = \mathbf{x}_{1:n_A+1}^{Z,2}$. Let $\mathbf{x}^* \in \arg \min_{\mathbf{x}} \|\mathbf{A}^Z \mathbf{x} - \mathbf{c}^Z\|_2$. We now bound the difference between our solution $\mathbf{x}$ and $\mathbf{x}^*$. Let $\mathbf{x}^{Z,2*} \in \arg \min_{\mathbf{x}} \|\mathbf{A}^{Z,2} \mathbf{x} - \mathbf{c}^{Z,2}\|_2$ of the form $(\mathbf{s}^*; \alpha; -\alpha)$. By the triangle inequality,

$$\|\mathbf{A}^Z (\mathbf{x} - \mathbf{x}^*)\|_2 \leq \|\mathbf{A}^Z (\mathbf{x} - \mathbf{s}^*)\|_2 + \|\mathbf{A}^Z (\mathbf{s}^* - \mathbf{x}^*)\|_2. \quad (38)$$

The second term is upper bound by Lemma 7.12. It remains to upper bound the first term. Let

$$\delta \stackrel{\text{def}}{=} \epsilon_2 \|\mathbf{\Pi}_{\mathbf{A}^{Z,2}} \mathbf{c}^{Z,2}\|_2. \quad (39)$$

Without loss of generality, we write $\mathbf{x}^{Z,2}$ as $(\mathbf{x}; \beta; -\beta)$, where $\mathbf{x}$ is the output of Algorithm 11.

$$\begin{aligned} \|\mathbf{A}^{Z,2} \mathbf{x}^{Z,2} - \mathbf{\Pi}_{\mathbf{A}^{Z,2}} \mathbf{c}^{Z,2}\|_2^2 &= \|\mathbf{A}^{Z,2} (\mathbf{x}^{Z,2} - \mathbf{x}^{Z,2*})\|_2^2 \\ &= \left\| \begin{pmatrix} \mathbf{A}^Z & \mathbf{a} & -\mathbf{a} \\ \mathbf{0} & w & -w \end{pmatrix} \begin{pmatrix} \mathbf{x} - \mathbf{s}^* \\ \beta - \alpha \\ \alpha - \beta \end{pmatrix} \right\|_2^2 \\ &= \left\| \begin{pmatrix} \mathbf{A}^Z (\mathbf{x} - \mathbf{s}^*) + 2(\beta - \alpha) \mathbf{a} \\ 2w(\alpha - \beta) \end{pmatrix} \right\|_2^2 \\ &= \|\mathbf{A} (\mathbf{x} - \mathbf{s}^*) + 2(\beta - \alpha) \mathbf{a}\|_2^2 + 4w^2 (\alpha - \beta)^2. \end{aligned}$$

Since $\|\mathbf{A}^{Z,2} \mathbf{x}^{Z,2} - \mathbf{\Pi}_{\mathbf{A}^{Z,2}} \mathbf{c}^{Z,2}\|_2 \leq \delta$, we have

$$4w^2 (\alpha - \beta)^2 \leq \delta^2,$$

that is,

$$(\alpha - \beta)^2 \leq \frac{\delta^2}{4w^2}. \quad (40)$$

Similarly, we have

$$\|\mathbf{A}^Z (\mathbf{x} - \mathbf{s}^*) + 2(\beta - \alpha) \mathbf{a}\|_2 \leq \delta.$$

By the triangle inequality,

$$\|\mathbf{A}^Z (\mathbf{x} - \mathbf{s}^*)\|_2 - 2\|(\beta - \alpha) \mathbf{a}\|_2 \leq \delta.$$

Plugging (40) and (39) into the above inequality, and rearranging it,

$$\|\mathbf{A}^Z (\mathbf{x} - \mathbf{s}^*)\|_2 \leq \left(1 + \frac{\|\mathbf{a}\|_2}{w}\right) \epsilon_2 \|\mathbf{c}^{Z,2}\|_2.$$

By Lemma 4.8,

$$\|\mathbf{A}^Z (\mathbf{x} - \mathbf{s}^*)\|_2 \leq \left(1 + \frac{\|\mathbf{a}\|_2}{w}\right) \epsilon_2 \sigma_{\max}(\mathbf{A}^Z) \|\mathbf{c}^{Z,2}\|_2 \|\mathbf{\Pi}_{\mathbf{A}^Z} \mathbf{c}^Z\|_2.$$

Together with Lemma 7.12 and Equation (38), we have

$$\|\mathbf{A}^Z (\mathbf{x} - \mathbf{x}^*)\|_2 \leq \left(\frac{\|\mathbf{a}\|_2}{w} + \left(1 + \frac{\|\mathbf{a}\|_2}{w}\right) \epsilon_2 \right) \sigma_{\max}(\mathbf{A}^Z) \|\mathbf{c}^Z\|_2 \|\mathbf{\Pi}_{\mathbf{A}^Z} \mathbf{c}^Z\|_2.$$

According to our setting of w and $\mathbf{a}$ in line 1 and line 5 of Algorithm 10,

$$w \geq \frac{\|\mathbf{a}\|_2}{\epsilon_2}.$$

This implies that

$$\|\mathbf{A}^Z \mathbf{x} - \mathbf{\Pi}_{\mathbf{A}^Z} \mathbf{c}^Z\|_2 \leq \epsilon_1 \|\mathbf{\Pi}_{\mathbf{A}^Z} \mathbf{c}^Z\|_2,$$

which completes the proof. $\square$

We then bound the singular values of $\mathbf{A}^{Z,2}$.

Claim 7.14. $\lambda_{\max}((\mathbf{A}^{Z,2})^\top \mathbf{A}^{Z,2}) \leq O\left(\epsilon_1^{-2} m \lambda_{\max}((\mathbf{A}^Z)^\top \mathbf{A}^Z) \|\mathbf{A}^Z\|_\infty^2 \|\mathbf{c}^Z\|_2^2\right).$

Proof. Let λ_1 be the largest eigenvalue of $(\mathbf{A}^{Z,2})^\top \mathbf{A}^{Z,2}$, and $\mathbf{y} = (\tilde{\mathbf{y}}; \alpha; -\alpha)$ be the associated eigenvector of unit length. Let μ_1 be the largest eigenvalue of $(\mathbf{A}^Z)^\top \mathbf{A}^Z$.

$$\begin{aligned} \lambda_1 &= \left\| \begin{pmatrix} \mathbf{A}^Z & \mathbf{a} & -\mathbf{a} \\ \mathbf{0} & w & -w \end{pmatrix} \mathbf{y} \right\|_2^2 \\ &= \left\| \begin{pmatrix} \mathbf{A}^Z & \mathbf{a} & -\mathbf{a} \\ \mathbf{0} & 1 & -1 \end{pmatrix} \mathbf{y} + \begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & w-1 & -w+1 \end{pmatrix} \mathbf{y} \right\|_2^2 \\ &= \tilde{\mathbf{y}}^\top (\mathbf{A}^Z)^\top \mathbf{A}^Z \tilde{\mathbf{y}} + 4\alpha \mathbf{a}^\top \mathbf{A}^Z \tilde{\mathbf{y}} + 4\alpha^2 (\mathbf{a}^\top \mathbf{a} + 1) + 8\alpha^2(w-1) + 4\alpha^2(w-1)^2. \end{aligned}$$

By the Courant-Fischer Theorem,

$$\tilde{\mathbf{y}}^\top (\mathbf{A}^Z)^\top \mathbf{A}^Z \tilde{\mathbf{y}} \leq \mu_1 \tilde{\mathbf{y}}^\top \tilde{\mathbf{y}} \leq \mu_1.$$

By Cauchy-Schwarz inequality,

$$|\mathbf{a}^\top \mathbf{A}^Z \tilde{\mathbf{y}}| \leq \|\mathbf{a}\|_2 \|\mathbf{A}^Z \tilde{\mathbf{y}}\|_2 \leq \|\mathbf{A}\|_\infty \sqrt{m} \sqrt{\mu_1}.$$

Thus,

$$\begin{aligned} \lambda_1 &\leq \mu_1 + 4|\alpha| \|\mathbf{A}^Z\|_\infty \sqrt{m} \sqrt{\mu_1} + 4\alpha^2 (\|\mathbf{A}^Z\|_\infty^2 m + 1) + 12\alpha^2 w^2 \\ &\leq (\sqrt{\mu_1} + 2\|\mathbf{A}^Z\|_\infty \sqrt{m})^2 + 12\alpha^2 w^2 \\ &\leq 2\mu_1 + 8m \|\mathbf{A}^Z\|_\infty^2 + 12\alpha^2 w^2. \end{aligned}$$

Our setting of w in line 1 of Algorithm 10 gives

$$\lambda_{\max}((\mathbf{A}^{Z,2})^\top \mathbf{A}^{Z,2}) \leq 2\mu_1 + 8m \|\mathbf{A}^Z\|_\infty^2 + 108\epsilon_1^{-2} m \mu_1 \|\mathbf{A}^Z\|_\infty^2 \|\mathbf{c}^Z\|_2^2.$$

This completes the proof. $\square$

Claim 7.15. $\lambda_{\min}((\mathbf{A}^{Z,2})^\top \mathbf{A}^{Z,2}) \geq \frac{2\lambda_{\min}((\mathbf{A}^Z)^\top \mathbf{A}^Z)}{2(\|\mathbf{A}^Z\|_\infty^2 m + 1) + \lambda_{\min}((\mathbf{A}^Z)^\top \mathbf{A}^Z)}.$

Proof. Let

$$\mathbf{C} = \begin{pmatrix} \mathbf{A}^Z & \mathbf{a} & -\mathbf{a} \\ \mathbf{0} & 1 & -1 \end{pmatrix}.$$

Note

$$\mathbf{A}^{Z,2} = \mathbf{C} + \begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & w-1 & -w+1 \end{pmatrix}.$$

We can check that

$$(\mathbf{A}^{Z,2})^\top \mathbf{A}^{Z,2} = \mathbf{C}^\top \mathbf{C} + \begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & w^2-1 & -(w^2-1) \\ \mathbf{0} & -(w^2-1) & w^2-1 \end{pmatrix}.$$

By our setting, $w^2 - 1 \geq 0$. The second matrix is a rank-one PSD matrix. Since $\mathbf{A}^{Z,2}$ and $\mathbf{C}$ has same null space,

$$\lambda_{\min} \left((\mathbf{A}^{Z,2})^\top \mathbf{A}^{Z,2} \right) \geq \lambda_{\min} \left(\mathbf{C}^\top \mathbf{C} \right).$$

It suffices to lower bound the smallest nonzero eigenvalue of $\mathbf{C}^\top \mathbf{C}$.

Let $\lambda_k \stackrel{\text{def}}{=} \lambda_{\min}(\mathbf{C}^\top \mathbf{C})$, and $\mathbf{y} = (\tilde{\mathbf{y}}; \alpha; -\alpha)$ be the associated eigenvector of unit length. Let $\mu_k \stackrel{\text{def}}{=} \lambda_{\min}((\mathbf{A}^Z)^\top \mathbf{A}^Z)$, and $\mathbf{x}$ be the associated eigenvector of unit length.

$$\begin{aligned} \lambda_k &= \tilde{\mathbf{y}}^\top (\mathbf{A}^Z)^\top \mathbf{A}^Z \tilde{\mathbf{y}} + 4\alpha \mathbf{a}^\top \mathbf{A}^Z \tilde{\mathbf{y}} + 4\alpha^2 (\mathbf{a}^\top \mathbf{a} + 1) \\ &= \|\mathbf{A}^Z \tilde{\mathbf{y}} + 2\alpha \mathbf{a}\|_2^2 + 4\alpha^2 \\ &\geq 4\alpha^2. \end{aligned} \tag{41}$$

On the other hand,

$$\begin{aligned} \lambda_k &= \left(2\alpha \sqrt{\mathbf{a}^\top \mathbf{a} + 1} + \frac{\mathbf{a}^\top \mathbf{A}^Z \tilde{\mathbf{y}}}{\sqrt{\mathbf{a}^\top \mathbf{a} + 1}} \right)^2 + \tilde{\mathbf{y}}^\top (\mathbf{A}^Z)^\top \mathbf{A}^Z \tilde{\mathbf{y}} - \frac{\tilde{\mathbf{y}}^\top (\mathbf{A}^Z)^\top \mathbf{a} \mathbf{a}^\top \mathbf{A}^Z \tilde{\mathbf{y}}}{\mathbf{a}^\top \mathbf{a} + 1} \\ &\geq \tilde{\mathbf{y}}^\top (\mathbf{A}^Z)^\top \left(\mathbf{I} - \frac{\mathbf{a} \mathbf{a}^\top}{\mathbf{a}^\top \mathbf{a} + 1} \right) \mathbf{A}^Z \tilde{\mathbf{y}}. \end{aligned}$$

Take eigen-decomposition of the matrix in the middle,

$$\lambda_k \geq \tilde{\mathbf{y}}^\top (\mathbf{A}^Z)^\top \mathbf{Q} \mathbf{D} \mathbf{Q}^\top \mathbf{A}^Z \tilde{\mathbf{y}},$$

where $\mathbf{D} := \text{DIAG} \left(1 - \frac{\mathbf{a}^\top \mathbf{a}}{\mathbf{a}^\top \mathbf{a} + 1}, 1, \dots, 1 \right)$. By Claim 7.10, $\mathbf{y} \perp \text{null}(\mathbf{A}^{Z,2})$ implies $\tilde{\mathbf{y}} \perp \text{null}(\mathbf{A}^Z)$. By the Courant-Fischer Theorem,

$$\lambda_k \geq \frac{1}{\mathbf{a}^\top \mathbf{a} + 1} \|\mathbf{A}^Z \tilde{\mathbf{y}}\|_2^2 \geq \frac{1}{\mathbf{a}^\top \mathbf{a} + 1} \mu_k \|\tilde{\mathbf{y}}\|_2^2.$$

Together with (41),

$$\lambda_k \geq \max \left\{ 4\alpha^2, \frac{\mu_k}{\mathbf{a}^\top \mathbf{a} + 1} \|\tilde{\mathbf{y}}\|_2^2 \right\}.$$

Since $\|\mathbf{y}\|_2^2 = \|\tilde{\mathbf{y}}\|_2^2 + 2\alpha^2 = 1$,

$$\lambda_k \geq \frac{2\mu_k}{2(\mathbf{a}^\top \mathbf{a} + 1) + \mu_k} \geq \frac{2\mu_k}{2(\|\mathbf{A}\|_\infty^2 m + 1) + \mu_k}.$$

This completes the proof. $\square$

The above two lemmas give the following bound on the condition number of $\mathbf{A}^{Z,2}$.

Lemma 7.16. $\kappa(\mathbf{A}^{Z,2}) = O(\epsilon_1^{-1} \sqrt{m} \|\mathbf{A}^Z\|_\infty \|\mathbf{c}^Z\|_2 (\sigma_{\max}(\mathbf{A}^Z) + \kappa(\mathbf{A}^Z) \sqrt{m} \|\mathbf{A}^Z\|_\infty)).$

Proof of Lemma 3.4. By Lemma 7.2 and 7.9, we have

$$\text{nnz}(\mathbf{A}^{Z,2}) = O(s).$$

The smallest nonzero entry does not change. The largest entry of $\mathbf{A}^{Z,2}$ is at most

$$\max\{O(\|\mathbf{A}\|_\infty), w\} = O(\epsilon^{-1} n^2 \sqrt{m} \sigma_{\max}(\mathbf{A}) \|\mathbf{A}\|_\infty \|\mathbf{c}^A\|_2).$$

By Claim B.1 and B.2, the largest entry is upper bounded by

$$O(\epsilon^{-1} s^{9/2} U^3).$$

By Lemma 7.8 and 7.16, and Claim B.1 and B.2

$$\kappa(\mathbf{A}^{Z,2}) = O(\epsilon^{-1} s^8 U^3 K).$$

By Lemma 7.4 and 7.13, we have

$$\epsilon_2^{-1} = O(s^{5/2} U^2) \epsilon^{-1}.$$

This completes the proof. □

8 2D Trusses

In this section, we show that the matrix $\mathbf{B}$ constructed by $\text{REDUCE } \mathcal{G}_{z,2} \text{ TO } \mathcal{MC}_2$ (Algorithm 1), is a 2D Truss Incidence Matrix defined in Definition 1.4. It follows that for any function f , $\mathcal{G} \leq_f \mathcal{MC}_2$ implies $\mathcal{G} \leq_f \mathcal{T}_2$. We assume that the algorithm $\text{REDUCE } \mathcal{G}_{z,2} \text{ TO } \mathcal{MC}_2$ is called on a matrix $\mathbf{A}$ with no two identical rows: if there are identical rows, these rows can be collapsed into one row without changing the associated normal equations, by reweighting the resulting row, similar to the technique used in the proof of Claim 9.4. Details are left to the reader. A key step is to show that a 2-commodity gadget in the reduction corresponds to a 2D truss subgraph, which we call the 2D-truss gadget.

Without loss of generality, we let $\mathbf{u}$ -variables correspond to the horizontal axis and $\mathbf{v}$ -variables to the vertical axis of the 2D plane. According to Definition 1.2 and 1.4:

1. an equation $\mathbf{u}_i - \mathbf{u}_j = 0$ in a 2-commodity linear system corresponds to a horizontal edge in the 2D plane;
2. an equation $\mathbf{v}_i - \mathbf{v}_j = 0$ in a 2-commodity linear system corresponds to a vertical edge in the 2D plane;
3. an equation $\mathbf{u}_i - \mathbf{v}_i - (\mathbf{u}_j - \mathbf{v}_j) = 0$ in a 2-commodity linear system corresponds to a diagonal edge in the 2D plane.

Note that our reduction here heavily relies on the ability to choose arbitrary weights. In particular, the weights on the elements are not related at all with the distances between the corresponding vertices.

Our strategy for picking the coordinates of the vertices of the constructed 2D truss is the following: we first pick the coordinates of the original n vertices randomly, and then determine the coordinates of the new vertices constructed in the reduction to satisfy all the truss equations.

For the n original vertices, we pick their $\mathbf{u}$ -coordinates arbitrarily and pick their $\mathbf{v}$ -coordinates randomly. We pick an n -dimensional random vector $\mathbf{y}$ uniformly distributed on the n -dimensional sphere centered at the origin and with radius $R = \|\mathbf{A}\|_1 n^{10}$. We then round each entry of $\mathbf{y}$ to have precision $\delta = 10^{-10}$, so that the total number of bits used to store an entry is at most $O(\log(n \|\mathbf{A}\|_1))$. Let $\tilde{\mathbf{y}}$ be the vector after rounding. We assign the $\mathbf{v}$ -coordinate of the i th vertex to be the i th entry of $\tilde{\mathbf{y}}$.

We then pick the coordinates of the new vertices in the order they are created. Note that each time we replace two vertices in the current equations, say $\mathbf{s}^{j_1}, \mathbf{s}^{j_2}$, whose coordinates have already been determined, we create a 2D truss gadget with 7 new vertices, say $\mathbf{s}^t, \mathbf{s}^{t+1}, \dots, \mathbf{s}^{t+6}$ (See Algorithm 2 $\mathcal{MC}_2\text{GADGET}$ for the construction.). According to the construction of this gadget, the new vertices $\mathbf{s}^{t+1}, \dots, \mathbf{s}^{t+6}$ only appear in this single gadget, whose coordinates do not affect other vertices. Figure 2 is the corresponding subgraph which satisfies all the equations in the 2D truss gadget. Note the two triangles $(\mathbf{s}^{t+3}, \mathbf{s}^{t+5}, \mathbf{s}^{t+6})$ and $(\mathbf{s}^{t+3}, \mathbf{s}^{t+4}, \mathbf{s}^{t+5})$ need to be isosceles right triangles, which implies $\mathbf{v}_t = (\mathbf{v}_{j_1} + \mathbf{v}_{j_2})/2$. Note also that we can assign $\mathbf{u}$ -coordinates to the new vertices which are not between the $\mathbf{u}$ -coordinates of $\mathbf{s}^{j_1}$ and $\mathbf{s}^{j_2}$. In fact, using an appropriate choice of $\mathbf{u}$ -coordinates and edge weights, we can always place $\mathbf{s}^t, \mathbf{s}^{t+1}, \dots, \mathbf{s}^{t+6}$ to get the desired equations, provided $\mathbf{v}_{j_1} \neq \mathbf{v}_{j_2}$, which we later will argue holds with high probability using Lemma 8.1. First, however, we state and prove Lemma 8.1.

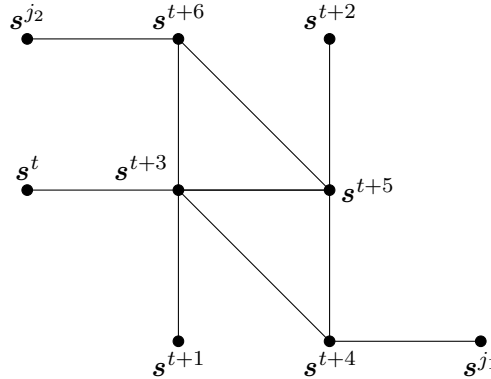


Figure 2: Geometric realization of the multicommodity flow gadget generated by as a truss matrix

Lemma 8.1. *Let $\mathbf{a} \in \mathbb{R}^n$ be a fixed vector such that $-2 \leq \mathbf{a}_i \leq 2, \forall i \in [n]$ and $\mathbf{a}^\top \mathbf{1} = 0$, and $\mathbf{a} \neq \mathbf{0}$. Let $\tilde{\mathbf{y}}$ be a vector picked as above. Then,*

$$\Pr(\mathbf{a}^\top \tilde{\mathbf{y}} = 0) \leq \frac{2\delta n^2}{\|\mathbf{a}\|_2 R}.$$

Proof. Let $\Delta \stackrel{\text{def}}{=} \tilde{\mathbf{y}} - \mathbf{y}$. Clearly, $-\delta \leq \Delta_i \leq \delta, \forall i \in [n]$.

$$\mathbf{a}^\top \tilde{\mathbf{y}} = \mathbf{a}^\top (\mathbf{y} + \Delta) \leq \mathbf{a}^\top \mathbf{y} + \sum_{i \in [n]} |\mathbf{a}_i| |\Delta_i| \leq \mathbf{a}^\top \mathbf{y} + 2\delta n.$$

Similarly, $\mathbf{a}^\top \tilde{\mathbf{y}} \geq \mathbf{a}^\top \mathbf{y} - 2\delta n$. Thus,

$$\Pr(\mathbf{a}^\top \tilde{\mathbf{y}} = 0) \leq \Pr(|\mathbf{a}^\top \mathbf{y}| \leq 2\delta n).$$

Since the distribution of $\mathbf{y}$ is rotation invariant, we assume without loss of generality $\mathbf{a} = (\|\mathbf{a}\|_2, 0, \dots, 0)$. Let A be the area of the n -dimensional sphere.

$$A = \frac{2\pi^{(n+1)/2}}{\Gamma(\frac{n+1}{2})} \cdot R^n.$$

Let A_δ be the area of $\{\mathbf{y} : \|\mathbf{y}\|_2 = R, |\mathbf{y}_1| \leq 2\delta n / \|\mathbf{a}\|_2\}$. Then,

$$A_\delta \leq \frac{2\pi^{n/2}}{\Gamma(\frac{n}{2})} \cdot R^{n-1} \cdot \frac{2\delta n}{\|\mathbf{a}\|_2}.$$

Thus, (assume n is even, the case of odd n can be checked similarly)

$$\begin{aligned} \Pr(|\mathbf{a}^\top \mathbf{y}| \leq \delta) &= \frac{A_\delta}{A} \\ &\leq \frac{2\delta n}{\|\mathbf{a}\|_2 \sqrt{\pi} R} \cdot \frac{\Gamma(\frac{n+1}{2})}{\Gamma(\frac{n}{2})} \\ &= \frac{2\delta n}{\|\mathbf{a}\|_2 R} \cdot \frac{n! \sqrt{\pi}}{n!!(n-2)!!} \\ &\leq \frac{2\delta n^2}{\|\mathbf{a}\|_2 R}. \end{aligned}$$

This completes the proof. $\square$

By our construction of the truss, for each vertex, its $\mathbf{v}$ -coordinate can be written as a fixed convex combination of $\tilde{\mathbf{y}}$, say $\mathbf{c}^\top \tilde{\mathbf{y}}$ in which $\mathbf{c}^\top \mathbf{1} = 1$ and $\mathbf{c}_i \geq 0, \forall i \in [n]$. Note that when algorithm $\text{REDUCE } \mathcal{G}_{z,2}\text{TO}\mathcal{MC}_2$ is applied to a matrix $\mathbf{A}$, it processes the j th row in k iterations where by Lemma 4.1 we have $k \leq \log \|\mathbf{A}_j\|_1$. Let $p \stackrel{\text{def}}{=} \log \|\mathbf{A}_j\|_1$. Given how the convex combination specified by $\mathbf{c}$ is formed, it follows that $2^p \mathbf{c}$ is an integer vector.

Next we argue that when two variables are chosen for pairing by the $\text{REDUCE } \mathcal{G}_{z,2}\text{TO}\mathcal{MC}_2$ algorithm as it processes some row $\mathbf{A}_j$, these two variables will have their $\mathbf{v}$ -coordinates represented as convex combinations $\mathbf{c}^\top \tilde{\mathbf{y}}$ and $\mathbf{d}^\top \tilde{\mathbf{y}}$ where crucially $\mathbf{c} \neq \mathbf{d}$. This ensures that $2^p(\mathbf{c} - \mathbf{d})$ is a non-zero integer vector, and hence $\|\mathbf{c} - \mathbf{d}\|_2 \geq 2^{-p} = 1 / \|\mathbf{A}_j\|_1$. This follows from stronger claim stated below, which we prove later.

Claim 8.2. *Suppose algorithm $\text{REDUCE } \mathcal{G}_{z,2}\text{TO}\mathcal{MC}_2$ is processing some row $\mathbf{A}_j$. Let V^l be the set of variables with non-zero coefficients in the main equation $\mathcal{A}_j$ at the l th iteration of the while-loop in Line 20 of $\text{REDUCE } \mathcal{G}_{z,2}\text{TO}\mathcal{MC}_2$. Let S^l be the set of associated vertices. Consider two arbitrary vertices $\mathbf{s}, \mathbf{t} \in S^l$, and let $\mathbf{c}, \mathbf{d} \in \mathbb{R}^n$ be the non-negative vectors with $\mathbf{c}^\top \mathbf{1} = \mathbf{d}^\top \mathbf{1} = 1$ s.t. the $\mathbf{v}$ -coordinates of $\mathbf{s}$ and $\mathbf{t}$ represented as convex combinations are $\mathbf{c}^\top \tilde{\mathbf{y}}$ and $\mathbf{d}^\top \tilde{\mathbf{y}}$ respectively. For every $i \in [n]$, we view entries $\mathbf{c}_i$ and $\mathbf{d}_i$ as fixed point binary numbers: $\mathbf{c}_i = \alpha_0^i \alpha_1^i \alpha_2^i \dots \alpha_p^i$ and $\mathbf{d}_i = \beta_0^i \beta_1^i \beta_2^i \dots \beta_p^i$, or equivalently $\mathbf{c}_i = \sum_{k=0}^p \alpha_k^i 2^{-k}$ and $\mathbf{d}_i = \sum_{k=0}^p \beta_k^i 2^{-k}$. Then there is no index k s.t. $1 = \alpha_k^i = \beta_k^i$, i.e. there is no index where the k th bit is 1 in both strings.*

These two vertices have same $\mathbf{v}$ -coordinate if and only if $(\mathbf{c} - \mathbf{d})^\top \tilde{\mathbf{y}} = 0$. Let $\mathbf{a} \stackrel{\text{def}}{=} \mathbf{c} - \mathbf{d}$. Then, $-2 \leq \mathbf{a}_i \leq 2, \forall i \in [n], \mathbf{a}^\top \mathbf{1} = 0$, and

$$\|\mathbf{a}\|_2 \geq 1/\|\mathbf{A}_j\|_1.$$

By Lemma 8.1,

$$\Pr(\mathbf{c}^\top \tilde{\mathbf{y}} = \mathbf{d}^\top \tilde{\mathbf{y}}) \leq \frac{2\delta n^2 \|\mathbf{A}_j\|_1}{R}.$$

By Lemma 4.1, the total number of the vertices in the truss is at most

$$O(n^2 \log n).$$

By a union bound, the probability that there exist two different vertices with same $\mathbf{v}$ -coordinate is at most

$$\frac{2\delta n^2 \|\mathbf{A}\|_1}{R} \cdot O(n^2 \log n)^2 = O\left(\frac{\log^2 n}{n^4}\right).$$

Proof of Lemma 3.8. Since the linear system for 2D trusses is the same as the linear system for 2-commodity, all complexity parameters of these two linear systems are the same. $\square$

Proof of Claim 8.2. We prove the claim by induction. Our induction hypothesis is simply that the claim holds in round l .

Note that by Lemma 4.1 every time a new vertex is created (during the l th iteration of while-loop in Line 20), it is always paired in the following iteration (iteration $l+1$) and then disappears (i.e. has zero coefficient) in the main equation in all following iterations.

Observe that the convex combination vector $\mathbf{a}$ for each vertex that corresponds to an original variable i is has $\mathbf{a}_i = 1$ and $\mathbf{a}_h = 0$ for all $h \neq i$. This proves the induction hypothesis for the base case of the variables in the main equation $\mathcal{A}_j$ before the first iteration of the while-loop (i.e. $l = 0$).

Suppose $\mathbf{c}, \mathbf{d}$ are the convex combination vectors for two variables that exist in some round l . Assume the induction hypothesis holds for round $l-1$. Write the binary strings for the i th of both vectors as $\mathbf{c}_i = \alpha_0^i \alpha_1^i \alpha_2^i \dots \alpha_p^i$ and $\mathbf{d}_i = \beta_0^i \beta_1^i \beta_2^i \dots \beta_p^i$, or equivalently $\mathbf{c}_i = \sum_{k=0}^p \alpha_k^i 2^{-k}$ and $\mathbf{d}_i = \sum_{k=0}^p \beta_k^i 2^{-k}$. Suppose for the sake of contradiction that there exists some k s.t. $\alpha_k^i = \beta_k^i = 1$. Trivially, it cannot be the case that the such a collision occurs if either variable is an original variable. So both variables must be new variables. The bit string for entry $\mathbf{c}_i$ is created by averaging two bit strings of variables from the main equation in round $l-1$, say $\eta_0^i \eta_1^i \eta_2^i \dots \eta_p^i$ and $\gamma_0^i \gamma_1^i \gamma_2^i \dots \gamma_p^i$. Similarly, bit string for entry $\mathbf{d}_i$ is created by averaging two bit strings of variables from the main equation in round $l-1$, say $\theta_0^i \theta_1^i \theta_2^i \dots \theta_p^i$ and $\sigma_0^i \sigma_1^i \sigma_2^i \dots \sigma_p^i$. Note that each variable can only be paired once in each iteration, so the four bit strings must come from distinct variables in round l .

$\alpha_k^i = 1$ requires that exactly one of the following conditions is true:

1. $\gamma_{k-1}^i = 1$
2. $\eta_{k-1}^i = 1$
3. A ‘‘carry’’ occurred when adding strings $\gamma_k^i \gamma_{k+1}^i \dots \gamma_p^i$ and $\eta_k^i \eta_{k+1}^i \dots \eta_p^i$.

But, Case 3 immediately leads to a contradiction, as a carry can only occur when there exists some bit position g s.t. $\gamma_g^i = \eta_g^i$. But this is false by the induction hypothesis. Thus we must be in either Case 1 or Case 2. By similar logic, we can conclude from $\beta_k^i = 1$ that exactly one of the following must be true: either $\theta_{k-1}^i = 1$ or $\sigma_{k-1}^i = 1$. All together, we have concluded that exactly two of the four bits $\eta_{k-1}^i, \gamma_{k-1}^i, \theta_{k-1}^i$, and σ_{k-1}^i must be set to 1. This contradicts the induction hypothesis. Having established a contradiction whenever the induction hypothesis fails at step l , we have shown that it holds at this step. $\square$

9 Connections with Interior Point Methods

In this section, we discuss how applications in scientific computing and combinatorial optimization produce the linear systems that we show are hard to solve. We first give a brief overview of interior point methods, with focus on how they generate linear systems in Section 9.1. Then we formalize the matrices that interior point methods produce when run on 2-commodity flow matrices 9.2 and isotropic total variation minimization 9.3.

9.1 Brief Overview of Interior Point Methods

Interior point methods [Wri97, Ye11, Nem, BV04, DS08, LS14, LS15] can be viewed as ways of solving convex optimization problems via a sequence of linear systems. For simplicity, we choose the log-barrier based interpretation from Chapter 11 of the book by Boyd and Vandenberghe as our starting point. The main idea is to represent a convex optimization problem with constraints as a sequence of linear programs with terms called barrier functions added to the objective. Specifically, turning the problem:

$$\begin{aligned} \min_{\mathbf{y}} \quad & f(\mathbf{y}) \\ \text{subject to:} \quad & \mathbf{M}\mathbf{y} = \mathbf{b} \\ & \mathbf{y} \geq 0 \end{aligned}$$

into the equality-constrained optimization problem:

$$\begin{aligned} \min_{\mathbf{y}} \quad & t \cdot f(\mathbf{y}) - \sum_i \log(\mathbf{y}_i) \\ \text{subject to:} \quad & \mathbf{M}\mathbf{y} = \mathbf{b} \end{aligned}$$

for a parameter t that is gradually increased (by factors of about $1 + n^{-1/2}$ throughout the course of the algorithm). The solution of the above optimization problem converges to an optimal solution of the original linear programming, as t goes to infinity. Between these increase steps, the algorithm performs Newton steps on this log barrier objective, which when combined with the equality constraint $\mathbf{M}\mathbf{y} = \mathbf{b}$ requires solving the problem:

$$\begin{aligned} \min_{\Delta \mathbf{y}} \quad & -\mathbf{g}(\mathbf{y})^\top \Delta \mathbf{y} \\ \text{subject to:} \quad & \mathbf{M}\Delta \mathbf{y} = \mathbf{0} \\ & \|\Delta \mathbf{y}\|_{\mathbf{H}(\mathbf{y})} \leq 0.1 \end{aligned}$$

which is to maximize the projection along the gradient subject to the second order term being at most 0.1 and staying in the null space. This can in turn be interpreted as a least squares problem, and solving the linear system:

$$\mathbf{M}\mathbf{H}(\mathbf{y})^{-1}\mathbf{M}^\top \mathbf{x} = -t\mathbf{M}\mathbf{H}(\mathbf{y})^{-1}\mathbf{g}(\mathbf{y}). \quad (42)$$

9.2 2-Commodity Flow

We now show that solving 2-commodity flow problems using interior point methods as described in Subsection 9.1 can lead to any system in the class $\mathcal{MC}_2^{>0}$. There are also many variants of the multicommodity flow problem [Mad10], and we work with the minimum cost version due to it being the most general.

Definition 9.1 (Min-cost 2-commodity flow problem). Given a directed graph $G = (V, E)$ with n vertices and m edges, a positive edge-capacity vector $\mathbf{z} \in \mathbb{R}^m$, a positive edge-cost vector $\mathbf{c} \in \mathbb{R}^{2m}$, and two vertex-demand vectors $\mathbf{d}^1, \mathbf{d}^2 \in \mathbb{R}^n$. The goal is to compute two flows $\mathbf{y}^1, \mathbf{y}^2$ such that,

1. $\mathbf{y}^1$ satisfies the demand $\mathbf{d}^1$, and $\mathbf{y}^2$ satisfies the demand $\mathbf{d}^2$,
2. for each edge $e \in E$, the sum of the two flows on e is no larger than the edge capacity $\mathbf{z}_e$, and
3. the total cost of the two flows is minimized.

To formulate this as a linear program, we let $\mathbf{y} = (\mathbf{y}^1; \mathbf{y}^2)$ be the two flows, and $\mathbf{N}$ be the edge-vertex incidence matrix of graph G .

$$\begin{aligned} \min \quad & \mathbf{c}^\top \mathbf{y} \\ \text{s.t.} \quad & \mathbf{N}^\top \mathbf{y}^1 = \mathbf{d}^1 \\ & \mathbf{N}^\top \mathbf{y}^2 = \mathbf{d}^2 \\ & \mathbf{y}^1, \mathbf{y}^2, \mathbf{z} - \mathbf{y}^1 - \mathbf{y}^2 \geq \mathbf{0} \end{aligned}$$

Write this linear programming as the following minimization problem with a logarithmic barrier function,

$$\begin{aligned} \min \quad & \mathbf{c}^\top \mathbf{y} - \frac{1}{t} \sum_{i \in [m]} \log \mathbf{y}_i^1 + \log \mathbf{y}_i^2 + \log(\mathbf{z}_i - \mathbf{y}_i^1 - \mathbf{y}_i^2) \\ \text{s.t.} \quad & \mathbf{N}^\top \mathbf{y}^1 = \mathbf{d}^1 \\ & \mathbf{N}^\top \mathbf{y}^2 = \mathbf{d}^2 \end{aligned}$$

where $t > 0$ is a parameter.

Definition 9.2. We say a linear system $\mathbf{B}^\top \mathbf{B} \mathbf{x} = \mathbf{B}^\top \mathbf{c}$ is a *Minimum Cost 2-commodity Flow IPM Linear System* if it can be obtained from as an instance of the Newton-Step Linear System of Equation (42) for some Min-cost 2-commodity flow problem.

The class of Minimum Cost 2-commodity Flow IPM Linear Systems is appears more restrictive than $\mathcal{MC}_2^{>0}$, but as we will see, it is essentially equivalent, and still sufficiently expressive that any linear system can be reduced to it. The main result that we will sketch in this section is:

Lemma 9.3. *For any linear system $\mathbf{A} \mathbf{x} = \mathbf{c}$ (with error parameter ϵ) with polynomially bounded sparse parameter complexity, there exists an efficient reduction to a Minimum Cost 2-commodity Flow IPM Linear System.*

Our reduction from a general linear system $(\mathbf{A}, \mathbf{c}, \epsilon)$, where $\mathbf{A} \in \mathcal{G}$, to a Minimum Cost 2-commodity Flow IPM Linear System is the same as our reduction to $\mathcal{MC}_2^{>0}$, except that before our chain of reductions, we first multiply the linear system by a diagonal matrix $\mathbf{S}$ with diagonal entries that are ± 1 . We will later specify how to choose these signs. This gives us the linear system $\mathbf{S} \mathbf{A} \mathbf{x} = \mathbf{S} \mathbf{c}$ with the same error parameter as before. It is easy to verify that this system has the same sparse parameter complexity as the original linear system, and an approximate solution to this system is an approximate solution to the original system with the same ϵ .

We now apply our usual chain of reductions to get a linear system in over a matrix in $\mathcal{MC}_2^{>0}$. We write this system as

$$(\mathbf{B}^{>0})^\top \mathbf{B}^{>0} \mathbf{x} = (\mathbf{B}^{>0})^\top \mathbf{c}^{\mathbf{B}^{>0}}.$$

The remainder of this Section is dedicated to showing that this linear system is a Minimum Cost 2-commodity Flow IPM Linear System, thus proving Lemma 9.3.

We can pull out the edge weights from $\mathbf{B}^{>0}$ by writing $(\mathbf{B}^{>0})^\top \mathbf{B}^{>0} = \widehat{\mathbf{B}}^\top \mathbf{W} \widehat{\mathbf{B}}$, where $\widehat{\mathbf{B}}$ is the unweighted 2-commodity edge-vertex incidence matrix with the same edge structure as $\mathbf{B}^{>0}$, and $\mathbf{W}$ is the diagonal matrix of the edge weights. Then, the linear system in $\mathcal{MC}_2^{>0}$ can be written as

$$\widehat{\mathbf{B}}^\top \mathbf{W} \widehat{\mathbf{B}} \mathbf{x} = \widehat{\mathbf{B}}^\top \mathbf{W}^{1/2} \mathbf{c}^{\mathbf{B}^{>0}}. \quad (43)$$

Before we prove Lemma 9.3, we explore some properties of the above linear system.

Claim 9.4. *There exist a 2-commodity edge-vertex incidence matrix $\widetilde{\mathbf{B}}$, a diagonal matrix $\widetilde{\mathbf{W}}$, and a vector $\widetilde{\mathbf{c}}$ such that*

1. $\widetilde{\mathbf{B}}^\top \widetilde{\mathbf{W}} \widetilde{\mathbf{B}} = \widehat{\mathbf{B}}^\top \mathbf{W} \widehat{\mathbf{B}}$ and $\widehat{\mathbf{B}}^\top \mathbf{W}^{1/2} \mathbf{c}^{\mathbf{B}^{>0}} = \widetilde{\mathbf{B}}^\top \widetilde{\mathbf{W}}^{1/2} \widetilde{\mathbf{c}}$,
2. all the rows of $\widetilde{\mathbf{B}}$ are distinct.

Proof. If all rows of $\widehat{\mathbf{B}}$ are distinct, then we set $\widetilde{\mathbf{B}} = \widehat{\mathbf{B}}$, $\widetilde{\mathbf{W}} = \mathbf{W}$ and $\widetilde{\mathbf{c}} = \mathbf{c}^{\mathbf{B}^{>0}}$. Otherwise, assume the i th row and the j th row of $\widehat{\mathbf{B}}$ are the same. Note

$$\widehat{\mathbf{B}}^\top \mathbf{W} \widehat{\mathbf{B}} = \sum_{k \neq i, j} \mathbf{W}_{kk} \widehat{\mathbf{B}}_k^\top \widehat{\mathbf{B}}_k + (\mathbf{W}_{ii} + \mathbf{W}_{jj}) \widehat{\mathbf{B}}_i^\top \widehat{\mathbf{B}}_i,$$

and

$$\widehat{\mathbf{B}}^\top \mathbf{W}^{1/2} \mathbf{c}^{\mathbf{B}^{>0}} = \sum_{k \neq i, j} \mathbf{W}_{kk}^{1/2} \mathbf{c}_k^{\mathbf{B}^{>0}} \widehat{\mathbf{B}}_k^\top + (\mathbf{W}_{ii}^{1/2} \mathbf{c}_i^{\mathbf{B}^{>0}} + \mathbf{W}_{jj}^{1/2} \mathbf{c}_j^{\mathbf{B}^{>0}}) \widehat{\mathbf{B}}_i^\top.$$

We can construct $\widetilde{\mathbf{B}}$ and $\widetilde{\mathbf{W}}$ by removing the j th row of $\widehat{\mathbf{B}}$, $\mathbf{W}$ and set the corresponding edge weight $\widetilde{\mathbf{W}}_{ii}$ to be $\mathbf{W}_{ii} + \mathbf{W}_{jj}$. We construct $\widetilde{\mathbf{c}}$ by removing the j th entry of $\mathbf{c}^{\mathbf{B}^{>0}}$ and set $\widetilde{\mathbf{c}}_i$ to be $(\mathbf{W}_{ii}^{1/2} \mathbf{c}_i^{\mathbf{B}^{>0}} + \mathbf{W}_{jj}^{1/2} \mathbf{c}_j^{\mathbf{B}^{>0}}) / (\mathbf{W}_{ii} + \mathbf{W}_{jj})^{1/2}$. We repeat the above process to merge the identical rows until we get the desired matrices and vector. $\square$

Remark. Let $\mathbf{x}^*$ be a minimizer of $\left\| \mathbf{W}^{1/2} \widehat{\mathbf{B}} \mathbf{x} - \mathbf{c}^{\mathbf{B}^{>0}} \right\|_2$. We know that $\mathbf{x}^*$ is a solution of the normal equations $\widehat{\mathbf{B}}^\top \mathbf{W} \widehat{\mathbf{B}} \mathbf{x} = \widehat{\mathbf{B}}^\top \mathbf{W}^{1/2} \mathbf{c}^{\mathbf{B}^{>0}}$. By Claim 9.4, $\mathbf{x}^*$ is a solution of $\widetilde{\mathbf{B}}^\top \widetilde{\mathbf{W}} \widetilde{\mathbf{B}} \mathbf{x} = \widetilde{\mathbf{B}}^\top \widetilde{\mathbf{W}}^{1/2} \widetilde{\mathbf{c}}$, which in turn gives that $\mathbf{x}^* \in \arg \min_{\mathbf{x}} \left\| \widetilde{\mathbf{W}}^{1/2} \widetilde{\mathbf{B}} \mathbf{x} - \widetilde{\mathbf{c}} \right\|_2$. Let $\mathbf{x}$ be an approximate solution such that

$$\|\mathbf{x} - \mathbf{x}^*\|_{\widehat{\mathbf{B}}^\top \mathbf{W} \widehat{\mathbf{B}}} \leq \epsilon \left\| \widehat{\mathbf{B}}^\top \mathbf{W}^{1/2} \mathbf{c}^{\mathbf{B}^{>0}} \right\|_{(\widehat{\mathbf{B}}^\top \mathbf{W} \widehat{\mathbf{B}})^\dagger}.$$

By Claim 9.4, the above equation is equivalent to

$$\|\mathbf{x} - \mathbf{x}^*\|_{\widetilde{\mathbf{B}}^\top \widetilde{\mathbf{W}} \widetilde{\mathbf{B}}} \leq \epsilon \left\| \widetilde{\mathbf{B}}^\top \widetilde{\mathbf{W}}^{1/2} \widetilde{\mathbf{c}} \right\|_{(\widetilde{\mathbf{B}}^\top \widetilde{\mathbf{W}} \widetilde{\mathbf{B}})^\dagger}.$$

It means that $\mathbf{x}$ is an approximate solution of $\min_{\mathbf{x}} \left\| \widetilde{\mathbf{W}}^{1/2} \widetilde{\mathbf{B}} \mathbf{x} - \widetilde{\mathbf{c}} \right\|_2$ with ϵ -accuracy, vice versa.

Thus, the two minimization problems before and after merging identical rows are equivalent.

Without the loss of generality, we assume that all the rows of $\widehat{\mathbf{B}}$ are distinct. We define the *underlying graph* of $\widehat{\mathbf{B}}$ to be the *simple* graph over n vertices where vertices i and j are connected if and only if $\widehat{\mathbf{B}}$ has at least one type of edges between i and j . According to the constructions

in $\text{REDUCE } \mathcal{G}_{z,2} \text{ TO } \mathcal{MC}_2$ (Algorithm 1) and $\text{REDUCE } \mathcal{MC}_2 \text{ TO } \mathcal{MC}_2^{>0}$ (Algorithm 4), each edge of the underlying graph has exactly 3 types of edges in $\hat{\mathbf{B}}$ (that is, type 1, type 2 and type 1 + 2), and the edge weights of $\mathbf{B}^{>0}$ are either at least 1 or equal to a tiny number δ assigned in line 1 of Algorithm 4. We call edge weights which are at least 1 as *large weights*.

Claim 9.5. *For each edge in the underlying graph of $\hat{\mathbf{B}}$, exactly one of its type 1, type 2, and type 1 + 2 edges has large weight.*

Proof. Note that the large-weight edges all appear in $\mathbf{B} \in \mathcal{MC}_2$ constructed in Algorithm 1. An edge in $\mathbf{B}$ is either a gadget edge in $\mathcal{B}$ or a non-gadget edge in $\mathcal{A}$.

1. By Algorithm 2, all edges in a single $\mathcal{MC}_2$ -gadget are distinct.
2. We show that each gadget edge is distinct from all other edges. By Algorithm 2, in a 2-commodity gadget, each “old” vertex is connected to a “new” vertex, and all new vertices except $\mathbf{x}_t$ are independent of the new vertices created in all other gadgets. However, in each gadget including $\mathbf{x}_t$ (as a new vertex or an odd vertex), $\mathbf{x}_t$ is connected to new vertices which are created in that gadget and do not appear in any other edge outside that gadget.
3. It is possible that two identical non-gadget edges exist when the original linear system instance has redundant or inconsistent constraints. But note all non-gadget edges are type 1 edges, it implies that two identical edges correspond to two identical rows in $\hat{\mathbf{B}}$. By Claim 9.4, we can always merge such identical rows so that all non-gadget edges are distinct.

This completes the proof. $\square$

Proof of Lemma 9.3. Note the matrices $\hat{\mathbf{B}}$, $\mathbf{W}$ and the vector $\mathbf{c}^{\mathbf{B}^{>0}}$ are given by the reductions from the general linear system instance $\mathbf{A}\mathbf{x} = \mathbf{c}^{\mathbf{A}}$. The goal of proving Lemma 9.3 is to determine the edge-vertex incidence matrix $\mathbf{N}$, the cost vector $\mathbf{c}$, the demand vectors $\mathbf{d}^1, \mathbf{d}^2$, the edge capacity vector $\mathbf{z}$, and the flows $\mathbf{y}^1, \mathbf{y}^2$ such that the intermediate linear systems that arise in an interior point method (42) for solving the Min-cost 2-commodity Flow Problem is exactly the above linear system (43).

We first make the coefficient matrices of the two linear systems in (42) and (43) to be the same, by choosing the matrix $\mathbf{N}$ and the vectors $\mathbf{y}^1, \mathbf{y}^2, \mathbf{d}^1, \mathbf{d}^2, \mathbf{z}$ properly. The coefficient matrix of the equality constraints of the 2-commodity linear programming is

$$\mathbf{M} = \begin{pmatrix} \mathbf{N}^\top & \\ & \mathbf{N}^\top \end{pmatrix}. \quad (44)$$

We choose $\mathbf{N}$ to be the edge-vertex incidence matrix of the underlying graph of $\hat{\mathbf{B}}$. By denoting the residue flow amount along an edge as

$$\mathbf{y}^r \stackrel{\text{def}}{=} \mathbf{z} - \mathbf{y}^1 - \mathbf{y}^2,$$

we can rewrite the Hessian as:

$$\mathbf{H} = \frac{1}{t} \sum_{i \in [m]} \frac{1}{(\mathbf{y}_i^1)^2} \mathbf{e}_{2i-1} \mathbf{e}_{2i-1}^\top + \frac{1}{(\mathbf{y}_i^2)^2} \mathbf{e}_{2i} \mathbf{e}_{2i}^\top + \frac{1}{(\mathbf{y}_i^r)^2} (\mathbf{e}_{2i-1} + \mathbf{e}_{2i}) (\mathbf{e}_{2i-1} + \mathbf{e}_{2i})^\top,$$

where $\mathbf{e}_i \in \mathbb{R}^{2m}$ is the standard basis vector.

We can rearrange the rows and columns of $\mathbf{H}$ such that, row $2i-1$ and column $2i-1$ correspond to the $\mathbf{y}^1$ -flow of the i th edge, and row $2i$ and column $2i$ correspond to the $\mathbf{y}^2$ -flow of the i th edge.

After this rearrangement, $\mathbf{H}$ becomes a block diagonal matrix, where each edge i corresponds to a one 2×2 block given by

$$\mathbf{H}_{2i-1:2i} \stackrel{\text{def}}{=} \begin{pmatrix} \frac{1}{(\mathbf{y}_i^1)^2} + \frac{1}{(\mathbf{y}_i^r)^2} & \frac{1}{(\mathbf{y}_i^r)^2} \\ \frac{1}{(\mathbf{y}_i^r)^2} & \frac{1}{(\mathbf{y}_i^2)^2} + \frac{1}{(\mathbf{y}_i^r)^2} \end{pmatrix}. \quad (45)$$

The block diagonal structure of $\mathbf{H}$ then gives:

$$\mathbf{H}^{-1} = \text{DIAG} \left(\mathbf{H}_{1:2}^{-1}, \dots, \mathbf{H}_{2m-1:2m}^{-1} \right),$$

and for each edge i we have:

$$\mathbf{H}_{2i-1:2i}^{-1} = \frac{1}{\alpha_i} \left(\begin{pmatrix} (\mathbf{y}_i^1)^2 (\mathbf{y}_i^r)^2 & 0 \\ 0 & (\mathbf{y}_i^2)^2 (\mathbf{y}_i^r)^2 \end{pmatrix} + (\mathbf{y}_i^1)^2 (\mathbf{y}_i^2)^2 \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} \right), \quad (46)$$

where $\alpha_i = (\mathbf{y}_i^1)^2 + (\mathbf{y}_i^2)^2 + (\mathbf{y}_i^r)^2$. It means that

1. the type 1 edge has weight $(\mathbf{y}_i^1)^2 (\mathbf{y}_i^r)^2 / \alpha_i$,
2. the type 2 edge has weight $(\mathbf{y}_i^2)^2 (\mathbf{y}_i^r)^2 / \alpha_i$, and
3. the type 1 + 2 edge has weight $(\mathbf{y}_i^1)^2 (\mathbf{y}_i^2)^2 / \alpha_i$.

Note these three types of weights are symmetric. Let $w_{(i,1)}, w_{(i,2)}, w_{(i,1+2)}$ be the edge weights of type 1, type 2 and type 1 + 2 edge of the i th edge, respectively. These edge weights are given by the corresponding diagonals of $\mathbf{W}$. We set the values of $\mathbf{y}_i^1, \mathbf{y}_i^2, \mathbf{y}_i^r$ such that

$$\begin{aligned} (\mathbf{y}_i^1)^2 (\mathbf{y}_i^r)^2 / \alpha_i &= w_{(i,1)}, \\ (\mathbf{y}_i^2)^2 (\mathbf{y}_i^r)^2 / \alpha_i &= w_{(i,2)}, \\ (\mathbf{y}_i^1)^2 (\mathbf{y}_i^2)^2 / \alpha_i &= w_{(i,1+2)}. \end{aligned}$$

Solving the above equations gives:

$$\begin{aligned} (\mathbf{y}_i^1)^2 &= \frac{w_{(i,1)} w_{(i,1+2)}}{w_{(i,2)}} + w_{(i,1)} + w_{(i,1+2)}, \\ (\mathbf{y}_i^2)^2 &= \frac{w_{(i,2)} w_{(i,1+2)}}{w_{(i,1)}} + w_{(i,2)} + w_{(i,1+2)}, \\ (\mathbf{y}_i^r)^2 &= \frac{w_{(i,1)} w_{(i,2)}}{w_{(i,1+2)}} + w_{(i,1)} + w_{(i,2)}. \end{aligned} \quad (47)$$

The $\mathbf{y}^1, \mathbf{y}^2$ and $\mathbf{y}^r$ determine the edge-capacity vector $\mathbf{z}$, and they together with $\mathbf{N}$ determine the vertex-demand vectors $\mathbf{d}^1$ and $\mathbf{d}^2$.

Since we rearranged the rows and columns of $\mathbf{H}$, we need do the same rearrangement for $\mathbf{M}$. After rearranging row and columns of $\mathbf{M}$, $\mathbf{M}$ is of the form that, column $2i - 1$ and column $2i$ correspond to the $\mathbf{y}^1$ -flow and $\mathbf{y}^2$ -flow of the i th edge, respectively, and row $2i - 1$ and row $2i$ correspond to the $\mathbf{u}$ -coordinate and $\mathbf{v}$ -coordinate of vertex i , respectively. In $\mathbf{M}$, each edge has a 2×2 identity matrix for one endpoint, and a negative 2×2 identity matrix for the other endpoint.

By the above setting, we can check that

$$\mathbf{M} \mathbf{H}^{-1} \mathbf{M}^\top = \widehat{\mathbf{B}}^\top \mathbf{W} \widehat{\mathbf{B}}. \quad (48)$$

Note that $\mathbf{M}$ has dimension $2n \times 2m$, and $\widehat{\mathbf{B}}$ has dimension $3m \times 2n$.

Secondly, we make the right hand side vectors of the two linear systems (42) and (43) to be the same, that is,

$$-t\mathbf{M}\mathbf{H}^{-1}\mathbf{g} = \widehat{\mathbf{B}}^\top \mathbf{W}^{1/2} \mathbf{c}^{\mathbf{B}^{>0}}, \quad (49)$$

by choosing the cost vector $\mathbf{c}$ properly. The gradient $\mathbf{g}$ is given by:

$$\mathbf{g} = \mathbf{c} - \frac{1}{t} \sum_{i \in [m]} \frac{1}{\mathbf{y}_i^1} \mathbf{e}_{2i-1} + \frac{1}{\mathbf{y}_i^2} \mathbf{e}_{2i} - \frac{1}{\mathbf{y}_i^r} (\mathbf{e}_{2i-1} + \mathbf{e}_{2i}).$$

For simplicity, let

$$\mathbf{f}(\mathbf{y}) \stackrel{\text{def}}{=} t(\mathbf{c} - \mathbf{g}).$$

$\mathbf{f}(\mathbf{y})$ is a vector in $2m$ -dimension such that, the i th edge corresponds to the 2×1 sub-vector

$$\begin{pmatrix} \frac{1}{\mathbf{y}_i^1} - \frac{1}{\mathbf{y}_i^r} \\ \frac{1}{\mathbf{y}_i^2} - \frac{1}{\mathbf{y}_i^r} \end{pmatrix}.$$

Then we have

$$-t\mathbf{M}\mathbf{H}^{-1}\mathbf{g} = \mathbf{M}\mathbf{H}^{-1}(\mathbf{f} - t\mathbf{c}).$$

Recall the right hand side vector of Equation (49) is $\widehat{\mathbf{B}}^\top \mathbf{W}^{1/2} \mathbf{c}^{\mathbf{B}^{>0}}$. According to our construction in $\text{REDUCE } \mathcal{G}_{z,2} \text{ TO } \mathcal{MC}_2$ (Algorithm 1) and $\text{REDUCE } \mathcal{MC}_2 \text{ TO } \mathcal{MC}_2^{>0}$ (Algorithm 4), $\mathbf{c}^{\mathbf{B}^{>0}} = (\mathbf{c}^{\mathbf{A}}; \mathbf{0})$, where $\mathbf{c}^{\mathbf{A}}$ is the right hand side vector of the general linear system instance. Note all the nonzero entries of $\mathbf{c}^{\mathbf{B}^{>0}}$ correspond to the type 1 edges in the main constraint set $\mathcal{A}$, see line 16 of Algorithm 1.

Recall the structure of $\mathbf{M}$ in Equation (44), the edge-vertex incidence matrix $\mathbf{N}$ contains all the m edges of the underlying graphs, which are exactly the set of edges in $\mathbf{B}$ constructed in $\text{REDUCE } \mathcal{G}_{z,2} \text{ TO } \mathcal{MC}_2$ (Algorithm 1). Note for each edge of $\mathbf{N}$, its large-weight edge can be any of the corresponding type 1, type 2 and type 1 + 2 edges. By Claim 9.5, we can partition the underlying graph edges into $E_1 \cup E_2 \cup E_{1+2}$ such that E_k contains all the underlying edges whose large-weight edges are the corresponding type k edges, $k \in \{1, 2, 1+2\}$. Note $\mathbf{c}_{(i,1)}^{\mathbf{B}^{>0}} \neq 0$ only if edge i is in E_1 .

We define a vector $\mathbf{c}' \in \mathbb{R}^{2m}$ such that the entry of $\mathbf{c}'$ which corresponds to the $\mathbf{y}^1$ -flow of the i th edge equals to $w_{(i,1)}^{1/2} \mathbf{c}_{(i,1)}^{\mathbf{B}^{>0}}$. This definition gives:

$$\widehat{\mathbf{B}}^\top \mathbf{W}^{1/2} \mathbf{c}^{\mathbf{B}^{>0}} = \mathbf{M}\mathbf{c}'. \quad (50)$$

By Equation (49),

$$\mathbf{M}\mathbf{H}^{-1}(\mathbf{f} - t\mathbf{c}) = \mathbf{M}\mathbf{c}'.$$

Rearranging it,

$$\mathbf{M}\mathbf{H}^{-1}(\mathbf{f} - t\mathbf{c} - \mathbf{H}\mathbf{c}') = \mathbf{0}.$$

According to $\mathcal{MC}_2 \text{ GADGET}$ (Algorithm 2), each gadget has a cycle containing both the 2 edges in E_{1+2} : $(t+3, t+4, t+2, t+5, t+6, t+1, t+3)$. Let E' be the set of all the edges in such gadget cycles. Let $\mathbf{p} \in \{0, 1\}^m$ such that each entry corresponding to an edge in E' has value 1. Let $\mathbf{q} \stackrel{\text{def}}{=} (\mathbf{p}; \mathbf{p})$. According to $\text{REDUCE } \mathcal{G}_{z,2} \text{ TO } \mathcal{MC}_2$ (Algorithm 1), every gadget edge i has $\mathbf{c}'_i = 0$. Thus, we are free to choose the sign of a column of $\mathbf{N}^\top$ which corresponds to a gadget edge, which

does not change Equations (48) and (50).⁷ Without loss of generality, we assume that after proper sign flipping, $\mathbf{M}\mathbf{q} = \mathbf{0}$. It gives that

$$\mathbf{M}\mathbf{H}^{-1}(\mathbf{f} - t\mathbf{c} - \mathbf{H}\mathbf{c}' + \mathbf{H}\mathbf{q}) = \mathbf{0}.$$

We set $t\mathbf{c} = \mathbf{f} - \mathbf{H}\mathbf{c}' + \lambda\mathbf{H}\mathbf{q}$, where $\lambda > 0$ is a sufficiently large constant to be determined later.

For each edge i , let $\mathcal{I}_i$ be an indicator whose value is 1 if edge i in the set E' and 0 otherwise. For the i th edge, its corresponding 2×1 sub-vector in $t\mathbf{c}$ is

$$t \begin{pmatrix} \mathbf{c}_i \\ \mathbf{c}_{m+i} \end{pmatrix} = \begin{pmatrix} \frac{1}{\mathbf{y}_i^1} - \frac{1}{\mathbf{y}_i^r} \\ \frac{1}{\mathbf{y}_i^2} - \frac{1}{\mathbf{y}_i^r} \end{pmatrix} - \mathbf{c}'_i \begin{pmatrix} \frac{1}{(\mathbf{y}_i^1)^2} + \frac{1}{(\mathbf{y}_i^r)^2} \\ \frac{1}{(\mathbf{y}_i^2)^2} \end{pmatrix} + \mathcal{I}_i \lambda \mathbf{s}^i,$$

where

$$\mathbf{s}^i \stackrel{\text{def}}{=} \begin{pmatrix} \frac{1}{(\mathbf{y}_i^1)^2} + \frac{2}{(\mathbf{y}_i^r)^2} \\ \frac{1}{(\mathbf{y}_i^2)^2} + \frac{2}{(\mathbf{y}_i^r)^2} \end{pmatrix}.$$

Note that $\mathbf{s}^i \geq \mathbf{0}$ (entry-wise). Plugging the solution of $\mathbf{y}$ from Equation (47) gives:

$$t \begin{pmatrix} \mathbf{c}_i \\ \mathbf{c}_{m+i} \end{pmatrix} = \begin{pmatrix} \sqrt{\frac{w(i,2)}{\gamma}} - \sqrt{\frac{w(i,1+2)}{\gamma}} \\ \sqrt{\frac{w(i,1)}{\gamma}} - \sqrt{\frac{w(i,1+2)}{\gamma}} \end{pmatrix} - \mathbf{c}'_i \begin{pmatrix} \frac{w(i,2)+w(i,1+2)}{\gamma} \\ \frac{w(i,1+2)}{\gamma} \end{pmatrix} + \mathcal{I}_i \lambda \mathbf{s}^i,$$

where $\gamma \stackrel{\text{def}}{=} w(i,1)w(i,1+2) + w(i,1)w(i,2) + w(i,2)w(i,1+2)$.

By Claim 9.5, each edge of the underlying graph has exactly one type of edges with large weight. We deal with each of these cases separately under the condition that two of the edge's weights are much smaller than that of the third one. We will also denote this ratio using $\epsilon > 0$:

1. If $w(i,2) = w(i,1+2) = \epsilon w(i,1)$, then

$$t \begin{pmatrix} \mathbf{c}_i \\ \mathbf{c}_{m+i} \end{pmatrix} \geq -\mathbf{c}'_i \begin{pmatrix} \frac{2\epsilon}{\mu} \\ \frac{\epsilon}{\mu} \end{pmatrix} + \begin{pmatrix} 0 \\ \frac{1-\sqrt{\epsilon}}{\sqrt{\mu}} \end{pmatrix},$$

where $\mu = (2\epsilon + \epsilon^2)w(i,1)$. This corresponds to an edge of E_1 . If $\mathbf{c}'_i \leq 0$, then $t\mathbf{c}_i, t\mathbf{c}_{m+i} \geq 0$. If $\mathbf{c}'_i > 0$, then $\mathbf{c}'_i = w_{(i,1)}^{1/2} \mathbf{c}_{(i,1)}^{B>0}$ where $\mathbf{c}_{(i,1)}^{B>0}$ comes from an entry of $\mathbf{c}^A$. We multiply -1 on both sides of the corresponding equation in the general linear system instance $\mathbf{A}\mathbf{x} = \mathbf{c}^A$ which creates this edge i , and will get $\mathbf{c}'_i < 0$ instead.⁸

2. If $w(i,1) = w(i,1+2) = \epsilon w(i,2)$, then

$$t \begin{pmatrix} \mathbf{c}_i \\ \mathbf{c}_{m+i} \end{pmatrix} \geq -\mathbf{c}'_i \begin{pmatrix} \frac{1+\epsilon}{\mu} \\ \frac{\epsilon}{\mu} \end{pmatrix} + \begin{pmatrix} \frac{1-\sqrt{\epsilon}}{\sqrt{\mu}} \\ 0 \end{pmatrix},$$

where $\mu = (2\epsilon + \epsilon^2)w(i,2)$. This corresponds to an edge of E_2 , in which $\mathbf{c}'_i = 0$. We can check that $t\mathbf{c}_i, t\mathbf{c}_{m+i} \geq 0$.

⁷Note the demand vectors $\mathbf{d}^1$ and $\mathbf{d}^2$ change after we change $\mathbf{N}$.

⁸We first flip signs of equations of $\mathbf{A}\mathbf{x} = \mathbf{c}^A$ so that $\mathbf{c}^A \leq \mathbf{0}$ (entry-wise), which determines $\mathbf{N}, \mathbf{M}$ and $\mathbf{c}'$. We then flip signs of columns of $\mathbf{N}$, which we guarantee does not violate Equation (48) and (50).

3. If $w_{(i,1)} = w_{(i,2)} = \epsilon w_{(i,1+2)}$, then

$$t \begin{pmatrix} \mathbf{c}_i \\ \mathbf{c}_{m+i} \end{pmatrix} = -\mathbf{c}'_i \begin{pmatrix} \frac{1+\epsilon}{\mu} \\ \frac{1}{\mu} \end{pmatrix} + \begin{pmatrix} \frac{\sqrt{\epsilon}-1}{\sqrt{\mu}} \\ \frac{\sqrt{\epsilon}-1}{\sqrt{\mu}} \end{pmatrix} + \mathcal{I}_i \lambda \mathbf{s}^i,$$

where $\mu = (2\epsilon + \epsilon^2)w_{(i,1+2)}$. This corresponds to an edge of E_{1+2} , in which $\mathbf{c}'_i = 0$ and $\mathcal{I}_i = 1$. We choose λ such that, for every edge i in this case, $t\mathbf{c}_i, t\mathbf{c}_{m+i} \geq 0$.

□

Note in IPMs, the values of t is a sequence of increasing values. Whenever $t \geq m/\epsilon'$, the optimality gap is smaller than ϵ' . This ensures that the cost vector $\mathbf{c}$'s entries are in a reasonable range.

9.3 Isotropic Total Variation Minimization

For the isotropic total variation problem, we follow the formulations given in [CMMP13a], namely given a graph $G = (V, E, \mathbf{w})$, we partition the sets into $S_1 \dots S_k$, and minimize the objective

$$\|\mathbf{y} - \mathbf{s}\|_2^2 + \sum_{1 \leq i \leq k} \mathbf{w}_i \sqrt{\sum_{(u,v) \in S_i} \mathbf{w}_{uv} (\mathbf{y}_u - \mathbf{y}_v)^2}.$$

where $\mathbf{s}$ is the input signal vector. Let $\mathbf{N}$ be the edge-vertex incidence matrix of the graph G , then the dual of the grouped flow problem is:

$$\begin{aligned} & \max \quad \mathbf{s}^\top \mathbf{N}^\top \mathbf{f} \\ \text{subject to:} \quad & \sum_{e \in S_i} \mathbf{w}_e^{-1} \mathbf{f}_e^2 \leq \mathbf{w}_i^{-1} \quad \forall 1 \leq i \leq k \end{aligned}$$

Both the primal and dual problems can be formulated into log barriers via second order cones [GY04]. In the case with minimizing vertex labels, we introduce a variable $\mathbf{y}_i$ for each cluster i , and instead minimize $\sum_i \mathbf{w}_i \mathbf{y}_i$ subject to the constraint

$$\mathbf{y}_i^2 \geq \sum_{(u,v) \in S_i} \mathbf{w}_e^{-1} (\mathbf{y}_u - \mathbf{y}_v)^2,$$

which in turn leads to the logarithmic barrier function

$$\log \left(\mathbf{y}_i^2 - \sum_{(u,v) \in S_i} \mathbf{w}_e^{-1} (\mathbf{y}_u - \mathbf{y}_v)^2 \right),$$

while in the flow case the barrier functions for each set of edges S_i is

$$\log \left(\mathbf{w}_i^{-1} - \sum_{e \in S_i} \mathbf{w}_e^{-1} \mathbf{f}_e^2 \right).$$

Due to the connections with the multicommodity flow problems, we will only state the connections through the flow version here. Recall that linear systems in $\mathcal{MC}_2$ have the form:

$$\mathbf{L}^1 \otimes \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} + \mathbf{L}^2 \otimes \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} + \mathbf{L}^{1+2} \otimes \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix},$$

where E_1, E_2, E_{1+2} denotes the edges in $\mathbf{L}^1, \mathbf{L}^2, \mathbf{L}^{1+2}$, respectively. Edges in $E_1 \cup E_2$ correspond to individual edges, so it remains to show that edges in E_{1+2} can correspond to the Hessian matrices of clusters containing pairs of edges. In a cluster with two edges $\mathbf{f}_1$ and $\mathbf{f}_2$, the gradient of the function

$$\log(\alpha - \mathbf{w}_1^{-1}\mathbf{f}_1^2 - \mathbf{w}_2^{-1}\mathbf{f}_2^2)$$

is

$$\frac{2}{r} \begin{pmatrix} -\mathbf{w}_1^{-1}\mathbf{f}_1 \\ -\mathbf{w}_2^{-1}\mathbf{f}_2 \end{pmatrix},$$

where we treat

$$r \stackrel{\text{def}}{=} \alpha - \mathbf{w}_1^{-1}\mathbf{f}_1^2 - \mathbf{w}_2^{-1}\mathbf{f}_2^2$$

as a new free variable because we are free to choose α in the IPM instance. Differentiating again (via the product rule) then gives the Hessian matrix:

$$\frac{-4}{r^2} \begin{pmatrix} \mathbf{w}_1^{-2}\mathbf{f}_1^2 & \mathbf{w}_1^{-1}\mathbf{w}_2^{-2}\mathbf{f}_1\mathbf{f}_2 \\ \mathbf{w}_1^{-1}\mathbf{w}_2^{-2}\mathbf{f}_1\mathbf{f}_2 & \mathbf{w}_2^{-2}\mathbf{f}_2^2 \end{pmatrix} + \frac{-2}{r} \begin{pmatrix} \mathbf{w}_1^{-1} & 0 \\ 0 & \mathbf{w}_2^{-1} \end{pmatrix}.$$

As we are free to choose $\mathbf{w}_1$ and $\mathbf{w}_2$, the second matrix can be any diagonal matrix. Then the freedom in choosing $\mathbf{f}_1$ and $\mathbf{f}_2$ then means the first matrix can be any rank 1 object. Therefore, this Hessian is equivalent to the block Hessian for two commodity flows generated in Equation (45) (up to a change of sign, since we are dealing with a maximization problem here), and its inverse as given in Equation (46) provides the characterization from Definition 1.5. We can also check more directly that the $1+2$ edge can be represented as one of these Hessian inverse blocks.

Claim 9.6. *For each edge (i, j) in E_{1+2} , there exist an edge-vertex incidence matrix $\mathbf{N}$, a diagonal matrix $\mathbf{W}$ and a vector $\mathbf{r}$ such that $\mathbf{W} \succcurlyeq \mathbf{r}\mathbf{r}^\top$ and*

$$\mathbf{L}_{ij}^{1+2} \otimes \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} = \mathbf{N}^\top (\mathbf{W} - \mathbf{r}\mathbf{r}^\top) \mathbf{N}.$$

Proof. For simplicity, we remove all zero rows and columns of $\mathbf{L}_{ij}^{1+2}$ so that the 2-commodity matrix for edge (i, j) only has dimension 4×4 . We pick

$$\mathbf{N} = \begin{pmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \end{pmatrix}, \mathbf{W} = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} \text{ and } \mathbf{r} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

We check the PSD condition,

$$\mathbf{W} - \mathbf{r}\mathbf{r}^\top = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} \succcurlyeq \mathbf{0}.$$

Then, we check that this decomposition equals to the 2-commodity matrix

$$\begin{aligned} \mathbf{N}^\top (\mathbf{W} - \mathbf{r}\mathbf{r}^\top) \mathbf{N} &= \begin{pmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} \begin{pmatrix} 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{pmatrix} \\ &= \begin{pmatrix} 1 \\ -1 \\ -1 \\ 1 \end{pmatrix} \begin{pmatrix} 1 & -1 & -1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & -1 & -1 & 1 \\ -1 & 1 & 1 & -1 \\ 1 & -1 & -1 & 1 \\ -1 & 1 & 1 & -1 \end{pmatrix} = \mathbf{L}_{ij}^{1+2} \otimes \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}. \end{aligned}$$

□

Proof of Lemma 3.9. Since the linear system related to the isotropic total variation minimization matrix is the same as the linear system for 2-commodity, all complexity parameters of these two linear systems are the same. $\square$

10 A Natural Decision Problem: Vector in the Image of a Matrix

When we are given a linear system $\mathbf{A}\mathbf{x} = \mathbf{c}$, a natural question is whether the system in fact has a solution. This is true whenever $\mathbf{c}$ lies in the image of $\mathbf{A}$, or equivalently, when $\Pi_{\mathbf{A}}\mathbf{c} = \mathbf{c}$.

We consider a slightly relaxed version of the question of whether $\mathbf{c}$ lies in the image of $\mathbf{A}$. We define the Linear System Decision Problem, abbreviated LSD, as follows:

Definition 10.1 (Linear System Decision Problem, LSD). Given a linear system $(\mathbf{A}, \mathbf{c})$ and an approximation parameter $\epsilon > 0$, we refer to the LSD problem for the triple $(\mathbf{A}, \mathbf{c}, \epsilon)$ as the problem of outputting

1. YES if there exists $\mathbf{x}$ s.t. $\mathbf{A}\mathbf{x} = \mathbf{c}$.
2. NO if for all $\mathbf{x}$, we have $\|\mathbf{A}\mathbf{x} - \mathbf{c}\| > \epsilon \|\mathbf{c}\|$.
3. Any output is acceptable if neither of the above two cases hold.

Algorithm 12 Reduce LSD to LSA.

- 1: Find $\mathbf{x}$ which is a solution to LSA instance $(\mathbf{A}, \mathbf{c}, \epsilon)$.
 - 2: **if** $\|\mathbf{A}\mathbf{x} - \mathbf{c}\| \leq \epsilon \|\mathbf{c}\|$ **then**
 - 3: **return** YES.
 - 4: **else**
 - 5: **return** NO.
 - 6: **end if**
-

Lemma 10.2. *Algorithm 12 solves LSD instance $(\mathbf{A}, \mathbf{c}, \epsilon)$.*

Proof. First, suppose we are in Case 1, i.e. there exists $\mathbf{x}$ s.t. $\mathbf{A}\mathbf{x} = \mathbf{c}$, or equivalently $\Pi_{\mathbf{A}}\mathbf{c} = \mathbf{c}$. From this and $\mathbf{x}$ being an ϵ -approximate solution to $(\mathbf{A}, \mathbf{c})$, it follows that

$$\|\mathbf{A}\mathbf{x} - \mathbf{c}\| = \|\mathbf{A}\mathbf{x} - \Pi_{\mathbf{A}}\mathbf{c}\| \leq \epsilon \|\Pi_{\mathbf{A}}\mathbf{c}\| = \epsilon \|\mathbf{c}\|.$$

So, the algorithm will return YES.

Second, suppose we are in Case 2, then for all $\mathbf{x}$ we have $\|\mathbf{A}\mathbf{x} - \mathbf{c}\| > \epsilon \|\mathbf{c}\|$, so the algorithm must return NO. $\square$

10.1 Why Not Solve the Exact Decision Problem?

Earlier in this section, we showed how to solve an approximate linear system decision problem, LSD by a reduction to an LSA problem. However, given matrix $\mathbf{A}$ and a vector $\mathbf{c}$ with polynomially bounded integer entries and condition number, it is natural to ask whether solving LSA to reasonably high accuracy could in fact let us solve exactly the decision problem of whether there exists an $\mathbf{x}$ s.t. $\mathbf{A}\mathbf{x} = \mathbf{c}$. Note that this is equivalent to $(\mathbf{I} - \Pi_{\mathbf{A}})\mathbf{c} = \mathbf{0}$. Lemma 4.8 implies that when $\mathbf{A}$ and $\mathbf{c}$ have integer valued entries, either $\Pi_{\mathbf{A}}\mathbf{c} = \mathbf{0}$ or $\|\Pi_{\mathbf{A}}\mathbf{c}\| \geq 1/\sigma_{\max}(\mathbf{A})$. If a similar separation result was true for $\|(\mathbf{I} - \Pi_{\mathbf{A}})\mathbf{c}\|$, we could use this to solve the exact decision problem.

However, as we will show below, there exists a matrix and vector pair $\mathbf{A} \in \mathbb{Z}^{(n+1) \times n}$ and a vector $\mathbf{c} \in \mathbb{Z}^{n+1}$ whose entries are polynomially bounded (in fact bounded between -1 and 2), and with polynomially bounded $\sigma_{\min}(\mathbf{A})$ and $\sigma_{\max}(\mathbf{A})$ s.t. $\|(\mathbf{I} - \Pi_{\mathbf{A}}) \mathbf{c}\|$ is exponentially small in n . This suggests that solving the exact decision problem of whether $(\mathbf{I} - \Pi_{\mathbf{A}}) \mathbf{c} = \mathbf{0}$ would require solving LSA to exponentially high precision, or adopting some radically different approach.

We now outline the construction of $\mathbf{A}$ and $\mathbf{c}$. Each column i of $\mathbf{A}$ has two nonzero entries: $\mathbf{A}_{i,i} = 2$ and $\mathbf{A}_{i,i+1} = -1$. The rank of $\mathbf{A}$ is n . Let $\mathbf{r}$ be the only unit vector of $\text{null}(\mathbf{A}^\top)$. Then,

$$\mathbf{I} - \Pi_{\mathbf{A}} = \mathbf{r} \mathbf{r}^\top.$$

We compute $\mathbf{r}$. Since $\mathbf{A}^\top \mathbf{r} = \mathbf{0}$, we have

$$\mathbf{r}_{i+1} = 2\mathbf{r}_i, \forall 1 \leq i \leq n.$$

Since $\|\mathbf{r}\|_2 = 1$, we have $\mathbf{r}_1 = \frac{1}{2^{n+2}-1}$. Let $\mathbf{c}$ be a vector whose first entry is 1 and all other entries are 0. Then,

$$\|(\mathbf{I} - \Pi_{\mathbf{A}}) \mathbf{c}\|_2 = \left| \mathbf{r}^\top \mathbf{c} \right|_2 = \frac{1}{2^{n+2}-1},$$

which is exponentially small.

We show that the nonzero singular values of $\mathbf{A}$ are bounded by constants. Note that $\mathbf{A}^\top \mathbf{A}$ is a tridiagonal matrix such that,

1. all entries on the main diagonal are 5,
2. all entries on the first diagonal below the main diagonal are -2, and
3. all entries on the first diagonal above the main diagonal are -2.

$\mathbf{A}^\top \mathbf{A}$ has full rank. $\mathbf{A}^\top \mathbf{A}$ can be written as $\mathbf{L} + \mathbf{D}$, where $\mathbf{L}$ is the graph Laplacian matrix for an undirected path with identical edge weight 2, and $\mathbf{D}$ is a diagonal matrix whose first and last diagonal entries are 3 and all other diagonal entries are 1. Thus,

$$\mathbf{L} + \mathbf{I} \preceq \mathbf{A}^\top \mathbf{A} \preceq \mathbf{L} + 3\mathbf{I}.$$

Since $\lambda_{\max}(\mathbf{L}) = 4$, we have for all eigenvalues of $\lambda(\mathbf{A}^\top \mathbf{A})$ that

$$1 \leq \lambda(\mathbf{A}^\top \mathbf{A}) \leq 11,$$

that is,

$$1 \leq \sigma_{\min}(\mathbf{A}) \leq \sigma_{\max}(\mathbf{A}) \leq \sqrt{11}.$$

Acknowledgements

We thank Richard Peng and Daniel Spielman for helpful comments and discussions and we thank Richard Peng for suggesting the example in Section 10.1 which demonstrates that the projection of a vector onto the null space of a matrix can be exponentially small despite well-conditionedness assumptions on the matrix-vector pair.

A Linear Algebra Background

Proof of Lemma 2.4. First observe

$$\begin{aligned}
\left\| \mathbf{A}^\top \mathbf{A} \mathbf{x} - \mathbf{A}^\top \mathbf{c} \right\|_{(\mathbf{A}^\top \mathbf{A})^\dagger}^2 &= (\mathbf{A}^\top \mathbf{A} \mathbf{x} - \mathbf{A}^\top \mathbf{c})^\top (\mathbf{A}^\top \mathbf{A})^\dagger (\mathbf{A}^\top \mathbf{A} \mathbf{x} - \mathbf{A}^\top \mathbf{c}) \\
&= (\mathbf{A} \mathbf{x} - \mathbf{c})^\top \mathbf{\Pi}_A (\mathbf{A} \mathbf{x} - \mathbf{c}) \\
&= (\mathbf{\Pi}_A \mathbf{A} \mathbf{x} - \mathbf{\Pi}_A \mathbf{c})^\top (\mathbf{\Pi}_A \mathbf{A} \mathbf{x} - \mathbf{\Pi}_A \mathbf{c}) \\
&= \|\mathbf{A} \mathbf{x} - \mathbf{\Pi}_A \mathbf{c}\|_2^2.
\end{aligned}$$

Taking $\mathbf{x} = \mathbf{0}$ it follows that $\|\mathbf{\Pi}_A \mathbf{c}\| = \|\mathbf{A}^\top \mathbf{c}\|_{(\mathbf{A}^\top \mathbf{A})^\dagger}$, and combining our observations immediately gives the second part of the Fact.

A similar argument gives claims for $\|\mathbf{x} - \mathbf{x}^*\|_{\mathbf{A}^\top \mathbf{A}}$.

$$\begin{aligned}
\|\mathbf{x} - \mathbf{x}^*\|_{\mathbf{A}^\top \mathbf{A}}^2 &= (\mathbf{x} - \mathbf{x}^*)^\top \mathbf{A}^\top \mathbf{A} (\mathbf{x} - \mathbf{x}^*) \\
&= \mathbf{x}^\top \mathbf{A}^\top \mathbf{A} \mathbf{x} - 2\mathbf{x}^\top \mathbf{A}^\top \mathbf{A} \mathbf{x}^* + (\mathbf{x}^*)^\top \mathbf{A}^\top \mathbf{A} \mathbf{x}^* \\
&= \mathbf{x}^\top \mathbf{A}^\top \mathbf{A} \mathbf{x} - 2\mathbf{x}^\top \mathbf{A}^\top \mathbf{\Pi}_A \mathbf{c} + \mathbf{c}^\top \mathbf{\Pi}_A^\top \mathbf{\Pi}_A \mathbf{c} \\
&= \|\mathbf{A} \mathbf{x} - \mathbf{\Pi}_A \mathbf{c}\|_2^2.
\end{aligned}$$

The third equality uses the fact that $\mathbf{A} \mathbf{x}^* = \mathbf{\Pi}_A \mathbf{c}$. This completes the proof. $\square$

B Using Complexity Parameters

In this section, we bound the norms, eigenvalues of matrix $\mathbf{A} \in \mathbb{R}^{n \times m}$ and the norms of vector $\mathbf{c} \in \mathbb{R}^n$, using the complexity parameters defined in Definition 2.5:

1. $s : \text{nnz}(\mathbf{A})$,
2. $U : \max \left(\|\mathbf{A}\|_{\max}, \|\mathbf{c}\|_{\max}, \frac{1}{\min_+(\mathbf{A})}, \frac{1}{\min_+(\mathbf{c})} \right)$,
3. $K : \kappa(\mathbf{A})$.

Note $n, m \leq s$.

Claim B.1. $\|\mathbf{A}\|_\infty \leq sU$.

Claim B.2. $\|\mathbf{c}\|_2 \leq \sqrt{s}U$.

Proof. This follows the relations of norms,

$$\|\mathbf{c}\|_2 \leq \sqrt{n} \|\mathbf{c}\|_\infty \leq \sqrt{s}U.$$

$\square$

Claim B.3. $\frac{1}{sU^2} \leq \lambda_{\max}(\mathbf{A}^\top \mathbf{A}) \leq sU^2$ and $\lambda_{\min}(\mathbf{A}^\top \mathbf{A}) \geq \frac{1}{sK^2U^2}$.

Proof. By the definition of Frobenius norm,

$$\|\mathbf{A}\|_F^2 = \sum_{i,j} \mathbf{A}_{ij}^2 = \sum_{i=1}^{\min\{m,n\}} \lambda_i(\mathbf{A}^\top \mathbf{A}).$$

Thus,

$$\lambda_{\max}(\mathbf{A}^\top \mathbf{A}) \leq \|\mathbf{A}\|_F^2 \leq sU^2,$$

and

$$\lambda_{\max}(\mathbf{A}^\top \mathbf{A}) \geq \frac{\|\mathbf{A}\|_F^2}{\min\{m, n\}} \geq \frac{1}{sU^2}.$$

By the definition of $\kappa(\mathbf{A}^\top \mathbf{A})$,

$$\lambda_{\min}(\mathbf{A}^\top \mathbf{A}) = \frac{\lambda_{\max}(\mathbf{A}^\top \mathbf{A})}{\kappa(\mathbf{A}^\top \mathbf{A})} \geq \frac{1}{sK^2U^2}.$$

This completes the proof. □

References

- [ABFM14] Boris Alexeev, Afonso S Bandeira, Matthew Fickus, and Dustin G Mixon. Phase retrieval with polarization. *SIAM Journal on Imaging Sciences*, 7(1):35–66, 2014.
- [ANKKS⁺12] Mica Arie-Nachimson, Shahar Z Kovalsky, Ira Kemelmacher-Shlizerman, Amit Singer, and Ronen Basri. Global motion estimation from point matches. In *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2012 Second International Conference on*, pages 81–88. IEEE, 2012.
- [BCPT05] Erik G Boman, Doron Chen, Ojas Parekh, and Sivan Toledo. On factor width and symmetric h-matrices. *Linear algebra and its applications*, 405:239–248, 2005.
- [BV04] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004. Available at https://web.stanford.edu/~boyd/cvxbook/bv_cvxbook.pdf.
- [CFM⁺14] Michael B. Cohen, Brittany Terese Fasy, Gary L. Miller, Amir Nayyeri, Richard Peng, and Noel Walkington. Solving 1-laplacians in nearly linear time: Collapsing and expanding a topological ball. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, pages 204–216, 2014.
- [CKM⁺14] Michael B. Cohen, Rasmus Kyng, Gary L. Miller, Jakub W. Pachocki, Richard Peng, Anup B. Rao, and Shen Chen Xu. Solving sdd linear systems in nearly $m \log 1/2n$ time. In *Proceedings of the Forty-sixth Annual ACM Symposium on Theory of Computing, STOC '14*, pages 343–352, New York, NY, USA, 2014. ACM.
- [CLM⁺16] Michael B Cohen, Yin Tat Lee, Gary Miller, Jakub Pachocki, and Aaron Sidford. Geometric median in nearly linear time. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing*, pages 9–21. ACM, 2016. Available at: <https://arxiv.org/abs/1606.05225>.
- [CMMP13a] Hui Han Chin, Aleksander Madry, Gary L. Miller, and Richard Peng. Runtime guarantees for regression problems. In *Proceedings of the 4th conference on Innovations in Theoretical Computer Science, ITCS '13*, pages 269–282, New York, NY, USA, 2013. ACM.

- [CMMP13b] Hui Han Chin, Aleksander Madry, Gary L Miller, and Richard Peng. Runtime guarantees for regression problems. In *Proceedings of the 4th conference on Innovations in Theoretical Computer Science*, pages 269–282. ACM, 2013.
- [CMSV17] Michael B. Cohen, Aleksander Madry, Piotr Sankowski, and Adrian Vladu. Negative-weight shortest paths and unit capacity minimum cost flow in $\tilde{O}(m^{10/7} \log w)$ time: (extended abstract). In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '17, pages 752–771, Philadelphia, PA, USA, 2017. Society for Industrial and Applied Mathematics.
- [CS05] Tony Chan and Jianhong Shen. *Image Processing and Analysis: Variational, PDE, Wavelet, and Stochastic Methods*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2005.
- [DS07] Samuel I. Daitch and Daniel A. Spielman. Support-graph preconditioners for 2-dimensional trusses. *CoRR*, abs/cs/0703119, 2007.
- [DS08] Samuel I. Daitch and Daniel A. Spielman. Faster approximate lossy generalized flow via interior point algorithms. In *Proceedings of the 40th annual ACM symposium on Theory of computing*, STOC '08, pages 451–460, New York, NY, USA, 2008. ACM. Available at <http://arxiv.org/abs/0803.0988>.
- [Fle00] Lisa K. Fleischer. Approximating fractional multicommodity flow independent of the number of commodities. *SIAM Journal on Discrete Mathematics*, 13:505–520, 2000.
- [GK98] Naveen Garg and Jochen Könemann. Faster and simpler algorithms for multicommodity flow and other fractional packing problems. In *In Proceedings of the 39th Annual Symposium on Foundations of Computer Science*, pages 300–309, 1998.
- [GY04] Donald Goldfarb and Wotao Yin. Second-order cone programming methods for total variation-based image restoration. *SIAM J. Sci. Comput.*, 27:622–645, 2004.
- [GY05] Donald Goldfarb and Wotao Yin. Second-order cone programming methods for total variation-based image restoration. *SIAM Journal on Scientific Computing*, 27(2):622–645, 2005.
- [KLOS14] Jonathan A. Kelner, Yin Tat Lee, Lorenzo Orecchia, and Aaron Sidford. An almost-linear-time algorithm for approximate max flow in undirected graphs, and its multicommodity generalizations. In *Proceedings of the Twenty-fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '14, pages 217–226, Philadelphia, PA, USA, 2014. Society for Industrial and Applied Mathematics.
- [KLP⁺16] Rasmus Kyng, Yin Tat Lee, Richard Peng, Sushant Sachdeva, and Daniel A. Spielman. Sparsified cholesky and multigrid solvers for connection laplacians. In *Proceedings of the Forty-eighth Annual ACM Symposium on Theory of Computing*, STOC '16, pages 842–850, New York, NY, USA, 2016. ACM.
- [KMP12] Jonathan A. Kelner, Gary L. Miller, and Richard Peng. Faster approximate multicommodity flow using quadratically coupled flows. In *Proceedings of the 44th symposium on Theory of Computing*, STOC '12, pages 1–18, New York, NY, USA, 2012. ACM. Available at <http://arxiv.org/abs/1202.3367>.

- [KMT11] Ioannis Koutis, Gary L Miller, and David Tolliver. Combinatorial preconditioners and multilevel solvers for problems in computer vision and image processing. *Computer Vision and Image Understanding*, 115(12):1638–1646, 2011.
- [LMP⁺91] Tom Leighton, Fillia Makedon, Serge Plotkin, Clifford Stein, Eva Tardos, and Spyros Tragoudas. Fast approximation algorithms for multicommodity flow problems. In *JOURNAL OF COMPUTER AND SYSTEM SCIENCES*, pages 487–496, 1991.
- [LS14] Yin Tat Lee and Aaron Sidford. Path finding methods for linear programming: Solving linear programs in $\tilde{O}(\sqrt{\text{rank}})$ iterations and faster algorithms for maximum flow. In *Foundations of Computer Science (FOCS), 2014 IEEE 55th Annual Symposium on*, pages 424–433. IEEE, 2014. Available at <http://arxiv.org/abs/1312.6677> and <http://arxiv.org/abs/1312.6713>.
- [LS15] Yin Tat Lee and Aaron Sidford. Efficient inverse maintenance and faster algorithms for linear programming. In *Foundations of Computer Science (FOCS), 2015 IEEE 56th Annual Symposium on*, pages 230–249. IEEE, 2015. Available at: <https://arxiv.org/abs/1503.01752>.
- [Mad10] Aleksander Madry. Faster approximation schemes for fractional multicommodity flow problems via dynamic graph algorithms. In *STOC '10: Proceedings of the 42nd ACM symposium on Theory of computing*, pages 121–130, New York, NY, USA, 2010. ACM.
- [Mad13] Aleksander Madry. Navigating central path with electrical flows: From flows to matchings, and back. In *Proceedings of the 2013 IEEE 54th Annual Symposium on Foundations of Computer Science, FOCS '13*, pages 253–262, Washington, DC, USA, 2013. IEEE Computer Society.
- [Mad16] Aleksander Madry. Computing maximum flow with augmenting electrical flows. In *Proceedings of the 2016 IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS '16*, Washington, DC, USA, 2016. IEEE Computer Society.
- [MTW14] Stefano Marchesini, Yu-Chao Tu, and Hau-tieng Wu. Alternating projection, psychographic imaging and phase synchronization. *arXiv preprint arXiv:1402.0550*, 2014.
- [Nem] Arkadi Nemirovski. Interior point polynomial time methods in convex programming.
- [OSB15] Onur Ozyesil, Amit Singer, and Ronen Basri. Stable camera motion estimation using convex programming. *SIAM Journal on Imaging Sciences*, 8(2):1220–1262, 2015.
- [Pen16] Richard Peng. Approximate undirected maximum flows in $o(\text{mpolylog}(n))$ time. In *Proceedings of the Twenty-seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '16*, pages 1862–1867, Philadelphia, PA, USA, 2016. Society for Industrial and Applied Mathematics.
- [Saa03] Y. Saad. *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2nd edition, 2003. Available at <http://www-users.cs.umn.edu/~saad/toc.pdf>.
- [She13] Jonah Sherman. Nearly maximum flows in nearly linear time. In *Proceedings of the 2013 IEEE 54th Annual Symposium on Foundations of Computer Science, FOCS '13*, pages 263–269, Washington, DC, USA, 2013. IEEE Computer Society.

- [Spi07] Daniel A Spielman. Spectral graph theory and its applications. In *Foundations of Computer Science, 2007. FOCS'07. 48th Annual IEEE Symposium on*, pages 29–38. IEEE, 2007.
- [Spi16] Daniel A. Spielman. Nsf award 1562041: Generalized algebraic graph theory: Algorithms and analysis. *ALGORITHMIC FOUNDATIONS*, 2016.
- [SS11] Amit Singer and Yoel Shkolnisky. Three-dimensional structure determination from common lines in cryo-em by eigenvectors and semidefinite programming. *SIAM journal on imaging sciences*, 4(2):543–572, 2011.
- [SS12] Yoel Shkolnisky and Amit Singer. Viewing direction estimation in cryo-em using synchronization. *SIAM journal on imaging sciences*, 5(3):1088–1110, 2012.
- [ST08] Gil Shklarski and Sivan Toledo. Rigidity in finite-element matrices: Sufficient conditions for the rigidity of structures and substructures. *SIAM J. Matrix Analysis Applications*, 30(1):7–40, 2008.
- [ST14] D. Spielman and S. Teng. Nearly linear time algorithms for preconditioning and solving symmetric, diagonally dominant linear systems. *SIAM Journal on Matrix Analysis and Applications*, 35(3):835–885, 2014. Available at <http://arxiv.org/abs/cs/0607105>.
- [Str69] Volker Strassen. Gaussian elimination is not optimal. *Numer. Math.*, 13(4):354–356, August 1969.
- [SZS⁺08] Richard Szeliski, Ramin Zabih, Daniel Scharstein, Olga Veksler, Vladimir Kolmogorov, Aseem Agarwala, Marshall Tappen, and Carsten Rother. A comparative study of energy minimization methods for markov random fields with smoothness-based priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30:1068–1080, 2008.
- [TBI97] Lloyd N Trefethen and David Bau III. *Numerical linear algebra*, volume 50. Siam, 1997.
- [Ten10] Shang-Hua Teng. The Laplacian Paradigm: Emerging Algorithms for Massive Graphs. In *Theory and Applications of Models of Computation*, pages 2–14, 2010.
- [Vai89] P. M. Vaidya. Speeding-up linear programming using fast matrix multiplication. In *Proceedings of the 30th Annual Symposium on Foundations of Computer Science*, pages 332–337, Washington, DC, USA, 1989. IEEE Computer Society.
- [Wil12] Virginia Vassilevska Williams. Multiplying matrices faster than coppersmith-winograd. In *Proceedings of the Forty-fourth Annual ACM Symposium on Theory of Computing*, STOC '12, pages 887–898, New York, NY, USA, 2012. ACM.
- [Woo14] David P Woodruff. Sketching as a tool for numerical linear algebra. *Theoretical Computer Science*, 10(1-2):1–157, 2014. Available at <http://arxiv.org/abs/1411.4357>.
- [WR07] B. Wohlberg and P. Rodriguez. An iteratively reweighted norm algorithm for minimization of total variation functionals. *Signal Processing Letters, IEEE*, 14(12):948–951, dec. 2007.

- [Wri97] Stephen J Wright. *Primal-dual interior-point methods*. SIAM, 1997.
- [WW10] Virginia Vassilevska Williams and Ryan Williams. Subcubic equivalences between path, matrix and triangle problems. In *Proceedings of the 2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, FOCS '10, pages 645–654, Washington, DC, USA, 2010. IEEE Computer Society.
- [WYYZ08] Yilun Wang, Junfeng Yang, Wotao Yin, and Yin Zhang. A new alternating minimization algorithm for total variation image reconstruction. *SIAM Journal on Imaging Sciences*, 1(3):248–272, 2008.
- [Ye11] Yinyu Ye. *Interior point algorithms: theory and analysis*, volume 44. John Wiley & Sons, 2011. Available at: <http://web.stanford.edu/~yyye/main.ps>.
- [ZS14] Zhizhen Zhao and Amit Singer. Rotationally invariant image representation for viewing direction classification in cryo-em. *Journal of structural biology*, 186(1):153–166, 2014.