Dimension Reduction for Smooth Convex Optimization via Color Refinement

Bryan Zhu

Washington, United States

Instructor: Ziang Chen

October 15, 2025

Dimension Reduction for Smooth Convex Optimization via Color Refinement

Author: Bryan Zhu

Abstract

Color refinement is a graph isomorphism routine that can be extended to matrices to produce equitable partitions. It can be used for dimension reduction for linear programs and convex linearly constrained quadratic programs. We propose a novel dimension reduction technique for smooth convex optimization problems. This approach is, to our knowledge, the first effort to extend techniques that exploit fractional automorphisms beyond quadratic forms to arbitrary smooth convex objectives and constraints. For polynomial cases, we further (i) generalize equitability from matrices to tensors of arbitrary rank, (ii) develop a higher-order color refinement algorithm to capture higher-dimensional symmetries, and (iii) prove the uniqueness of the resulting coloring. Our reductions are exact, preserving feasibility, optimality, and convexity. Experiments show that our method achieves substantial reduction in both dimension and runtime on existing convex quadratically constrained linear programs under rounding to various precisions, as well as synthetic convex quadratically constrained quadratic programs.

Keywords: color refinement, convex optimization, dimension reduction, polynomial optimization

Contents

1 Introduction

An optimization problem with n variables and m constraints takes the form

$$\begin{aligned} &\text{minimize } F(x) \\ &\text{subject to } G_i(x) \leq b_i, \\ &\qquad l \leq x \leq u \end{aligned}$$

for all $1 \leq i \leq m$. For a polynomial optimization problem of degree d, $F(x)$ is in the form

$$\sum_{k=1}^{d} \frac{1}{k} \sum_{1 \leq j_1, \ldots, j_k \leq n} A^{(k)}_{j_1 \ldots j_k} \prod_{r=1}^{k} x_{j_r},$$

and each $G_i(x)$ can be expressed as

$$\sum_{k=1}^{d} \frac{1}{k} \sum_{1 \leq j_1, \ldots, j_k \leq n} \left(P_i^{(k)}\right)_{j_1 \ldots j_k} \prod_{r=1}^{k} x_{j_r}.$$

Here, the $\frac{1}{k}$ simplifies the gradient of each expression and follows convention for quadratic programs, and $A^{(k)}, P_i^{(k)} \in \mathbb{R}^{n^k}$ refer to rank k coefficient tensors that we assume are invariant under any permutation of indices. We also define $P^{(k)} \in \mathbb{R}^{m \times n^k}$ as a rank $k+1$ tensor given by stacking $P_i^{(k)}$ along a new leading dimension so that

$$P^{(k)}_{ij_1 \ldots j_k} = \left(P_i^{(k)}\right)_{j_1 \ldots j_k}.$$

Standard algorithms for convex polynomial optimization exhibit rapidly growing runtimes with instance size. For example, the simplex method for linear programming takes exponential time in the worst case [10]. For convex quadratic programs, active-set methods can also have exponential complexity [12], while interior point algorithms have complexity $O(n^3 L)$, where L is the total bit length of all entries [7]. Our method, by enabling dimension reduction, can reduce memory and runtime costs without sacrificing convexity guarantees, thus improving tractability of solvers.

Prior work has shown that dimension reduction in optimization can be achieved by exploiting symmetry through color refinement, a classic graph isomorphism algorithm that assigns colors to vertices in a way that reflects structural equivalence [8]. This approach has been applied to linear programs (LPs), which minimize a linear objective subject to linear constraints, allowing redundant variables and constraints to be removed. An extension has been developed for convex linearly constrained quadratic programs (LCQPs), where the objective is a convex quadratic function [13]. In this work, we generalize dimension reduction via color refinement to all smooth convex optimization problems, with particular emphasis on polynomial instances.

The remainder of this paper is organized as follows. Section 2 introduces necessary background and notation, and reviews prior work. Section 3 presents our generalization of these methods to convex polynomial and non-polynomial optimization problems, and proves the effectiveness of our method. Section 4 reports experimental results on convex quadratically constrained linear programs and quadratically constrained quadratic programs, a class of problems not addressed by previous approaches. Section 5 outlines possible directions for future research, and Section 6 concludes the paper.

2 Preliminaries

2.1 Color Refinement

Given a graph G and its vertex set $V(G)$, a *coloring function* $\eta : V(G) \to \mathbb{N}_0$ induces a *coloring*, or partition, of $V(G)$ into disjoint colors $\mathcal{C} = \{\eta^{-1}(c) \mid c \in \eta(V(G))\}$. We call $\mathcal{C}$ *stable* if

$$|N(v_1) \cap C_2| = |N(v_2) \cap C_2|$$

for all $v_1, v_2 \in C_1$ and $C_1, C_2 \in \mathcal{C}$ [2].

A coloring $\mathcal{C}_1$ *refines* another coloring $\mathcal{C}_2$ if, for all $C_1 \in \mathcal{C}_1$, there exists some $C_2 \in \mathcal{C}_2$ such that $C_1 \subseteq C_2$. Here, C_1 is *finer*, and C_2 is *coarser*. Additionally, we call $\mathcal{C}$ the *unit coloring* if $|\mathcal{C}| = 1$.

Color refinement aims to find the stable $\mathcal{C}$ that minimizes $|\mathcal{C}|$, which can be shown to be unique [2]. This is called the *coarsest stable partition*. Given an initial $\mathcal{C}$, color refinement gradually refines $\mathcal{C}$ by progressively splitting all vertices $v \in C_1$ based on $|N(v) \cap C_2|$. Each split for all $C_1 \in \mathcal{C}$ and some fixed $C_2 \in \mathcal{C}$ is called a *refinement operation*. Although this algorithm fails to accurately classify all non-isomorphic graphs, it can be useful for isomorphism testing because if, in the coarsest stable partition $\mathcal{C}$ of $G_1 \cup G_2$ for two graphs G_1 and G_2, there exists some $C \in \mathcal{C}$ such that $|V(G_1) \cap C| \neq |V(G_2) \cap C|$, then G_1 and G_2 are not isomorphic.

The time complexity of color refinement is

$$O((n+m)\log n),$$

where n is the number of vertices and m is the number of edges.

2.2 Dimension Reduction for Linear Programs

A *linear program* (LP) is defined as

$$\begin{aligned} &\text{minimize } c^\top x \\ &\text{subject to } Ax \leq b, \\ &\qquad l \leq x \leq u. \end{aligned}$$

We refer to $l \leq x \leq u$ as the *box domain*, and we call an x in the box domain *feasible* if it satisfies the constraint $Ax \leq b$.

Color refinement can be applied to matrices for LP dimension reduction [8]. Instead of aiming to find a stable partition, color refinement for a matrix A aims to find an *equitable partition* $(\mathcal{P}, \mathcal{Q})$, where $\mathcal{P}$ is a partition of the rows of A and $\mathcal{Q}$ is a partition of its columns. The partition is equitable if for any $S \in \mathcal{P}$ and $T \in \mathcal{Q}$,

$$\sum_{j \in T} A_{ij}$$

is constant across all $i \in S$, and

$$\sum_{i \in S} A_{ij}$$

is constant across all $j \in T$. Note how this is essentially stability for the edge-weighted bipartite graph with adjacency matrix A. In this graph, one set of vertices corresponds to the rows of A and another corresponds to its columns. The time complexity of using color refinement to find the coarsest equitable partition of a matrix $A \in \mathbb{R}^{V \times W}$ is $O((n+m)\log n)$ for $n = |V| + |W|$ and m denoting the total bit length of all nonzero entries of A.

For reduction, a partition $\mathcal{P}$ of a set V is associated with the *partition matrix* $\Pi_{\mathcal{P}} \in \{0,1\}^{V \times \mathcal{P}}$, with

$$(\Pi_{\mathcal{P}})_{vS} = \begin{cases} 1 & \text{if } v \in S, \\ 0 & \text{otherwise.} \end{cases}$$

Given Π, its *scaled transpose* Π^S is defined by taking $\Pi^\top$ and normalizing every row so that it becomes stochastic.

Now, suppose colorings $\mathcal{P}$ and $\mathcal{Q}$ satisfy the following conditions:

- For all $T \in \mathcal{Q}$, (c_j, l_j, u_j) is equal for all $j \in T$.
- $(\mathcal{P}, \mathcal{Q})$ is equitable on A.
- For all $S \in \mathcal{P}$, b_i is equal for all $i \in S$.

Additionally, let

$$c' = \Pi_{\mathcal{Q}}^{\top} c,$$
$$A' = \Pi_{\mathcal{P}}^{S} A \Pi_{\mathcal{Q}},$$
$$b' = \Pi_{\mathcal{P}}^{S} b,$$
$$x' = \Pi_{\mathcal{Q}}^{S} x,$$
$$l' = \Pi_{\mathcal{Q}}^{S} l,$$

and

$$u' = \Pi_{\mathcal{Q}}^{S} u,$$

and define an LP with these reduced terms. Then x is an optimum for the original LP if and only if x' is an optimum for the reduced LP.

2.3 Dimension Reduction for Linearly Constrained Quadratic Programs

Color refinement can also be used to reduce the dimension of convex *linearly constrained quadratic programs* (LCQPs), which take the form

$$\begin{aligned} \text{minimize } & \frac{1}{2} x^{\top} Q x + c^{\top} x \\ \text{subject to } & Ax \leq b, \\ & l \leq x \leq u \end{aligned}$$

for some positive semidefinite Q [1]. The process of reduction is nearly identical to that of LPs [13], except that $(\mathcal{Q}, \mathcal{Q})$ must also be equitable on Q, and in the reduced LCQP, we have

$$Q' = \Pi_{\mathcal{Q}}^{\top} Q \Pi_{\mathcal{Q}}.$$

3 Methods

3.1 Smooth Convex Optimization and Convex Polynomial Optimization

A general optimization problem with n variables and m constraints takes the form

$$\begin{aligned} \text{minimize } & F(x) \\ \text{subject to } & G_i(x) \leq b_i, \\ & l \leq x \leq u \end{aligned}$$

for all $1 \leq i \leq m$. We call F *smooth* if $\frac{\partial F}{\partial x_j}$ is continuous for all $1 \leq j \leq n$. Our methods apply to cases where F and all G_i are convex and smooth.

For polynomial optimization, problems beyond degree 2 naturally require coefficient tensors of rank higher than 2. Let $A^{(k)} \in \mathbb{R}^{n^k}$ denote a rank k tensor representing coefficients of degree k terms. We call $A^{(k)}$ *symmetric* if, for any permutation σ and any choice of $1 \leq j_1, \ldots, j_k \leq n$, $A_{j_1 \ldots j_k} = A_{\sigma(j_1, \ldots, j_k)}$.

We can express a convex polynomial optimization problem with degree d, n variables, and m constraints as:

$$\begin{aligned} \text{minimize } & \sum_{k=1}^{d} \frac{1}{k} \sum_{1 \leq j_1, \ldots, j_k \leq n} A^{(k)}_{j_1 \ldots j_k} \prod_{r=1}^{k} x_{j_r} \\ \text{subject to } & \sum_{k=1}^{d} \frac{1}{k} \sum_{1 \leq j_1, \ldots, j_k \leq n} \left(P_i^{(k)} \right)_{j_1 \ldots j_k} \prod_{r=1}^{k} x_{j_r} \leq b_i, \\ & l \leq x \leq u \end{aligned}$$

for all $1 \le i \le m$. Here, $A^{(k)}, P_i^{(k)} \in \mathbb{R}^{n^k}$ are rank k coefficient tensors; for convenience, we assume they are symmetric. We will also use a rank $k+1$ tensor $P^{(k)} \in \mathbb{R}^{m \times n^k}$ defined by

$$P^{(k)}_{ij_1 \dots j_k} = \left(P_i^{(k)}\right)_{j_1 \dots j_k}.$$

Because

$$\begin{aligned}
\frac{\partial}{\partial x_j}\left(\sum_{1 \le j_1,\dots,j_k \le n} A^{(k)}_{j_1\dots j_k} \prod_{r=1}^{k} x_{j_r}\right) &= \sum_{s=1}^{k} s x_j^{s-1} \sum_{j_1,\dots,j_{k-s} \in \{1,\dots,n\}\setminus\{j\}} \binom{k}{s} A^{(k)}_{j\dots jj_1\dots j_{k-s}} \prod_{r=1}^{k-s} x_{j_r} \\
&= \sum_{s=1}^{k} s x_j^{s-1} \sum_{j_1,\dots,j_{k-s} \in \{1,\dots,n\}\setminus\{j\}} \frac{k}{s}\binom{k-1}{s-1} A^{(k)}_{j\dots jj_1\dots j_{k-s}} \prod_{r=1}^{k-s} x_{j_r} \\
&= \sum_{s=1}^{k} k x_j^{s-1} \sum_{j_1,\dots,j_{k-s} \in \{1,\dots,n\}\setminus\{j\}} \binom{k-1}{s-1} A^{(k)}_{j\dots jj_1\dots j_{k-s}} \prod_{r=1}^{k-s} x_{j_r} \\
&= k \sum_{1 \le j_1,\dots,j_{k-1} \le n} A^{(k)}_{jj_1\dots j_{k-1}} \prod_{r=1}^{k-1} x_{j_r},
\end{aligned} \tag{1}$$

the $\frac{1}{k}$ removes constants from the gradient of each expression. It also follows the convention of expressing the quadratic term in LCQPs as $\frac{1}{2}x^\top Q x$.

3.2 Tensor Equitability and Reduction Theorem

We first provide some definitions for general convex optimization, then define equitability on tensors and provide analogous definitions for convex polynomial optimization. Finally, we state our reduction theorem, which applies to both cases.

Definition. A coloring $\mathcal{P}$ of the constraints and a coloring $\mathcal{Q}$ of the variables of a convex optimization problem is a *reduction coloring* if it satisfies the following conditions for all $\hat{x}$ in the box domain that satisfy $\hat{x}_{j_1} = \hat{x}_{j_2}$ if j_1 and j_2 share a color in $\mathcal{Q}$:

- If $T \in \mathcal{Q}$ and $j_1, j_2 \in T$, then
$$\left.\frac{\partial F}{\partial x_{j_1}}\right|_{x=\hat{x}} = \left.\frac{\partial F}{\partial x_{j_2}}\right|_{x=\hat{x}}.$$
- If $S \in \mathcal{P}$, $T \in \mathcal{Q}$, and $j_1, j_2 \in T$, then
$$\left.\frac{\partial}{\partial x_{j_1}}\left(\sum_{i \in S} G_i\right)\right|_{x=\hat{x}} = \left.\frac{\partial}{\partial x_{j_2}}\left(\sum_{i \in S} G_i\right)\right|_{x=\hat{x}}.$$
- For all i that share some color in $\mathcal{P}$, $G_i(\hat{x})$ is equal.
- For each $S \in \mathcal{P}$, b_i is equal for every $i \in S$.
- For each $T \in \mathcal{Q}$, (l_j, u_j) is equal for every $j \in T$.

Definition. Given a reduction coloring $(\mathcal{P}, \mathcal{Q})$ of a convex optimization problem, we define a *reduced problem*:

- The reduced objective is given by
$$F' = F\big|_{x_j \mapsto x_T \text{ for all } j \in T \text{ and } T \in \mathcal{Q}}.$$

- Reduced constraints are given by

$$G'_S = G_i\big|_{x_j \mapsto x_T \text{ for all } j \in T \text{ and } T \in \mathcal{Q}}$$

  for any $i \in S$.

- For x, l, and u, we left-multiply by $\Pi^S_{\mathcal{Q}}$ to yield the reduced vectors.
- For b, we use $b' = \Pi^S_{\mathcal{P}} b$.

Now, we extend the definition of an equitable partition of a matrix to all tensors. Matrix equitability is essentially a condition of equating sums of 1-dimensional "slices" (vectors) of 2-dimensional submatrices given by a partition. Hence, for tensors of rank $d > 2$, we generalize equitability by equating sums of rank $d-1$ slices of rank d subtensors.

Definition. We call a partition $(\mathcal{Q}_1, \ldots, \mathcal{Q}_k)$ of a rank k tensor Q *equitable* if, for any index $1 \leq r \leq k$, fixed $T_r \in \mathcal{Q}_r$, and $T_s \in \mathcal{Q}_s$ for all $s \neq r$, the value

$$\sum_{j_1 \in T_1, \ldots, j_{r-1} \in T_{r-1}, j_{r+1} \in T_{r+1}, \ldots, j_k \in T_k} Q_{j_1 \ldots j_k}$$

is equal for all $j_r \in T_r$.

Next, we define a reduction coloring and a reduced problem for convex polynomial optimization.

Definition. For a convex polynomial optimization problem, $(\mathcal{P}, \mathcal{Q})$ is a *reduction coloring* if it satisfies the following:

- For all $1 \leq k \leq d$, $(\mathcal{Q}, \ldots, \mathcal{Q})$ is equitable on $A^{(k)}$.
- For all $1 \leq k \leq d$, $(\mathcal{P}, \mathcal{Q}, \ldots, \mathcal{Q})$ is equitable on $P^{(k)}$.
- For each $S \in \mathcal{P}$, b_i is equal for every $i \in S$.
- For each $T \in \mathcal{Q}$, (l_j, u_j) is equal for every $j \in T$.

Definition. Given a reduction coloring $(\mathcal{P}, \mathcal{Q})$ of a convex polynomial optimization problem, we define a *reduced problem*:

- For each $A^{(k)}$ and $P^{(k)}$, the reduced tensors are given by

$$A^{(k)'}_{T_1 \ldots T_k} = \sum_{j_1 \in T_1, \ldots, j_k \in T_k} A^{(k)}_{j_1 \ldots j_k}$$

  and

$$\left(P^{(k)'}_S\right)_{T_1 \ldots T_k} = \sum_{j_1 \in T_1, \ldots, j_k \in T_k} \left(P^{(k)}_i\right)_{j_1 \ldots j_k}$$

  for any $i \in S$.

- For x, l, and u, we left-multiply by $\Pi^S_{\mathcal{Q}}$ to yield the reduced vectors.
- For b, we use $b' = \Pi^S_{\mathcal{P}} b$.

Finally, we state our reduction theorem. The statement of this theorem is identical for general convex optimization and convex polynomial optimization, and applies in both cases.

Theorem 1. *Consider any convex (polynomial) optimization problem. Let $(\mathcal{P}, \mathcal{Q})$ be a reduction coloring. If x is an optimum for the original problem, then $x' = \Pi^S_{\mathcal{Q}} x$ is an optimum for the reduced problem. If x' is an optimum for the reduced problem, then $x = \Pi_{\mathcal{Q}} x'$ is an optimum for the original problem.*

We will prove Theorem 1 in the following section.

Example 3.1 (Convex quadratically constrained quadratic program reducing to two variables and two constraints)**.** *Consider the quadratically constrained quadratic program (QCQP)*

$$\begin{aligned} &\text{minimize } x^\top Q x + c^\top x \\ &\text{subject to } x^\top P_i x \le b_i, \\ &\qquad \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \le x \le \begin{bmatrix} 5 \\ 5 \\ 3 \\ 3 \end{bmatrix} \end{aligned}$$

for $1 \le i \le 4$*, where*

$$Q = \begin{bmatrix} 14 & 2 & 1 & 5 \\ 2 & 14 & 5 & 1 \\ 1 & 5 & 13 & 1 \\ 5 & 1 & 1 & 13 \end{bmatrix},$$

$$c = \begin{bmatrix} 2 \\ 2 \\ 7 \\ 7 \end{bmatrix},$$

$$b = \begin{bmatrix} 100 \\ 100 \\ 90 \\ 90 \end{bmatrix},$$

and

$$P_1 = \begin{bmatrix} 4 & 2 & 2 & 2 \\ 2 & 4 & 2 & 2 \\ 2 & 2 & 7 & 3 \\ 2 & 2 & 3 & 7 \end{bmatrix}, \qquad P_2 = \begin{bmatrix} 5 & 1 & 2 & 2 \\ 1 & 5 & 2 & 2 \\ 2 & 2 & 6 & 4 \\ 2 & 2 & 4 & 6 \end{bmatrix},$$

$$P_3 = \begin{bmatrix} 3 & 1 & 1 & 1 \\ 1 & 3 & 1 & 1 \\ 1 & 1 & 6 & 6 \\ 1 & 1 & 6 & 6 \end{bmatrix}, \qquad P_4 = \begin{bmatrix} 2 & 2 & 1 & 1 \\ 2 & 2 & 1 & 1 \\ 1 & 1 & 8 & 4 \\ 1 & 1 & 4 & 8 \end{bmatrix}.$$

Then $\mathcal{P} = \{S_1, S_2\}$ *with* $S_1 = \{1, 2\}$ *and* $S_2 = \{3, 4\}$ *and* $\mathcal{Q} = \{T_1, T_2\}$ *with* $T_1 = \{1, 2\}$ *and* $T_2 = \{3, 4\}$ *form a reduction coloring.*

By Theorem 1,

$$Q' = \begin{bmatrix} 32 & 12 \\ 12 & 28 \end{bmatrix}$$

and

$$c' = \begin{bmatrix} 4 \\ 14 \end{bmatrix},$$

and each constraint color $S \in \mathcal{P}$ *yields a reduced form*

$$P'_{S_1} = \begin{bmatrix} 12 & 8 \\ 8 & 20 \end{bmatrix},$$

$$P'_{S_2} = \begin{bmatrix} 8 & 4 \\ 4 & 24 \end{bmatrix},$$

and

$$b' = \begin{bmatrix} 100 \\ 90 \end{bmatrix}.$$

The optimum for both the original and the reduced problems is the origin.

Theorem 1, when applied to convex polynomial optimization, generalizes prior work on dimension reduction for LPs [8] and for convex LCQPs [13]. In all cases, equal b_i for all $i \in S$ and $S \in \mathcal{P}$ and equal (l_j, u_j) for all $j \in T$ and $T \in \mathcal{Q}$ are required. Additionally, each of our other conditions for a reduction coloring can be simplified for LPs and LCQPs. Equitability on $A^{(k)}$ simplifies to equal c_j for all $j \in T$ and $T \in \mathcal{Q}$ and, for LCQPs, equitability of $(\mathcal{Q}, \mathcal{Q})$ on Q. Equitability on $P^{(k)}$ simplifies to equitability of $(\mathcal{P}, \mathcal{Q})$ on A. These are precisely the conditions on $\mathcal{P}$ and $\mathcal{Q}$ in the literature.

Example 3.2 (Explicit example of a reduction coloring for non-polynomial optimization)**.** *We demonstrate that Theorem 1 extends naturally to smooth convex non-polynomial optimization by considering the binary logistic regression problem*

$$\text{minimize } F(w,b) = -\sum_{i=1}^{m}(y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)),$$

where $\hat{y} = \sigma(Xw+b)$ is the predicted probability vector, $X \in \mathbb{R}^{m\times n}$ contains the input samples, σ represents the sigmoid function, and $y \in \{0,1\}^m$ are the ground truth labels. At some point $\hat{w}$ with $\hat{w}_{j_1} = \hat{w}_{j_2}$ if j_1 and j_2 share a color in $\mathcal{Q}$,

$$\begin{aligned}\left.\frac{\partial F}{\partial w_j}\right|_{w=\hat{w}} &= \sum_{i=1}^{m} X_{ij}\,(\hat{y}_i - y_i)\\ &= \sum_{i=1}^{m} X_{ij}\left(\sigma\left(b + \sum_{k=1}^{n} X_{ik}w_k\right) - y_i\right)\\ &= \sum_{i=1}^{m} X_{ij}\left(\sigma\left(b + \sum_{T\in\mathcal{Q}} w_T \sum_{k\in T} X_{ik}\right) - y_i\right),\end{aligned}$$

where we use w_T to denote the constant value of w_j for all $j \in T$.

We call $\mathcal{Q}$ a reduction coloring for logistic regression if the following conditions are met:

- *Define $\mathcal{P}$ as the coloring given by partitioning all $1 \le i \le m$ by*

$$\left\{ \left(T, \sum_{j\in T} X_{ij} \right) \,\middle|\, T \in \mathcal{Q} \right\}.$$

  We require that $(\mathcal{P}, \mathcal{Q})$ is equitable on X.

- *If j_1 and j_2 share a color in $\mathcal{Q}$, then*

$$\sum_{i=1}^{m} X_{ij_1} y_i = \sum_{i=1}^{m} X_{ij_2} y_i.$$

Under these conditions, for any $T \in \mathcal{Q}$ and $j_1, j_2 \in T$,

$$\begin{aligned}\left(\left.\frac{\partial F}{\partial w_{j_1}}\right|_{w=\hat{w}}\right) - \left(\left.\frac{\partial F}{\partial w_{j_2}}\right|_{w=\hat{w}}\right) &= \sum_{i=1}^{m}(X_{ij_1} - X_{ij_2})\left(\sigma\left(b + \sum_{T\in\mathcal{Q}} w_T \sum_{k\in T} X_{ik}\right) - y_i\right)\\ &= \sum_{i=1}^{m}(X_{ij_1} - X_{ij_2})\,\sigma\left(b + \sum_{T\in\mathcal{Q}} w_T \sum_{j\in T} X_{ij}\right) - \left(\sum_{i=1}^{m} X_{ij_1} y_i - X_{ij_2} y_i\right)\\ &= \sum_{i=1}^{m}(X_{ij_1} - X_{ij_2})\,\sigma\left(b + \sum_{T\in\mathcal{Q}} w_T \sum_{j\in T} X_{ij}\right).\end{aligned}$$

Due to equitability of $(\mathcal{P}, \mathcal{Q})$ on X, we have constant $\sum_{j\in T} X_{ij}$ for all $i \in S$ and constant $\sum_{i\in S} X_{ij}$ for all $j \in T$ if $S \in \mathcal{P}$ and $T \in \mathcal{Q}$. Hence, the logit for each sample,

$$z_i = b + \sum_{T\in\mathcal{Q}} w_T \sum_{j\in T} X_{ij},$$

is equal for all i that share a color $S \in \mathcal{P}$, and we can denote this value as z_S. Using this, we can further simplify:

$$\begin{aligned}\sum_{i=1}^{m}(X_{ij_1} - X_{ij_2})\,\sigma\left(b + \sum_{T\in\mathcal{Q}} w_T \sum_{j\in T} X_{ij}\right) &= \sum_{S\in\mathcal{P}} \sigma(z_S)\left(\left(\sum_{i\in S} X_{ij_1}\right) - \sum_{i\in S} X_{ij_2}\right)\\ &= \sum_{S\in\mathcal{P}} \sigma(z_S)\cdot 0\\ &= 0.\end{aligned}$$

Hence $\frac{\partial F}{\partial w_{j_1}}\Big|_{w=\hat{w}} = \frac{\partial F}{\partial w_{j_2}}\Big|_{w=\hat{w}}$ for all $j_1, j_2 \in T$.

If each feature j is normalized so that

$$\sum_{i=1}^{m} X_{ij} y_i = \sum_{i|y_i=1} X_{ij} = 1,$$

then our second condition is automatically satisfied. Hence, with proper normalization, the only requirement for reduction is our first condition.

3.3 Proof of Reduction Theorem

In this section, we will occasionally use color-based indexing in cases where all elements of a color are known to be equal. For example, we might say x_T for $T \in \mathcal{Q}$ if it is known that x_j is equal for all $j \in T$.

3.3.1 General Convex Optimization

Suppose that $(\mathcal{P}, \mathcal{Q})$ is a reduction coloring. Let $\hat{x} = \Pi_{\mathcal{Q}}\Pi_{\mathcal{Q}}^{S} x$. Note that if $j_1 \in T$ and $T \in \mathcal{Q}$, then

$$\hat{x}_{j_1} = \frac{\sum_{j_2\in T} x_{j_2}}{|T|}.$$

We prove our reduction theorem in three steps:

1. If x is an optimum for the problem, then so is $\hat{x}$.
2. If x is feasible, $\hat{x}$ must be too.
3. The above statements imply equivalence of the original and reduced problems.

Firstly, we show that $F(\hat{x}) \le F(x)$ using the Lagrangian, inspired by [3]. Suppose that x is fixed, and we wish to minimize $F(z)$ such that

$$\sum_{j\in T} z_j = \sum_{j\in T} x_j$$

for all $T \in \mathcal{Q}$. It suffices to show that $z = \hat{x}$ is a minimum, and hence, since the objective is convex and it is obvious that $z = \hat{x}$ satisfies our condition, that the partial derivative of the Lagrangian

$$L = F(z) - \sum_{T\in\mathcal{Q}} \lambda_T \left(\left(\sum_{j\in T} z_j\right) - \sum_{j\in T} x_j\right)$$

with respect to each z_{j_s} is 0 if $z = \hat{x}$. We have that

$$\frac{\partial L}{\partial z_j} = \frac{\partial F}{\partial z_j} - \lambda_T$$

if $j \in T$. Then our claim is true if we have equal $\left.\frac{\partial F}{\partial z_j}\right|_{z=\hat{x}}$ for all j in the same color T, which is one of our reduction coloring conditions.

Secondly, we show that $\hat{x}$ is feasible. Due to our requirement that each $T \in \mathcal{Q}$ has the same (l_j, u_j) for all $j \in T$, we immediately have that $\hat{x}$ is in the box domain. Additionally, we know that

$$\sum_{i \in S} G_i(\hat{x}) \leq \sum_{i \in S} G_i(x) \leq |S| b_S.$$

The first inequality is true by similar logic as above, and the second inequality is true by assumption and due to our requirement that b_i is equal for all $i \in S$. In fact, $G_i(\hat{x}) \leq b_i$ for all individual $1 \leq i \leq m$ because we require that $G_i(\hat{x})$ is equal for all $i \in S$.

Finally, we show that the above implies optimality of x' on the reduced problem. Since F' is given simply by replacing each x_j in F with x_T where $j \in T$, we have that

$$F'(x') = F(\hat{x}),$$

so if x' is not optimal, then $\hat{x}$ and hence x is not optimal, which contradicts our assumption. We can use similar logic with the reduced constraints: $G'(x') = G(\hat{x})$, so x' is feasible too.

For the reverse direction, note that $\hat{x} = \Pi_{\mathcal{Q}} x'$. Hence, if x' is optimal, it is impossible for $\hat{x}$ to not be optimal. Otherwise, there exists some feasible y with $F(y) < F(\hat{x}) \implies F'(y') < F'(x')$, contradiction.

3.3.2 Convex Polynomial Optimization

It is obvious that for convex polynomial optimization problems, our process for reduction is equivalent to that for the general case. Hence, it suffices to show that our two definitions of a reduction coloring are equivalent.

We have from Equation 1 that

$$\begin{aligned}\left.\frac{\partial F}{\partial z_j}\right|_{z=\hat{x}} &= \sum_{k=1}^{d} \sum_{1 \leq j_1, \ldots, j_{k-1} \leq n} A^{(k)}_{j j_1 \ldots j_{k-1}} \prod_{r=1}^{k-1} z_{j_r} \\ &= \sum_{k=1}^{d} \sum_{T_1, \ldots, T_{k-1} \in \mathcal{Q}} \left(\prod_{r=1}^{k-1} z_{T_r} \right) \sum_{j_1 \in T_1, \ldots, j_{k-1} \in T_{k-1}} A^{(k)}_{j j_1 \ldots j_{k-1}}\end{aligned}$$

if $j \in T$. (Note how we use z_{T_r} to refer to the value of z_j for $j \in T_r$, which is equal for all $j \in T_r$.) Then $\left.\frac{\partial F}{\partial z_j}\right|_{z=\hat{x}}$ is equal across $j \in T$ for each $T \in \mathcal{Q}$ if we have equal

$$\sum_{j_1 \in T_1, \ldots, j_{k-1} \in T_{k-1}} A^{(k)}_{j j_1 \ldots j_{k-1}}$$

for all $j \in T$ regardless of $T_1, \ldots, T_{k-1} \in \mathcal{Q}$, which is exactly our equitability condition.

Additionally, note that equitability of $(\mathcal{P}, \mathcal{Q}, \ldots, \mathcal{Q})$ on $P^{(k)}$ is equivalent to the following two conditions:

- For all $S \in \mathcal{P}$, $(\mathcal{Q}, \ldots, \mathcal{Q})$ is equitable on $\sum_{i \in S} P^{(k)}_i$.
- If we fix $S \in \mathcal{P}$ and $T_1, \ldots, T_k \in \mathcal{Q}$, the value

$$\sum_{j_1 \in T_1, \ldots, j_k \in T_k} \left(P^{(k)}_i \right)_{j_1 \ldots j_k}$$

  is constant for all $i \in S$.

Requiring equitability on $\sum_{i \in S} P_i^{(k)}$ is equivalent to requiring equal $\frac{\partial}{\partial x_j} \left(\sum_{i \in S} G_i \right) \Big|_{x=\hat{x}}$ for all j in the same color T. Also, we have that

$$\begin{aligned} G_i(\hat{x}) &= \sum_{k=1}^{d} \frac{1}{k} \sum_{1 \le j_1, \ldots, j_k \le n} \left(P_i^{(k)} \right)_{j_1 \ldots j_k} \prod_{r=1}^{k} \hat{x}_{j_r} \\ &= \sum_{k=1}^{d} \frac{1}{k} \sum_{T_1, \ldots, T_k \in \mathcal{Q}} \left(\prod_{r=1}^{k} \hat{x}_{T_r} \right) \sum_{j_1 \in T_1, \ldots, j_k \in T_k} \left(P_i^{(k)} \right)_{j_1 \ldots j_k} \end{aligned}$$

is equal for all $i \in S$ if

$$\sum_{j_1 \in T_1, \ldots, j_k \in T_k} \left(P_i^{(k)} \right)_{j_1 \ldots j_k}$$

is equal across all $i \in S$ for any choices of $T_1, \ldots, T_k$.

Remark. To see why the reduced objective and constraints are convex, note that every reduced expression is given by a scaled intersection of the original curve and the affine space given by setting all x_j equal for all $j \in T$ and any fixed $T \in \mathcal{Q}$. If a function is convex over a convex set, then it is also convex over any affine subspace of this set [11], showing that the reduced expressions are convex.

3.4 Necessity of Each Condition

For each condition we impose on a convex polynomial optimization reduction coloring, we establish its necessity by providing a counterexample: if the condition is removed, the resulting coloring is invalid. These counterexamples demonstrate that none of the conditions can be omitted. They also prove the necessity of each condition we set for a reduction coloring for the general case. Note that if there does not exist an optimum with all x_j equal for all $j \in T$ for any fixed $T \in \mathcal{Q}$ in a given coloring, then that coloring is automatically invalid.

- **For equitability on** $A^{(k)}$: minimize $x_1^2 + 4x_2^2$ such that $-x_1 - x_2 \le -1$. Without this condition, we get the unit coloring. But the unique optimum is $x_1 = 0.8$ and $x_2 = 0.2$.
- **For equitability on** $P^{(k)}$: minimize $x_1^2 + x_2^2$ such that $x_1^2 + 4x_1 \le -2$ and $2x_2^2 + 4x_2 \le -2$. Without this condition, we get the unit coloring. But the unique optimum is $x_1 = \sqrt{2} - 2$ and $x_2 = -1$.
- **For equal** b_i **in each** $S \in \mathcal{P}$: minimize $x_1^2 + x_2^2$ such that $-x_1 - x_2 \le -1$ and $-x_1 - x_2 \le -2$. The unique optimum is $x_1 = x_2 = 1$. Without this condition, we get the unit coloring. As with equal $\sum_{j_1 \in T_1, \ldots, j_k \in T_k} \left(P_i^{(k)} \right)_{j_1 \ldots j_k}$, our method no longer applies, as the values of b'_S become ambiguous. If we take the average, the only constraint becomes $-2x \le -1.5$, with the unique optimum being $x = x_1 = x_2 = 0.75$. If we take the maximum, the only constraint becomes $-2x \le -1$, with the unique optimum being $x = x_1 = x_2 = 0.5$. These are both incorrect.

  We will propose a different example to show why taking the minimum does not work either. Suppose that we wish to minimize $(x_1 - 10)^2 + (x_2 - 10)^2$ such that $x_1^2 \le 1$ and $x_2^2 \le 4$. The unique optimum is $x_1 = 1$ and $x_2 = 2$, but without this condition, if we take the minimum, the only constraint becomes $x^2 \le 1$, with the unique optimum being $x = x_1 = x_2 = 1$. This is also incorrect.
- **For equal** (l_j, u_j) **in each** $T \in \mathcal{Q}$: minimize $x_1^2 + x_2^2$ such that $0 \le x_1 \le 1$ and $2 \le x_2 \le 3$. Without this condition, we get the unit coloring. But the unique optimum is $x_1 = 0$ and $x_2 = 2$.

3.5 Generalized Color Refinement

We begin by presenting the pseudocode for our color refinement algorithm, which generalizes for convex polynomial optimization.

Algorithm 1 Color refinement for polynomial cases

Initialize: Partition $\mathcal{P}$ by (b_i), and $\mathcal{Q}$ by $\left(l_j, u_j, A^{(1)}_j\right)$
Add all colors in $\mathcal{P} \cup \mathcal{Q}$ to a stack S_{refine}
while $S_{\text{refine}} \neq \emptyset$ **and** ($|\mathcal{P}| < m$ **or** $|\mathcal{Q}| < n$) **do**
    $R \leftarrow \textsc{Pop}(S_{\text{refine}})$
    if $R \in \mathcal{P}$ **then**
        for all $T \in \mathcal{Q}$ **do**
            Refine $\mathcal{Q}$: partition all $j_1 \in T$ by

$$\left\{ \left(T_2, \ldots, T_k, \sum_{j_2 \in T_2, \ldots, j_k \in T_k} \sum_{i \in R} \left(P^{(k)}_i\right)_{j_1 \ldots j_k} \right) \;\middle|\; T_2, \ldots, T_k \in \mathcal{Q}, 1 \leq k \leq d \right\}$$

        end for
        Push all newly created colors into S_{refine}
    else if $R \in \mathcal{Q}$ **then**
        for all $T \in \mathcal{Q}$ **do**
            Refine $\mathcal{Q}$: partition all $j_1 \in T$ by

$$\left\{ \left(T_3, \ldots, T_k, \sum_{j_2 \in R, j_3 \in T_3, \ldots, j_k \in T_k} A^{(k)}_{j_1 \ldots j_k} \right) \;\middle|\; T_3, \ldots, T_k \in \mathcal{Q}, 2 \leq k \leq d \right\}$$

            Refine $\mathcal{Q}$: partition all $j_1 \in T$ by

$$\left\{ \left(S, T_3, \ldots, T_k, \sum_{j_2 \in R, j_3 \in T_3, \ldots, j_k \in T_k} \sum_{i \in S} \left(P^{(k)}_i\right)_{j_1 \ldots j_k} \right) \;\middle|\; S \in \mathcal{P}, T_3, \ldots, T_k \in \mathcal{Q}, 2 \leq k \leq d \right\}$$

        end for
        for all $S \in \mathcal{P}$ **do**
            Refine $\mathcal{P}$: partition all $i \in S$ by

$$\left\{ \left(T_2, \ldots, T_k, \sum_{j_1 \in R, j_2 \in T_2, \ldots, j_k \in T_k} \left(P^{(k)}_i\right)_{j_1 \ldots j_k} \right) \;\middle|\; T_2, \ldots, T_k \in \mathcal{Q}, 1 \leq k \leq d \right\}$$

        end for
        Push all newly created colors into S_{refine}
    end if
end while
return $(\mathcal{P}, \mathcal{Q})$

When we split a color R into sub-colors $\mathcal{S} = \{\gamma, \gamma', \gamma'', \dots\}$ (we assume without loss of generality that $|\gamma| \geq |\gamma'| \geq |\gamma''| \geq \dots$), γ keeps the R label, while each of $\mathcal{S} \setminus \{\gamma\}$ is assigned a distinct new label.

Now, we discuss the motivations behind various parts of Algorithm 1 and why it converges to a reduction coloring. Note that our requirement of equitability on $A^{(1)}$ is equivalent to requiring that $A^{(1)}_j$ is equal for all j that share the same color in $\mathcal{Q}$. It is easy to satisfy this condition, in addition to equal b_i within each $S \in \mathcal{P}$ and equal (l_j, u_j) within each $T \in \mathcal{Q}$, by simply initializing $\mathcal{P}$ and $\mathcal{Q}$ to account for these requirements, as seen in our pseudocode for Algorithm 1.

In the original color refinement algorithm, when refining a stable coloring $\mathcal{P}$ of a graph, S_{refine} is always maintained as a *sufficient refining color set*. This means that for all $S_1, S_2 \in \mathcal{P}$, if there exist $v_1, v_2 \in S_1$ such that $|N(v_1) \cap S_2| \neq |N(v_2) \cap S_2|$, then this inequality also holds for some $S_2 \in S_{\text{refine}}$ [2]. For convenience, we say that in this scenario, S_1 *has an objection with* S_2. Informally, S_{refine} is a sufficient refining color set

if any color that has objections, has an objection with at least one color in S_{refine}.

For each color R popped out of S_{refine}, color refinement essentially resolves any objections other colors have with R by splitting those other colors as needed. Because this may lead to new objections with newly split colors, all of the newly created colors are pushed into S_{refine} to ensure that S_{refine} remains a sufficient refining color set. When the algorithm terminates, $|S_{\text{refine}}| = 0$, implying no objections and therefore stability of $\mathcal{P}$.

We provide several analogous definitions of objections, or possible violations of our reduction coloring conditions, for our generalized algorithm. For all $T_1 \in \mathcal{Q}$, if, for some $k \geq 2$, there exist $j^{(1)}, j^{(2)} \in T_1$ and $T_2, \ldots, T_k \in \mathcal{Q}$ such that

$$\sum_{j_2 \in T_2, \ldots, j_k \in T_k} A^{(k)}_{j^{(1)} j_2 \ldots j_k} \neq \sum_{j_2 \in T_2, \ldots, j_k \in T_k} A^{(k)}_{j^{(2)} j_2 \ldots j_k},$$

then T_1 has an objection with T_2. (Note that in this case, T_1 also has an objection with $T_3, \ldots, T_k$.)

Similarly, T_1 has an objection with T_2 if, for some $k \geq 2$, there exist $j^{(1)}, j^{(2)} \in T_1$, colors $T_2, \ldots, T_k \in \mathcal{Q}$, and $S \in \mathcal{P}$ such that

$$\sum_{j_2 \in T_2, \ldots, j_k \in T_k} \sum_{i \in S} \left(P^{(k)}_i\right)_{j^{(1)} j_2 \ldots j_k} \neq \sum_{j_2 \in T_2, \ldots, j_k \in T_k} \sum_{i \in S} \left(P^{(k)}_i\right)_{j^{(2)} j_2 \ldots j_k}.$$

In this case, T_1 also has an objection with S.

Finally, if there exist $i^{(1)}, i^{(2)} \in S$ and $T_1, \ldots, T_k \in \mathcal{Q}$ such that

$$\sum_{j_1 \in T_1, \ldots, j_k \in T_k} \left(P^{(k)}_{i^{(1)}}\right)_{j_1 \ldots j_k} \neq \sum_{j_1 \in T_1, \ldots, j_k \in T_k} \left(P^{(k)}_{i^{(2)}}\right)_{j_1 \ldots j_k},$$

then S has an objection with every color in $\{T_1, \ldots, T_k\}$.

In general, to explain all of our objection definitions, suppose we require equitability of $(\mathcal{Q}_1, \ldots, \mathcal{Q}_k)$ on a rank k tensor $Q \in \{A^{(k)}, P^{(k-1)}\}$. If there exist fixed $T_r \in \mathcal{Q}_r$ and $T_s \in \mathcal{Q}_s$ for all $s \neq r$ such that

$$\sum_{j \in T_1, \ldots, j_{r-1} \in T_{r-1}, j_{r+1} \in T_{r+1}, \ldots, j_k \in T_k} Q_{j_1 \ldots j_k}$$

is not equal for all $j_r \in T_r$, then T_r has an objection with every color in $\{T_s \mid s \neq r\}$.

Note that for each R popped out of S_{refine}, our generalized algorithm still resolves any objections other colors have with R.

Lemma 1. *Using all definitions of objections, S_{refine} remains a sufficient refining color set after each iteration of the loop.*

Proof. By definition of sufficient refining color set, it suffices to show that for each of our three definitions of objections, it is always guaranteed after each iteration of the loop that if any color R_1 has an objection with another color R_2, then R_1 has an objection with some $R_3 \in S_{\text{refine}}$.

Suppose for the sake of contradiction that some color has an objection with a color $R \notin S_{\text{refine}}$, but does not have an objection with any colors in S_{refine}. However, note that R must have been in S_{refine} at some point. Additionally, after one iteration after R was popped out of S_{refine}, no other color had an objection with R.

Hence, there are only two ways in which our claim could fail:

1. Some $R \notin S_{\text{refine}}$ is split into $\{\gamma, \gamma', \gamma'', \ldots\}$ at some point (with all resulting colors except γ pushed into S_{refine}), leading to some other color having an objection with γ but not any colors in S_{refine}, including $\gamma', \gamma'', \cdots \in S_{\text{refine}}$.

2. After one color's objections with $\gamma', \gamma'', \ldots$ are resolved, that color still has an objection with γ.

We prove that both of these are impossible for any definition of an objection. In particular, we show that if a color R' does not have an objection with R but does have an objection with γ, it must also have an objection with at least one color in $\mathcal{S} \setminus \{\gamma\}$, which also proves that after all objections by R' with colors

in $\mathcal{S} \setminus \{\gamma\}$ are resolved, R' no longer has an objection with γ. We assume that R' is not split, but it is easy to see how our proof still applies if it is.

Without loss of generality, suppose we are refining for equitability of $(\mathcal{Q}_1, \ldots, \mathcal{Q}_k)$ on Q, and $R' \in \mathcal{Q}_1$ has an objection with $\gamma \subset R$ for $R \in \mathcal{Q}_2$. Because R' does not have an objection with R, we have for any $T_3 \in \mathcal{Q}_3, \ldots, T_k \in \mathcal{Q}_k$ and $j^{(1)}, j^{(2)} \in R'$ with

$$\sum_{j_2 \in \gamma, j_3 \in T_3, \ldots, j_k \in T_k} Q_{j^{(1)} j_2 j_3 \ldots j_k} \neq \sum_{j_2 \in \gamma, j_3 \in T_3, \ldots, j_k \in T_k} Q_{j^{(2)} j_2 j_3 \ldots j_k}$$

that

$$\begin{aligned}
\sum_{\rho \in \mathcal{S} \setminus \{\gamma\}} \sum_{j_2 \in \rho, j_3 \in T_3, \ldots, j_k \in T_k} Q_{j^{(1)} j_2 \ldots j_k} &= \left(\sum_{j_2 \in R, j_3 \in T_3, \ldots, j_k \in T_k} Q_{j^{(1)} j_2 \ldots j_k} \right) - \sum_{j_2 \in \gamma, j_3 \in T_3, \ldots, j_k \in T_k} Q_{j^{(1)} j_2 \ldots j_k} \\
&\neq \left(\sum_{j_2 \in R, j_3 \in T_3, \ldots, j_k \in T_k} Q_{j^{(2)} j_2 \ldots j_k} \right) - \sum_{j_2 \in \gamma, j_3 \in T_3, \ldots, j_k \in T_k} Q_{j^{(2)} j_2 \ldots j_k} \\
&= \sum_{\rho \in \mathcal{S} \setminus \{\gamma\}} \sum_{j_2 \in \rho, j_3 \in T_3, \ldots, j_k \in T_k} Q_{j^{(2)} j_2 \ldots j_k},
\end{aligned}$$

so this inequality also must hold for at least one individual $\rho \in \mathcal{S} \setminus \{\gamma\}$. Hence, R' also has an objection with some other color in $\mathcal{S}$ besides γ. □

Lemma 1 shows that Algorithm 1 converges to a reduction coloring because when Algorithm 1 terminates, there are no more objections and therefore no more violations of our conditions for a reduction coloring.

3.6 Time Complexity of Generalized Color Refinement

The first time the color R of some index $1 \leq i \leq m$ or $1 \leq j \leq n$ is pushed into S_{refine}, it has size $|R|$. If the color of i or j ever gets pushed into S_{refine} again, then at some point, R must have been split, with i or j being assigned to some color in $\mathcal{S} \setminus \{\gamma\}$. Hence, the current color of i or j has size at most $\frac{|R|}{2}$. This implies that each variable can be pushed into S_{refine} at most $\log_2(n)$ times, and each constraint can be pushed at most $\log_2(m)$ times.

When we pop a color R from S_{refine} and process it, we need to examine all entries of $A^{(k)}$ and $P^{(k)}$ for each $1 \leq k \leq d$ that include an index in R. We need to examine up to n^k entries for $A^{(k)}$ and up to mn^k entries for $P^{(k)}$. Because each entry in $P^{(k)}$ has one constraint index and k variable indices, it is "touched" at most $O(k \log(n) + \log(m))$ times. Similarly, each entry in $A^{(k)}$ is "touched" at most $O(k \log(n))$ times. Hence, the total time complexity is

$$\begin{aligned}
&O\left(\left(\sum_{k=1}^{d} k n^k \log(n) \right) + \sum_{k=1}^{d} m n^k (k \log(n) + \log(m)) \right) \\
&= O\left((\log(n) + m \log(n)) \left(\sum_{k=1}^{d} k n^k \right) + m \log(m) \left(\sum_{k=1}^{d} n^k \right) \right) \\
&= O\left((m+1) \log(n) \left(\sum_{k=1}^{d} k n^k \right) + m \log(m) \frac{n^{d+1} - n}{n-1} \right) \\
&= O\left((m+1) \log(n) \left(\sum_{k=1}^{d} k n^k \right) + m n^d \log(m) \right),
\end{aligned}$$

and since

$$\begin{aligned}
\sum_{k=1}^{d} kn^k &= \sum_{k_1=1}^{d} \sum_{k_2=k_1}^{d} n^{k_2} \\
&= \sum_{k_1=1}^{d} \frac{n^{d+1} - n^{k_1}}{n-1} \\
&= \frac{1}{n-1} \left(\sum_{k_1=1}^{d} n^{d+1} - \sum_{k_1=1}^{d} n^{k_1} \right) \\
&= \frac{1}{n-1} \left(dn^{d+1} - \sum_{k_1=1}^{d} n^{k_1} \right) \\
&= \frac{1}{n-1} \left(dn^{d+1} - \frac{n^{d+1} - n}{n-1} \right) \\
&= \Theta(dn^d),
\end{aligned}$$

the time complexity of Algorithm 1 is

$$O\left((m+1)dn^d \log(n) + mn^d \log(m)\right) = O\left(n^d((m+1)d\log(n) + m\log(m))\right).$$

3.7 Uniqueness of Coarsest Reduction Coloring

Inspired by [2], we generalize the definition of a refining operation as follows:

Definition. A *refinement operation* does one of the following:

- Refines $\mathcal{Q}$ for equitability on $A^{(k)}$ by splitting each $T \in \mathcal{Q}$ based on

$$\left\{ \left(T_3, \ldots, T_k, \sum_{j_2 \in R, j_3 \in T_3, \ldots, j_k \in T_k} A^{(k)}_{j_1 \ldots j_k} \right) \,\middle|\, T_3, \ldots, T_k \in \mathcal{Q} \right\}$$

  for $j_1 \in T$ and some fixed $R \in \mathcal{Q}$.

- Refines $\mathcal{Q}$ for equitability on $P^{(k)}$ by splitting each $T \in \mathcal{Q}$ based on

$$\left\{ \left(S, T_3, \ldots, T_k, \sum_{j_2 \in R, j_3 \in T_3, \ldots, j_k \in T_k} \sum_{i \in S} \left(P^{(k)}_i \right)_{j_1 \ldots j_k} \right) \,\middle|\, S \in \mathcal{P}, T_3, \ldots, T_k \in \mathcal{Q} \right\}$$

  for $j_1 \in T$ and some fixed $R \in \mathcal{Q}$.

- Refines $\mathcal{Q}$ for equitability on $P^{(k)}$ by splitting each $T \in \mathcal{Q}$ based on

$$\left\{ \left(T_2, \ldots, T_k, \sum_{j_2 \in T_2, \ldots, j_k \in T_k} \sum_{i \in R} \left(P^{(k)}_i \right)_{j_1 \ldots j_k} \right) \,\middle|\, T_2, \ldots, T_k \in \mathcal{Q} \right\}$$

  for $j_1 \in T$ and some fixed $R \in \mathcal{P}$.

- Refines $\mathcal{P}$ for equitability on $P^{(k)}$ by splitting each $S \in \mathcal{P}$ based on

$$\left\{ \left(T_2, \ldots, T_k, \sum_{j_1 \in R, j_2 \in T_2, \ldots, j_k \in T_k} \left(P^{(k)}_i \right)_{j_1 \ldots j_k} \right) \,\middle|\, T_2, \ldots, T_k \in \mathcal{Q} \right\}$$

  for $i \in S$ and some fixed $R \in \mathcal{Q}$.

In general, a refinement operation refines $\mathcal{Q}_r$ for equitability of $(\mathcal{Q}_1, \ldots, \mathcal{Q}_k)$ on Q by splitting each $T_r \in \mathcal{Q}_r$ by

$$\left\{ \left(T_1, \ldots, T_k, \sum_{j_s \in T_s, s \notin \{r,s'\}} \sum_{j_{s'} \in R} Q_{j_1 \ldots j_k} \right) \,\middle|\, T_s \in \mathcal{Q}_s, s \notin \{r, s'\} \right\}$$

for $j_r \in T_r$ and some fixed $R \in \mathcal{Q}_{s'}$ where $s' \neq r$. Note that in Algorithm 1, a refinement operation consists of a line beginning with `refine` and the innermost loop that surrounds it. In each case, the loop iterates through colors in either $\mathcal{P}$ or $\mathcal{Q}$ and refines the corresponding coloring as needed based on other colors.

We now prove the uniqueness of the coarsest reduction coloring in two steps.

Lemma 2. *Suppose that $(\mathcal{P}', \mathcal{Q}')$ refines $(\mathcal{P}, \mathcal{Q})$, and $(\mathcal{P}', \mathcal{Q}')$ is a reduction coloring. After any refinement operation on $(\mathcal{P}, \mathcal{Q})$, $(\mathcal{P}', \mathcal{Q}')$ still refines $(\mathcal{P}, \mathcal{Q})$.*

Proof. It suffices to show that if $i^{(1)}$ and $i^{(2)}$ share a color $S' \in \mathcal{P}'$, then they also share a color in $(\mathcal{P}, \mathcal{Q})$ after any refinement operation, and likewise for any $j^{(1)}$ and $j^{(2)}$ that share a color $T' \in \mathcal{Q}'$. The crucial insight here is that refinement operations split colors in $\mathcal{P}$ and $\mathcal{Q}$ only if certain conditions are true, and these conditions are not met in $(\mathcal{P}, \mathcal{Q})$. For example, for the refinement operation for equitability on $A^{(k)}$, $j^{(1)}$ and $j^{(2)}$ are split only if there exist $R, T_2, \ldots, T_k \in \mathcal{Q}$ such that

$$\sum_{j_2 \in R, j_3 \in T_3, \ldots, j_k \in T_k} A^{(k)}_{j^{(1)} j_2 \ldots j_k} \neq \sum_{j_2 \in R, j_3 \in T_3, \ldots, j_k \in T_k} A^{(k)}_{j^{(2)} j_2 \ldots j_k}.$$

However, since $(\mathcal{P}', \mathcal{Q}')$ refines $(\mathcal{P}, \mathcal{Q})$, we can write

$$\begin{aligned}
\sum_{j_2 \in R, j_3 \in T_3, \ldots, j_k \in T_k} A^{(k)}_{j^{(1)} j_2 \ldots j_k} &= \sum_{\substack{R', T_3', \ldots, T_k' \in \mathcal{Q}' \\ R' \subseteq R, T_3' \subseteq T_3, \ldots, T_k' \subseteq T_k}} \sum_{j_2 \in R', j_3 \in T_3', \ldots, j_k \in T_k'} A^{(k)}_{j^{(1)} j_2 \ldots j_k} \\
&= \sum_{\substack{R', T_3', \ldots, T_k' \in \mathcal{Q}' \\ R' \subseteq R, T_3' \subseteq T_3, \ldots, T_k' \subseteq T_k}} \sum_{j_2 \in R', j_3 \in T_3', \ldots, j_k \in T_k'} A^{(k)}_{j^{(2)} j_2 \ldots j_k} \\
&= \sum_{j_2 \in R, j_3 \in T_3, \ldots, j_k \in T_k} A^{(k)}_{j^{(2)} j_2 \ldots j_k}
\end{aligned}$$

due to equitability of $(\mathcal{Q}', \ldots, \mathcal{Q}')$ on $A^{(k)}$. Hence, $j^{(1)}$ and $j^{(2)}$ are not split by this refinement operation. Similar logic holds for the other refinement operations. □

Lemma 3. *The coarsest reduction coloring $(\mathcal{P}, \mathcal{Q})$ of any convex polynomial optimization problem is unique.*

Proof. Since it is obvious that only a finite set of distinct colors can appear in S_{refine} more than once, S_{refine} is always a sufficient refining color set by Lemma 1, and only finitely many distinct colors can exist in $(\mathcal{P}, \mathcal{Q})$, our color refinement algorithm eventually converges to a reduction coloring, which is refined by every other reduction coloring according to Lemma 2. Clearly, there cannot exist two coarsest reduction colorings with one refining the other. □

4 Experiments

4.1 Dataset

Our dataset is QPLIB, a library of 453 quadratic programs [6]. Excluding non-convex and/or LCQP instances, there are 27 convex quadratically constrained linear programs (QCLPs).

4.2 Results

We run experiments on small QCLPs from QPLIB, rounding coefficients and bounds to various powers of 10. Some linear and concave quadratic constraints in QPLIB problems have lower bounds, which do not appear in our original format for convex polynomial optimization problems. We negate these constraints to ensure that they are bounded above, not below. Additionally, some linear constraints have both lower bounds and upper bounds. We treat each of these as two separate constraints.

For the QCLPs we study, preprocessing consists of the following:

- All coefficients are rounded to a certain power of ten.
- A synthetic QCQP is generated from the QCLP by setting $A^{(2)} = 2I$; this way, the quadratic term in the objective is simply $x^\top x$.
- In cases where rounding makes individual constraints infeasible, we generate both a QCLP and its associated QCQP by removing all infeasible constraints.

All QCLPs and QCQPs were modeled in CVXPY [4] and solved using the ECOS solver [5]. Reduction was implemented in C++. Figures were generated with Matplotlib [9]. Runtime with dimension reduction (the sum of the time taken to reduce and the time taken to solve the reduced problem) as a percentage of runtime on the original problem is visualized below for each instance. Full results are provided in Appendix A.

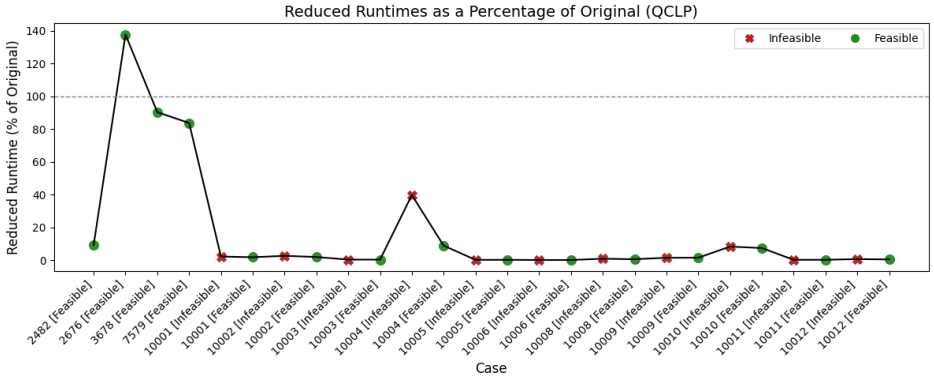

Figure 1: Percentage of original runtime after dimension reduction (QCLPs).

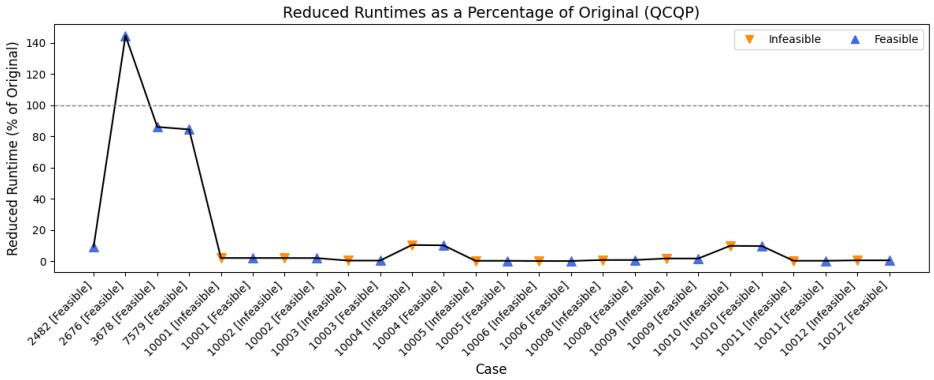

Figure 2: Percentage of original runtime after dimension reduction (QCQPs).

5 Future Work

Several directions remain open for future investigation.

5.1 Affine Transformations for Coarser Colorings

Among the most restrictive conditions for a reduction coloring are the requirements that (i) (l_j, u_j) is equal for all j in the same $T \in \mathcal{Q}$; (ii) for convex polynomial optimization, that $A_j^{(1)}$ is equal for all j in the same $T \in \mathcal{Q}$. However, prior to applying reduction, we can scale (multiply by some constant) all x_j so that all nonzero $A_j^{(1)}$ take the same value. Alternatively, we can apply some affine transformation to x so that (l_j, u_j) is constant across all j (we assume that $l_j \neq u_j$ for all j; otherwise, we can treat all "fixed" x_j as coefficients). For polynomial cases, we can do both. We show the order of transformations does not matter.

Suppose we apply equal $A_j^{(1)}$ first. Here, we assume that $A_j^{(1)} > 0$ for all j. We ignore j where $A_j^{(1)} = 0$ as we can clearly transform these x_j to any (l_j, u_j), and because these x_j cannot be grouped with other x_j with $A_j^{(1)} \neq 0$ due to our requirement that $\mathcal{Q}$ is equitable on $A^{(1)}$. Additionally, we can simply negate x_j where $A_j^{(1)} < 0$ as needed.

Let C denote the value we aim to scale each $A_j^{(1)}$ to (not to be confused with a color C in a graph coloring, from Section 2). Then we have

$$x_j \mapsto \frac{\left|A_j^{(1)}\right|}{C} x_j.$$

We can still add any constant term to each x_j to try and set as many (l_j, u_j) as possible equal, though we are unable to change the size of each pair of bounds, $u_j - l_j$. Hence, this is equivalent to two x_j being assigned the same $T \in \mathcal{Q}$ only if they have equal $\frac{\left|A_j^{(1)}\right|(u_j - l_j)}{C}$, which is equivalent to equal $\left|A_j^{(1)}(u_j - l_j)\right|$.

Alternatively, suppose we apply equal (l_j, u_j) first, and we aim to change every bound to $L \leq x_j \leq U$. We scale using

$$x_j \mapsto \frac{U - L}{u_j - l_j} x_j + \frac{L u_j - U l_j}{u_j - l_j},$$

which also changes each $A_j^{(1)}$:

$$\left|A_j^{(1)}\right| \mapsto \frac{\left|A_j^{(1)}\right|(u_j - l_j)}{U - L}.$$

This means that two x_j are assigned the same $T \in \mathcal{Q}$ only if they have the same $\left|A_j^{(1)}(u_j - l_j)\right|$.

These two are the same condition, implying that it does not matter whether we scale to match $A_j^{(1)}$ first or to match (l_j, u_j) first. For simplicity, we can scale each x_j so that all (l_j, u_j) are equal, and this method applies for both polynomial and general convex optimization problems.

We can also scale each constraint so that all positive b_i have the same value and all negative b_i have the same value, though we note that this must occur after affine transformations of x as those transformations change b by introducing constant terms into each constraint.

We now show that these methods preserve or coarsen the coarsest reduction coloring $(\mathcal{P}, \mathcal{Q})$ by proving that $(\mathcal{P}, \mathcal{Q})$ still satisfies all of our reduction coloring conditions after affine transformations. Equal b_i within each $S \in \mathcal{P}$ and (l_j, u_j) within each $T \in \mathcal{Q}$ are trivially satisfied, so it suffices to show that $\frac{\partial F}{\partial x_j}\Big|_{x=\hat{x}}$ remains equal across all $j \in T$ for any $T \in \mathcal{Q}$, and that $G_i(\hat{x})$ remains constant across all $i \in S$ for any $S \in \mathcal{P}$.

Note that since transformations of x_j are solely determined by (l_j, u_j), which is guaranteed to be constant within each $T \in \mathcal{Q}$, x_j is transformed the same way for each $j \in T$. Denote the transformation for each $T \in \mathcal{Q}$ as $\mathcal{T}_T$, and denote by $\mathcal{T}$ the transformation of x given by applying $\mathcal{T}_T$ to each x_j for all $j \in T$ and $T \in \mathcal{Q}$.

Let $\overline{x} = \mathcal{T}(\hat{x})$ denote some $\hat{x}$ after transformation. Then the new objective function is

$$F\left(\mathcal{T}^{-1}(\overline{x})\right),$$

and we have that

$$\begin{aligned}\frac{\partial}{\partial \overline{x}_j} F\left(\mathcal{T}^{-1}(\overline{x})\right) &= \sum_{j_2=1}^{n} \left(\left.\frac{\partial F}{\partial x_{j_2}}\right|_{x=\mathcal{T}^{-1}(\overline{x})}\right)\left(\frac{\partial \mathcal{T}_{j_2}^{-1}(\overline{x})}{\partial \overline{x}_j}\right) \\ &= \left(\left.\frac{\partial F}{\partial x_j}\right|_{x=\mathcal{T}^{-1}(\overline{x})}\right)\left(\frac{d\mathcal{T}_j^{-1}(\overline{x})}{d\overline{x}_j}\right) \\ &= \left(\left.\frac{\partial F}{\partial x_j}\right|_{x=\mathcal{T}^{-1}(\overline{x})}\right)\left(\frac{d\mathcal{T}_T^{-1}(\overline{x}_j)}{d\overline{x}_j}\right)\end{aligned}$$

if $j \in T$. But $\mathcal{T}_T^{-1}$ is clearly also an affine transformation, so $\frac{d\mathcal{T}_T^{-1}(\overline{x}_j)}{d\overline{x}_j}$ is a constant that depends only on T, not j. Similarly, we note that $\left.\frac{\partial F}{\partial x_j}\right|_{x=\mathcal{T}^{-1}(\overline{x})} = \left.\frac{\partial F}{\partial x_j}\right|_{x=\hat{x}}$ is equal across all $j \in T$ by assumption. Hence, $\left.\frac{\partial F}{\partial x_j}\right|_{x=\hat{x}}$ remains equal across all $j \in T$.

For equal $G_i(\hat{x})$, note that each $\mathcal{T}_T$ is bijective. Hence, $\overline{x}_j$ is equal for all $j \in T$ given some $T \in \mathcal{Q}$ if and only if x satisfies the same condition. Since we are given that x satisfies equal $G_i(\hat{x})$, $\overline{x}$ must too, finishing our proof that the coarsest reduction coloring can only get coarser, not finer, after we apply affine transformations.

While these methods can lead to a coarser coloring in theory, they may not be optimal in practice for several reasons.

Firstly, these transformations can inadvertently destroy sparsity. As seen in Table 1, in the QCLPs from QPLIB that we used, the vast majority of entries (coefficients or bounds) take on some default values.

Table 1: Sparsity of QCLP instances

Problem number	**Entries**	**Non-default entries**	**Percentage non-default**
7579	27,452,404	11,004	$\left(4.0 \times 10^{-2}\right)\%$
3678	217,804,804	42,004	$\left(1.9 \times 10^{-2}\right)\%$
2482	10,433,232,689	12,196	$\left(1.2 \times 10^{-4}\right)\%$
3088	83,791,398,183	22,450	$\left(2.7 \times 10^{-5}\right)\%$
2784	167,376,548,283	25,150	$\left(1.5 \times 10^{-5}\right)\%$
2519	197,872,438,713	33,280	$\left(1.7 \times 10^{-5}\right)\%$

However, transformations aiming for a coarser coloring introduce dense non-default values, significantly increasing preprocessing time. Additionally, it may be expensive to compute how F and G_i change.

Despite these tradeoffs, potential directions for future research include the following:

- Aim for a balance between sparsity and modifying the data to make it easier for $(\mathcal{P}, \mathcal{Q})$ to satisfy the equal (l_j, u_j) and equal b_i conditions.
- Study non-affine transformations of each x_j.
- For polynomial cases, scale constraints with $b_i = 0$, aiming to make it easier to satisfy equitability on $P^{(k)}$, and/or scale constraints with $b_i \neq 0$, balancing equitability on $P^{(k)}$, the equal b_i condition, and sparsity.
- For polynomial cases, scale variables to promote equitability on $A^{(k)}$ and $P^{(k)}$. In order to be effective, this method will necessarily interfere with our existing transformations of x to aim for equal (l_j, u_j), so also study how to balance equitability and equal (l_j, u_j).

5.2 Other Directions

- Extend color refinement, which we have established for convex polynomial optimization problems, to certain non-polynomial cases.
- Improve the efficiency of color refinement for convex polynomial optimization, aiming to avoid an exponential time complexity in the polynomial degree.

6 Conclusion

We generalized color refinement techniques for dimension reduction to all smooth convex optimization. We built on prior methods for LPs and convex LCQPs, proving a reduction theorem that holds under a set of coloring conditions. We also proposed a modified color refinement algorithm for polynomial cases capable of refining double colorings $(\mathcal{P}, \mathcal{Q})$ while preserving key convexity, feasibility, and optimality properties. We proved the uniqueness of the coarsest coloring in this setting and demonstrated how our procedure converges to it. Experiments on QCLPs from QPLIB and synthetic QCQPs confirmed that our approach yields significant dimension reduction, particularly under rounding. We also proposed several directions for future work, including variable and constraint scaling heuristics and non-affine transformations of variables. Ultimately, our method provides a computationally effective preprocessing procedure that can exploit symmetries in convex optimization.

7 Acknowledgments

I would like to thank the MIT Program for Research in Mathematics, Engineering, and Science (PRIMES) for making this project possible. I am also grateful to my mentor, Ziang Chen, for his extensive guidance and help.

References

[1] A. A. Ahmadi. ORF 523 lecture 9. Lecture notes, Princeton University. Available at `https://www.princeton.edu/~aaa/Public/Teaching/ORF523/ORF523_Lec9.pdf`.

[2] C. Berkholz, P. Bonsma, and M. Grohe. Tight lower and upper bounds for the complexity of canonical colour refinement. Theory of Computing Systems, vol. 60, pages 581–614, 2017, DOI:10.1007/s00224-016-9686-0.

[3] Z. Chen, X. Chen, J. Liu, X. Wang, and W. Yin. Expressive power of graph neural networks for (mixed-integer) quadratic programs. Poster presented at the International Conference on Machine Learning, 2025.

[4] S. Diamond and S. Boyd. CVXPY: A Python-embedded modeling language for convex optimization. Journal of Machine Learning Research, vol. 17, no. 83, pages 1-5, 2016.

[5] A. Domahidi, E. Chu, and S. Boyd. ECOS: An SOCP solver for embedded systems. European Control Conference, pages 3071-3076, 2013.

[6] F. Furini, E. Traversi, P. Belotti, A. Frangioni, A. Gleixner, N. Gould, L. Liberti, A. Lodi, R. Misener, H. Mittelmann, N. Sahinidis, S. Vigerske, and A. Wiegele. QPLIB: A library of quadratic programming instances. Mathematical Programming Computation, 2018, DOI:10.1007/s12532-018-0147-4.

[7] D. Goldfarb and S. Liu. An $O(n^3L)$ primal interior point algorithm for convex quadratic programming. Mathematical Programming, vol. 49, pages 325–340, 1990, DOI:10.1007/BF01588795.

[8] M. Grohe, K. Kersting, M. Mladenov, and E. Selman. Dimension reduction via colour refinement. Algorithms - ESA 2014, vol. 8737, pages 505-516, 2014, DOI:10.1007/978-3-662-44777-2_42.

[9] J. D. Hunter. Matplotlib: A 2D graphics environment. Computing in Science & Engineering, vol. 9, no. 3, pages 90-95, 2007, DOI:10.1109/MCSE.2007.55.

[10] V. Klee and G. Minty. How good is the simplex algorithm? Inequalities III, pages 159–175, 1972.

[11] P. D. Loewen. Convex analysis with applications. Lecture notes, University of British Columbia, 2001. Available at `https://www.math.ubc.ca/~loew/m604/web-ho/convex.pdf`.

[12] C. M. Maes. A regularized active-set method for sparse convex quadratic programming. PhD thesis, Stanford University, 2010. Available at `https://web.stanford.edu/group/SOL/dissertations/maes-thesis.pdf`.

[13] M. Mladenov, L. Kleinhans, and K. Kersting. Lifted inference for convex quadratic programs. Proceedings of the AAAI Conference on Artificial Intelligence, vol. 31, no. 1, 2017.

A Full Results

Table 2 provides the full runtime impact of dimension reduction across all QCLPs and QCQPs studied. Runtime is given in seconds.

Table 2: Runtime impact of dimension reduction. Feas. = feasibility/feasible; Infeas. = infeasible; Orig. = original problem; Red. = reduction time; Red.Opt. = reduced optimizer time; Orig. = original optimizer time; Tot. = Red. + Red.Opt.; %Orig. = Tot./Orig. $\times 100$. Feasibility does not change the reduction time.

Case	Round	Class	Feas.	Red.	Red.Opt.	Orig.	Tot.	%Orig.
2482	10^{-1}	QCLP	Feas. (Orig.)	93.00	0.01	1006.03	93.01	9.245
2482	10^{-1}	QCQP	Feas.	92.92	0.01	990.44	92.93	9.383
2676	10^{-1}	QCLP	Feas. (Orig.)	245.84	351.84	434.07	597.68	137.69
2676	10^{-1}	QCQP	Feas.	273.46	342.95	426.05	616.41	144.68
3678	10^{-1}	QCLP	Feas. (Orig.)	4.92	4.57	10.76	9.49	88.2
3678	10^{-1}	QCQP	Feas.	4.92	4.40	10.83	9.32	86.1
7579	10^{-1}	QCLP	Feas. (Orig.)	0.54	0.49	1.28	1.03	80.5
7579	10^{-1}	QCQP	Feas.	0.55	0.56	1.31	1.11	84.7
10001	10^{1}	QCLP	Infeas. (Orig.)	0.05	0.01	2.61	0.06	2
			Feas.		0.01	3.46	0.06	2
10001	10^{1}	QCQP	Infeas.	0.06	0.01	3.48	0.07	2
			Feas.		0.01	3.50	0.07	2
10002	10^{1}	QCLP	Infeas. (Orig.)	0.06	0.01	2.59	0.07	3
			Feas.		0.01	3.51	0.07	2
10002	10^{1}	QCQP	Infeas.	0.06	0.01	3.46	0.07	2
			Feas.		0.01	3.49	0.07	2
10003	10^{1}	QCLP	Infeas. (Orig.)	0.15	0.01	53.72	0.16	0.30
			Feas.		0.01	58.31	0.16	0.27
10003	10^{1}	QCQP	Infeas.	0.18	0.01	58.13	0.19	0.33
			Feas.		0.01	57.91	0.19	0.33
10004	10^{1}	QCLP	Infeas. (Orig.)	0.06	0.01	0.17	0.07	40
			Feas.		0.01	0.75	0.07	10
10004	10^{1}	QCQP	Infeas.	0.07	0.01	0.76	0.08	10
			Feas.		0.01	0.77	0.08	10
10005	10^{1}	QCLP	Infeas. (Orig.)	0.50	0.01	330.02	0.51	0.15
			Feas.		0.01	339.16	0.51	0.15
10005	10^{1}	QCQP	Infeas.	0.60	0.01	334.41	0.61	0.18
			Feas.		0.01	332.42	0.61	0.18
10006	10^{1}	QCLP	Infeas. (Orig.)	1.53	0.01	2820.49	1.54	0.0546
			Feas.		0.01	2951.25	1.54	0.0522
10006	10^{1}	QCQP	Infeas.	1.78	0.01	2898.90	1.79	0.0617
			Feas.		0.01	2838.59	1.79	0.0631
10008	10^{1}	QCLP	Infeas. (Orig.)	0.09	0.01	11.55	0.10	0.87
			Feas.		0.01	17.95	0.10	0.56
10008	10^{1}	QCQP	Infeas.	0.11	0.01	16.86	0.12	0.71
			Feas.		0.01	16.82	0.12	0.71
10009	10^{1}	QCLP	Infeas. (Orig.)	0.06	0.01	4.69	0.07	1
			Feas.		0.01	4.64	0.07	2
10009	10^{1}	QCQP	Infeas.	0.07	0.01	4.62	0.08	2
			Feas.		0.01	4.70	0.08	2
10010	10^{1}	QCLP	Infeas. (Orig.)	0.03	0.01	0.45	0.04	9
			Feas.		0.01	0.50	0.04	8
10010	10^{1}	QCQP	Infeas.	0.04	0.01	0.50	0.05	10
			Feas.		0.01	0.50	0.05	10
10011	10^{1}	QCLP	Infeas. (Orig.)	0.21	0.01	125.02	0.22	0.18
			Feas.		0.01	124.08	0.22	0.18
10011	10^{1}	QCQP	Infeas.	0.25	0.01	123.03	0.26	0.21
			Feas.		0.01	122.61	0.26	0.21
10012	10^{1}	QCLP	Infeas. (Orig.)	0.12	0.01	21.24	0.13	0.60
			Feas.		0.01	30.95	0.13	0.42
10012	10^{1}	QCQP	Infeas.	0.14	0.01	30.81	0.15	0.49
			Feas.		0.01	30.81	0.15	0.49