RiNNAL+: a Riemannian ALM Solver for SDP-RLT Relaxations of Mixed-Binary QP

Di Hou

Department of Mathematics, National University of Singapore

Joint work with Tianyun Tang (NUS), Kim-Chuan Toh (NUS)

ICCOPT, July 22, 2025

Outline

1. Problem: SDP-RLT relaxations of mixed-binary QP
2. RiNNAL+: Riemannian ALM solver
3. Numerical experiments

Mixed-binary QP

$$\begin{aligned} \min_{x} \quad & x^\top Q x + 2c^\top x \\ \text{s.t.} \quad & Ax = b,\ Bx \le d,\ x \in \{0,1\}^n. \end{aligned} \tag{MBQP}$$

- **Applications:** Max-Cut, clutering, portfolio optimization ...
- **Generalization:** mixed-binary, QCQP.
- **Global optimality:** solve convex relaxations to obtain lower bound.
- **Challenge:** tradeoff between tightness and computational cost.

SDP Relaxation

$$\begin{aligned} \min_{x} \quad & x^\top Q x + 2c^\top x \\ \text{s.t.} \quad & Ax = b,\ Bx \le d,\ x \in \{0,1\}^n. \end{aligned} \tag{MBQP}$$

$$\begin{aligned} \min_{X,\, x} \quad & \langle Q, X \rangle + 2c^\top x \\ \text{s.t.} \quad & Ax = b,\ Bx \le d,\ \operatorname{diag}(X) = x, \\ & \begin{bmatrix} 1 & x^\top \\ x & X \end{bmatrix} \succeq 0. \end{aligned} \tag{SDP}$$

Relaxation	Size	Bound	Speed
SDP	$\Omega(n^2)$	Loose	Fast

DNN Relaxation

$$\begin{aligned} \min_{x} \quad & x^\top Q x + 2c^\top x \\ \text{s.t.} \quad & Ax = b,\ Bx + s = d,\ x \in \{0,1\}^n,\ (x, s) \geq 0. \end{aligned} \tag{MBQP}$$

$$\begin{aligned} \min_{X,\, x} \quad & \langle Q, X \rangle + 2c^\top x \\ \text{s.t.} \quad & \begin{bmatrix} -b & A & 0 \\ -d & B & I \end{bmatrix} \widehat{X} = 0,\ \operatorname{diag}(X) = x, \\ & \widehat{X} = \begin{bmatrix} 1 & x^\top & s^\top \\ x & X & Z^\top \\ s & Z & W \end{bmatrix} \succeq 0 \text{ and } \geq 0. \end{aligned} \tag{DNN}$$

Relaxation	**Size**	**Bound**	**Speed**
SDP	$\Omega(n^2)$	Loose	Fast
DNN[1]	$\Omega((n+\ell)^2)$	Tight	Slow

[1] Burer, 2010.

SDP-RLT Relaxation

$$
\begin{aligned}
\min_{X,\,x} \quad & \langle Q, X\rangle + 2c^\top x \\
\text{s.t.} \quad & Ax = b,\ Bx \le d,\ \mathrm{diag}(X) = x, \\
& AX - bx^\top = 0,\ X \ge 0, \\
& dx^\top - BX \ge 0, \\
& BXB^\top - Bxd^\top - dx^\top B^\top + dd^\top \ge 0, \\
& \begin{bmatrix} 1 & x^\top \\ x & X \end{bmatrix} \succeq 0.
\end{aligned}
\qquad \text{(SDP-RLT)}
$$

Relaxation	Size	Bound	Speed
SDP	$\Omega(n^2)$	Loose	Fast
DNN[1]	$\Omega((n+\ell)^2)$	Tight	Slow
SDP-RLT[2]	$\Omega(n^2)$	Tight	Fast

[1] Burer, 2010.

[2] Sherali and Adams, 1990.

Tightness Comparison

Theorem

The DNN and SDP-RLT relaxations of (MBQP) yield the same optimal value.

Tightness Comparison

Theorem

The DNN and SDP-RLT relaxations of (MBQP) yield the same optimal value.

Tightening Strategy:

$$\min\left\{x^\top Qx + 2c^\top x : \; x \in \{0,1\}^n\right\} \tag{BIQ}$$

Strengthen by adding redundant bound constraint:

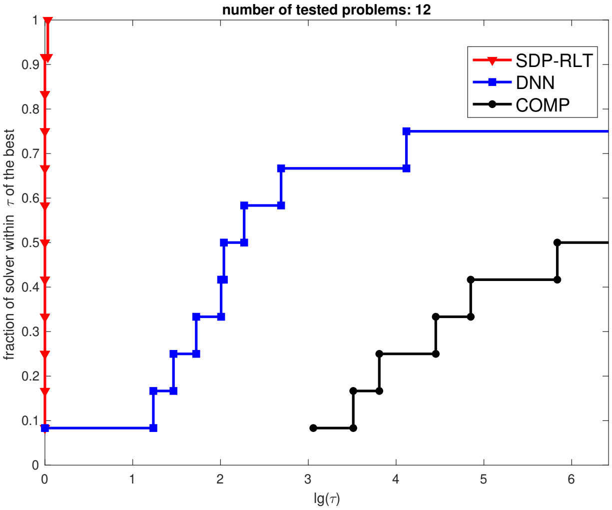
$$\min\left\{x^\top Qx + 2c^\top x : \; {\color{red} x \le e}, \; x \in \{0,1\}^n\right\} \tag{BIQ-S}$$

Data: bqp100.1	(BIQ)		(BIQ-S)	
UB: -7.97e3	lower bound	% gap	lower bound	% gap
(SDP)	-8.72e3	9.424	-8.72e3	9.424
(DNN)	-8.38e3	5.149	-8.04e3	0.836
(SDP-RLT)	-8.38e3	5.149	-8.04e3	0.836

Tighter

Time Comparison

- Solving SDP-RLT is faster due to lower dimensionality.



How to Solve SDP-RLT?

Conclusion: SDP-RLT delivers the same relaxation bound as the DNN approach but with substantially lower computational cost.

How to Solve SDP-RLT?

Conclusion: SDP-RLT delivers the same relaxation bound as the DNN approach but with substantially lower computational cost.

Challenges:

- $\Omega(n^2)$ matrix variables.
- $\Omega(nm)$ equality constraints from $Ax = b$.
- $\Omega(n\ell)$ inequality constraints from $Bx \leq d$.
- $\Omega(n^2)$ nonnegativity constraints $X \geq 0$.

How to Solve SDP-RLT?

Conclusion: SDP-RLT delivers the same relaxation bound as the DNN approach but with substantially lower computational cost.

Challenges:

- $\Omega(n^2)$ matrix variables.
- $\Omega(nm)$ equality constraints from $Ax = b$.
- $\Omega(n\ell)$ inequality constraints from $Bx \leq d$.
- $\Omega(n^2)$ nonnegativity constraints $X \geq 0$.

Limitations of existing solvers:

- Interior-point methods: $\mathcal{O}(n^6)$ per iteration.
- ADMM/ALM: costly eigenvalue decompositions.
- RiNNAL: applicable to DNN, not to SDP–RLT.

How to Solve SDP-RLT?

Conclusion: SDP-RLT delivers the same relaxation bound as the DNN approach but with substantially lower computational cost.

Challenges:

- $\Omega(n^2)$ matrix variables.
- $\Omega(nm)$ equality constraints from $Ax = b$.
- $\Omega(n\ell)$ inequality constraints from $Bx \leq d$.
- $\Omega(n^2)$ nonnegativity constraints $X \geq 0$.

Limitations of existing solvers:

- Interior-point methods: $\mathcal{O}(n^6)$ per iteration.
- ADMM/ALM: costly eigenvalue decompositions.
- RiNNAL: applicable to DNN, not to SDP–RLT.

Contribution:

- RiNNAL+: a two-phase Riemannian ALM method.
- Exploits low-rank structure and handles many constraints efficiently.

SDP-RLT Relaxation

Compact form

$$\begin{aligned} \min_{Y} \quad & \langle C, Y \rangle \\ \text{s.t.} \quad & Y \in \mathcal{I} \cap \mathcal{E} \cap \mathbb{S}_{+}^{n+1}. \end{aligned} \tag{SDP-RLT}$$

Variables and data:

$$C := \begin{bmatrix} 0 & c^\top \\ c & Q \end{bmatrix}, \quad Y = \begin{bmatrix} z & x^\top \\ x & X \end{bmatrix}.$$

Feasible set:

$$\text{(Equalities)} \quad \mathcal{E} := \left\{ Y \in \mathbb{S}^{n+1} : \ \mathcal{A}(X) = g \right\},$$

$$\text{(Inequalities)} \quad \mathcal{I} := \left\{ Y \in \mathbb{S}^{n+1} : \ \mathcal{B}(X) \geq h \right\}.$$

Augmented Lagrangian Method for (SDP-RLT)

Reformulation:

$$\begin{aligned} \min_{Y} \quad & \langle C, Y \rangle + \delta_{\mathcal{E} \cap \mathbb{S}^{n+1}_{+}}(Y) \\ \text{s.t.} \quad & \mathcal{B}(Y) \geq h. \end{aligned}$$

Augmented Lagrangian Method for (SDP-RLT)

Reformulation:

$$\begin{aligned} \min_{Y} \quad & \langle C, Y \rangle + \delta_{\mathcal{E}\cap\mathbb{S}^{n+1}_{+}}(Y) \\ \text{s.t.} \quad & \mathcal{B}(Y) \geq h. \end{aligned}$$

Augmented Lagrangian function:

$$L_\sigma(Y;\lambda) := \langle C, Y \rangle + \frac{\sigma}{2}\|\sigma^{-1}\lambda - (\mathcal{B}(Y) - h)\|^2.$$

Augmented Lagrangian Method for (SDP-RLT)

Reformulation:

$$\begin{aligned} \min_{Y} \quad & \langle C, Y \rangle + \delta_{\mathcal{E}\cap\mathbb{S}_+^{n+1}}(Y) \\ \text{s.t.} \quad & \mathcal{B}(Y) \geq h. \end{aligned}$$

Augmented Lagrangian function:

$$L_\sigma(Y;\lambda) := \langle C, Y \rangle + \frac{\sigma}{2}\|\sigma^{-1}\lambda - (\mathcal{B}(Y) - h)\|^2.$$

ALM Iteration

$$Y^{k+1} = \arg\min\left\{\phi(Y) := L_{\sigma_k}(Y;\lambda^k) \;\middle|\; Y \in \mathcal{E} \cap \mathbb{S}_+^{n+1}\right\}, \qquad \text{(ALM-sub)}$$

$$\lambda^{k+1} = \lambda^k - \sigma_k(\mathcal{B}(Y^{k+1}) - g).$$

Challenge: How to solve (ALM-sub) efficiently?

A Hybrid Method to Solve ALM Subproblem

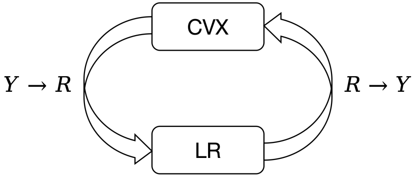


Figure: Two-phase transition.

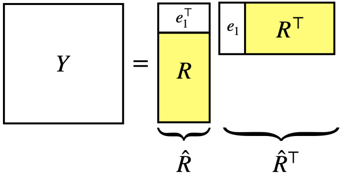


Figure: Low-Rank factorization.

A Hybrid Method to Solve ALM Subproblem

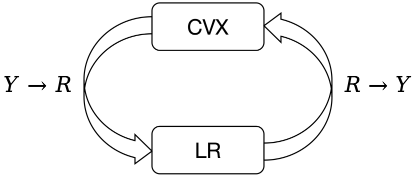


Figure: Two-phase transition.

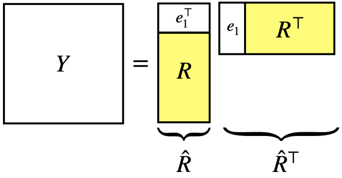


Figure: Low-Rank factorization.

CVX phase

Projected gradient descent on Y

$$\begin{aligned} \min_{Y} \quad & \phi(Y) \\ \text{s.t.} \quad & Y \in \mathcal{E} \cap \mathbb{S}_+^{n+1}. \end{aligned}$$

- Ensures global convergence.
- Automatically adjusts the rank.

LR phase

Riemannian gradient descent on R

$$\begin{aligned} \min_{R} \quad & \phi(\widehat{R}\,\widehat{R}^\top) \\ \text{s.t.} \quad & \widehat{R}\,\widehat{R}^\top \in \mathcal{E}. \end{aligned}$$

- Reduces dim and #constraints.
- Reduces the objective value.

CVX Phase: Computation

Perform one PG step: (Lee et al., 2024)

$$\begin{aligned} G &= Y - t\nabla\phi(Y), \\ Y^+ &= \Pi_{\mathcal{E}\cap\mathbb{S}_+^{n+1}}(G). \end{aligned}$$

CVX Phase: Computation

Perform one PG step: (Lee et al., 2024)

$$\begin{aligned} G &= Y - t\nabla\phi(Y), \\ Y^+ &= \Pi_{\mathcal{E}\cap\mathbb{S}^{n+1}_+}(G). \end{aligned}$$

Projection subproblem: Generalized nearest correlation matrix problem:

$$Y^+ = \operatorname{argmin}\left\{\frac{1}{2}\|Y - G\|^2 :\ \operatorname{diag}(Y) = e,\ \langle NN^\top, Y\rangle = 0,\ Y \succeq 0\right\}, \qquad \text{(Proj-sub)}$$

efficiently solved via Semismooth Newton-CG method (Qi and Sun, 2006).

CVX Phase: Computation

Perform one PG step: (Lee et al., 2024)

$$\begin{aligned} G &= Y - t\nabla\phi(Y), \\ Y^+ &= \Pi_{\mathcal{E}\cap\mathbb{S}_+^{n+1}}(G). \end{aligned}$$

Projection subproblem: Generalized nearest correlation matrix problem:

$$Y^+ = \operatorname{argmin}\left\{\frac{1}{2}\|Y - G\|^2 : \ \operatorname{diag}(Y) = e, \ \langle NN^\top, Y\rangle = 0, \ Y \succeq 0\right\}, \qquad \text{(Proj-sub)}$$

efficiently solved via Semismooth Newton-CG method (Qi and Sun, 2006).

Acceleration: (10+ times faster)

- Preprocessing: transform $\operatorname{diag}(X) = x$ to $\operatorname{diag}(Y) = e$.
- Warm-start: recover dual variables of (Proj-sub) from LR phase.

CVX Phase: Auto-rank

Rank-adaptive strategy:

- Rank Increase: Escape saddle points by following the eigenvector direction with the smallest eigenvalue. (Journée et al., 2010)
- Rank Decrease: Reduce computational cost by truncating small singular values. (Gao and Absil, 2022)
- Frequently used: (Boumal, 2015; J. Wang and Hu, 2025; Y. Wang et al., 2023)

CVX Phase: Auto-rank

Rank-adaptive strategy:

- Rank Increase: Escape saddle points by following the eigenvector direction with the smallest eigenvalue. (Journée et al., 2010)
- Rank Decrease: Reduce computational cost by truncating small singular values. (Gao and Absil, 2022)
- Frequently used: (Boumal, 2015; J. Wang and Hu, 2025; Y. Wang et al., 2023)

Issues:

- Requires careful parameter tuning (step sizes, thresholds).
- Truncating small singular values may increase the function value.

CVX Phase: Auto-rank

Rank-adaptive strategy:

- Rank Increase: Escape saddle points by following the eigenvector direction with the smallest eigenvalue. (Journée et al., 2010)
- Rank Decrease: Reduce computational cost by truncating small singular values. (Gao and Absil, 2022)
- Frequently used: (Boumal, 2015; J. Wang and Hu, 2025; Y. Wang et al., 2023)

Issues:

- Requires careful parameter tuning (step sizes, thresholds).
- Truncating small singular values may increase the function value.

PG Step:

- Automatically adjust the rank.
- parameter-free.
- monotonically decreasing.

CVX Phase: Auto-rank

BIQ experiment:

- Data: bqp500.1.
- observation: PG identifies the correct rank rapidly in both scenarios.

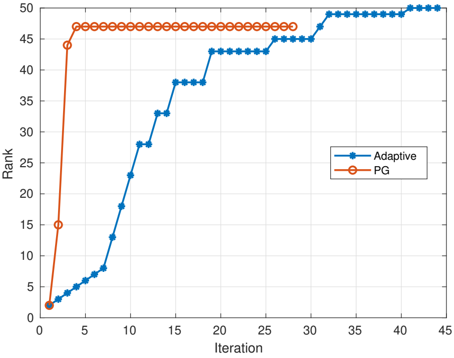


Figure: Initial rank: $r = 2$.

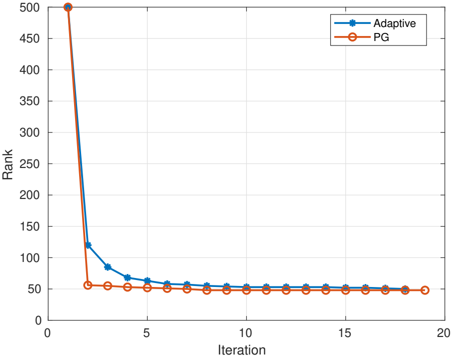


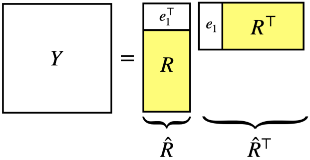
Figure: Initial rank: $r = n$.

LR Phase: RiNNAL Framework

Manifold:

$$Y \in \mathcal{E} \cap \mathbb{S}_+^{n+1} \iff R \in \mathcal{M}_r,$$

$$\color{red}{\mathcal{M}_r} := \left\{ R \in \mathbb{R}^{n \times r} : AR = be_1^\top,\ \mathrm{diag}(RR^\top) = Re_1 \right\}.$$

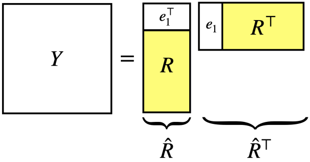


LR Phase: RiNNAL Framework

Manifold:

$$Y \in \mathcal{E} \cap \mathbb{S}_+^{n+1} \iff R \in \mathcal{M}_r,$$

$$\color{red}{\mathcal{M}_r} := \left\{ R \in \mathbb{R}^{n \times r} : AR = be_1^\top,\ \mathrm{diag}(RR^\top) = Re_1 \right\}.$$



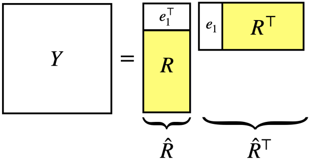
- Dim: Reduce $\mathcal{O}(n^2) \to \mathcal{O}(nr)$.
- Constraints: Reduce $\Omega(mn) \to \Omega(mr)$.

LR Phase: RiNNAL Framework

Manifold:

$$Y \in \mathcal{E} \cap \mathbb{S}_+^{n+1} \iff R \in \mathcal{M}_r,$$

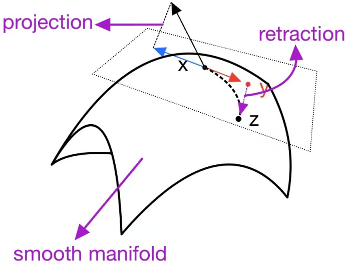
$$\mathcal{M}_r := \left\{ R \in \mathbb{R}^{n \times r} : AR = be_1^\top,\ \mathrm{diag}(RR^\top) = Re_1 \right\}.$$



- Dim: Reduce $\mathcal{O}(n^2) \to \mathcal{O}(nr)$.
- Constraints: Reduce $\Omega(mn) \to \Omega(mr)$.

Properties:

- Smoothness: Random perturbation.
- Projection: Schur-based reduction.
- Retraction: Convex reformulation (solved by Weiszfeld algorithm).
- Dual recovery: (LR) KKT $\Rightarrow$ partial (CVX) KKT.



LR Phase: Smoothness

Spherical formulation of $\mathcal{M}_r$:

$$\mathcal{M}_r \ = \ \Big\{ R \in \mathbb{R}^{n\times r} : \ AR = be_1^\top, \ \mathrm{diag}_B\big((2R - ee_1^\top)(2R - ee_1^\top)^\top\big) = e \Big\}.$$

LR Phase: Smoothness

Spherical formulation of $\mathcal{M}_r$:

$$\mathcal{M}_r \ = \ \left\{ R \in \mathbb{R}^{n\times r} : \ AR = be_1^\top, \ \mathrm{diag}_B\big((2R - ee_1^\top)(2R - ee_1^\top)^\top\big) = e \right\}.$$

Perturbed manifold $\mathcal{M}_{r,v}$:

$$\mathcal{M}_{r,v} \ = \ \left\{ R \in \mathbb{R}^{n\times r} : \ AR = be_1^\top, \ \mathrm{diag}_B\big((2R - ee_1^\top)(2R - ee_1^\top)^\top\big) = e + v \right\},$$

where $v \in \mathbb{R}^p$ is random with $\|v\| = \varepsilon > 0$.

LR Phase: Smoothness

Spherical formulation of $\mathcal{M}_r$:

$$\mathcal{M}_r \; = \; \Big\{ R \in \mathbb{R}^{n \times r} : \; AR = be_1^\top, \; \mathrm{diag}_B\big((2R - ee_1^\top)(2R - ee_1^\top)^\top\big) = e \Big\}.$$

Perturbed manifold $\mathcal{M}_{r,v}$:

$$\mathcal{M}_{r,v} \; = \; \Big\{ R \in \mathbb{R}^{n \times r} : \; AR = be_1^\top, \; \mathrm{diag}_B\big((2R - ee_1^\top)(2R - ee_1^\top)^\top\big) = e + v \Big\},$$

where $v \in \mathbb{R}^p$ is random with $\|v\| = \varepsilon > 0$.

Lemma (Tang and Toh, 2024b)

For a generic v, every point of $\mathcal{M}_{r,v}$ satisfies LICQ.

Implementation: switch to $\mathcal{M}_{r,v}$ if encounter nonsmooth point.

Projection

To compute $\mathrm{grad} f_r(R)$, we need to compute projection onto $\mathrm{T}_R \mathcal{M}_r$, i.e.

$$\mathrm{grad}\, f_r(R) = \mathrm{Proj}_{\mathcal{T}_R \mathcal{M}_r}(\nabla f_r(R)) = \nabla f_r(R) - h_R^*(\lambda, \mu),$$

where $\mathrm{T}_R \mathcal{M}_r$ is the tangent space, $h_R : \mathbb{R}^{n\times r} \to \mathbb{R}^{m\times r} \times \mathbb{R}^{|B|}$ is the Fréchet differential mapping of the constraints defining $\mathcal{M}_r$ at R:

$$h_R(H) := (AH; 2\,\mathrm{diag}_B(HR^\top) - H_B e_1),$$

and $(\lambda, \mu) \in \mathbb{R}^{m\times r} \times \mathbb{R}^{|B|}$ is the solution to the following linear system:

$$h_R(h_R^*(\lambda, \mu)) = h_R(\nabla f_r(R)). \qquad \text{(Proj-sub)}$$

Projection

To compute $\mathrm{grad} f_r(R)$, we need to compute projection onto $\mathrm{T}_R\,\mathcal{M}_r$, i.e.

$$\operatorname{grad} f_r(R) = \operatorname{Proj}_{\mathcal{T}_R\mathcal{M}_r}(\nabla f_r(R)) = \nabla f_r(R) - h_R^*(\lambda, \mu),$$

where $\mathrm{T}_R\,\mathcal{M}_r$ is the tangent space, $h_R : \mathbb{R}^{n\times r} \to \mathbb{R}^{m\times r} \times \mathbb{R}^{|B|}$ is the Fréchet differential mapping of the constraints defining $\mathcal{M}_r$ at R:

$$h_R(H) := (AH; 2\,\operatorname{diag}_B(HR^\top) - H_B e_1),$$

and $(\lambda, \mu) \in \mathbb{R}^{m\times r} \times \mathbb{R}^{|B|}$ is the solution to the following linear system:

$$h_R(h_R^*(\lambda, \mu)) = h_R(\nabla f_r(R)). \tag{Proj-sub}$$

By leveraging the special structure of (Proj-sub) with Schur complement and SMW formula, we reduce the complexity for solving (Proj-sub) to

$$\mathcal{O}\left(\min\left\{|B|^3 + m^2 r + mr|B|, (mr)^2|B| + (mr)^3\right\}\right),$$

which is smaller than vanilla $\mathcal{O}\left((|B| + mr)^3\right)$ when either $|B|$ or mr is small.

Retraction

We use the following projection retraction to compute the projection onto $\mathcal{M}_r$:

$$\mathrm{Rtr}_{\bar{R}}(H) := \mathrm{Proj}_{\mathcal{M}_r}(V) = \arg\min \left\{ \|R - V\|_F^2 : \; R \in \mathcal{M}_r \right\},$$

where $\bar{R} \in \mathcal{M}_r$, $H \in \mathcal{T}_{\bar{R}}\mathcal{M}_r$ and $V := \bar{R} + H$. This problem is nonconvex, but can be transformed into a convex problem.

Retraction

We use the following projection retraction to compute the projection onto $\mathcal{M}_r$:

$$\mathrm{Rtr}_{\bar{R}}(H) := \mathrm{Proj}_{\mathcal{M}_r}(V) = \arg\min \left\{ \|R - V\|_F^2 : \ R \in \mathcal{M}_r \right\},$$

where $\bar{R} \in \mathcal{M}_r$, $H \in \mathcal{T}_{\bar{R}}\mathcal{M}_r$ and $V := \bar{R} + H$. This problem is nonconvex, but can be transformed into a convex problem.Define spherical manifold $\mathcal{B}_r$ as

$$\mathcal{B}_r := \left\{ R \in \mathbb{R}^{n\times r} : \mathrm{diag}_B(RR^\top) - R_B e_1 = 0 \right\},$$

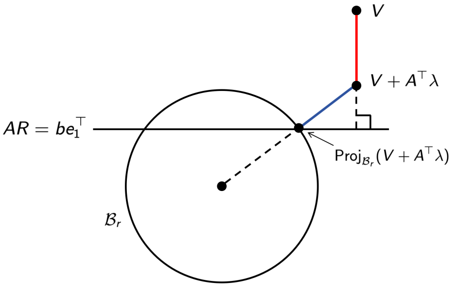
$$\mathcal{M}_r := \left\{ R \in \mathbb{R}^{n\times r} : \ AR = be_1^\top, \ \mathrm{diag}_B(RR^\top) - R_B e_1 = 0 \right\}.$$

Lemma 1. For any $V \in \mathbb{R}^{n\times r}$, if there exists $\lambda \in \mathbb{R}^{m\times r}$ such that

$$A\,\mathrm{Proj}_{\mathcal{B}_r}(V + A^\top\lambda) = be_1^\top,$$

then

$$\mathrm{Proj}_{\mathcal{M}_r}(V) = \mathrm{Proj}_{\mathcal{B}_r}(V + A^\top\lambda).$$



Retraction Subproblem

λ is solution to the following unconstrained nonsmooth convex problem:

$$\boxed{\min_{\lambda \in \mathbb{R}^{m \times r}} \sum_{i \in B} \|(V' + A^\top \lambda)_i\| + \sum_{i \in [n] \setminus B} \|(V' + A^\top \lambda)_i\|^2 + \langle b', A^\top \lambda \rangle,}$$

where $V' = V - ee_1^\top/2$, $b' \in \mathbb{R}^{m \times r}$ such that $(Ae - 2b)e_1^\top = Ab'$, and R_i denotes the i-th row of R. This is a generalization of the geometric median problem, which can be solved by the Weiszfeld algorithm (Weiszfeld 1937).

Retraction Subproblem

λ is solution to the following unconstrained nonsmooth convex problem:

$$\min_{\lambda \in \mathbb{R}^{m \times r}} \sum_{i \in B} \|(V' + A^\top \lambda)_i\| + \sum_{i \in [n] \setminus B} \|(V' + A^\top \lambda)_i\|^2 + \langle b', A^\top \lambda \rangle,$$

where $V' = V - ee_1^\top / 2$, $b' \in \mathbb{R}^{m \times r}$ such that $(Ae - 2b)e_1^\top = Ab'$, and R_i denotes the i-th row of R. This is a generalization of the geometric median problem, which can be solved by the Weiszfeld algorithm (Weiszfeld 1937).

Generalized Weiszfeld algorithm: assume every row of $(V' + A^\top \lambda^k)_B$ is non-zero, we design the following algorithm with convergence guarantee:

$$\lambda^{k+1} = T(\lambda^k) := -(A \operatorname{diag}(\mu^k) A^\top)^{-1} A(b' + \operatorname{diag}(\mu^k) V'),$$

where $\mu^k \in \mathbb{R}^n$ is the vector such that $\mu_i^k = 1 / \|(V' + A^\top \lambda^k)_i\|$ for any $i \in [B]$, and $\mu_i^k = 2$ otherwise.

We also use the Newton method to accelerate the convergence.

Numerical Experiments

Machine: All experiments were conducted using MATLAB R2023b on a workstation with an Intel Xeon E5-2680 v3 @ 2.50GHz CPU and 96GB RAM.

Tolerance for relative KKT residue: $= 10^{-6}$; Maxtime: $= 3600s$.

Solvers comparison:

Solver	**LR Factorization**	**Handle Ineq**
ManiSDP (J. Wang and Hu, 2025)	Yes	No
HALLaR (Monteiro, Sujanani, and Cifuentes, 2024)	Yes	No
LoRADS (Han et al., 2024)	Yes	No
SDPF (Tang and Toh, 2024a)	Yes	No
RiNNAL (Hou, Tang, and Toh, 2025)	Yes	No
SketchyCGAL (Yurtsever et al., 2021)	No	Yes
SDPNAL+ (Yang, Sun, and Toh, 2015)	No	Yes
SDPLR (Burer and Monteiro, 2003)	Yes	Yes
SDPDAL (Y. Wang et al., 2023)	Yes	Yes
RiNNAL+ (ours)	Yes	Yes

- Not comparing SDPDAL: code unavailable.
- Not Comparing SDPLR/SketchyCGAL: slower than SDPNAL+.

Binary Integer Nonconvex Quadratic Programming

$$\min \left\{ x^\top Q x + 2c^\top x : \ x \le e, \ x \in \{0,1\}^n \right\}$$

Problem	Algorithm	Iteration	Rank	R_{max}	Objective	Time	TPG
$n = 500$	RiNNAL+	16, 850, 3	50	8.70e-07	-1.2259545e+05	8.0	0.9
	SDPNAL+	44, 97, 1535	76	9.98e-07	-1.2259531e+05	613.1	–
$n = 1000$	RiNNAL+	12, 650, 3	79	9.58e-07	-3.8983061e+05	19.3	2.3
	SDPNAL+	–	–	–	–	–	–
$n = 2500$	RiNNAL+	11, 600, 2	134	7.28e-07	-1.6096512e+06	133.3	15.9
	SDPNAL+	–	–	–	–	–	–
$n = 5000$	RiNNAL+	12, 650, 3	177	8.61e-07	-4.6838013e+06	1103.1	254.8
	SDPNAL+	–	–	–	–	–	–

- PG step is efficient due to preprocessing and warm-start.

Maximum Stable Set Problems

$$\max \left\{ x^\top x : \; x_i x_j = 0 \; \forall (i,j) \in E, \; x \leq e, \; x \in \{0,1\}^n \right\}$$

Problem	Algorithm	Iteration	Rank	R_{max}	Objective	Time	TPG
G11	RiNNAL+	56, 2850, 11	8	6.41e-07	-4.0000014e+02	47.7	6.4
$n = 800$	SDPNAL+	482, 722, 9358	2	2.46e-07	-4.0000005e+02	3008.3	–
G18	RiNNAL+	34, 1750, 7	75	9.08e-07	-2.7900052e+02	34.0	5.5
$n = 800$	SDPNAL+	–	–	–	–	–	–
G54	RiNNAL+	46, 2350, 10	168	9.01e-07	-3.4100141e+02	68.4	11.3
$n = 1000$	SDPNAL+	–	–	–	–	–	–
G30	RiNNAL+	13, 700, 3	98	9.00e-07	-5.7703872e+02	87.7	10.4
$n = 2000$	SDPNAL+	–	–	–	–	–	–
G50	RiNNAL+	24, 1250, 5	134	9.39e-07	-1.4940617e+03	754.8	149.3
$n = 3000$	SDPNAL+	–	–	–	–	–	–
1tc.1024	RiNNAL+	116, 5850, 24	288	9.38e-07	-2.0420520e+02	192.4	28.1
$n = 1024$	SDPNAL+	–	–	–	–	–	–
1tc.2048	RiNNAL+	94, 4750, 26	520	9.85e-07	-3.7049061e+02	820.7	162.8
$n = 2048$	SDPNAL+	–	–	–	–	–	–

- Efficient even on problems with high-rank solutions.

Quadratic Knapsack Problems

$$\max\left\{x^\top Qx:\ a^\top x\le\tau,\ x\le e,\ x\in\{0,1\}^n\right\}$$

n	Iteration	Rank	R_{max}	Objective	Time	TPG
500	87, 4400, 18	26	9.91e-07	-1.1420448e+06	40.5	4.3
1000	27, 1400, 6	37	9.29e-07	-4.5986965e+06	53.1	5.5
2000	15, 800, 3	42	6.84e-07	-1.8316334e+07	123.7	22.7
5000	18, 950, 4	77	7.89e-07	-1.1399134e+08	1761.8	517.4

- SDPNAL+ fails to solve all instances.
- Projection and retraction nontrivial, but taking only 10–20% of total time.

Cardinality-Constrained Minimum SOS Clustering

$$\min\left\{\sum_{j=1}^{k}\frac{1}{c_j}\sum_{s=1}^{m}\sum_{t=1}^{m} d_{st}\pi_j^{(s)}\pi_j^{(t)}\ :\ \pi\in\mathcal{S},\ \pi\le e_{m\times k},\ \pi\in\{0,1\}^{m\times k}\right\}$$

where $\pi=[\pi^{(1)},\dots,\pi^{(k)}]$ with $\pi^{(j)}\in\{0,1\}^m$, and

$$\mathcal{S}:=\left\{\pi\in\mathbb{R}^{m\times k}:\sum_{j=1}^{k}\pi_i^{(j)}=1\ \forall i\in[m],\quad \sum_{i=1}^{m}\pi_i^{(j)}=c_j\ \forall j\in[k]\right\}$$

Problem	Algorithm	Iteration	Rank	R_{max}	Objective	Time	TPG
1	RiNNAL+	9, 500, 2	5	9.27e-07	7.3254609e+04	14.3	1.4
$n=720$	SDPNAL+	-	-	-	-	-	-
2	RiNNAL+	3, 200, 1	7	8.78e-07	4.0536895e+09	12.6	1.1
$n=902$	SDPNAL+	54, 246, 2312	1	7.91e-07	4.0536886e+09	2313.4	-
3	RiNNAL+	12, 511, 3	21	3.90e-07	8.5151301e+02	1500.3	193.9
$n=1662$	SDPNAL+	-	-	-	-	-	-
4	RiNNAL+	2, 150, 1	28	5.87e-07	7.3610392e+08	42.2	13.7
$n=1800$	SDPNAL+	-	-	-	-	-	-
5	RiNNAL+	51, 2600, 11	61	5.42e-07	1.0500893e+06	2215.9	170.8
$n=2250$	SDPNAL+	-	-	-	-	-	-

- Efficient for many constraints (up to 3.7M eq/ineq and 5M nonneg).

Sparse StQP

$$\min \left\{ x^\top Q x : \ e^\top x = 1, \ \|x\|_0 \le \rho, \ x \in \mathbb{R}^m_+ \right\},$$

Big-M reformulation:

$$\min \left\{ x^\top Q x : \ e^\top x = 1, \ e^\top u = \rho, \ x \le u \le e, \ x \in \mathbb{R}^m_+, \ u \in \{0,1\}^m \right\}.$$

Problem	Algorithm	Iteration	Rank	R_{max}	Objective	Time	TPG
COP	RiNNAL+	125, 6300, 25	102	6.64e-07	-1.0190589e-01	30.7	2.1
$n = 200$	SDPNAL+	76, 112, 1170	102	8.59e-07	-1.0196083e-01	40.1	-
PSD	RiNNAL+	374, 18750, 75	5	8.59e-07	1.4174794e-02	80.7	14.5
$n = 200$	SDPNAL+	801, 1058, 17306	2	8.50e-07	1.4174999e-02	468.1	-
SPN	RiNNAL+	747, 37400, 150	107	9.31e-07	1.3592384e-02	204.0	20.8
$n = 200$	SDPNAL+	801, 1032, 29187	12	9.94e-07	1.4887842e-02	866.1	-
COP	RiNNAL+	300, 15050, 60	202	9.99e-07	-1.0402158e-01	230.5	15.8
$n = 400$	SDPNAL+	103, 145, 1511	202	9.24e-07	-1.0445084e-01	170.7	-
PSD	RiNNAL+	360, 18050, 72	108	9.50e-07	9.0349366e-03	231.0	19.4
$n = 400$	SDPNAL+	-	-	-	-	-	-
SPN	RiNNAL+	1812, 90650, 363	248	9.97e-07	-1.2530973e-02	1556.1	145.7
$n = 400$	SDPNAL+	-	-	-	-	-	-
PSD	RiNNAL+	507, 25400, 102	234	9.88e-07	2.9188026e-03	1838.0	152.6
$n = 1000$	SDPNAL+	-	-	-	-	-	-

Quadratic Minimum Spanning Tree Problem

$$\min\left\{x^\top Qx : x \in \mathcal{T}'\right\}.$$

where

$$\mathcal{T}' := \left\{x \in \{0,1\}^m : x \le e,\ \sum_{e\in E} x_e = n-1,\ \sum_{e\in\partial S} x_e \ge 1,\ \forall S \subsetneq V,\ |S| = 1\right\},$$

Problem	Algorithm	Iteration	Rank	R_{max}	Objective	Time	TPG
vsym	RiNNAL+	29, 1480, 6	5	4.86e-07	7.7804931e+04	12.9	1.7
$m = 435$	SDPNAL+	148, 230, 3713	1	2.91e-07	7.7804694e+04	299.7	-
sym	RiNNAL+	103, 5190, 21	197	7.70e-07	5.3262719e+03	99.3	2.8
$m = 435$	SDPNAL+	96, 109, 1486	195	1.09e-06	5.3262716e+03	130.5	-
esym	RiNNAL+	4912, 254289, 4912	49	9.85e-07	7.4273402e+03	3456.3	931.4
$m = 435$	SDPNAL+	529, 692, 10111	46	9.55e-07	7.4273883e+03	1083.3	-
vsym	RiNNAL+	96, 4850, 22	8	7.99e-07	1.6444786e+05	289.2	64.3
$m = 1225$	SDPNAL+	-	-	-	-	-	-
sym	RiNNAL+	84, 4275, 17	509	8.07e-07	1.4104759e+04	484.0	21.2
$m = 1225$	SDPNAL+	192, 199, 2628	507	1.22e-06	1.4104759e+04	1987.4	-

Conclusion

- We justify the use of SDP-RLT over DNN: same tightness, smaller dimension.
- We propose RiNNAL+: a two-phase Riemannian ALM for solving SDP-RLT.
- Efficient for problems with many constraints and even high-rank solutions.
- The framework is broadly applicable to other structured SDPs.

Conclusion

- We justify the use of SDP-RLT over DNN: same tightness, smaller dimension.
- We propose RiNNAL+: a two-phase Riemannian ALM for solving SDP-RLT.
- Efficient for problems with many constraints and even high-rank solutions.
- The framework is broadly applicable to other structured SDPs.

Paper: arXiv: 2507.13776

Code: `https://github.com/HouDiOpt/RiNNALplus`

Thank you

References I

- Boumal, Nicolas (2015). "A Riemannian low-rank method for optimization over semidefinite matrices with block-diagonal constraints". In: *arXiv preprint arXiv:1506.00575*.
- Burer, Samuel (2010). "Optimizing a polyhedral-semidefinite relaxation of completely positive programs". In: *Mathematical Programming Computation* 2.1, pp. 1–19.
- Burer, Samuel and Renato DC Monteiro (2003). "A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization". In: *Mathematical Programming* 95.2, pp. 329–357.
- Gao, Bin and P-A Absil (2022). "A Riemannian rank-adaptive method for low-rank matrix completion". In: *Computational Optimization and Applications* 81, pp. 67–90.
- Han, Qiushi et al. (2024). "A Low-Rank ADMM Splitting Approach for Semidefinite Programming". In: *arXiv preprint arXiv:2403.09133*.

References II

- Hou, Di, Tianyun Tang, and Kim-Chuan Toh (2025). "A low-rank augmented Lagrangian method for doubly nonnegative relaxations of mixed-binary quadratic programs". In: *arXiv preprint arXiv:2502.13849.*
- Journée, Michel et al. (2010). "Low-rank optimization on the cone of positive semidefinite matrices". In: *SIAM Journal on Optimization* 20.5, pp. 2327–2351.
- Lee, Ching-pei et al. (2024). "Accelerating nuclear-norm regularized low-rank matrix optimization through Burer-Monteiro decomposition". In: *Journal of Machine Learning Research* 25.379, pp. 1–52.
- Monteiro, Renato DC, Arnesh Sujanani, and Diego Cifuentes (2024). "A low-rank augmented Lagrangian method for large-scale semidefinite programming based on a hybrid convex-nonconvex approach". In: *arXiv preprint arXiv:2401.12490.*
- Qi, Houduo and Defeng Sun (2006). "A quadratically convergent Newton method for computing the nearest correlation matrix". In: *SIAM Journal on Matrix Analysis and Applications* 28.2, pp. 360–385.

References III

- Sherali, Hanif D and Warren P Adams (1990). "A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems". In: *SIAM Journal on Discrete Mathematics* 3.3, pp. 411–430.
- Tang, Tianyun and Kim-Chuan Toh (2024a). "A feasible method for general convex low-rank SDP problems". In: *SIAM Journal on Optimization* 34.3, pp. 2169–2200.
- — (2024b). "A feasible method for solving an SDP relaxation of the quadratic knapsack problem". In: *Mathematics of Operations Research* 49.1, pp. 19–39.
- Wang, Jie and Liangbing Hu (2025). "Solving low-rank semidefinite programs via manifold optimization". In: *Journal of Scientific Computing* 104.1, p. 33.
- Wang, Yifei et al. (2023). "A decomposition augmented Lagrangian method for low-rank semidefinite programming". In: *SIAM Journal on Optimization* 33.3, pp. 1361–1390.

References IV

Yang, Liuqin, Defeng Sun, and Kim-Chuan Toh (2015). "SDPNAL+: a majorized semismooth Newton-CG augmented Lagrangian method for semidefinite programming with nonnegative constraints". In: *Mathematical Programming Computation* 7.3, pp. 331–366.

Yurtsever, Alp et al. (2021). "Scalable semidefinite programming". In: *SIAM Journal on Mathematics of Data Science* 3.1, pp. 171–200.