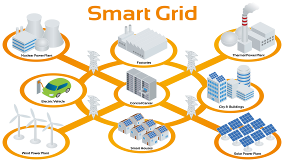
COSMO.jl

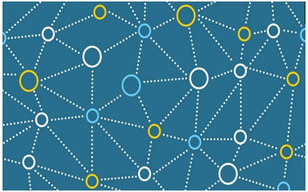
A conic operator splitting method for large convex problems

Michael Garstka · Paul Goulart · Mark Cannon

International Conference on Continuous Optimization, Berlin
6th August 2019

Why do we care about solving large convex conic problems?





- Problem dimensions grow drastically
- State-of-the-art (interior point) solver do not scale well

COSMO.jl A quadratic objective conic solver

implemented in pure Julia

- ADMM solver for large convex conic problems
- Support of major convex cones:

Zero cone	Second order cone	Power cone
Nonnegatives	Positive semidefinite cone	
Hyperbox	Exponential cone	

- Quadratic cost function and conic constraints
- Implemented in Julia

Overview

COSMO.jl

Example: Nearest Correlation Matrix problem

Conic Problem Format

ADMM Algorithm

Chordal decomposition of PSD constraints

Customisable and extensible code

Conclusion

Example: Nearest correlation matrix problem

- Given data matrix $C \in \mathbb{R}^{n \times n}$ find the nearest correlation matrix X:

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\|X - C\|_F^2 \\ \text{subject to} & X_{ii} = 1, \quad i = 1, \dots, n \\ & X \in \mathbb{S}_+^n, \end{array}$$

- The objective function can be rewritten as

$$\frac{1}{2}\|X - C\|_F^2 = \frac{1}{2} x^\top x - c^\top x + \frac{1}{2} c^\top c$$

with $x = \text{vec}(X)$ and $c = \text{vec}(C)$

Example: Nearest correlation matrix problem

- We can solve this with a few lines of code with JuMP and COSMO:

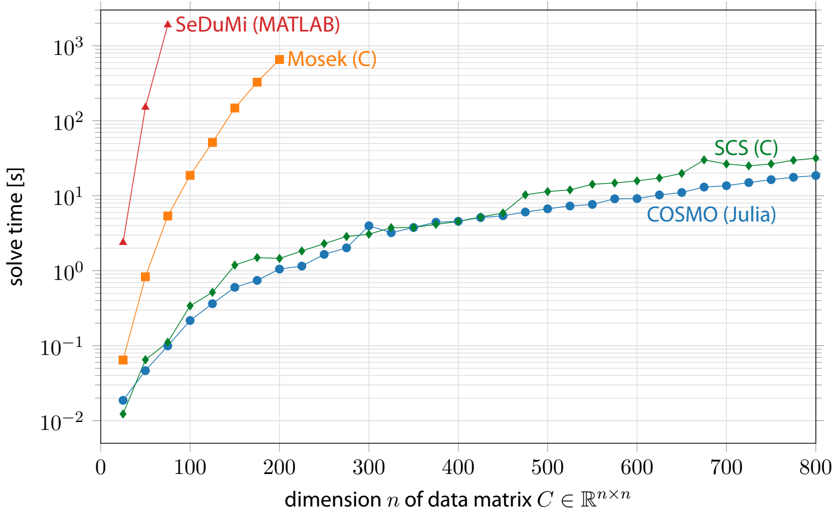
```
C = Symmetric(rand(n, n));
c = vec(C);

m = JuMP.Model(with_optimizer(COSMO.Optimizer));
@variable(m, X[1:n, 1:n], PSD);
x = vec(X);

@objective(m, Min, 0.5 * x' * x  - c' * x + 0.5 * c' * c)

@constraint(m, [i = 1: n], X[i, i] == 1.)

JuMP.optimize!(m)
```

$X \in \mathcal{S}_n^+$

$\frac{1}{2}\|X - C\|_F^2$

$X_{ii} = 1,\ i = 1, \ldots, n$

Example: Nearest correlation matrix problem



Problem Format

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}x^\top P x + q^\top x \\ \text{subject to} & Ax + s = b \\ & s \in \mathcal{K} \end{array}$$

- Decision variables: $x \in \mathbb{R}^n$, $s \in \mathbb{R}^m$
- Problem data: real matrices $P \succeq 0$, A, and real vectors q, b
- Convex cone $\mathcal{K}$ which can be a Cartesian product of cones:

$$\mathcal{K} = \mathcal{K}_1 \times \mathcal{K}_2 \times \cdots \times \mathcal{K}_N$$

Problem Format

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}x^\top P x + q^\top x \\ \text{subject to} & Ax + s = b \\ & s \in \{0\}^{m_1} \times \mathbb{R}_+^{m_2} \end{array}$$

Linear Program

- Decision variables: $x \in \mathbb{R}^n$, $s \in \mathbb{R}^m$
- Problem data: real matrices $P \succeq 0$, A, and real vectors q, b
- Convex cone $\mathcal{K}$ which can be a Cartesian product of cones:

$$\mathcal{K} = \mathcal{K}_1 \times \mathcal{K}_2 \times \cdots \times \mathcal{K}_N$$

Problem Format

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}x^\top P x + q^\top x \\ \text{subject to} & Ax + s = b \\ & \text{mat}(s) \succeq 0 \end{array}$$

Semidefinite Program

- Decision variables: $x \in \mathbb{R}^n$, $s \in \mathbb{R}^m$
- Problem data: real matrices $P \succeq 0$, A, and real vectors q, b
- Convex cone $\mathcal{K}$ which can be a Cartesian product of cones:

$$\mathcal{K} = \mathcal{K}_1 \times \mathcal{K}_2 \times \cdots \times \mathcal{K}_N$$

Generic ADMM

$$\begin{array}{ll}\text{minimize} & f(x) + g(z) \\ \text{subject to} & Ax + Bz = c\end{array}$$

- Augmented Lagrangian:

$$L_\rho(x, z, y) = f(x) + g(z) + y^\top(Ax + Bz - c) + \frac{\rho}{2}\|Ax + Bz - c\|_2^2,$$

- ADMM steps:

Generic ADMM

$$\begin{array}{ll} \text{minimize} & f(x) + g(z) \\ \text{subject to} & Ax + Bz = c \end{array}$$

- Augmented Lagrangian:

$$L_\rho(x, z, y) = f(x) + g(z) + y^\top (Ax + Bz - c) + \frac{\rho}{2}\|Ax + Bz - c\|_2^2,$$

- ADMM steps:

$$x^{k+1} := \operatorname*{argmin}_x L_\rho(x, z^k, y^k)$$

Generic ADMM

$$\begin{array}{ll}\text{minimize} & f(x) + g(z) \\ \text{subject to} & Ax + Bz = c\end{array}$$

- Augmented Lagrangian:

$$L_\rho(x, z, y) = f(x) + g(z) + y^\top (Ax + Bz - c) + \frac{\rho}{2}\|Ax + Bz - c\|_2^2,$$

- ADMM steps:

$$x^{k+1} := \underset{x}{\text{argmin}}\, L_\rho(x, z^k, y^k)$$
$$z^{k+1} := \underset{z}{\text{argmin}}\, L_\rho(x^{k+1}, z, y^k)$$

Generic ADMM

$$\begin{array}{ll} \text{minimize} & f(x) + g(z) \\ \text{subject to} & Ax + Bz = c \end{array}$$

- Augmented Lagrangian:

$$L_\rho(x, z, y) = f(x) + g(z) + y^\top (Ax + Bz - c) + \frac{\rho}{2}\|Ax + Bz - c\|_2^2,$$

- ADMM steps:

$$\begin{aligned} x^{k+1} &:= \operatorname*{argmin}_x L_\rho(x, z^k, y^k) \\ z^{k+1} &:= \operatorname*{argmin}_z L_\rho(x^{k+1}, z, y^k) \\ y^{k+1} &:= y^k + \rho(Ax^{k+1} + Bz^{k+1} - c) \end{aligned}$$

Splitting method

$$\begin{aligned} \text{minimize} \quad & \tfrac{1}{2}x^\top P x + q^\top x \\ \text{subject to} \quad & Ax + s = b \\ & s \in \mathcal{K} \end{aligned}$$

$$\text{minimize} \quad \overbrace{\tfrac{1}{2}\tilde{x}^\top P \tilde{x} + q^\top \tilde{x} + I_{Ax+s=b}(\tilde{x}, \tilde{s})}^{f(\tilde{x}, \tilde{s})} + \overbrace{I_{\mathcal{K}}(s)}^{g(x, s)}$$

subject to $(\tilde{x}, \tilde{s}) = (x, s)$

ADMM algorithm

1: **Input:** Initial values x^0, s^0, y^0, step sizes σ, ρ

2: **Do**

3: $$(\tilde{x}^{k+1}, \tilde{s}^{k+1}) = \underset{\tilde{x},\tilde{s}}{\text{argmin}}\, L_\rho\left(\tilde{x}, \tilde{s}, x^k, s^k, y^k\right)$$

equality constrained QP → linear KKT system

4: $$x^{k+1} = \Pi_{\mathbb{R}^n}\left(\tilde{x}^{k+1}\right)$$

5: $$s^{k+1} = \Pi_{\mathcal{K}}\left(\tilde{s}^{k+1} + \tfrac{1}{\rho} y^k\right)$$

projection onto $\mathcal{K}$

6: $$y^{k+1} = y^k + \rho\left(\tilde{s}^{k+1} - s^{k+1}\right)$$

7: **while** *termination criteria not satisfied*

Chordal Decomposition

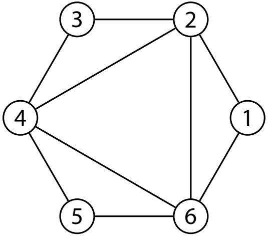
$$\begin{aligned} \text{minimize} \quad & \tfrac{1}{2}x^\top P x + q^\top x \\ \text{subject to} \quad & \sum_{i=1}^{m} \mathcal{A}_i x_i + S = B \\ & S \in \mathbb{S}^r_+ \end{aligned}$$

$$\begin{bmatrix} S_{11} & S_{12} & 0 & 0 & 0 & S_{16} \\ S_{21} & S_{22} & S_{23} & S_{24} & 0 & S_{26} \\ 0 & S_{32} & S_{33} & S_{34} & 0 & 0 \\ 0 & S_{42} & S_{43} & S_{44} & S_{45} & S_{46} \\ 0 & 0 & 0 & S_{54} & S_{55} & S_{56} \\ S_{61} & S_{62} & 0 & S_{64} & S_{65} & S_{66} \end{bmatrix}$$

Chordal Decomposition

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}x^\top P x + q^\top x \\ \text{subject to} & \sum_{i=1}^{m} \mathcal{A}_i x_i + S = B \\ & S \in \mathbb{S}^r_+ \end{array}$$

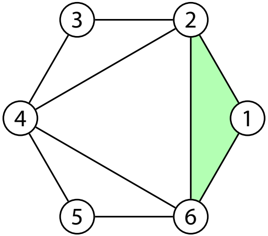
$$\begin{bmatrix} S_{11} & S_{12} & 0 & 0 & 0 & S_{16} \\ S_{21} & S_{22} & S_{23} & S_{24} & 0 & S_{26} \\ 0 & S_{32} & S_{33} & S_{34} & 0 & 0 \\ 0 & S_{42} & S_{43} & S_{44} & S_{45} & S_{46} \\ 0 & 0 & 0 & S_{54} & S_{55} & S_{56} \\ S_{61} & S_{62} & 0 & S_{64} & S_{65} & S_{66} \end{bmatrix}$$



Chordal Decomposition

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}x^\top P x + q^\top x \\ \text{subject to} & \sum_{i=1}^{m} \mathcal{A}_i x_i + S = B \\ & S \in \mathbb{S}^r_+ \end{array}$$

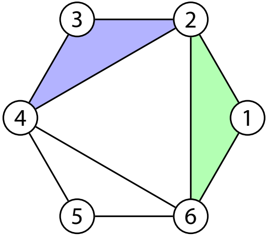
$$\begin{bmatrix} S_{11} & S_{12} & 0 & 0 & 0 & S_{16} \\ S_{21} & S_{22} & S_{23} & S_{24} & 0 & S_{26} \\ 0 & S_{32} & S_{33} & S_{34} & 0 & 0 \\ 0 & S_{42} & S_{43} & S_{44} & S_{45} & S_{46} \\ 0 & 0 & 0 & S_{54} & S_{55} & S_{56} \\ S_{61} & S_{62} & 0 & S_{64} & S_{65} & S_{66} \end{bmatrix}$$



Chordal Decomposition

$$\begin{aligned} \text{minimize} \quad & \tfrac{1}{2}x^\top P x + q^\top x \\ \text{subject to} \quad & \sum_{i=1}^{m} \mathcal{A}_i x_i + S = B \\ & S \in \mathbb{S}_+^r \end{aligned}$$

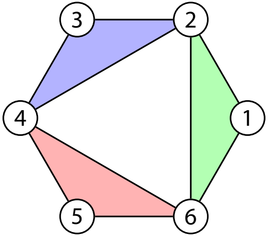
$$\begin{bmatrix} S_{11} & S_{12} & 0 & 0 & 0 & S_{16} \\ S_{21} & S_{22} & S_{23} & S_{24} & 0 & S_{26} \\ 0 & S_{32} & S_{33} & S_{34} & 0 & 0 \\ 0 & S_{42} & S_{43} & S_{44} & S_{45} & S_{46} \\ 0 & 0 & 0 & S_{54} & S_{55} & S_{56} \\ S_{61} & S_{62} & 0 & S_{64} & S_{65} & S_{66} \end{bmatrix}$$



Chordal Decomposition

$$\begin{aligned} \text{minimize} \quad & \tfrac{1}{2}x^\top P x + q^\top x \\ \text{subject to} \quad & \sum_{i=1}^{m} \mathcal{A}_i x_i + S = B \\ & S \in \mathbb{S}^r_+ \end{aligned}$$

$$\begin{bmatrix} S_{11} & S_{12} & 0 & 0 & 0 & S_{16} \\ S_{21} & S_{22} & S_{23} & S_{24} & 0 & S_{26} \\ 0 & S_{32} & S_{33} & S_{34} & 0 & 0 \\ 0 & S_{42} & S_{43} & S_{44} & S_{45} & S_{46} \\ 0 & 0 & 0 & S_{54} & S_{55} & S_{56} \\ S_{61} & S_{62} & 0 & S_{64} & S_{65} & S_{66} \end{bmatrix}$$



Chordal Decomposition

minimize $\frac{1}{2}x^\top Px + q^\top x$

subject to $\sum_{i=1}^{m} \mathcal{A}_i x_i + S = B$

$S \in \mathbb{S}_+^r$

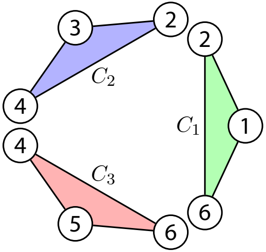
minimize $\frac{1}{2}x^\top Px + q^\top x$

subject to $\sum_{i=1}^{m} \mathcal{A}_i x_i + \sum_{\ell=1}^{p} T_\ell^\top S_\ell T_\ell = B$

$S_\ell \in \mathbb{S}_+^{|C_\ell|}, \quad \ell = 1, \dots, p$

Agler's theorem

$$\begin{bmatrix} S_{11} & S_{12} & 0 & 0 & 0 & S_{16} \\ S_{21} & S_{22} & S_{23} & S_{24} & 0 & S_{26} \\ 0 & S_{32} & S_{33} & S_{34} & 0 & 0 \\ 0 & S_{42} & S_{43} & S_{44} & S_{45} & S_{46} \\ 0 & 0 & 0 & S_{54} & S_{55} & S_{56} \\ S_{61} & S_{62} & 0 & S_{64} & S_{65} & S_{66} \end{bmatrix}$$

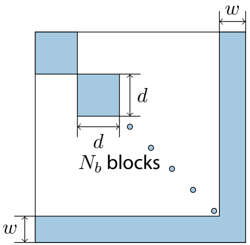


Example: Nearest correlation matrix problem

- Given data matrix $C \in \mathbb{R}^{n \times n}$ find the nearest correlation matrix X:

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\|X - C\|_F^2 \\ \text{subject to} & X_{ii} = 1, \quad i = 1, \dots, n \\ & X \in \mathbb{S}_+^n, \end{array}$$

- Let's assume that C has a chordal sparsity structure with $G(V, E)$:

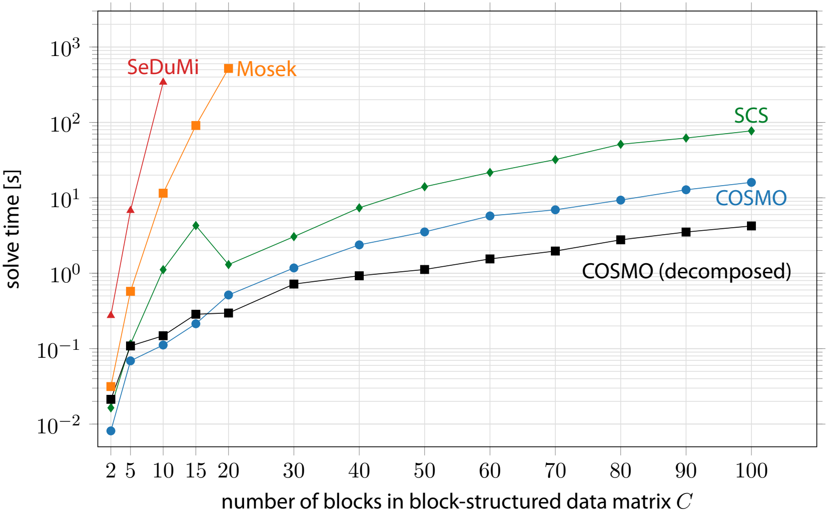


- We want X to keep the same sparsity structure $X \in \mathbb{S}_+^n(E, 0)$

Example: Nearest correlation matrix problem

```
m = JuMP.Model(with_optimizer(COSMO.Optimizer, decompose = true));
@variable(m, X[1:n, 1:n]);
x = vec(X);
@objective(m, Min, 0.5 * x' * x  - c' * x + 0.5 * c' * c)
@constraint(m, [i = 1: n], X[i, i] == 1.)


@constraint(m, A * x  in MOI.PositiveSemidefiniteConeTriangle(n));

JuMP.optimize!(m)
```

Example: Nearest correlation matrix problem



ADMM algorithm

1: **Input:** Initial values x^0, s^0, y^0, step sizes σ, ρ

2: **Do**

3: $$\left(\tilde{x}^{k+1}, \tilde{s}^{k+1}\right) = \underset{\tilde{x},\tilde{s}}{\text{argmin}}\, L_\rho\left(\tilde{x}, \tilde{s}, x^k, s^k, y^k\right)$$

equality constrained QP $\rightarrow$ linear KKT system

4: $$x^{k+1} = \Pi_{\mathbb{R}^n}\left(\tilde{x}^{k+1}\right)$$

5: $$s^{k+1} = \Pi_{\mathcal{K}}\left(\tilde{s}^{k+1} + \frac{1}{\rho} y^k\right)$$

projection onto $\mathcal{K}$

6: $$y^{k+1} = y^k + \rho\left(\tilde{s}^{k+1} - s^{k+1}\right)$$

7: **while** *termination criteria not satisfied*

Customisable and extensible code

- Custom solver for KKT system
- User-defined convex sets:

```
# Define new convex set
struct MyConvexSet <: COSMO.AbstractConvexSet
  dim::Int
end

# define a projection function
function COSMO.project!(x, convex_set::MyConvexSet)
  # projection code for x onto convex_set
end
```

COSMO.jl A quadratic objective conic solver
implemented in pure Julia

Conclusion:

- open source ADMM-based solver written in Julia
- supports quadratic objectives
- supports major convex cones
- infeasibility detection
- chordal decomposition of PSD constraints
- allows user-defined convex sets

COSMO.jl A quadratic objective conic solver
implemented in pure Julia

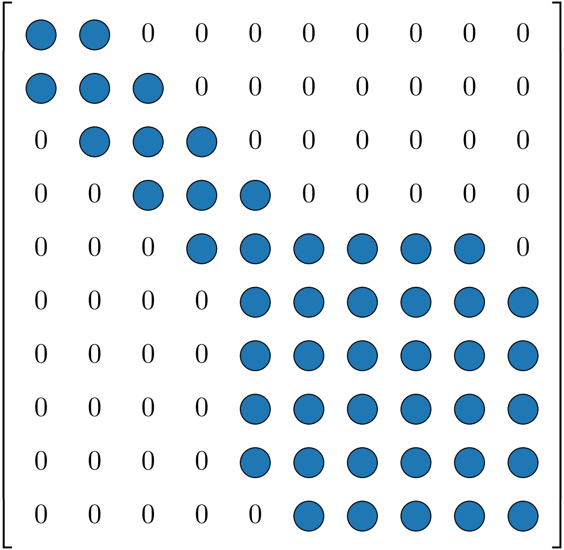
Conclusion:

- open source ADMM-based solver written in Julia
- supports quadratic objectives
- supports major convex cones
- infeasibility detection
- chordal decomposition of PSD constraints
- allows user-defined convex sets

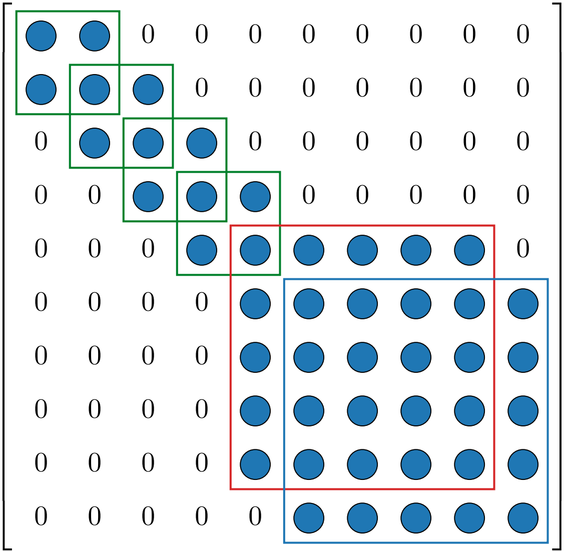
Future work:

- Acceleration methods
- Approximate projections
- Parallel Implementation of projections

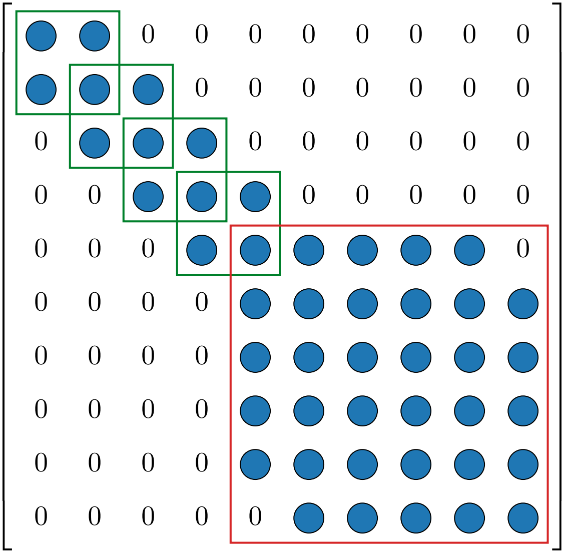
Future Work: Smart Clique Merging



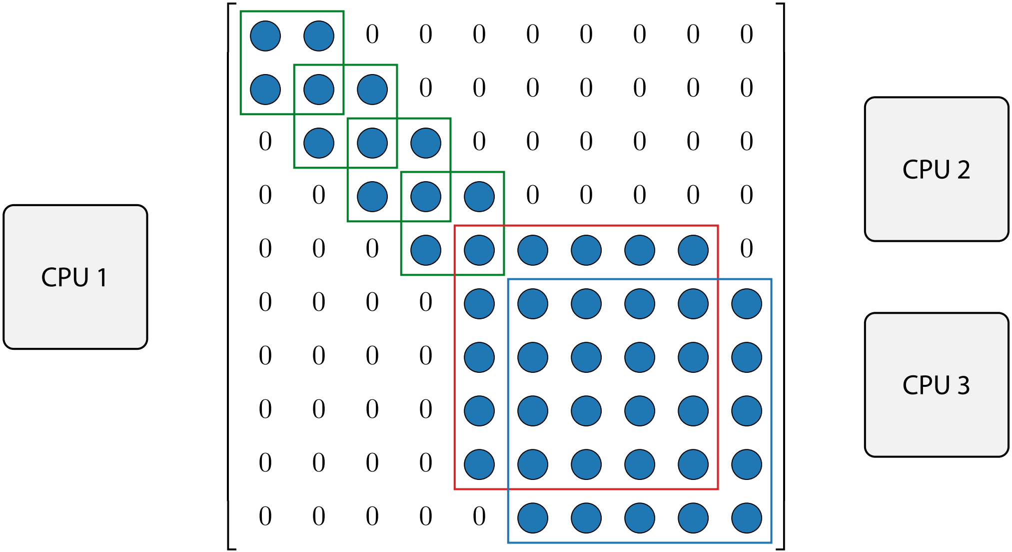
Future Work: Smart Clique Merging



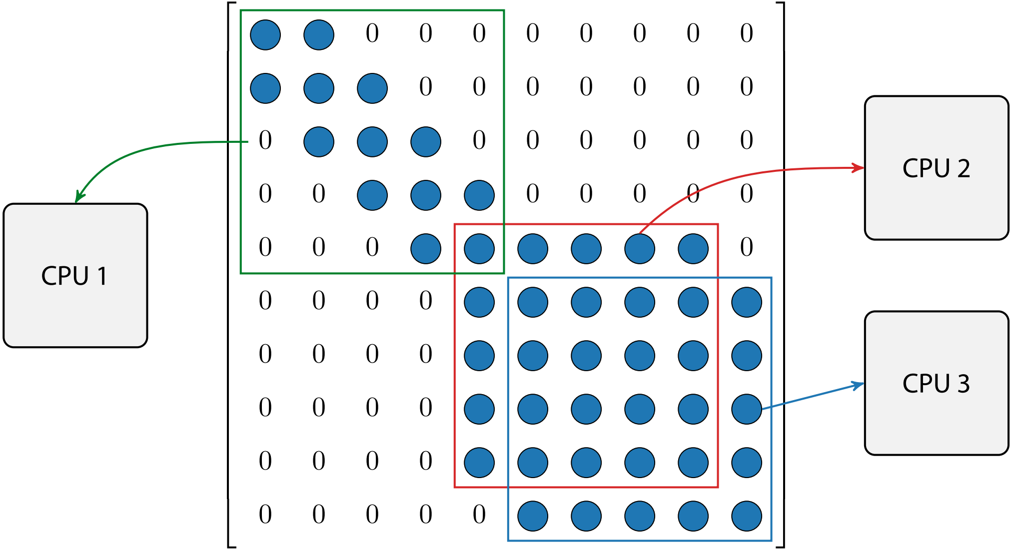
Future Work: Smart Clique Merging



Future Work: Smart Clique Merging

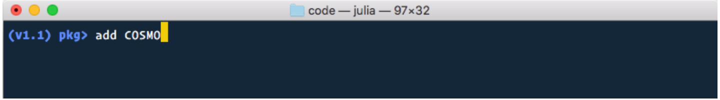


Future Work: Smart Clique Merging

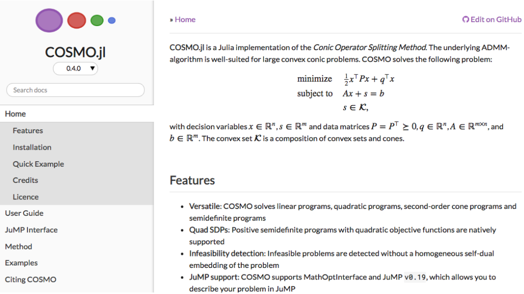


COSMO.jl Package

- Installation via the Julia package manager



- Code and documentation available at:
  `https://github.com/oxfordcontrol/COSMO.jl`


COSMO.jl
0.4.0
Search docs

Home
Features
Installation
Quick Example
Credits
Licence
User Guide
JuMP Interface
Method
Examples
Citing COSMO

» Home   Edit on GitHub

COSMO.jl is a Julia implementation of the *Conic Operator Splitting Method*. The underlying ADMM-algorithm is well-suited for large convex conic problems. COSMO solves the following problem:

$$\begin{aligned} \text{minimize} \quad & \tfrac{1}{2}x^\top Px + q^\top x \\ \text{subject to} \quad & Ax + s = b \\ & s \in \mathcal{K}, \end{aligned}$$

with decision variables $x \in \mathbb{R}^n, s \in \mathbb{R}^m$ and data matrices $P = P^\top \succeq 0, q \in \mathbb{R}^n, A \in \mathbb{R}^{m\times n}$, and $b \in \mathbb{R}^m$. The convex set $\mathcal{K}$ is a composition of convex sets and cones.

Features

- Versatile: COSMO solves linear programs, quadratic programs, second-order cone programs and semidefinite programs
- Quad SDPs: Positive semidefinite programs with quadratic objective functions are natively supported
- Infeasibility detection: Infeasible problems are detected without a homogeneous self-dual embedding of the problem
- JuMP support: COSMO supports MathOptInterface and JuMP `v0.19`, which allows you to describe your problem in JuMP