OSQP.jl

A Julia wrapper for the Operator Splitting QP solver

Bartolomeo Stellato

joint work with Goran Banjac,

Nicholas Moehle, Paul Goulart, Alberto Bemporad, Stephen Boyd

JuMP Developers Meetup, 13 Jun 2017

Why quadratic programming?

AN ALGORITHM FOR QUADRATIC PROGRAMMING

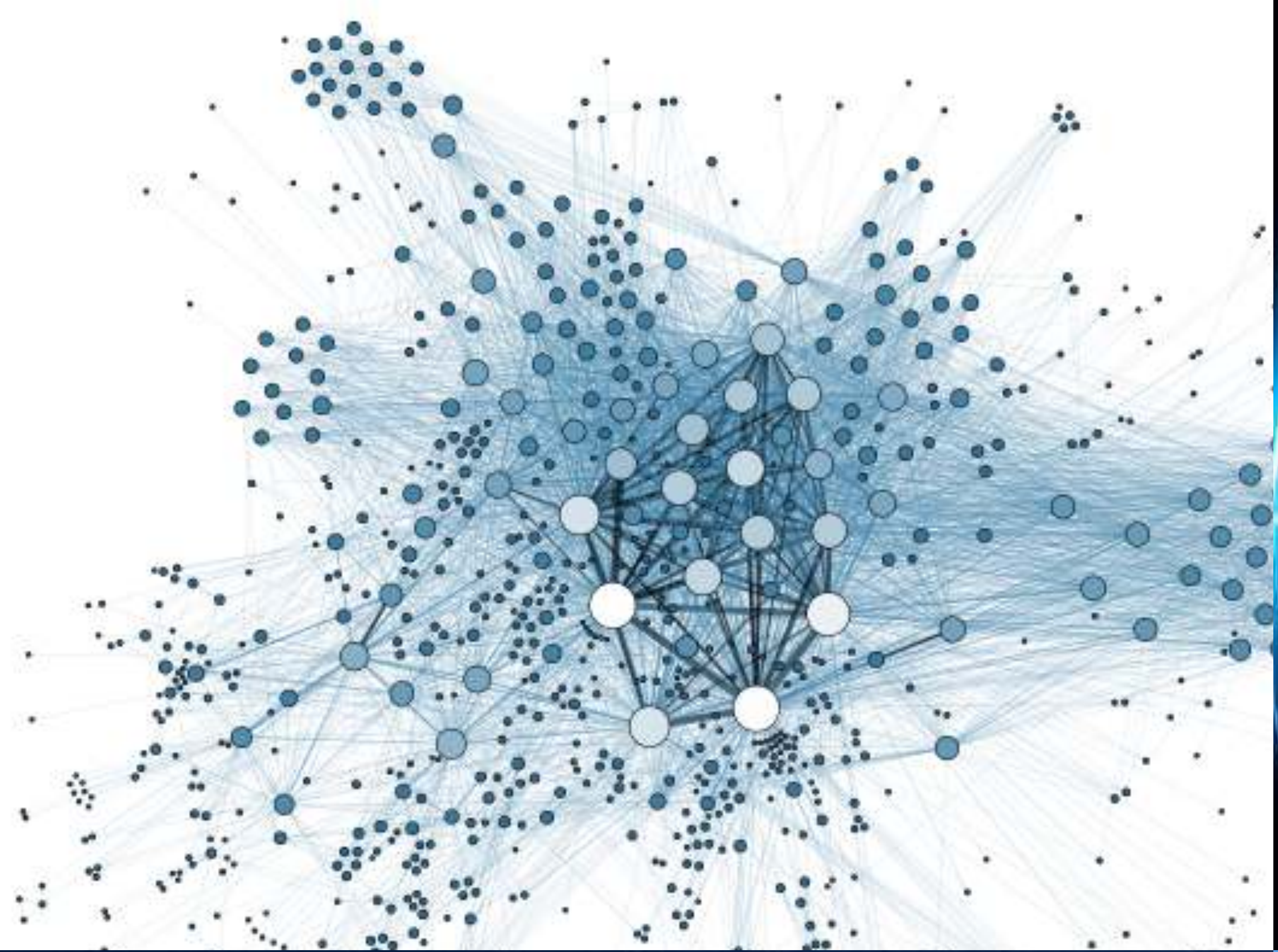
Marguerite Frank and Philip Wolfe[1]
Princeton University

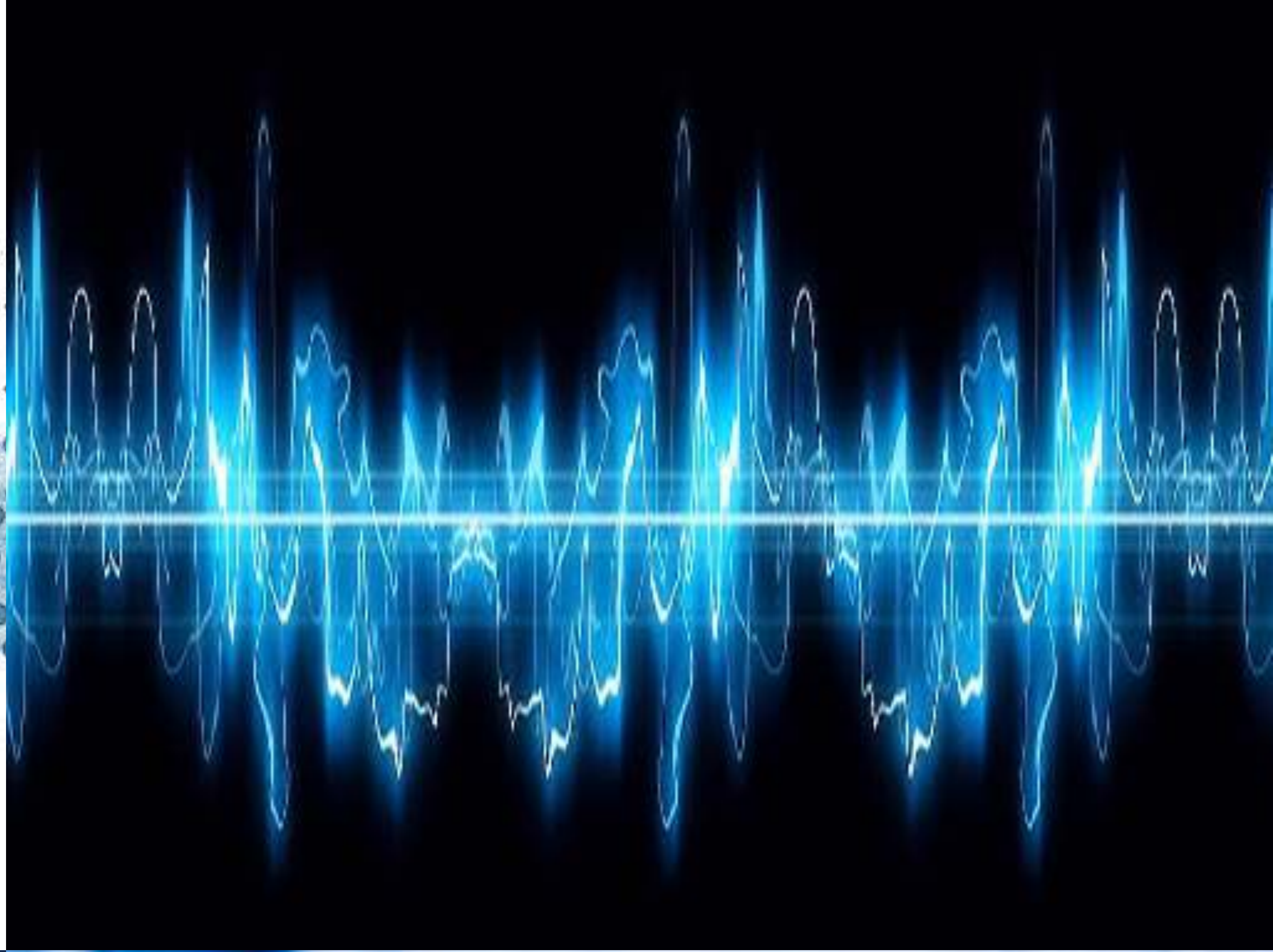
A finite iteration method for calculating the solution of quadratic programming problems is described. Extensions to more general non-linear problems are suggested.

1. INTRODUCTION

The problem of maximizing a concave quadratic function whose variables are subject to linear inequality constraints has been the subject of several recent studies, from both the computational side and the theoretical (see Bibliography). Our aim here has been to develop a method for solving this non-linear programming problem which should be particularly well adapted to high-speed machine computation.

March 1956!






123.74
145
140
135
130
125
Tues.
142.78
Chg

First-order methods

Pros	Cons
Warm starting	Low accuracy solutions
Handle large-scale problems	Don't detect infeasibility
Embeddable	Problem data dependent

General Purpose QP Solver

Based on first-order methods

Robust

Accurate

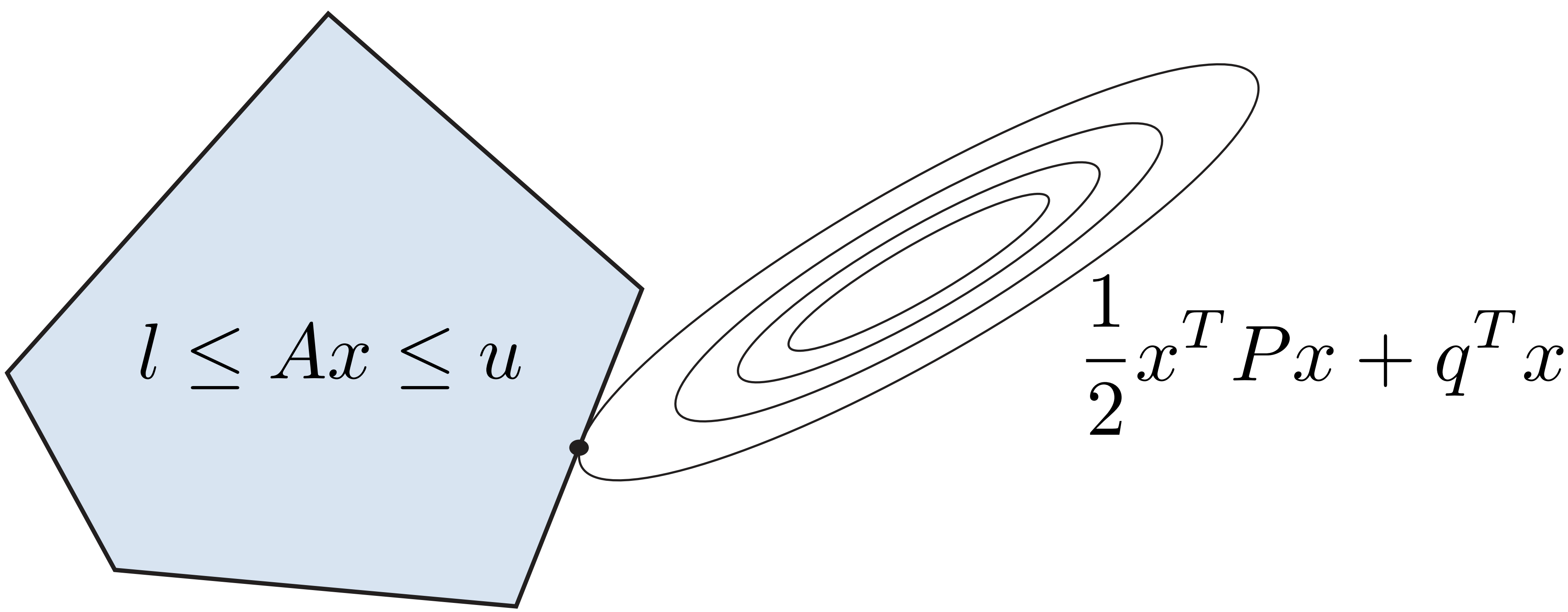
Detects Infeasibility

The OSQP Solver

The problem

Quadratic Program

minimize $\frac{1}{2}x^T P x + q^T x$
subject to $l \leq Ax \leq u$



ADMM

minimize $\quad f(x) + g(x)$ $\longrightarrow$

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

ADMM

minimize $f(x) + g(x)$ → minimize $f(\tilde{x}) + g(x)$ subject to $\tilde{x} = x$

1. $$\tilde{x}^{k+1} \leftarrow \underset{\tilde{x}}{\operatorname{argmin}} \left(f(\tilde{x}) + \frac{\rho}{2} \left\| \tilde{x} - x^k + \frac{y^k}{\rho} \right\|^2 \right)$$

2. $$x^{k+1} \leftarrow \underset{x}{\operatorname{argmin}} \left(g(x) + \frac{\rho}{2} \left\| x - \tilde{x}^{k+1} - \frac{y^k}{\rho} \right\|^2 \right)$$

3. $$y^{k+1} \leftarrow y^k + \rho \left(\tilde{x}^{k+1} - x^{k+1} \right)$$

How to split the QP?

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}x^T P x + q^T x \\ \text{subject to} & Ax = z \\ & l \le z \le u \end{array}$$

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\tilde{x}^T P \tilde{x} + q^T \tilde{x} + \mathcal{I}_{Ax=z}(\tilde{x}, \tilde{z}) + \mathcal{I}_{[l,u]}(z) \\ \text{subject to} & (\tilde{x}, \tilde{z}) = (x, z) \end{array}$$

How to split the QP?

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}x^T P x + q^T x \\ \text{subject to} & Ax = z \\ & l \leq z \leq u \end{array} \qquad f$$

$$f$$

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\tilde{x}^T P \tilde{x} + q^T \tilde{x} + \mathcal{I}_{Ax=z}(\tilde{x}, \tilde{z}) + \mathcal{I}_{[l,u]}(z) \\ \text{subject to} & (\tilde{x}, \tilde{z}) = (x, z) \end{array}$$

How to split the QP?

$$\begin{array}{ll} \text{minimize} & \tfrac{1}{2}x^T P x + q^T x \\ \text{subject to} & Ax = z \\ & l \leq z \leq u \end{array}$$

f: $\tfrac{1}{2}x^T P x + q^T x$, $Ax = z$

g: $l \leq z \leq u$

$$\begin{array}{ll} \text{minimize} & \underbrace{\tfrac{1}{2}\tilde{x}^T P \tilde{x} + q^T \tilde{x} + \mathcal{I}_{Ax=z}(\tilde{x}, \tilde{z})}_{f} + \underbrace{\mathcal{I}_{[l,u]}(z)}_{g} \\ \text{subject to} & (\tilde{x}, \tilde{z}) = (x, z) \end{array}$$

OSQP

Problem

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}x^T P x + q^T x \\ \text{subject to} & l \leq Ax \leq u \end{array}$$

Algorithm

1
$$\begin{cases} (x^{k+1}, \nu^{k+1}) \leftarrow \text{solve} \begin{bmatrix} P + \sigma I & A^T \\ A & -\frac{1}{\rho} I \end{bmatrix} \begin{bmatrix} x^{k+1} \\ \nu^{k+1} \end{bmatrix} = \begin{bmatrix} \sigma x^k - q \\ z^k - \frac{1}{\rho} y^k \end{bmatrix} \\ \tilde{z}^{k+1} \leftarrow z^k + \frac{1}{\rho}(\nu^{k+1} - y^k) \end{cases}$$

2
$$\left\{ z^{k+1} \leftarrow \Pi\left(\tilde{z}^{k+1} + \frac{1}{\rho} y^k\right) \right.$$

3
$$\left\{ y^{k+1} \leftarrow y^k + \rho\left(\tilde{z}^{k+1} - z^{k+1}\right) \right.$$

OSQP

Problem

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}x^T P x + q^T x \\ \text{subject to} & l \leq Ax \leq u \end{array}$$

Algorithm

Linear system solve

$$(x^{k+1}, \nu^{k+1}) \leftarrow \text{solve} \begin{bmatrix} P + \sigma I & A^T \\ A & -\frac{1}{\rho} I \end{bmatrix} \begin{bmatrix} x^{k+1} \\ \nu^{k+1} \end{bmatrix} = \begin{bmatrix} \sigma x^k - q \\ z^k - \frac{1}{\rho} y^k \end{bmatrix}$$

$$\tilde{z}^{k+1} \leftarrow z^k + \frac{1}{\rho}(\nu^{k+1} - y^k)$$

$$z^{k+1} \leftarrow \Pi\left(\tilde{z}^{k+1} + \frac{1}{\rho} y^k\right)$$

$$y^{k+1} \leftarrow y^k + \rho\left(\tilde{z}^{k+1} - z^{k+1}\right)$$

OSQP

Problem

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}x^T P x + q^T x \\ \text{subject to} & l \leq Ax \leq u \end{array}$$

Algorithm

Linear system solve

$$(x^{k+1}, \nu^{k+1}) \leftarrow \text{solve} \begin{bmatrix} P + \sigma I & A^T \\ A & -\frac{1}{\rho} I \end{bmatrix} \begin{bmatrix} x^{k+1} \\ \nu^{k+1} \end{bmatrix} = \begin{bmatrix} \sigma x^k - q \\ z^k - \frac{1}{\rho} y^k \end{bmatrix}$$

Easy operations

$$\tilde{z}^{k+1} \leftarrow z^k + \frac{1}{\rho}(\nu^{k+1} - y^k)$$

$$z^{k+1} \leftarrow \Pi\left(\tilde{z}^{k+1} + \frac{1}{\rho} y^k\right)$$

$$y^{k+1} \leftarrow y^k + \rho\left(\tilde{z}^{k+1} - z^{k+1}\right)$$

OSQP

Problem

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}x^T P x + q^T x \\ \text{subject to} & l \leq Ax \leq u \end{array}$$

Algorithm

Linear system solve

$$(x^{k+1}, \nu^{k+1}) \leftarrow \text{solve} \begin{bmatrix} P + \sigma I & A^T \\ A & -\frac{1}{\rho}I \end{bmatrix} \begin{bmatrix} x^{k+1} \\ \nu^{k+1} \end{bmatrix} = \begin{bmatrix} \sigma x^k - q \\ z^k - \frac{1}{\rho}y^k \end{bmatrix}$$

Easy operations

$$\tilde{z}^{k+1} \leftarrow z^k + \frac{1}{\rho}(\nu^{k+1} - y^k)$$

$$z^{k+1} \leftarrow \Pi\left(\tilde{z}^{k+1} + \frac{1}{\rho}y^k\right)$$

$$y^{k+1} \leftarrow y^k + \rho\left(\tilde{z}^{k+1} - z^{k+1}\right)$$

Factorization caching

OSQP

Problem

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}x^T P x + q^T x \\ \text{subject to} & l \leq Ax \leq u \end{array}$$

Algorithm

Linear system solve

$$(x^{k+1}, \nu^{k+1}) \leftarrow \text{solve} \begin{bmatrix} P + \sigma I & A^T \\ A & -\frac{1}{\rho} I \end{bmatrix} \begin{bmatrix} x^{k+1} \\ \nu^{k+1} \end{bmatrix} = \begin{bmatrix} \sigma x^k - q \\ z^k - \frac{1}{\rho} y^k \end{bmatrix}$$

Easy operations

$$\tilde{z}^{k+1} \leftarrow z^k + \frac{1}{\rho}(\nu^{k+1} - y^k)$$

$$z^{k+1} \leftarrow \Pi\left(\tilde{z}^{k+1} + \frac{1}{\rho} y^k\right)$$

$$y^{k+1} \leftarrow y^k + \rho\left(\tilde{z}^{k+1} - z^{k+1}\right)$$

Warm starting

Factorization caching

OSQP

Problem

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}x^T P x + q^T x \\ \text{subject to} & l \leq Ax \leq u \end{array}$$

Algorithm

Linear system solve

$$(x^{k+1}, \nu^{k+1}) \leftarrow \text{solve} \begin{bmatrix} P + \sigma I & A^T \\ A & -\frac{1}{\rho} I \end{bmatrix} \begin{bmatrix} x^{k+1} \\ \nu^{k+1} \end{bmatrix} = \begin{bmatrix} \sigma x^k - q \\ z^k - \frac{1}{\rho} y^k \end{bmatrix}$$

Easy operations

$$\tilde{z}^{k+1} \leftarrow z^k + \frac{1}{\rho}(\nu^{k+1} - y^k)$$

$$z^{k+1} \leftarrow \Pi\left(\tilde{z}^{k+1} + \frac{1}{\rho} y^k\right)$$

$$y^{k+1} \leftarrow y^k + \rho\left(\tilde{z}^{k+1} - z^{k+1}\right)$$

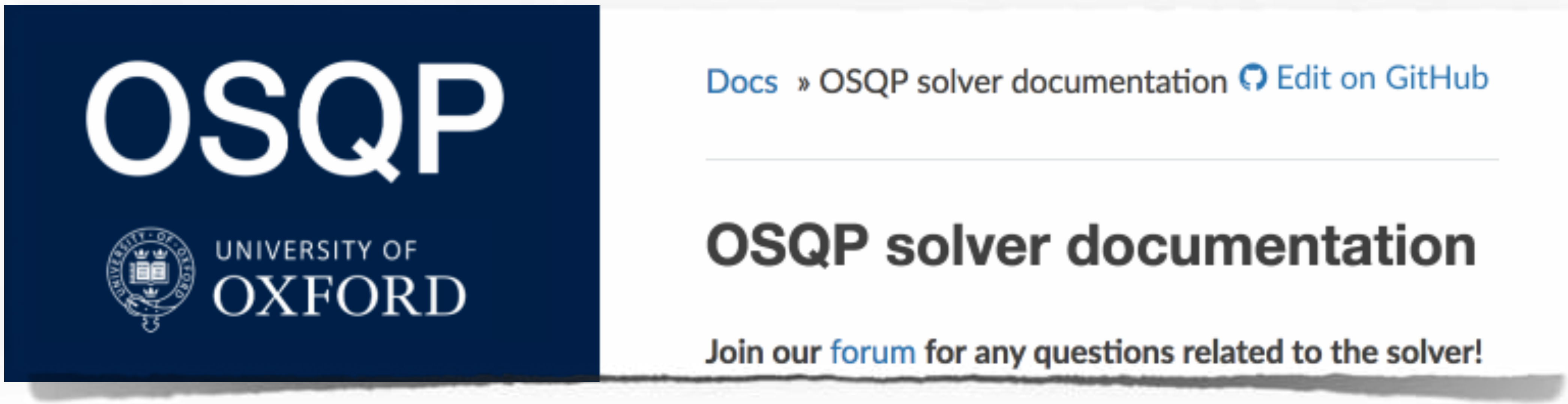
Solution polishing

Warm starting

Factorization caching

OSQP

`osqp.readthedocs.io`



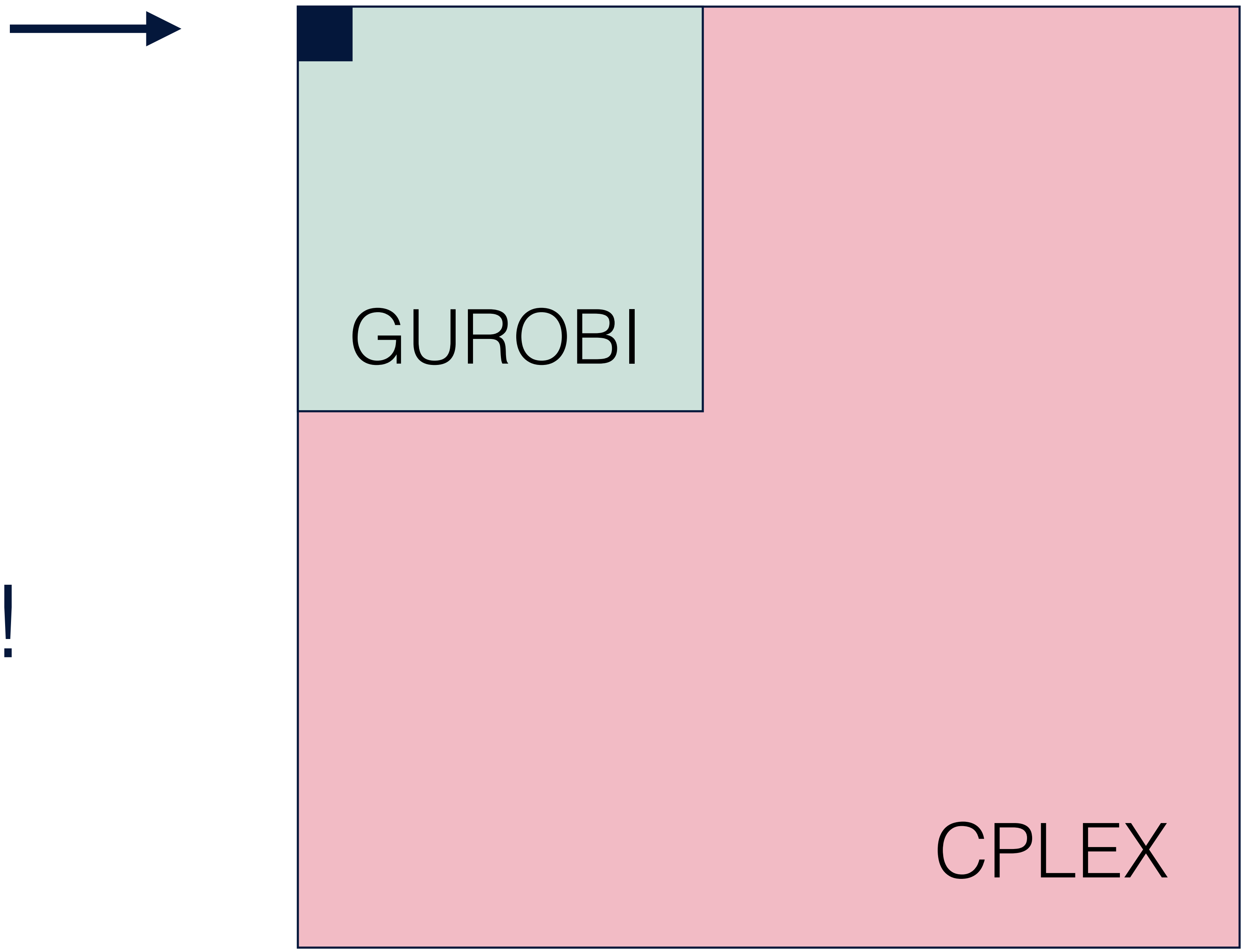
Library free

Detects Infeasibility

Embeddable

Compiled code size ~80kb

OSQP



300x
Reduction!

Interfaces

Languages

julia

python™

MATLAB®

Parsers

JuMP

CVXPY

YALMIP

OSQP interface

python

```python
# Create OSQP object
m = osqp.OSQP()

# Initialize solver
m.setup(P, q, A, l, u,
             settings)

# Solve
results = m.solve()

# Update cost with q_new
m.update(q=q_new)

# Solve again
results_new = m.solve()
```

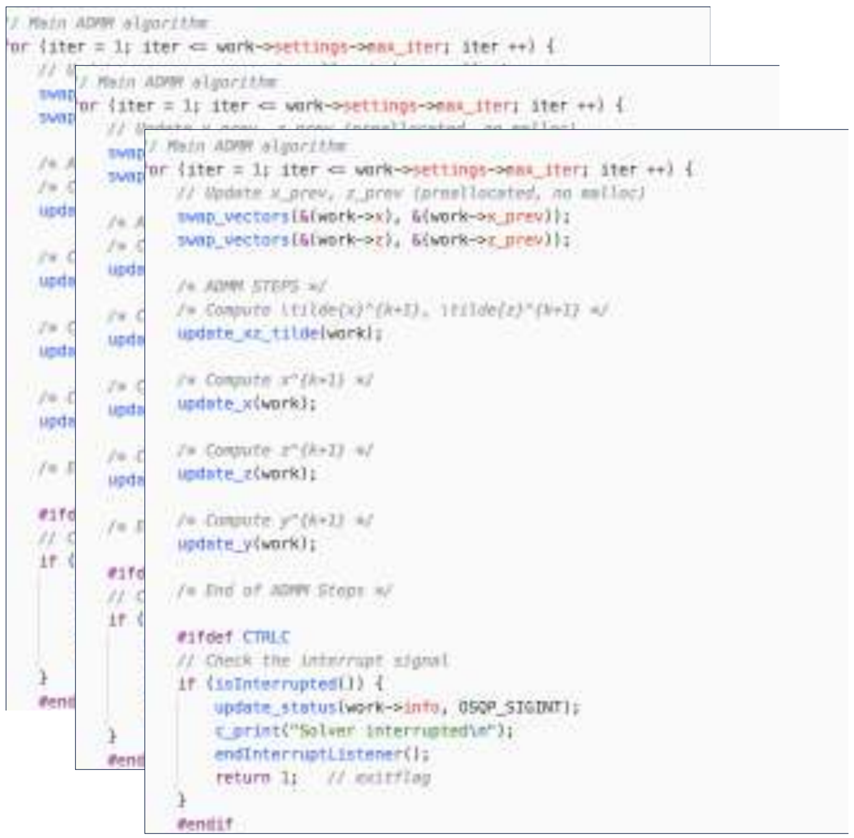
MATLAB

```matlab
% Create OSQP object
m = osqp();

% Initialize solver
m.setup(P, q, A, l, u,
             settings);

% Solve
results = m.solve();

% Update cost with q_new
m.update('q', q_new);

% Solve again
results_new = m.solve();
```

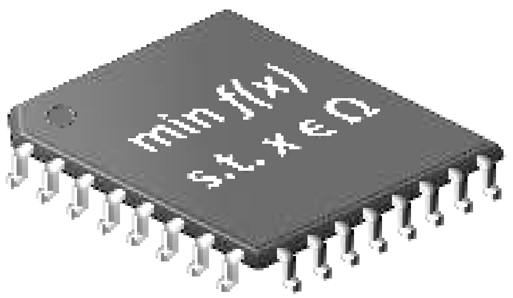
Code generation

```
# Create OSQP object
m = osqp.OSQP()

# Initialize solver
m.setup(P, q, A, l, u,
            settings)

# Generate C code
m.codegen('folder_name')
```

Optimized
C code



Embedded
hardware



Numerical Example

Lasso

minimize $\quad \|Ax - b\|_2^2 + \lambda \|x\|_1$

Features
n

Data points
$m = 100n$

Lasso

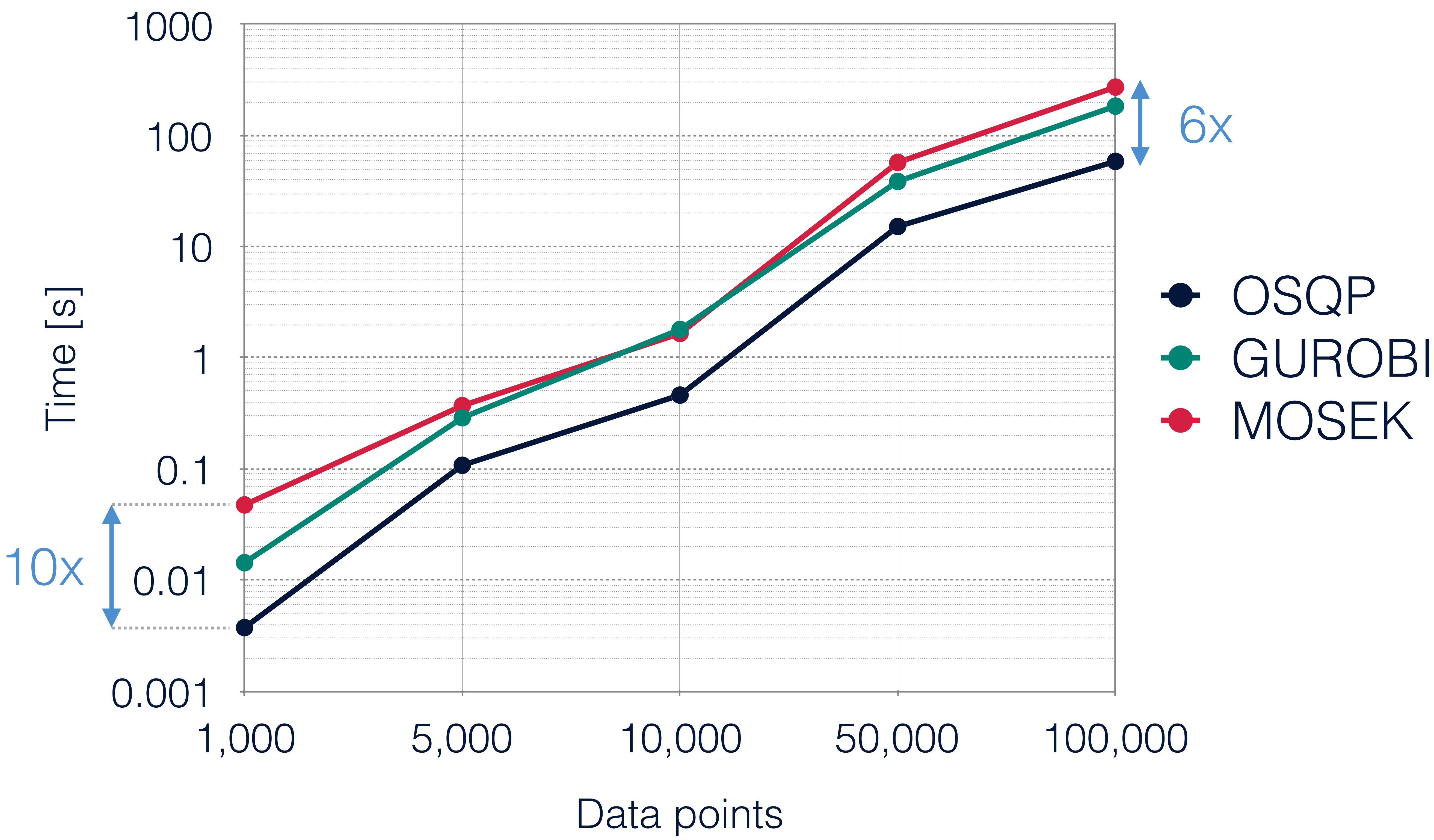
Weighting parameter

$$\text{minimize} \quad \|Ax - b\|_2^2 + \lambda \|x\|_1$$

Features
n

Data points
$m = 100n$

Lasso timings



OSQP
in
“meta-algorithms”

MIQP

$$
\begin{array}{ll}
\text{minimize} & \frac{1}{2}x^T P x + q^T x \\
\text{subject to} & l \leq Ax \leq u \\
 & x_i \in \mathbf{Z} \quad \forall i \in \mathbf{I}
\end{array}
$$

MIQP

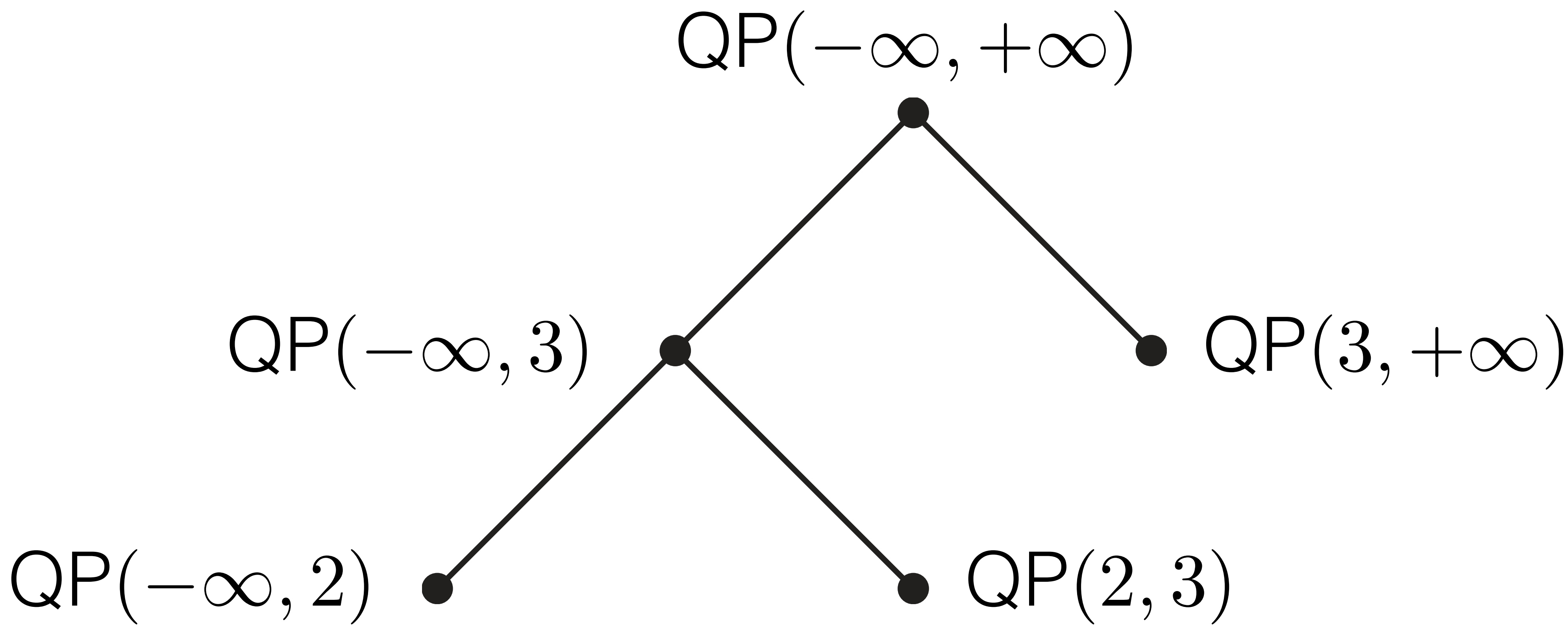
$$\begin{array}{ll} \text{minimize} & \frac{1}{2}x^T P x + q^T x \\ \text{subject to} & l \leq Ax \leq u \\ & x_i \in \mathbf{Z} \quad \forall i \in \mathbf{I} \end{array}$$

Integer constraints

MIQP

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}x^T P x + q^T x \\ \text{subject to} & l \leq Ax \leq u \\ & x_i \in \mathbf{Z} \quad \forall i \in \mathbf{I} \end{array}$$

Integer constraints



Inner QPs

$\text{QP}(\underline{x}, \overline{x})$

$$
\begin{array}{ll}
\text{minimize} & \frac{1}{2}x^T P x + q^T x \\
\text{subject to} & l \leq Ax \leq u \\
 & \underline{x}_i \leq x_i \leq \overline{x}_i \quad \forall i \in \mathbf{I}
\end{array}
$$

Inner QPs

QP$(\underline{x}, \overline{x})$

minimize $\frac{1}{2}x^T P x + q^T x$
subject to $l \leq Ax \leq u$
$\underline{x}_i \leq x_i \leq \overline{x}_i \quad \forall i \in \mathbf{I}$

Changing bounds

↓

Reusing factorization

Saving computations

Factorization caching

+

Warm starting

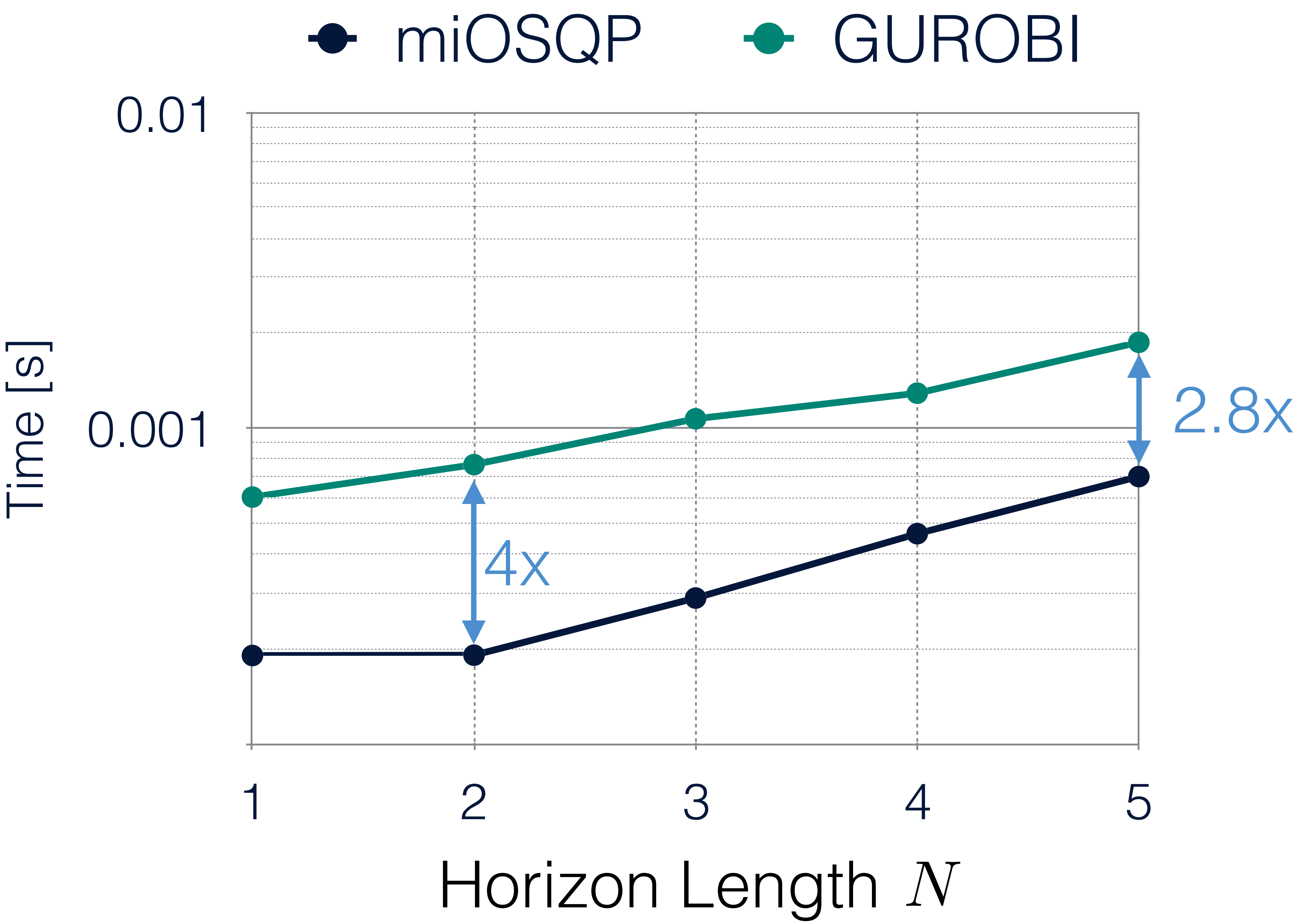
ADMM

QP$(\underline{x}, \overline{x})$

Repeated MIQPs

MIQP Timings

Hybrid Model Predictive Control



Simple
Python
Implementation!

Conclusions

Acknowledgements

Goran Banjac

Oxford

Nicholas Moehle

Stanford

Paul Goulart

Oxford

Alberto Bemporad

IMT Lucca

Stephen Boyd

Stanford

Final remarks

OSQP

Simple

Robust

Embeddable

Julia interface

Exploit Initialization

C code generation

New high-level algorithms

References

B. Stellato, G. Banjac, P. Goulart, A. Bemporad and S. Boyd. *OSQP: An Operator Splitting Solver for Quadratic Programs.* *(Coming soon!)*

G. Banjac, P. Goulart, B. Stellato, and S. Boyd. *Infeasibility detection in the alternating direction method of multipliers for convex optimization.* optimization-online.org, 2017

G. Banjac, B.Stellato, N. Moehle, P. Goulart, A. Bemporad and S. Boyd. *Embedded code generation using the OSQP solver.* IEEE Conference on Decision and Control (CDC) (submitted), 2017

B. Stellato, V. Naik, A. Bemporad, P. Goulart, and S. Boyd. *Embedded mixed-integer quadratic optimization using the OSQP solver.* IEEE Conference on Decision and Control (CDC) (submitted), 2017