

# Package ‘ROI.plugin.osqp’

October 12, 2022

**Version** 1.0-0

**Title** 'osqp' Plugin for the 'R' Optimization Infrastructure

**Description** Enhances the 'R' Optimization Infrastructure ('ROI') package with the quadratic solver 'OSQP'. More information about 'OSQP' can be found at <https://osqp.org>.

**Imports** methods, slam, ROI (>= 0.2-5), osqp, Matrix

**License** GPL-3

**URL** <http://roi.r-forge.r-project.org/>,  
<https://r-forge.r-project.org/projects/roi/>

**NeedsCompilation** no

**Author** Florian Schwendinger [aut, cre]

**Maintainer** Florian Schwendinger <FlorianSchwendinger@gmx.at>

**Repository** CRAN

**Date/Publication** 2020-09-14 10:20:03 UTC

## R topics documented:

ROI.plugin.osqp-package . . . . .	1
Example-1 . . . . .	3

<b>Index</b>	<b>4</b>
--------------	----------

---

ROI.plugin.osqp-package  
*osqp*

---

## Description

This package provides an interface to OSQP. The OSQP solver is a numerical optimization package or solving convex quadratic programs written in C and based on the alternating direction method of multipliers.

## Control Arguments

The following description of the control parameters is mostly copied from the **osqp** manual.

- `[] rho` ADMM step rho
- `[] sigma` ADMM step sigma
- `[] max_iter` maximum iterations
- `[] abs_tol` absolute convergence tolerance
- `[] rel_tol` relative convergence tolerance
- `[] eps_prim_inf` primal infeasibility tolerance
- `[] eps_dual_inf` dual infeasibility tolerance
- `[] alpha` relaxation parameter
- `[] linsys_solver` which linear systems solver to use, 0=QDLDL, 1=MKL Pardiso
- `[] delta` regularization parameter for polish
- `[] polish` boolean, polish ADMM solution
- `[] polish_refine_iter` iterative refinement steps in polish
- `[] verbose` boolean, write out progress
- `[] scaled_termination` boolean, use scaled termination criteria
- `[] check_termination` integer, check termination interval. If 0, termination checking is disabled
- `[] warm_start` boolean, warm start
- `[] scaling` heuristic data scaling iterations. If 0, scaling disabled
- `[] adaptive_rho` boolean, is rho step size adaptive?
- `[] adaptive_rho_interval` Number of iterations between rho adaptations rho. If 0, it is automatic
- `[] adaptive_rho_tolerance` Tolerance X for adapting rho. The new rho has to be X times larger or 1/X times smaller than the current one to trigger a new factorization
- `[] adaptive_rho_fraction` Interval for adapting rho (fraction of the setup time)

## References

Bartolomeo Stellato and Goran Banjac and Paul Goulart and Alberto Bemporad and Stephen Boyd. OSQP: An Operator Splitting Solver for Quadratic Programs <https://arxiv.org/abs/1711.08013>, 2017

Bartolomeo Stellato and Goran Banjac. OSQP “webpage” <https://osqp.org/>, 2019

**Description**

$$\text{maximize } x_1^2 + x_2^2 + x_3^2 - 5x_2$$

subject to :

$$-4x_1 - 3x_2 + \geq -8$$

$$2x_1 + x_2 + \geq 2$$

$$-2x_2 + x_3 \geq 0$$

$$x_1, x_2, x_3 \geq 0$$

**Examples**

```
require("ROI")
require("ROI.plugin.osqp")

A <- cbind(c(-4, -3, 0),
          c( 2,  1, 0),
          c( 0, -2, 1))
x <- OP(Q_objective(diag(3), L = c(0, -5, 0)),
       L_constraint(L = t(A),
                   dir = rep(">=", 3),
                   rhs = c(-8, 2, 0)))

opt <- ROI_solve(x, solver = "osqp", abs_tol = 1e-8, rel_tol = 1e-8)
opt
## Optimal solution found.
## The objective value is: -2.380952e+00
solution(opt)
## [1] 0.4761905 1.0476191 2.0952381
```

# Index

Example-1, [3](#)

ROI.plugin.osqp-package, [1](#)