

# Package: ROI.plugin.ecos (via r-universe)

August 14, 2024

**Version** 1.0-0

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

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

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

**Description** Enhances the 'R' Optimization Infrastructure ('ROI') package with the Embedded Conic Solver ('ECOS') for solving conic optimization problems.

**Imports** methods, slam, Matrix, ROI (>= 0.3-0), ECOSolveR (>= 0.5.0)

**License** GPL-3

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

**Repository** <https://r-forge.r-universe.dev>

**RemoteUrl** <https://github.com/r-forge/roi>

**RemoteRef** HEAD

**RemoteSha** f089cbe8d2717ead4862edf2c866ead61659e1f6

## Contents

Example-1 . . . . .	2
Example-2 . . . . .	2
Example-3 . . . . .	3
<b>Index</b>	<b>5</b>

---

 Example-1

 SOCP 1
 

---

**Description**

$$\begin{aligned} & \text{maximize } x + y \\ & \text{subject to } x^2 + y^2 \leq 1 \\ & x \geq 0, y \geq 0 \end{aligned}$$

**Examples**

```

library(ROI)
obj <- L_objective(c(1, 1))
## NOTE: chol(diag(2)) == diag(2)
con <- C_constraint(L = rbind(0, -diag(2)),
                   cones = K_soc(3),
                   rhs = c(1, 0, 0))
op <- OP(obj, con, maximum=TRUE)
x <- ROI_solve(op, solver="ecos")
x
## Optimal solution found.
## The objective value is: 1.414214e+00
solution(x)
## [1] 0.7071068 0.7071068

```

---

 Example-2

 SOCP 2
 

---

**Description**

The following example is also known as Problem 10 from the Hock-Schittkowski-Collection Hock and Schittkowski (1981).

$$\begin{aligned} & \text{minimize } x - y \\ & \text{subject to } -3x^2 + 2xy + 1 \geq 0 \end{aligned}$$

**References**

W. Hock, K. Schittkowski (1981): Test Examples for Nonlinear Programming Codes, Lecture Notes in Economics and Mathematical Systems, Vol. 187, Springer

**Examples**

```

library(ROI)
obj <- L_objective(c(1, -1))
L <- chol(rbind(c(3, -1), c(-1, 1)))
con <- C_constraint(L = rbind(0, -L),
                   cones = K_soc(3),
                   rhs = c(1, 0, 0))
op <- OP(objective = obj, constraints = con,
         bounds = V_bound(li=1:2, lb=rep(-Inf, 2)))
x <- ROI_solve(op, solver="ecos")
x
## Optimal solution found.
## The objective value is: -1.000000e+00
solution(x)
## [1] 1.996387e-10 1.000000e+00

```

Example-3

SOCP 3

**Description**

The following example is originally from the CVXOPT (<http://cvxopt.org/userguide/coneprog.html>) homepage.

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

subject to

$$\left\| \begin{array}{l} -13x_1 + 3x_2 + 5x_3 - 3 \\ -12x_1 + 12x_2 - 6x_3 - 2 \end{array} \right\|_2 \leq -12x_1 - 6x_2 + 5x_3 - 12$$

$$\left\| \begin{array}{l} -3x_1 + 6x_2 + 2x_3 \\ x_1 + 9x_2 + 2x_3 + 3 \\ -x_1 - 19x_2 + 3x_3 - 42 \end{array} \right\|_2 \leq -3x_1 + 6x_2 - 10x_3 + 27$$

**References**

[CVXOPT] Andersen, Martin S and Dahl, Joachim and Vandenberghe, Lieven (2016) CVXOPT: A Python package for convex optimization, version 1.1.8, <http://cvxopt.org/>

**Examples**

```

library(ROI)
lo <- L_objective(c(-2, 1, 5))
lc1 <- rbind(c(12, 6, -5), c(13, -3, -5), c(12, -12, 6))
lc2 <- rbind(c(3, -6, 10), c(3, -6, -2), c(-1, -9, -2), c(1, 19, -3))
lc <- C_constraint(L = rbind(lc1, lc2), cones = K_soc(c(3, 4)),
                  rhs=c(c(-12, -3, -2), c(27, 0, 3, -42)))
vb <- V_bound(li=1:3, lb=rep(-Inf, 3))
op <- OP(objective = lo, constraints = lc, bounds = vb)
x <- ROI_solve(op, solver="ecos")

```

```
x
## Optimal solution found.
## The objective value is: -3.834637e+01
solution(x)
## [1] -5.014767 -5.766924 -8.521796
```

# Index

Example-1, 2

Example-2, 2

Example-3, 3