Package 'ROI.plugin.deoptim'

July 21, 2025

Version 1.0-2

Title 'DEoptim' and 'DEoptimR' Plugin for the 'R' Optimization Interface

Description Enhances the R Optimization Infrastructure ('ROI') package with the 'DEoptim' and 'DEoptimR' package. 'DEoptim' is used for unconstrained optimization and 'DEoptimR' for constrained optimization.

Imports methods, stats, utils, ROI (>= 1.0-0), DEoptim, DEoptimR (>= 1.0-10)

License GPL-3

URL https://roigrp.gitlab.io,

https://gitlab.com/roigrp/solver/ROI.plugin.deoptim

NeedsCompilation no

Author Florian Schwendinger [aut, cre]

Maintainer Florian Schwendinger <FlorianSchwendinger@gmx.at>

Repository CRAN

Date/Publication 2023-07-25 17:33:42 UTC

Contents

`ROI.plugin.deoptim-package` *deoptimr*

Description

This package is part of the R Optimization Infrastructure **ROI**

References

Babu, B. V. and Angira, R. (2006) Modified differential evolution (MDE) for optimization of non-linear chemical processes. *Computers and Chemical Engineering* **30**, 989–1002.

Brest, J., Greiner, S., Boskovic, B., Mernik, M. and Zumer, V. (2006) Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems. *IEEE Transactions on Evolutionary Computation* **10**, 646–657.

Lampinen, J. and Zelinka, I. (1999). Mechanical engineering design optimization by differential evolution; in Corne, D., Dorigo, M. and Glover, F., Eds., *New Ideas in Optimization*. McGraw-Hill, pp. 127–146.

Price, K. V., Storn, R. M. and Lampinen, J. A. (2005) *Differential Evolution: A practical approach to global optimization*. Springer, Berlin, pp. 117–118.

Storn, R. (2008) Differential evolution research — trends and open questions; in Chakraborty, U. K., Ed., *Advances in differential evolution*. SCI 143, Springer-Verlag, Berlin, pp. 11–12.

Storn, R. and Price, K. (1997) Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization* **11**, 341–359.

Wu, G., Pedrycz, W., Suganthan, P. N. and Mallipeddi, R. (2015) A variable reduction strategy for evolutionary algorithms handling equality constraints. *Applied Soft Computing* **37**, 774–786.

Zhang, H. and Rangaiah, G. P. (2012) An efficient constraint handling method with integrated differential evolution for numerical and engineering optimization. *Computers and Chemical Engineering* **37**, 74–88.

Zielinski, K. and Laur, R. (2008) Stopping criteria for differential evolution in constrained single-objective optimization; in Chakraborty, U. K., Ed., *Advances in differential evolution*. SCI 143, Springer-Verlag, Berlin, pp. 111–138.

See Also

Function `JDEoptim()` in the **DEoptimR** package.

Example-1 *Banana*

Description

The following example is also known as Rosenbrock's banana function (https://en.wikipedia.org/wiki/Rosenbrock_function).

$$minimize\ f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$$

Solution: `c(1, 1)`

Examples

```
Sys.setenv(ROI_LOAD_PLUGINS = FALSE)
library(ROI)
library(ROI.plugin.deoptim)


f <- function(x) {
    return( 100 * (x[2] - x[1]^2)^2 + (1 - x[1])^2 )
}

x <- OP( objective = F_objective(f, n=2L, names=c("x_1", "x_2")),
         bounds = V_bound(li=1:2, ui=1:2, lb=c(-3, -3), ub=c(3,  3)) )

nlp <- ROI_solve(x, solver = "deoptim")
nlp
## Optimal solution found.
## The objective value is: 3.828383e-22
solution(nlp)
## x_1 x_2
##   1   1
```

Example-2 *Hock-Schittkowski-Collection Problem 16*

Description

The following example solves problem 16 from the Hock-Schittkowski-Collection.

$$minimize\ f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$$

$$subject\ to: \quad x_1 + x_2^2 \geq 0 \quad x_1^2 + x_2 \geq 0$$

$$-2 \geq x_1 \geq 0.5 \quad x_2 \geq 1$$

Solution: `c(0.5, 0.25)`

Examples

```
Sys.setenv(ROI_LOAD_PLUGINS = FALSE)
library(ROI)
library(ROI.plugin.deoptim)

f <- function(x) {
    return( 100 * (x[2] - x[1]^2)^2 + (1 - x[1])^2 )
}

f.gradient <- function(x) {
    return( c( -400 * x[1] * (x[2] - x[1] * x[1]) - 2 * (1 - x[1]),
               200 * (x[2] - x[1] * x[1])) )
}

x <- OP( objective = F_objective(f, n=2L, G=f.gradient),
         constraints = c(F_constraint(F=function(x) x[1] + x[2]^2, ">=", 0,
                                      J=function(x) c(1, 2*x[2])),
                         F_constraint(F=function(x) x[1]^2 + x[2], ">=", 0,
                                      J=function(x) c(2*x[1], x[2]))),
         bounds = V_bound(li=1:2, ui=1:2, lb=c(-2, -Inf), ub=c(0.5,  1)) )

nlp <- ROI_solve(x, solver="deoptimr", start=c(0.4, 0.3))
nlp
## Optimal solution found.
## The objective value is: 2.499999e-01
solution(nlp)
## [1] 0.5000001 0.2499994
```

Example-3 *Hock-Schittkowski-Collection Problem 36*

Description

The following example solves exmaple 36 from the Hock-Schittkowski-Collection.

$$minimize \ -x_1x_2x_3$$

$$subject\ to: \ x_1 + 2x_2 + x_3 \leq 72$$

$$0 \leq x_1 \leq 20, \ 0 \leq x_2 \leq 11, \ 0 \leq x_3 \leq 42$$

Examples

```
Sys.setenv(ROI_LOAD_PLUGINS = FALSE)
library(ROI)
library(ROI.plugin.deoptim)

hs036_obj <- function(x) {
    -x[1] * x[2] * x[3]
}
```

```
hs036_con <- function(x) {
    x[1] + 2 * x[2] + 2 * x[3]
}


x <- OP( objective = F_objective(hs036_obj, n = 3L),
         constraints = F_constraint(hs036_con, "<=", 72),
         bounds = V_bound(ub = c(20, 11, 42)) )

nlp <- ROI_solve(x, solver = "deoptimr", start = c(10, 10, 10),
                 max_iter = 2000)
nlp
## Optimal solution found.
## The objective value is: -3.300000e+03
solution(nlp, "objval")
## [1] -3300
solution(nlp)
## [1] 20 11 15
```

Index