Package 'ROI.plugin.neos'

January 20, 2025

Version 1.0-2

Title 'NEOS' Plug-in for the 'R' Optimization Interface

Description Enhances the 'R' Optimization Infrastructure ('ROI') package with a connection to the 'neos' server. 'ROI' optimization problems can be directly be sent to the 'neos' server and solution obtained in the typical 'ROI' style.

Imports stats, methods, utils, ROI (>= 1.0-0), xmlrpc2, xml2

Suggests slam

License GPL-3

Encoding UTF-8

URL https://roigrp.gitlab.io,
https://gitlab.com/roigrp/solver/ROI.plugin.neos

RoxygenNote 7.2.3

NeedsCompilation no

Author Ronald Hochreiter [aut],
Florian Schwendinger [aut, cre]

Maintainer Florian Schwendinger <FlorianSchwendinger@gmx.at>

Repository CRAN

Date/Publication 2023-11-25 22:10:02 UTC

Contents

Example-1 *Linear Problem 1*

Description

$$maximize \;\; 2x_1 + 4x_2 + 3x_3$$

$$subject\ to:$$

$$3x_1 + 4x_2 + 2x_3 \leq 60$$

$$2x_1 + x_2 + 2x_3 \leq 40$$

$$x_1 + 3x_2 + 2x_3 \leq 80$$

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

Examples

```
## Not run:
library(ROI)
mat <- matrix(c(3, 4, 2,
                2, 1, 2,
                1, 3, 2), nrow=3, byrow=TRUE)
x <- OP(objective = c(2, 4, 3),
        constraints = L_constraint(L = mat,
                                   dir = c("<=", "<=", "<="),
                                   rhs = c(60, 40, 80)),
        maximum = TRUE)


opt <- ROI_solve(x, solver = "neos", method = "scip")
opt
## Optimal solution found.
## The objective value is: 7.666667e+01
solution(opt)
## [1]  0.000000  6.666667 16.666667

## End(Not run)
```

neos_control *Neos Control Variables*

Description

The control variables for ROI.plugin.neos.

Usage

```
neos_control(
  method = "auto",
  wait = TRUE,
  email = "",
  password = "",
  user = "rneos",
  dry_run = FALSE,
  options = "",
  parameters = "",
  gdx = "",
  restart = "",
  wantgdx = "",
  wantlst = "",
  wantlog = "",
  comments = ""
)
```

Arguments

method	a chracter string giving the name of the solver to be selected on the NEOS server.
wait	a logical indicating whether the R interpreter should wait for the command to finish, or run it asynchronously. If TRUE **ROI** returns an object of class "neos_job".
email	a character string giving the email address.
password	a character string giving the account password.
user	a character string giving the username.
dry_run	a logical if TRUE **ROI** returns the solver call.
options	a character string (default is "") passed to options tag of the GAMS solver template.
parameters	a character string (default is "") passed to parameters tag of the GAMS solver template.
gdx	a character string (default is "") passed to gdx tag of the GAMS solver template.
restart	a character string (default is "") passed to restart tag of the GAMS solver template.

wantgdx	a character string (default is "") passed to wantgdx tag of the GAMS solver template.
wantlst	a character string (default is "") passed to wantlst tag of the GAMS solver template.
wantlog	a character string (default is "") passed to wantlog tag of the GAMS solver template.
comments	a character string (default is "") passed to comments tag of the GAMS solver template.

---

to_gams *Translate to GAMS*

Description

Translate a ROI OP to GAMS code. This function can translate optimization problems with linear or quadratic objective and linear or quadratic constraints.

Usage

```
to_gams(x)
```

Arguments

x	an **ROI** object of class OP.

Value

a character string giving the GAMS optimization model.

Examples

```
library("ROI")
mat <- matrix(c(3, 4, 2, 2, 1, 2, 1, 3, 2), nrow=3, byrow=TRUE)
x <- OP(objective = c(2, 4, 3),
        constraints = L_constraint(L = mat,
                                   dir = c("<=", "<=", "<="),
                                   rhs = c(60, 40, 80)),
        bounds = V_bound(ui = seq_len(3), ub = c(1000, Inf, 1000), nobj = 3),
        maximum = TRUE)
writeLines(to_gams(x))
```

Index