Package ‘highs’

August 30, 2022

Type Package

Title 'HiGHS' Optimization Solver

Version 0.1-2

Description R interface to 'HiGHS', an optimization solver for solving mixed integer optimization problems with quadratic or linear objective and linear constraints.

License GPL (>= 2)

Imports Rcpp (>= 1.0.7), checkmate

SystemRequirements Bash, PkgConfig, ZLIB (>=1.2.3), CMAKE (>=3.15), C++11

Suggests tinytest

LinkingTo Rcpp

RoxygenNote 7.2.0

NeedsCompilation yes

Author Florian Schwendinger [aut, cre],
Dirk Schumacher [aut],
Julian Hall [cph],
Ivet Galabova [cph],
Leona Gottwald [cph],
Michael Feldmeier [cph]

Maintainer Florian Schwendinger <florianschwendinger@gmx.at>

Repository CRAN

Date/Publication 2022-08-30 15:00:02 UTC

R topics documented:

highs_available_solver_options *Available Solver Options*

Description

Reference for the available solver options.

Usage

```
highs_available_solver_options()
```

Value

A `data.frame` containing the available solver options.

Examples

```
highs_available_solver_options()
```

highs_solve *Solve an Optimization Problems*

Description

Solve linear and quadratic mixed integer optimization problems.

Usage

```
highs_solve(
  Q = NULL,
  L,
  lower,
  upper,
  A,
  lhs,
  rhs,
  types,
  maximum = FALSE,
  offset = 0,
  control = list(),
  dry_run = FALSE
)
```

Arguments

`Q` a numeric symatric matrix giving the quadratic part of the objective.

`L` a numeric vector giving the linear part of the objective function.

`lower` a numeric vector giving the lower bounds of the variables.

`upper` a numeric vector giving the upper bounds of the variables.

`A` a numeric matrix giving the quadratic part of the objective.

`lhs` a numeric vector giving the left hand-side of the linear constraints.

`rhs` a numeric vector giving the right hand-side of the linear constraints.

`types` a integer vector or character vector giving the variable types. `'C'` or `'1'` for continuous, `'I'` or `'2'` for integer, `'SC'` or `'3'` for semi continuous, `'SI'` or `'4'` for semi integer and `'II'` or `'5'` for implicit integer.

`maximum` a logical if `TRUE` the solver searches for a maximum, if `FALSE` the solver searches for a minimum.

`offset` a numeric value giving the offset (default is `0`).

`control` a list giving additional options for the solver, see highs_available_solver_options or the `README` file for a list of all available options.

`dry_run` a logical if true only the model is returned.

Value

A `list` containing the result provided by the solver, containing the following named objects:

`primal_solution`
a numeric vector giving the primal solution.

`objective_value`
a numeric giving the objective value.

`status` an integer giving the status code

`status_message` a character string giving the status message (explaination of the `status_code`).

`solver_msg` a list giving the original (not canonicalized) solver message.

`info` a list giving additional information provided by the solver.

Additional information on can be found in the `README` file.

Examples

```
library("highs")
# Minimize:
#  x_0 +  x_1 + 3
# Subject to:
#               x_1 <=  7
#  5 <=  x_0 + 2x_1 <= 15
#  6 <= 3x_0 + 2x_1
#  0 <= x_0 <= 4
#  1 <= x_1
A <- rbind(c(0, 1), c(1, 2), c(3, 2))
```

```
s <- highs_solve(L = c(1.0, 1), lower = c(0, 1), upper = c(4, Inf),
                 A = A, lhs = c(-Inf, 5, 6), rhs = c(7, 15, Inf),
                 offset = 3)
s[["objective_value"]]
s[["primal_solution"]]

# Minimize:
#  -x_2 - 3x_3 + (1/2) * (2 x_1^2 - 2 x_1x_3 + 0.2 x_2^2 + 2 x_3^2)
# Subject to:
#  x_1 + x_3 <= 2
#  0 <= x
L <- c(0, -1, -3)
Q <- rbind(c(2, 0.0, -1), c(0, 0.2, 0), c(-1, 0.0, 2))
A <- cbind(1, 0, 1)
s <- highs_solve(Q = Q, L = L, lower = 0, A = A, rhs = 2)
s[["objective_value"]]
s[["primal_solution"]]
```

Index