Package: nlopt (via r-universe)

December 12, 2024

Type Package

Title Call Optimization Solvers with .nl Files

Version 0.1.1

Description The purpose of this library is to to call different optimization solvers (such as Gonzalez Rodriguez et al. (2022) <doi:10.1007/s10898-022-01229-w>, Tawarmalani and Sahinidis (2005) <doi:10.1007/s10107-005-0581-8>, and Byrd et al. (2006) <doi:10.1007/0-387-30065-1_4>) to solve problems given by a standard nl file.

License GPL-3

Language en-US

Imports stringr

RoxygenNote 7.2.3

Encoding UTF-8

NeedsCompilation no

Author Brais Gonzalez-Rodriguez [aut, cre]

Maintainer Brais Gonzalez-Rodriguez <braisgonzalez.rodriguez@usc.es>

Repository CRAN

Date/Publication 2023-11-17 20:00:05 UTC

Config/pak/sysreqs libicu-dev

Contents

`check_solver` *Checks if solver exists*

Description

Function that checks if the solver exists and it can be run without any issue

Usage

```
check_solver(solver)
```

Arguments

`solver`	name of the solver (it has to be in the PATH)

Value

TRUE if the solver exists and FALSE otherwise

Examples

```
check_solver(solver = "ipopt")
```

`example_sol_file` *Returns an example.sol file*

Description

Function that returns an example.sol file

Usage

```
example_sol_file()
```

Value

An example.sol file

Examples

```
example_sol_file()
```

get_available_options *Gets the available options for a solver*

Description

Function that gets a vector with all the available options for a solver

Usage

```
get_available_options(solver)
```

Arguments

solver	name of the solver (it has to be in the PATH)

Value

vector with all the available options for a solver

Examples

```
get_available_options(solver = "ipopt")
```

get_minlplib_problem *Gets problem from MINLPLib*

Description

Function that gets the corresponding problem from MINLPLib library

Usage

```
get_minlplib_problem(name)
```

Arguments

name	name of the problem

Value

TRUE if there is no error getting the problem and FALSE otherwise

Examples

```
get_minlplib_problem(name = "alkyl")
```

optimize *Optimizes the problem*

Description

Function that calls the corresponding solver with custom options to solve the problem given by the .nl file

Usage

```
optimize(solver, file, options)
```

Arguments

solver	name of the solver (it has to be in the PATH)
file	.nl file with the optimization problem. If the name of the file starts with MINLPLib:: then the problem will be downloaded from MINLPLib library
options	list with the options for the solver

Value

list with a string of the output given by the solver (output), the optimal value of the problem (objective), the status returned by the solver (status), the optimal primal solution (primal_solution), and the optimal dual solution (dual_solution)

Examples

```
optimize(solver = "ipopt", file = "MINLPLib::alkyl", options=list(max_cpu_time=300, outlev=3))
```

print_available_options *Prints the available options of the solver*

Description

Function that prints the available options of the solver

Usage

```
print_available_options(solver)
```

Arguments

solver	name of the solver (it has to be in the PATH)

Value

No return value

Examples

```
print_available_options(solver = "ipopt")
```

---

print_help *Prints the help of the solver*

Description

Function that prints the help of the solver

Usage

```
print_help(solver)
```

Arguments

solver	name of the solver (it has to be in the PATH)

Value

No return value

Examples

```
print_help(solver = "ipopt")
```

---

print_solver_version *Prints the version of the solver*

Description

Function that prints the version of the solver

Usage

```
print_solver_version(solver)
```

Arguments

solver	name of the solver (it has to be in the PATH)

Value

No return value

Examples

```
print_solver_version(solver = "ipopt")
```

`read_sol_file` *Reads the .sol file*

Description

Function that reads the corresponding .sol file

Usage

```
read_sol_file(solfile)
```

Arguments

`solfile`	.sol file with the solution

Value

list with a string of the output given by the solver (output), the optimal value of the problem (objective), the status returned by the solver (status), the optimal primal solution (primal_solution), and the optimal dual solution (dual_solution)

Examples

```
read_sol_file(solfile = "example.sol")
```

Index