Package 'Rglpk'

October 12, 2022

Version 0.6-4

Title R/GNU Linear Programming Kit Interface

Description R interface to the GNU Linear Programming Kit.
'GLPK' is open source software for solving large-scale linear programming (LP), mixed integer linear programming ('MILP') and other related problems.

Depends slam (>= 0.1-9)

SystemRequirements GLPK library package (e.g., libglpk-dev on Debian/Ubuntu)

License GPL-2 | GPL-3

URL http://R-Forge.R-project.org/projects/rglp/,
http://www.gnu.org/software/glpk/

NeedsCompilation yes

Author Stefan Theussl [aut, cre],
Kurt Hornik [aut],
Christian Buchta [ctb],
Florian Schwendinger [ctb],
Heinrich Schuchardt [ctb]

Maintainer Stefan Theussl <Stefan.Theussl@R-project.org>

Repository CRAN

Date/Publication 2019-02-09 15:33:16 UTC

R topics documented:

Rglpk_read_file *Interface to GLPK's file reader*

Description

High level R interface to the CPLEX_LP, MATHPROG and MPS reader of the GNU Linear Programming Kit (GLPK). Example data from the GLPK release is included in the `'./examples/'` sub-directory.

Usage

```
## File reader for various formats
Rglpk_read_file(file, type = c("MPS_fixed", "MPS_free", "CPLEX_LP", "MathProg"),
ignore_first_row = FALSE, verbose = FALSE)

## print method
## S3 method for class 'MP_data_from_file'
print(x, ...)
```

Arguments

`file`	a character string specifying the relative or absolute path to the model file.
`type`	a character string specifying the file format. This can be either `"MPS_fixed"`, `"MPS_free"`, `"CPLEX_LP"`, and GNU `"MathProg"`.
`ignore_first_row`	a logical indicating whether the first row of the model file should be ignored or not. Default: `FALSE`.
`verbose`	a logical for turning on/off additional solver output. Default: `FALSE`.
`x`	an object of class `"MP_data_from_file"`.
`...`	further arguments passed on to the print method.

Details

`Rglpk_read_file()` takes the path to a file as an argument and calls GLPK's file reader. The description of the linear or mixed integer linear program is returned as an object of class `"MP_data_from_file"`.

Value

`Rglpk_read_file()` returns the specification of a (mixed integer) linear program defined in `file` as an object of class `"MP_data_from_file"`. The returned object is a list containing the following components.

`objective`	a `"simple_triplet_matrix"` representing the coefficients to x in the objective function.

constraints	a list with three elements: a "simple_triplet_matrix" of coefficients, a character vector of constraint directions, and a numeric vector representing the right hand side.
bounds	a list containing two elements: lower and upper. Each of which contain a list specifying indices (ind) and corresponding bounds (val).
types	a character vector specifying whether the corresponding objective variable is of type binary ("B"), continuous ("C"), or integer ("I").
maximum	a logical indicating whether a minimum or a maximum is sought.

Further meta data is provided as attributes to the object.

Author(s)

Stefan Theussl

Examples

```
## read a CPLEX LP file
x <- Rglpk_read_file( system.file(file.path("examples", "plan.lp"), package
= "Rglpk"), type = "CPLEX_LP")
x
## optimal solution: 296.2166
Rglpk_solve_LP(x$objective, x$constraints[[1]], x$constraints[[2]],
               x$constraints[[3]], x$bounds, x$types, x$maximum)
## read a MATHPROG file
x <- Rglpk_read_file( system.file(file.path("examples", "assign.mod"), package
= "Rglpk"), type = "MathProg")
x
## optimal solution: 76
Rglpk_solve_LP(x$objective, x$constraints[[1]], x$constraints[[2]],
               x$constraints[[3]], x$bounds, x$types, x$maximum)
## read a MATHPROG file
x <- Rglpk_read_file( system.file(file.path("examples", "plan.mps"), package
= "Rglpk"), type = "MPS_fixed")
x
## optimal solution: 296.2166
Rglpk_solve_LP(x$objective, x$constraints[[1]], x$constraints[[2]],
               x$constraints[[3]], x$bounds, x$types, x$maximum)
```

Rglpk_solve_LP *Linear and Mixed Integer Programming Solver Using GLPK*

Description

High level R interface to the GNU Linear Programming Kit (GLPK) for solving linear as well as mixed integer linear programming (MILP) problems.

Usage

```
Rglpk_solve_LP(obj, mat, dir, rhs, bounds = NULL, types = NULL, max = FALSE,
               control = list(), ...)
```

Arguments

obj	a numeric vector representing the objective coefficients.
mat	a numeric vector or a (sparse) matrix of constraint coefficients. If the optimization problem is unconstrained then a matrix of dimension 0 times the number of objective variables is required.
dir	a character vector with the directions of the constraints. For a nonzero number of constraints each element must be one of "<", "<=", ">", ">=", or "==". Note, however, that the GLPK API only allows for non-strict inequalities. Strict inequalities are handled the same way as non-strict inequalities.
rhs	a numeric vector representing the right hand side of the constraints.
bounds	NULL (default) or a list with elements upper and lower containing the indices and corresponding bounds of the objective variables. The default for each variable is a bound between 0 and Inf.
types	a character vector indicating the types of the objective variables. types can be either "B" for binary, "C" for continuous or "I" for integer. By default NULL, taken as all-continuous. Recycled as needed.
max	a logical giving the direction of the optimization. TRUE means that the objective is to maximize the objective function, FALSE (default) means to minimize it.
control	a list of parameters to the solver. See *Details*.
...	a list of control parameters (overruling those specified in control).

Details

GLPK is open source. The current version can be found at https://www.gnu.org/software/glpk/glpk.html. Package **Rglpk** provides a high level solver function using the low level C interface of the GLPK solver. R interface packages which port all low level C routines of the GLPK API to R are also available. Consult the 'See Also' Section for references.

Matrix mat and obj may be sparse arrays or matrices (simple_triplet_matrix) as provided by the **slam** package.

The control argument can be used to set GLPK's many parameters. See the respective method section of the *GNU Linear Programming Kit Reference Manual* for further details. The following parameters are supported:

verbose: turn GLPK terminal output on (TRUE) or off (FALSE, the default).

presolve: turn presolver on (TRUE) or off (FALSE, the default).

tm_limit: time limit in milliseconds of call to optimizer. Can be any nonnegative integer. Default: 0 (use GLPK default).

canonicalize_status: a logical indicating whether to canonicalize GLPK status codes (on success Rglpk_solve_LP() returns code 0) or not (1). Default: TRUE.

Value

A list containing the optimal solution, with the following components.

`solution`	the vector of optimal coefficients
`objval`	the value of the objective function at the optimum
`status`	an integer with status information about the solution returned. If the control parameter `canonicalize_status` is set (the default) then it will return 0 for the optimal solution being found, and non-zero otherwise. If the control parameter is set to `FALSE` it will return the GLPK status codes.
`solution_dual`	variable reduced cost, if available (`NA` otherwise).
`auxiliary`	a list with two vectors each containing the values of the auxiliary variable associated with the respective constraint at solution, primal and dual (if available, `NA` otherwise).

Author(s)

Stefan Theussl and Kurt Hornik

References

GNU Linear Programming Kit (https://www.gnu.org/software/glpk/glpk.html).

GLPK Interface to R (https://cran.R-project.org/package=Rglpk).

See Also

glpk and **glpkAPI** for C API bindings; `lp` in package **lpSolve**; `ROI_solve` in package **ROI**; `Rsymphony_solve_LP` in package **Rsymphony**.

Examples

```
## Simple linear program.
## maximize:   2 x_1 + 4 x_2 + 3 x_3
## subject to: 3 x_1 + 4 x_2 + 2 x_3 <= 60
##             2 x_1 +   x_2 + 2 x_3 <= 40
##               x_1 + 3 x_2 + 2 x_3 <= 80
##               x_1, x_2, x_3 are non-negative real numbers

obj <- c(2, 4, 3)
mat <- matrix(c(3, 2, 1, 4, 1, 3, 2, 2, 2), nrow = 3)
dir <- c("<=", "<=", "<=")
rhs <- c(60, 40, 80)
max <- TRUE

Rglpk_solve_LP(obj, mat, dir, rhs, max = max)

## Simple mixed integer linear program.
## maximize:    3 x_1 + 1 x_2 + 3 x_3
## subject to: -1 x_1 + 2 x_2 +   x_3 <= 4
##                      4 x_2 - 3 x_3 <= 2
```

```
##                x_1 - 3 x_2 + 2 x_3 <= 3
##                x_1, x_3 are non-negative integers
##                x_2 is a non-negative real number

obj <- c(3, 1, 3)
mat <- matrix(c(-1, 0, 1, 2, 4, -3, 1, -3, 2), nrow = 3)
dir <- c("<=", "<=", "<=")
rhs <- c(4, 2, 3)
types <- c("I", "C", "I")
max <- TRUE

Rglpk_solve_LP(obj, mat, dir, rhs, types = types, max = max)

## Same as before but with bounds replaced by
## -Inf <  x_1 <= 4
##    0 <= x_2 <= 100
##    2 <= x_3 <  Inf

bounds <- list(lower = list(ind = c(1L, 3L), val = c(-Inf, 2)),
               upper = list(ind = c(1L, 2L), val = c(4, 100)))
Rglpk_solve_LP(obj, mat, dir, rhs, bounds, types, max)

## Examples from the GLPK manual
## Solver output enabled

## 1.3.1
## maximize:   10 x_1 + 6 x_2 + 4 x_3
## subject to:    x_1 +   x_2 +   x_3 <= 100
##             10 x_1 + 4 x_2 + 5 x_3 <= 600
##              2 x_1 + 2 x_2 + 6 x_3 <= 300
##                x_1,    x_2,    x_3 are non-negative real numbers

obj <- c(10, 6, 4)
mat <- matrix(c(1, 10, 2, 1, 4, 2, 1, 5, 6), nrow = 3)
dir <- c("<=", "<=", "<=")
rhs <- c(100, 600, 300)
max <- TRUE

Rglpk_solve_LP(obj, mat, dir, rhs, max = max, control = list("verbose" =
TRUE, "canonicalize_status" = FALSE))
```

Index