Package 'lpmodeler'

February 20, 2015

Title Modeler for linear programs (LP) and mixed integer linear programs (MILP)

Version 0.2-1

Date 2014-02-20

Author Cyrille Szymanski <cnszym@gmail.com> [aut]

Maintainer Cyrille Szymanski <cnszym@gmail.com>

Description lpmodeler is a set of user-friendly functions to simplify the modelling of linear programs (LP) and mixed integer programs (MIP). It provides a unified interface compatible with optimization packages: Rsymphony.

License GPL (>= 2) | BSD_2_clause + file LICENSE

Suggests Rsymphony (>= 0.1-17)

Imports slam (>= 0.1-31)

Depends R (>= 2.15.0)

NeedsCompilation no

Repository CRAN

Date/Publication 2014-02-21 05:59:10

R topics documented:

lpmodeler-package *Modeler for linear programs (LP) and mixed integer programs (MIP)*

Description

lpmodeler is a set of user-friendly functions to simplify the modelling of linear programs (LP) and mixed integer programs (MIP). It provides a unified interface compatible with optimization packages: Rsymphony.

Details

TODO

Author(s)

Cyrille Szymanski <cnszym@gmail.com>

References

TODO: Papers, books

See Also

TODO: R packages TODO: other software

Examples

```
# TODO
```

addConstraint *Add a new constraint to a LP or MIP*

Description

addConstraint creates a new constraint (<, >, <=, >=, ==) and adds it to a linear program (LP) or mixed integer program (MIP) represented by an object of class lpmodeler.

Usage

```
addConstraint(p, sense, rhs, coefs = NULL, name = NULL)
```

Arguments

p	an object of class lpmodeler
sense	sense of the constraint (<, >, <=, >=, == or !=)
rhs	right hand side of the constraint
coefs	optional coefficients of the variables in the left hand side of the constraint
name	an optional string to name the new constraint

Details

TODO

Value

An object of class lpmodeler.

Author(s)

Cyrille Szymanski <cnszym at gmail.com>

See Also

TODO

Examples

```
p <- newProblem()

# add variables x and y
p <- addVariable(p, "C", 5, "x")
p <- addVariable(p, "C", 4, "y")

# add the constraint: x + 2y >= 5
p <- addConstraint(p, ">=", 5, c(1, 2), name = "x+2y greater or equal than 5")

# add the empty constraint: <= 10
p <- addConstraint(p, "<=", 10, name = "less or equal than 10")
```

---

addVariable *Add a new variable to a LP or MIP*

Description

addVariable creates a new variable (continuous, integer or binary) and adds it to a linear program (LP) or mixed integer program (MIP) represented by an object of class lpmodeler.

Usage

```
addVariable(p, t = c("C", "I", "B"), o = 0, name = NULL)
```

Arguments

p	an object of class lpmodeler
t	type of the variable to create, C = continuous (default), I = integer, B = binary
o	numeric value representing the coefficient of the variable in the objective function (objective point), 0 by default
name	an optional string to name the new variable

Details

TODO

Value

An object of class lpmodeler.

Author(s)

Cyrille Szymanski <cnszym at gmail.com>

See Also

TODO

Examples

```
p <- newProblem()

# add an integer variable called x to the
# problem, and set its coefficient in the
# objective function to a value of 5.
p <- addVariable(p, "I", 5, "x")
```

---

checkDims *Check the consistency of the dimensions of a LP or MIP*

Description

checkDims checks the consistency of the dimensions of the matrices and vectors of a linear program (LP) or a mixed integer program (MIP) reprensented by an object of class lpmodeler.

Usage

```
checkDims(p)
```

Arguments

p	an object of class lpmodeler

Author(s)

Cyrille Szymanski <cnszym at gmail.com>

See Also

TODO

`mipSolve` *Solve a LP or a MIP*

Description

Solve a linear program (LP) or a mixed integer program (MIP). Find the values of the objective function and the associated variables using the specified solver.

Usage

```
mipSolve(p, solver = c("Rsymphony"), ...)
```

Arguments

`p`	an object of class `lpmodeler`
`solver`	name of the solver to use: Rsymphony (default)
`...`	other parameters passed to the solver

Value

The object returned by the solver

Author(s)

Cyrille Szymanski <cnszym at gmail.com>

See Also

TODO

Examples

```
# create and solve the following linear program:
# Simple mixed integer linear program.
# max:   3 x1 + 1 x2 + 3 x3
# s.t.: -1 x1 + 2 x2 +   x3 <= 4
#               4 x2 - 3 x3 <= 2
#          x1 - 3 x2 + 2 x3 <= 3
#          x1 >= 0 (integer)
#          x2 >= 0 (real)
#          x3 >= 0 (integer)
```

```
p <- newProblem()
p <- addVariable(p, "I", 3)
p <- addVariable(p, "C", 1)
p <- addVariable(p, "I", 3)
p <- addConstraint(p, "<=", 4, c(-1, 2, 1))
p <- addConstraint(p, "<=", 2, c(0, 4, -3))
p <- addConstraint(p, "<=", 3, c(1, -3, 2))
p <- addConstraint(p, ">=", 0, c(1, 0, 0))
p <- addConstraint(p, ">=", 0, c(0, 1, 0))
p <- addConstraint(p, ">=", 0, c(0, 0, 1))

if(require(Rsymphony))
  mipSolve(p)
```

`newProblem` *Create a new LP or MIP*

Description

`newProblem` creates a new and empty linear program (LP) or mixed integer program (MIP).

Usage

```
newProblem(max = T)
```

Arguments

`max`	`TRUE` (default) for a maximization problem, `FALSE` for a minization problem

Value

An object of class `lpmodeler`.

Author(s)

Cyrille Szymanski <cnszym at gmail.com>

See Also

TODO

Examples

```
p <- newProblem()
```

print.lpmodeler *Print a LP or MIP problem*

Description

Prints general information about a linear program (LP) or mixed integer program (MIP) represented by an object of class `lpmodeler`.

Usage

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

Arguments

`x`	an object of class `lpmodeler`
`...`	further arguments passed to or from other methods

Author(s)

Cyrille Szymanski <cnszym at gmail.com>

See Also

TODO

Examples

```
p <- newProblem()
print(p)
```

setCoeff *Set the coefficient of a variable in a contraint of a LP or MIP given their indexes*

Description

`setCoef` sets the coefficient of a variable in a contraint of a linear program (LP) or mixed integer program (MIP) given its numeric indexes in the problem matrix.

Usage

```
setCoeff(p, v, c, x)
```

Arguments

p	an object of class lpmodeler
v	index of the variable in the problem matrix (column)
c	index of the constraint in the problem matrix (row)
x	value of the coefficient

Value

An object of class lpmodeler.

Author(s)

Cyrille Szymanski <cnszym at gmail.com>

See Also

TODO

Examples

```
# TODO
```

---

setPoint *Set the coefficient of a variable in a contraint of a LP or MIP given their names*

Description

setPoint sets the coefficient of a variable in a contraint of a linear program (LP) or mixed integer program (MIP) given their names.

Usage

```
setPoint(p, v, c, x)
```

Arguments

p	an object of class lpmodeler
v	name of the variable in the problem matrix (column)
c	name of the constraint in the problem matrix (row)
x	value of the coefficient

Value

An object of class lpmodeler.

Author(s)

Cyrille Szymanski <cnszym at gmail.com>

See Also

TODO

Examples

```
# TODO
```

Index