Package ‘gMOIP’

February 20, 2017

Type Package

Title '2D plots of linear or integer programming models'

Version 1.1.0

Date 2017-01-22

URL https://github.com/relund/gMOIP/

BugReports https://github.com/relund/gMOIP/issues

Description Make 2D plots of the polyeder of a LP or IP problem, including integer points and iso profit curve. Can also make a plot of a bi-objective criterion space.

License GPL (>= 3.3.2)

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

Imports ggrepel

Depends ggplot2

Suggests tikzDevice

NeedsCompilation no

Author Lars Relund [aut, cre]

Maintainer Lars Relund <lars@relund.dk>

Repository CRAN

Date/Publication 2017-02-20 15:12:14

R topics documented:

`cornerPoints` *Calculate the corner points for the polytope Ax<=b, x>=0 (used some code from the* `intpoint` *package)*

Description

Calculate the corner points for the polytope Ax<=b, x>=0 (used some code from the `intpoint` package)

Usage

```
cornerPoints(A, b)
```

Arguments

`A`	A matrix.
`b`	Right hand side.

Value

A data frame with a corner point in each row.

Author(s)

Lars Relund <`lars@relund.dk`>

Examples

```
# Define the LP max/min coeff*x st. Ax<=b, x>=0
A <- matrix(c(9,10,2,4,-3,2), ncol = 2, byrow = TRUE)
b <- c(90,27,3)
coeff <- c(7.75, 10)

# Corner points of the polytope
cPoints<-cornerPoints(A, b)
# Integer points in the polytope
iPoints<-integerPoints(A, b)

# Plot of the polytope
plotPolytope(cPoints)
plotPolytope(points = cPoints)
plotPolytope(points = iPoints)
plotPolytope(cPoints, cPoints, iso = coeff, crit = "max")
plotPolytope(cPoints, iPoints, iso = coeff, crit = "max")
plotPolytope(cPoints, iPoints, showLbl = TRUE)
plotPolytope(cPoints, iPoints, iso = c(3,-3), crit = "min")
plotPolytope(cPoints, iPoints) + xlab("x") + ylab("y")
```

```
# Plot of criterion points given a bi-objective vector
zPoints<-criterionPoints(iPoints, c1 = c(coeff[1], 0), c2 = c(0, coeff[2]), crit = "max")
head(zPoints)
plotCriterion(zPoints)
plotCriterion(zPoints, addHull = FALSE, addTriangles = TRUE)
# mimimize
zPoints<-criterionPoints(zPoints, c1 = c(-1, 1), c2 = c(1, -1), crit = "min")
plotCriterion(zPoints, addHull = TRUE, addTriangles = TRUE, crit = "min")
# identify solutions (x1,x2) corresponding to (z1,z2)
plotPolytope(cPoints, zPoints, showLbl = TRUE)
plotCriterion(zPoints, addHull = TRUE, addTriangles = TRUE, crit = "min", showLbl = TRUE)
# max and min
zPoints<-criterionPoints(iPoints, c1 = c(-2, -1), c2 = c(1, 4), crit = "max")
plotPolytope(cPoints, zPoints, showLbl = TRUE, shape = zPoints$nD)
plotCriterion(zPoints, addHull = TRUE, addTriangles = TRUE, crit = "max", showLbl = TRUE)
zPoints<-criterionPoints(iPoints, c1 = c(-2, -1), c2 = c(1, 4), crit = "min")
plotCriterion(zPoints, addHull = TRUE, addTriangles = TRUE, crit = "min", showLbl = TRUE)

## Not run:
# Generate tikz file for LaTeX
library(tikzDevice)
tikz(file = "plot_polytope.tex", standAlone=F, width = 7, height = 6)
plotPolytope(cPoints, zPoints, showLbl = TRUE)
dev.off()

## End(Not run)
```

`criterionPoints` *Calculate the criterion points of a set of points and find the set of non-dominated points (pareto points) and classify them into extreme supported, non-extreme supported, non-supported.*

Description

Calculate the criterion points of a set of points and find the set of non-dominated points (pareto points) and classify them into extreme supported, non-extreme supported, non-supported.

Usage

```
criterionPoints(points, c1, c2, crit)
```

Arguments

`points`	A data frame with columns x1 and x2.
`c1`	2D vector for first criterion.
`c2`	2D vector for second criterion.
`crit`	Either max or min.

Value

A data frame with columns x1, x2, z1 = c1*x, z2 = c2*x, z = z1 + z2, lbl = label, nD = non-dominated, ext = extreme, nonExt = non-extreme supported.

Author(s)

Lars Relund <lars@relund.dk>

Examples

```
# Define the LP max/min coeff*x st. Ax<=b, x>=0
A <- matrix(c(9,10,2,4,-3,2), ncol = 2, byrow = TRUE)
b <- c(90,27,3)
coeff <- c(7.75, 10)

# Corner points of the polytope
cPoints<-cornerPoints(A, b)
# Integer points in the polytope
iPoints<-integerPoints(A, b)

# Plot of the polytope
plotPolytope(cPoints)
plotPolytope(points = cPoints)
plotPolytope(points = iPoints)
plotPolytope(cPoints, cPoints, iso = coeff, crit = "max")
plotPolytope(cPoints, iPoints, iso = coeff, crit = "max")
plotPolytope(cPoints, iPoints, showLbl = TRUE)
plotPolytope(cPoints, iPoints, iso = c(3,-3), crit = "min")
plotPolytope(cPoints, iPoints) + xlab("x") + ylab("y")

# Plot of criterion points given a bi-objective vector
zPoints<-criterionPoints(iPoints, c1 = c(coeff[1], 0), c2 = c(0, coeff[2]), crit = "max")
head(zPoints)
plotCriterion(zPoints)
plotCriterion(zPoints, addHull = FALSE, addTriangles = TRUE)
# mimimize
zPoints<-criterionPoints(zPoints, c1 = c(-1, 1), c2 = c(1, -1), crit = "min")
plotCriterion(zPoints, addHull = TRUE, addTriangles = TRUE, crit = "min")
# identify solutions (x1,x2) corresponding to (z1,z2)
plotPolytope(cPoints, zPoints, showLbl = TRUE)
plotCriterion(zPoints, addHull = TRUE, addTriangles = TRUE, crit = "min", showLbl = TRUE)
# max and min
zPoints<-criterionPoints(iPoints, c1 = c(-2, -1), c2 = c(1, 4), crit = "max")
plotPolytope(cPoints, zPoints, showLbl = TRUE, shape = zPoints$nD)
plotCriterion(zPoints, addHull = TRUE, addTriangles = TRUE, crit = "max", showLbl = TRUE)
zPoints<-criterionPoints(iPoints, c1 = c(-2, -1), c2 = c(1, 4), crit = "min")
plotCriterion(zPoints, addHull = TRUE, addTriangles = TRUE, crit = "min", showLbl = TRUE)

## Not run:
# Generate tikz file for LaTeX
library(tikzDevice)
tikz(file = "plot_polytope.tex", standAlone=F, width = 7, height = 6)
```

```
plotPolytope(cPoints, zPoints, showLbl = TRUE)
dev.off()

## End(Not run)
```

gMOIP *2D plots of linear/integer programming models.*

Description

Make 2D plots of the polyeder of a LP or IP problem, including integer points and iso profit curve. Can also make a plot of a bi-objective criterion space.

Author(s)

Lars Relund <lars@relund.dk> [aut, cre].

See Also

plotPolytope and plotCriterion.

Examples

```
# Define the LP max/min coeff*x st. Ax<=b, x>=0
A <- matrix(c(9,10,2,4,-3,2), ncol = 2, byrow = TRUE)
b <- c(90,27,3)
coeff <- c(7.75, 10)

# Corner points of the polytope
cPoints<-cornerPoints(A, b)
# Integer points in the polytope
iPoints<-integerPoints(A, b)

# Plot of the polytope
plotPolytope(cPoints)
plotPolytope(points = cPoints)
plotPolytope(points = iPoints)
plotPolytope(cPoints, cPoints, iso = coeff, crit = "max")
plotPolytope(cPoints, iPoints, iso = coeff, crit = "max")
plotPolytope(cPoints, iPoints, showLbl = TRUE)
plotPolytope(cPoints, iPoints, iso = c(3,-3), crit = "min")
plotPolytope(cPoints, iPoints) + xlab("x") + ylab("y")

# Plot of criterion points given a bi-objective vector
zPoints<-criterionPoints(iPoints, c1 = c(coeff[1], 0), c2 = c(0, coeff[2]), crit = "max")
head(zPoints)
plotCriterion(zPoints)
plotCriterion(zPoints, addHull = FALSE, addTriangles = TRUE)
# mimimize
```

```
zPoints<-criterionPoints(zPoints, c1 = c(-1, 1), c2 = c(1, -1), crit = "min")
plotCriterion(zPoints, addHull = TRUE, addTriangles = TRUE, crit = "min")
# identify solutions (x1,x2) corresponding to (z1,z2)
plotPolytope(cPoints, zPoints, showLbl = TRUE)
plotCriterion(zPoints, addHull = TRUE, addTriangles = TRUE, crit = "min", showLbl = TRUE)
# max and min
zPoints<-criterionPoints(iPoints, c1 = c(-2, -1), c2 = c(1, 4), crit = "max")
plotPolytope(cPoints, zPoints, showLbl = TRUE, shape = zPoints$nD)
plotCriterion(zPoints, addHull = TRUE, addTriangles = TRUE, crit = "max", showLbl = TRUE)
zPoints<-criterionPoints(iPoints, c1 = c(-2, -1), c2 = c(1, 4), crit = "min")
plotCriterion(zPoints, addHull = TRUE, addTriangles = TRUE, crit = "min", showLbl = TRUE)

## Not run:
# Generate tikz file for LaTeX
library(tikzDevice)
tikz(file = "plot_polytope.tex", standAlone=F, width = 7, height = 6)
plotPolytope(cPoints, zPoints, showLbl = TRUE)
dev.off()

## End(Not run)
```

`integerPoints` *Integer points inside the feasible region (Ax<=b).*

Description

Integer points inside the feasible region (Ax<=b).

Usage

```
integerPoints(A, b)
```

Arguments

`A`	A matrix.
`b`	Right hand side.

Value

A data frame with all integer points inside the feasible region (columns x1, x2, lbl).

Note

Do a simple enumeration of all integer points between min and max.

Author(s)

Lars Relund <`lars@relund.dk`>

Examples

```
# Define the LP max/min coeff*x st. Ax<=b, x>=0
A <- matrix(c(9,10,2,4,-3,2), ncol = 2, byrow = TRUE)
b <- c(90,27,3)
coeff <- c(7.75, 10)

# Corner points of the polytope
cPoints<-cornerPoints(A, b)
# Integer points in the polytope
iPoints<-integerPoints(A, b)

# Plot of the polytope
plotPolytope(cPoints)
plotPolytope(points = cPoints)
plotPolytope(points = iPoints)
plotPolytope(cPoints, cPoints, iso = coeff, crit = "max")
plotPolytope(cPoints, iPoints, iso = coeff, crit = "max")
plotPolytope(cPoints, iPoints, showLbl = TRUE)
plotPolytope(cPoints, iPoints, iso = c(3,-3), crit = "min")
plotPolytope(cPoints, iPoints) + xlab("x") + ylab("y")

# Plot of criterion points given a bi-objective vector
zPoints<-criterionPoints(iPoints, c1 = c(coeff[1], 0), c2 = c(0, coeff[2]), crit = "max")
head(zPoints)
plotCriterion(zPoints)
plotCriterion(zPoints, addHull = FALSE, addTriangles = TRUE)
# mimimize
zPoints<-criterionPoints(zPoints, c1 = c(-1, 1), c2 = c(1, -1), crit = "min")
plotCriterion(zPoints, addHull = TRUE, addTriangles = TRUE, crit = "min")
# identify solutions (x1,x2) corresponding to (z1,z2)
plotPolytope(cPoints, zPoints, showLbl = TRUE)
plotCriterion(zPoints, addHull = TRUE, addTriangles = TRUE, crit = "min", showLbl = TRUE)
# max and min
zPoints<-criterionPoints(iPoints, c1 = c(-2, -1), c2 = c(1, 4), crit = "max")
plotPolytope(cPoints, zPoints, showLbl = TRUE, shape = zPoints$nD)
plotCriterion(zPoints, addHull = TRUE, addTriangles = TRUE, crit = "max", showLbl = TRUE)
zPoints<-criterionPoints(iPoints, c1 = c(-2, -1), c2 = c(1, 4), crit = "min")
plotCriterion(zPoints, addHull = TRUE, addTriangles = TRUE, crit = "min", showLbl = TRUE)

## Not run:
# Generate tikz file for LaTeX
library(tikzDevice)
tikz(file = "plot_polytope.tex", standAlone=F, width = 7, height = 6)
plotPolytope(cPoints, zPoints, showLbl = TRUE)
dev.off()

## End(Not run)
```

`plotCriterion` *Create a plot of criterion space*

Description

Create a plot of criterion space

Usage

```
plotCriterion(points, showLbl = FALSE, addTriangles = FALSE,
  addHull = TRUE, crit = "max", latex = FALSE)
```

Arguments

`points`	Data frame with criterion points
`showLbl`	Add labels to the points.
`addTriangles`	Add triangles to the non-dominated points
`addHull`	Add the convex hull of the non-dominated points and rays.
`crit`	Either min or max. The objective the criterion points are classified as. Note must be the same as used in criterionPoints.
`latex`	If true make latex math labels for TikZ.

Value

The ggplot2 object.

Author(s)

Lars Relund `<lars@relund.dk>`

Examples

```
# Define the LP max/min coeff*x st. Ax<=b, x>=0
A <- matrix(c(9,10,2,4,-3,2), ncol = 2, byrow = TRUE)
b <- c(90,27,3)
coeff <- c(7.75, 10)

# Corner points of the polytope
cPoints<-cornerPoints(A, b)
# Integer points in the polytope
iPoints<-integerPoints(A, b)

# Plot of the polytope
plotPolytope(cPoints)
plotPolytope(points = cPoints)
plotPolytope(points = iPoints)
plotPolytope(cPoints, cPoints, iso = coeff, crit = "max")
plotPolytope(cPoints, iPoints, iso = coeff, crit = "max")
plotPolytope(cPoints, iPoints, showLbl = TRUE)
plotPolytope(cPoints, iPoints, iso = c(3,-3), crit = "min")
plotPolytope(cPoints, iPoints) + xlab("x") + ylab("y")

# Plot of criterion points given a bi-objective vector
```

```
zPoints<-criterionPoints(iPoints, c1 = c(coeff[1], 0), c2 = c(0, coeff[2]), crit = "max")
head(zPoints)
plotCriterion(zPoints)
plotCriterion(zPoints, addHull = FALSE, addTriangles = TRUE)
# mimimize
zPoints<-criterionPoints(zPoints, c1 = c(-1, 1), c2 = c(1, -1), crit = "min")
plotCriterion(zPoints, addHull = TRUE, addTriangles = TRUE, crit = "min")
# identify solutions (x1,x2) corresponding to (z1,z2)
plotPolytope(cPoints, zPoints, showLbl = TRUE)
plotCriterion(zPoints, addHull = TRUE, addTriangles = TRUE, crit = "min", showLbl = TRUE)
# max and min
zPoints<-criterionPoints(iPoints, c1 = c(-2, -1), c2 = c(1, 4), crit = "max")
plotPolytope(cPoints, zPoints, showLbl = TRUE, shape = zPoints$nD)
plotCriterion(zPoints, addHull = TRUE, addTriangles = TRUE, crit = "max", showLbl = TRUE)
zPoints<-criterionPoints(iPoints, c1 = c(-2, -1), c2 = c(1, 4), crit = "min")
plotCriterion(zPoints, addHull = TRUE, addTriangles = TRUE, crit = "min", showLbl = TRUE)

## Not run:
# Generate tikz file for LaTeX
library(tikzDevice)
tikz(file = "plot_polytope.tex", standAlone=F, width = 7, height = 6)
plotPolytope(cPoints, zPoints, showLbl = TRUE)
dev.off()

## End(Not run)
```

`plotPolytope` *Create a plot of a polytope (bounded convex set)*

Description

If an iso profit line added then the max/min is obtained among the `points`.

Usage

```
plotPolytope(cPoints = NULL, points = NULL, showLbl = FALSE, iso = NULL,
  crit = "max", latex = FALSE, ...)
```

Arguments

`cPoints`	Corner points in the polytope.
`points`	Points to plot (e.g integer points inside the polytope or corner points).
`showLbl`	Add labels to the points (only if points have a `lbl` column).
`iso`	NULL or if 2D vector add the iso profit line the the solution plot.
`crit`	Either max or min (only used if add the iso profit line)
`latex`	If true make latex math labels for TikZ.
`...`	Arguments passed to the aes function in geom_point.

Value

The ggplot2 object.

Author(s)

Lars Relund <lars@relund.dk>

Examples

```
# Define the LP max/min coeff*x st. Ax<=b, x>=0
A <- matrix(c(9,10,2,4,-3,2), ncol = 2, byrow = TRUE)
b <- c(90,27,3)
coeff <- c(7.75, 10)

# Corner points of the polytope
cPoints<-cornerPoints(A, b)
# Integer points in the polytope
iPoints<-integerPoints(A, b)

# Plot of the polytope
plotPolytope(cPoints)
plotPolytope(points = cPoints)
plotPolytope(points = iPoints)
plotPolytope(cPoints, cPoints, iso = coeff, crit = "max")
plotPolytope(cPoints, iPoints, iso = coeff, crit = "max")
plotPolytope(cPoints, iPoints, showLbl = TRUE)
plotPolytope(cPoints, iPoints, iso = c(3,-3), crit = "min")
plotPolytope(cPoints, iPoints) + xlab("x") + ylab("y")

# Plot of criterion points given a bi-objective vector
zPoints<-criterionPoints(iPoints, c1 = c(coeff[1], 0), c2 = c(0, coeff[2]), crit = "max")
head(zPoints)
plotCriterion(zPoints)
plotCriterion(zPoints, addHull = FALSE, addTriangles = TRUE)
# mimimize
zPoints<-criterionPoints(zPoints, c1 = c(-1, 1), c2 = c(1, -1), crit = "min")
plotCriterion(zPoints, addHull = TRUE, addTriangles = TRUE, crit = "min")
# identify solutions (x1,x2) corresponding to (z1,z2)
plotPolytope(cPoints, zPoints, showLbl = TRUE)
plotCriterion(zPoints, addHull = TRUE, addTriangles = TRUE, crit = "min", showLbl = TRUE)
# max and min
zPoints<-criterionPoints(iPoints, c1 = c(-2, -1), c2 = c(1, 4), crit = "max")
plotPolytope(cPoints, zPoints, showLbl = TRUE, shape = zPoints$nD)
plotCriterion(zPoints, addHull = TRUE, addTriangles = TRUE, crit = "max", showLbl = TRUE)
zPoints<-criterionPoints(iPoints, c1 = c(-2, -1), c2 = c(1, 4), crit = "min")
plotCriterion(zPoints, addHull = TRUE, addTriangles = TRUE, crit = "min", showLbl = TRUE)

## Not run:
# Generate tikz file for LaTeX
library(tikzDevice)
tikz(file = "plot_polytope.tex", standAlone=F, width = 7, height = 6)
plotPolytope(cPoints, zPoints, showLbl = TRUE)
```

```
dev.off()

## End(Not run)
```

Index