

# Package ‘orloca’

October 24, 2017

**Type** Package

**Depends** methods, png, ucminf

**Suggests** grDevices, graphics

**Title** Operations Research LOCational Analysis Models

**Version** 4.3

**Date** 2017-10-24

**Author** Manuel Munoz-Marquez <manuel.munoz@uca.es>

**Maintainer** Manuel Munoz-Marquez <manuel.munoz@uca.es>

**Description** Objects and methods to handle and solve the min-sum location problem, also known as Fermat-Weber problem. The min-sum location problem search for a point such that the weighted sum of the distances to the demand points are minimized. See “The Fermat-Weber location problem revisited” by Brimberg, *Mathematical Programming*, 1, pg. 71-76, 1995. <DOI:10.1007/BF01592245>. General global optimization algorithms are used to solve the problem, along with the ad-hoc Weiszfeld method, see “Sur le point pour lequel la Somme des distances de n points donnees est minimum”, by Weiszfeld, *Tohoku Mathematical Journal, First Series*, 43, pg. 355-386, 1937.

**License** GPL (>= 3)

**URL** <http://knuth.uca.es/orloca>

**RoxygenNote** 6.0.1

**Repository** CRAN

**Repository/R-Forge/Project** orloca

**Repository/R-Forge/Revision** 25

**Repository/R-Forge/DateTimeStamp** 2017-10-24 15:15:40

**Date/Publication** 2017-10-24 21:45:53 UTC

**NeedsCompilation** no

## R topics documented:

orloca-package	2
andalusia-data	3
as-methods	4
contour.loca.p	5
loca.p	6
persp.loca.p	7
plot	8
rloca.p	9
zsum	10
zsumgra	11
zsummin	12

<b>Index</b>	<b>14</b>
--------------	-----------

---

orloca-package	<i>Operations Research LOCational Analysis Models</i>
----------------	---

---

### Description

Objects and methods to handle and solve the min-sum location problem, also known as Fermat-Weber problem.

### Details

The min-sum location problem search for a point such that the weighted sum of the distances to the demand points are minimized. See "The Fermat-Weber location problem revisited" by Brimberg, *Mathematical Programming*, 1, pg. 71-76, 1995. DOI: 10.1007/BF01592245.

General global optimization algorithms are used to solve the problem, along with the adhoc Weiszfeld method, see "Sur le point pour lequel la Somme des distances de n points donnees est minimum", by Weiszfeld, *Tohoku Mathematical Journal, First Series*, 43, pg. 355-386, 1937.

Package: orloca

Type: Package

Version: 4.3

Date: 2017-10-20

License: GPL (>= 3)

The package provides a class (`loca.p`) that represents a location problem with a finite set of demand points over the plane. Also, it is possible to plot the points and the objective function. Such objective function is the total weighted distances travelled by all the customers to the service.

Non-planar location problems could be handle in future versions of the package.

For a demo, load the package with `library(orloca)`, and use `demo(orloca)`.

The package is ready for internationalization. The authors ask for translated version of the `.mo` file to include in the package.

### Author(s)

Manuel Munoz-Marquez <manuel.munoz@uca.es>

Mantainer: Manuel Munoz-Marquez <manuel.munoz@uca.es>

### References

[1] Love, R. F., Morris, J. G., Wesolowsky, G. O. *Facilities Location: Chapter 2: Introduction to Single-Facility Location*, 1988, North-Holland

[2] <http://knuth.uca.es/orloca>

### See Also

Para la version en espanol, instale el paquete `orloca.es` y consulte la ayuda sobre [orloca.es-package](#). (For the spanish version, install the package `orloca.es` and see the help about [orloca.es-package](#)).

### Examples

```
# A new unweighted loca.p object
o <- loca.p(x = c(-1, 1, 1, -1), y = c(-1, -1, 1, 1))

# Compute the sum of distances to point (3, 4)
zsum(o, 3, 4)

# Compute the sum of distances to point (3, 4) using lp norm
zsum(o, 3, 4, lp=2.5)

# Solve the optimization problem
zsummin(o)
# Contour plot
contour(o)

# Make a demo of the package
demo(orloca)
```

---

andalusia-data

*Cities of Andalusia*

---

### Description

The `'andalusia'` data frame has 12 rows and 4 columns, which are the geographical position of the main capital cities of andalusia.

**Format**

name: The name of the city or relative position label.

x: The x coordinate of points.

y: The y coordinate of points.

city: If yes the point is a city in other case is a limit.

**Usage**

```
data('andalusia')
```

**Source**

The data are taken from wikipedia.

**See Also**

See also [orloca-package](#).

---

as-methods

*as-methods*


---

**Description**

Conversions between loca.p class and some others

**Arguments**

x is the object to convert to the new class object.

row.names Unused.

optional Unused.

... Other arguments, unused.

**Details**

Methods to convert from and to loca.p class.

NA's values are not allowed in any of the arguments.

The matrix to convert into loca.p must have at least two columns. The first column will be consider as the x coordinates, the second as the y coordinates, and the third (if given) as the values of w.

The data.frame to convert into loca.p must have at least an x column for x coordinates, and an y column for y coordinates. Optionally, it can have w column, as the values of w.

**Value**

If the arguments have valid values, it returns a new object of the new class.

**See Also**

See also [loca.p](#)

**Examples**

```
# A new unweighted loca.p object
loca <- loca.p(x = c(-1, 1, 1, -1), y = c(-1, -1, 1, 1))

# Conversion to matrix
m <- as.matrix(loca)

# Show matrix
m

# Conversion from matrix
as.loca.p(m)
```

---

contour.loca.p

*Plots of the min-sum objective function*


---

**Description**

contour provides a graphical representations of min-sum function (zsum).

**Usage**

```
## S3 method for class 'loca.p'
contour(x, lp = numeric(0), xmin = min(min(x@x), xleft),
        xmax = max(max(x@x), xright), ymin = min(min(x@y), ybottom),
        ymax = max(max(x@y), ytop), n = 100, img = NULL, xleft = min(x@x),
        ybottom = min(x@y), xright = max(x@x), ytop = max(x@y), ...)
```

**Arguments**

x	The loca.p object to compute the objective.
lp	If given, then $l_p$ norm will be used instead of the Euclidean norm.
xmin	The minimum value for x axis.
xmax	The maximum value for x axis.
ymin	The minimum value for y axis.
ymax	The maximum value for y axis.
n	The number of divisions for grid.
img	A raster image to plot on background.
xleft	The left position of the image.
ybottom	The bottom position of the image.
xright	The right position of the image.
ytot	The top position of the image.
...	Other options.

**Details**

If  $p < 1$  then  $l_p$  are not a norm, so only  $p \geq 1$  are valid values.

**Value**

`contour.loca.p` plots a contour plot of min-sum function (`zsum`).

**See Also**

See also [orloca-package](#), [plot.loca.p](#) and [loca.p](#).

**Examples**

```
# A new unweighted loca.p object
loca <- loca.p(x = c(-1, 1, 1, -1), y = c(-1, -1, 1, 1))

# The contour plot of min-sum function for loca (a loca.p object)
contour(loca)
```

---

loca.p

*loca.p class for Operations Research LOCational Analysis*


---

**Description**

An object of class `loca.p` represents a weighted location problem with a finite demand points set. The [orloca-package](#) is mainly devoted to deals with location problems.

**Arguments**

<code>x</code>	is a vector of the x coordinates of the demand points.
<code>y</code>	is a vector of the y coordinates of the demand points.
<code>w</code>	is a vector of weights of the demand points. If <code>w</code> is omitted then all weights are considered as 1.
<code>label</code>	If given, it is the label of the new object.

**Details**

The main generator of the `loca.p` class is `loca.p(x, y, w = numeric(0), label = "")`. An alternative form is `new("loca.p", x, y, w = numeric(0), label = "")`.

The lengths of `x` and `y` vector must be equals. The length of `w` must be equal to the previous ones or must be 0. NA's values are not allowed at any of the arguments.

**Value**

If the arguments have valid values, it returns a new object of class `loca.p`, else it returns an error. `summary(x)` returns a summary of the `x` `loca.p` object and `print(x)` prints a summary of the `x` `loca.p` object.

**See Also**

See also [orloca-package](#).

**Examples**

```
# A new unweighted loca.p object
loca <- loca.p(x = c(-1, 1, 1, -1), y = c(-1, -1, 1, 1))
# or
loca <- new("loca.p", x = c(-1, 1, 1, -1), y = c(-1, -1, 1, 1))

# An example with weights and name
locb <- new("loca.p", x = c(-1, 1, 1, -1), y = c(-1, -1, 1, 1),
w = c(1, 2, 1, 2), label = "Weighted case")
```

---

persp.loca.p

*Plots of the min-sum objective function*

---

**Description**

persp provides a graphical representations of min-sum function (zsum).

**Usage**

```
## S3 method for class 'loca.p'
persp(x, lp = numeric(0), xmin = min(x@x),
      xmax = max(x@x), ymin = min(x@y), ymax = max(x@y), n = 100, ...)
```

**Arguments**

x	The loca.p object to compute the objective.
lp	If given, then $l_p$ norm will be used instead of the Euclidean norm.
xmin	The minimum value for x axis.
xmax	The maximum value for x axis.
ymin	The minimum value for y axis.
ymax	The maximum value for y axis.
n	The number of divisions for grid.
...	Other options.

**Details**

If  $p < 1$  then  $l_p$  are not a norm, so only  $p \geq 1$  are valid values.

**Value**

A plot a 3D plot or min-sum function.

**See Also**

See also [orloca-package](#), [plot.loca.p](#) and [loca.p](#).

**Examples**

```
# A new unweighted loca.p object
loca <- loca.p(x = c(-1, 1, 1, -1), y = c(-1, -1, 1, 1))

# The 3D graphics
persp(loca)
```

---

plot

*plot of loca.p class objects*


---

**Description**

This method provides a graphical representations of an object of class `loca.p`.

**Usage**

```
## S3 method for class 'loca.p'
plot(x, xlab = "", ylab = "",
     main = gettext("Plot of loca.p object", domain = "R-orloca"), img = NULL,
     xlim = c(min(xleft, min(x@x)), max(xright, max(x@x))),
     ylim = c(min(ybottom, min(x@y)), max(ytop, max(x@y))), xleft = min(x@x),
     ybottom = min(x@y), xright = max(x@x), ytop = max(x@y), ...)
```

**Arguments**

<code>x</code>	The <code>loca.p</code> object to plot.
<code>xlab</code>	The label for x axis.
<code>ylab</code>	The label for y axis.
<code>main</code>	The main title for the plot.
<code>img</code>	A raster image to plot on background.
<code>xlim</code>	Limit over the x axes of the plot.
<code>ylim</code>	Limit over the y axes of the plot.
<code>xleft</code>	The left position of the image.
<code>ybottom</code>	The bottom position of the image.
<code>xright</code>	The right position of the image.
<code>ytop</code>	The top position of the image.
<code>...</code>	Other graphical options.



**Details**

The function plots the demand points with automatic limits evaluation.

**Value**

The function plots the required graphics.

**See Also**

See also [orloca-package](#), [loca.p](#) and [plot](#).

**Examples**

```
# A new unweighted loca.p object
loca <- loca.p(x = c(-1, 1, 1, -1), y = c(-1, -1, 1, 1))
# The plot of loca object
plot(loca)
```

---

rloca.p

*Random instances generator of loca.p class object*


---

**Description**

rloca.p function returns a random instance of loca.p class object at a given rectangular region.

**Usage**

```
rloca.p(n, xmin = 0, xmax = 1, ymin = 0, ymax = 1, groups = 0,
        xgmin = xmin, xgmax = xmax, ygmin = ymin, ygmax = ymax)
```

**Arguments**

n	The number of demand points.
xmin	Minimum value for the x coordinates of the demand points.
xmax	Maximum value for the x coordinates of the demand points.
ymin	Minimum value for the y coordinates of the demand points.
ymax	Maximum value for the y coordinates of the demand points.
groups	The number of (almost) equal size groups to generate, or a list size of the groups to generate. In the second case n will be ignored.
xgmin	Minimum value for the x coordinate of demand points with respect to the group reference point.
xgmax	Maximum value for the x coordinate of demand points with respect to the group reference point.
ygmin	Minimum value for the y coordinate of demand points with respect to the group reference point.
ygmax	Maximum value for the y coordinate of demand points with respect to the group reference point.

**Details**

n must be at least 1.

xmin must be less or equal than xmax.

ymin must be less or equal than ymax.

If a non zero value is given for groups parameter, then a reference point for each group are generated. At second stage, the offset part for each demand point are generated, and added to the reference point generated at the first stage.

Note that groups = 1 is not equivalent to the default value groups = 0, because in the first case a reference point are generated at the first stage.

**Value**

If the arguments are valid values, it returns a new object of loca.p class, else it returns an error.

**See Also**

See also [orloca-package](#) and loca.p.

**Examples**

```
# A random loca.p object at unit square with 5 demand points
rloca.p(5)
# At another region
rloca.p(10, xmin=-2, xmax=2, ymin=-2, ymax=2)
# Five groups
rloca.p(48, groups=5)
# Three unequal groups
rloca.p(1, groups=c(10, 7, 2))
```

---

zsum

*Computes zsum function*


---

**Description**

The objective function and the gradient function for the min-sum location problem.

**Usage**

```
zsum(o, x = 0, y = 0, lp = numeric(0))
```

**Arguments**

o	An object of loca.p class.
x	The x coordinate of the point to be evaluated.
y	The y coordinate of the point to be evaluated.
lp	If given, then $l_p$ norm will be used instead of the Euclidean norm.

**Value**

zsum returns the objective function of the min-sum location problem,  $\sum_{a_i \in o} w_i d(a_i, (x, y))$ , where  $d(a_i, (x, y))$  gives the euclidean or the  $l_p$  distances between  $a_i$  and the point  $(x, y)$ .

**See Also**

See also [orloca-package](#) and [zsummin](#).

**Examples**

```
# A new unweighted loca.p object
loca <- loca.p(x = c(-1, 1, 1, -1), y = c(-1, -1, 1, 1))
# Evaluation of zsum at (0, 0)
zsum(loca)

# Evaluation of zsum at (1, 3)
zsum(loca, 1, 3)
# Compute the objective function at point (3, 4) using lp norm and p = 2.5
zsum(loca, 3, 4, lp=2.5)
# The gradient function at (1,3)
zsumgra(loca, 1, 3)
```

---

zsumgra

*Computes the gradient of zsum function*


---

**Description**

The gradient function for the min-sum location problem.

**Usage**

```
zsumgra(o, x = 0, y = 0, lp = numeric(0), partial = F)
```

**Arguments**

o	An object of loca.p class.
x	The x coordinate of the point to be evaluated.
y	The y coordinate of the point to be evaluated.
lp	If given, then $l_p$ norm will be used instead of the Euclidean norm.
partial	If (x,y) is a demand point partial=T means ignore such point to compute the gradient. This option is mainly for internal use.

**Value**

zsumgra returns the gradient vector of the function of the min-sum location problem,  $\sum_{a_i \in o} w_i d(a_i, (x, y))$ , where  $d(a_i, (x, y))$  gives the euclidean or the  $l_p$  distances between  $a_i$  and the point  $(x, y)$ .

**See Also**

See also [orloca-package](#) and [zsum](#).

**Examples**

```
# A new unweighted loca.p object
loca <- loca.p(x = c(-1, 1, 1, -1), y = c(-1, -1, 1, 1))
# Evaluation of zsum at (0, 0)
zsum(loca)

# Evaluation of zsum at (1, 3)
zsum(loca, 1, 3)
# Compute the objective function at point (3, 4) using lp norm and p = 2.5
zsum(loca, 3, 4, lp=2.5)
# The gradient function at (1,3)
zsumgra(loca, 1, 3)
```

---

zsummin	<i>zsummin at orloca package returns the solution of the minimization problem</i>
---------	---

---

**Description**

Solve the min-sum location problem for a given `loca.p` class object.

**Usage**

```
zsummin(o, x = 0, y = 0, lp = numeric(0), max.iter = 100, eps = 0.001,
        verbose = FALSE, algorithm = "weiszfeld", ...)
```

**Arguments**

o	An object of <code>loca.p</code> class.
x	The x coordinate of the starting point.
y	The y coordinate of the starting point.
lp	If given, the $l_p$ norm will be used instead of the Euclidean norm.
max.iter	Maximum number of iterations allowed.
eps	The module of the gradient in the stop rule.
verbose	If TRUE the function produces detailed output.
algorithm	The algorithm to be use. For this version of the package, the valid values are: "gradient" or "g" for a gradient based method, "search" or "s" for local search method, "ucminf" or "u" for optimization with ucminf from ucminf package, and "weiszfeld" or "w" for the Weiszfeld method or any of the valid method for optim function, now "Nelder-Mead", "BFGS", "CG", "L-BFGS-B", "SANN". "weiszfeld" is the default value.
...	Other options for optimization algorithms.

**Details**

If  $p < 1$  thus  $l_p$  is not a norm, so, only  $p \geq 1$  are valid values.

**Value**

zsummin returns an array with the coordinates of the solution point.

**See Also**

See also [orloca-package](#), [loca.p](#) and [zsum](#).

**Examples**

```
# A new unweighted loca.p object
loca <- loca.p(x = c(-1, 1, 1, -1), y = c(-1, -1, 1, 1))
# Compute the minimum
sol<-zsummin(loca)

# Show the result
sol

# Evaluation of the objective function at solution point
zsum(loca, sol[1], sol[2])
```

# Index

- \*Topic **andalusia**
    - andalusia-data, 3
  - \*Topic **classes**
    - as-methods, 4
    - contour.locap, 5
    - locap, 6
    - persp.locap, 7
    - plot, 8
    - zsum, 10
    - zsumgra, 11
    - zsummin, 12
  - \*Topic **datagen**
    - rlocap, 9
  - \*Topic **data**
    - andalusia-data, 3
  - \*Topic **hplot**
    - contour.locap, 5
    - persp.locap, 7
    - plot, 8
  - \*Topic **methods**
    - as-methods, 4
  - \*Topic **optimize**
    - locap, 6
    - orlocap-package, 2
    - zsum, 10
    - zsumgra, 11
    - zsummin, 12
  - \*Topic **package**
    - orlocap-package, 2
- andalusia (andalusia-data), 3  
andalusia-data, 3  
as-methods, 4  
as.data.frame (as-methods), 4  
as.data.frame, locap-method (as-methods), 4  
as.data.frame.locap (as-methods), 4  
as.locap (as-methods), 4  
as.locap, data.frame-method (as-methods), 4  
as.locap, matrix-method (as-methods), 4  
as.locap.data.frame (as-methods), 4  
as.locap.matrix (as-methods), 4  
as.matrix (as-methods), 4  
as.matrix, locap-method (as-methods), 4  
as.matrix.locap (as-methods), 4  
contour, locap-method (contour.locap), 5  
contour.locap, 5  
initialize, locap-method (locap), 6  
locap, 5, 6, 6, 8, 9, 13  
locap-class (locap), 6  
orlocap-package, 2  
persp, locap-method (persp.locap), 7  
persp.locap, 7  
plot, 8, 9  
plot, locap-method (plot), 8  
plot-methods (plot), 8  
plot.locap, 6, 8  
plot.locap (plot), 8  
print, locap-method (locap), 6  
print.locap (locap), 6  
rlocap, 9  
summary, locap-method (locap), 6  
summary-method (locap), 6  
zsum, 10, 12, 13  
zsum, locap-method (zsum), 10  
zsumgra, 11  
zsumgra, locap-method (zsumgra), 11  
zsummin, 11, 12  
zsummin, locap-method (zsummin), 12