

# Package ‘orloca.es’

April 19, 2009

**Type** Package

**Depends** methods, orloca

**Suggests** grDevices, graphics, snow

**Title** Spanish version of orloca package

**Version** 2.0

**Date** 2008-11-25

**Author** Fernando Fernandez-Palacin <fernando.fernandez@uca.es> and Manuel Munoz-Marquez <manuel.munoz@uca.es>

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

**Description** Version espanola del paquete orloca que trata de los modelos de Localizacion en Investigacion Operativa (Operations Research LOCational Analysis). En particular, el problema de encontrar el punto que minimiza la suma ponderada de las distancias a un conjunto de puntos dados.

**License** GPL (>= 2)

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

**Repository** CRAN

**Date/Publication** 2008-11-21 08:25:53

## R topics documented:

orloca.es-package . . . . .	2
as-methods . . . . .	3
czsum . . . . .	4
czsummin . . . . .	5
loca.p-class . . . . .	6
plot-methods . . . . .	7
plot.zsum . . . . .	8
rloca.p . . . . .	9
zsum . . . . .	10
zsummin . . . . .	11

**Index****13**


---

orloca.es-package *Spanish version of orloca package - Version española del paquete orloca*

---

**Description**

El paquete ofrece modelos de Análisis de Localización. Esta versión aborda problemas de localización min-sum. El problema min-sum, también conocido como problema de Weber, localiza un punto tal que la suma ponderada de las distancias entre dicho punto y los puntos de demanda sea mínima.

**Details**

```
Package:  orloca.es
Type:    Package
Version:  2.0
Date:    2008-11-20
License:  GPL (>= 2)
```

El paquete proporciona una clase (`loca.p`) que representa un problema de localización con un conjunto finito de puntos de demanda sobre el plano. También es posible representar los puntos y la función objetivo. Dicha función objetivo representa la suma de los desplazamientos de los usuarios a un servicio.

Para una demostración, cargue el paquete con `library(orloca.es)` y use `demo(orloca)`.

**Index:**

```
loca.p:      descripción de la clase loca.p.
rloca.p:    instancias aleatorias de objetos de la clase loca.p.
zsum:       función para evaluar la función objetivo para modelos min-sum.
zsummin:    para encontrar el mínimo del problema de localización.
plot.loca.p: para hacer gráficos de los objetos de la clase loca.p.
plot.zsum:  para hacer gráficos de la función objetivo.
```

**Author(s)**

Fernando Fernandez-Palacin <fernando.fernandez@uca.es> and Manuel Munoz-Marquez <manuel.munoz@uca.es>

Maintainer: 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

This is the spanish version of the orloca package. More information at <http://knuth.uca.es/orloca>

## Examples

```
# Un objeto loca.p no ponderado
o <- new("loca.p", x = c(-1, 1, 1, -1), y = c(-1, -1, 1, 1))

# Calcula la función objetivo en el punto c(3, 4)
zsum(o, 3, 4)

# Resuelve el problema de localización
zsummin(o)

# Curvas de nivel
contour(o)

# Ejecuta una demo del paquete
demo(orloca)
```

---

as-methods

*Conversion entre objetos de la clase loca.p y otros objetos*

---

## Description

Métodos para convertir desde y hacia la clase `loca.p`.

## Usage

```
## S3 method for class 'data.frame':
as.loca.p(x, ...)
## S3 method for class 'matrix':
as.loca.p(x, ...)
## S3 method for class 'loca.p':
as.data.frame(x, row.names = NULL, optional = FALSE, ...)
## S3 method for class 'loca.p':
as.matrix(x, ...)
```

## Arguments

<code>x</code>	es el objeto para convertir a una nueva clase.
<code>row.names</code>	No usado.
<code>optional</code>	No usado.
<code>...</code>	Otros argumentos, no usados.

## Details

Valores NA's no están permitidos en ninguno de los argumentos.

La `matrix` `a` a convertir en `loca.p` debe tener al menos dos columnas. La primera columna será considerada como la coordenada `x`, la segunda como la coordenada `y`, y la tercera (si existe) como los valores de `w`.

El `data.frame` para convertir en `loca.p` debe tener al menos una columna `x` para la coordenada `x`, y una columna `y` para la coordenada `y`. Opcionalmente, puede tener una columna `w`, como los valores de `w`.

## Value

Si los argumentos tienen valores válidos, devuelve un nuevo objeto de la nueva clase.

## See Also

Ver también [loca.p](#).

## Examples

```
# Un objeto loca.p no ponderado
loca <- loca.p(x = c(-1, 1, 1, -1), y = c(-1, -1, 1, 1))

# Conversión a matriz
m <- as.matrix(loca)
m

# Conversión desde una matriz
as.loca.p(m)
```

---

czsum

*czsum y czsumgra en el paquete orloca*

---

## Description

Función objetivo y función gradiente para problemas de localización min-sum.

## Usage

```
czsum(o, cluster, x=0, y=0)
czsumgra(o, cluster, x=0, y=0)
```

## Arguments

<code>o</code>	Un objeto de la clase <code>loca.p</code> .
<code>cluster</code>	Un objeto <code>snow cluster</code> .
<code>x</code>	La coordenada <code>x</code> del punto para ser evaluada.
<code>y</code>	La coordenada <code>y</code> del punto para ser evaluada.

**Value**

`czsum` devuelve la función objetivo del problema de localización mini-sum,  $\sum_{a_i \in o} w_i d(a_i, (x, y))$ , donde  $d(a_i, (x, y))$  representa la distancia euclídea entre  $a_i$  y  $(x, y)$ .

`czsumgra` devuelve el vector gradiente de la función `zsum`.

El cálculo se realiza en el cluster dado.

**See Also**

Ver <http://www.stat.uiowa.edu/~luke/R/cluster/cluster.html> para información sobre computación en clusters.

Ver también `zsum`, `zsumgra` y `zsummin`

---

 czsummin

*czsummin en el paquete orloca*


---

**Description**

Resuelve el problema de localización min-sum para un objeto de la clase `loca.p`.

**Usage**

```
czsummin(o, cluster, x=0, y=0, max.iter=100, eps=1.e-3,
         verbose=FALSE, algorithm="gradient")
```

**Arguments**

<code>o</code>	Un objeto de la clase <code>loca.p</code> .
<code>cluster</code>	Un objeto <code>snow cluster</code> .
<code>x</code>	La coordenada x del punto inicial.
<code>y</code>	La coordenada y del punto inicial.
<code>max.iter</code>	Máximo número de iteraciones permitido.
<code>eps</code>	El módulo del gradiente en la regla de parada.
<code>verbose</code>	Si es TRUE la función proporciona una salida detallada.
<code>algorithm</code>	El método que se usará. Los valores válidos son: "gradient" o "g" para el método del gradiente, y "search" o "s" para el método de busca local. "gradient" es el valor por defecto.

**Value**

`czsummin` devuelve una matriz con las coordenadas del punto solución.

Los cálculos se realizan en el cluster dado.

**See Also**

Ver <http://www.stat.uiowa.edu/~luke/R/cluster/cluster.html> para información sobre computación en clusters.

Ver también `zsummin`, `loca.p` y `zsum`.

---

`loca.p-class`*Clase de objetos `loca.p` para Localización en Investigación Operativa*

---

**Description**

Un objeto de la clase `loca.p` representa un problema de localización ponderado con un conjunto de puntos de demanda. El `orloca-package` está principalmente dedicado a abordar problemas de localización.

**Details**

Las longitudes de los vectores `x` e `y` deben ser iguales. La longitud de `w` debe ser igual a los anteriores o 0. Los valores NA no están permitidos en ninguno de los argumentos.

**Value**

Si los argumentos son valores válidos, devuelve un objeto de la clase `loca.p`, en caso contrario devuelve un error. `summary(x)` devuelve un resumen del objeto `x` de la clase `loca.p` y `print(x)` imprime un resumen del objeto `x` de la clase `loca.p`.

**Generators**

El principal generador es `loca.p(x, y, w = numeric(0), label = "")` o alternativamente `new("loca.p", x, y, w = numeric(0), label = "")`.

`x` es un vector con las coordenadas `x` de los puntos de demanda.

`y` es un vector con las coordenadas `y` de los puntos de demanda.

`w` es un vector de pesos de los puntos de demanda. Si `w` se omite entonces todos los pesos se consideran iguales a 1.

**label** Si se explicita, es la etiqueta del nuevo objeto.

**See Also**

Véase también `orloca-package`.

## Examples

```
# Un objeto loca.p no ponderado
loca <- loca.p(x = c(-1, 1, 1, -1), y = c(-1, -1, 1, 1))
# o
loca <- new("loca.p", x = c(-1, 1, 1, -1), y = c(-1, -1, 1, 1))

# Un ejemplo con pesos y nombre
locb <- new("loca.p", x = c(-1, 1, 1, -1), y = c(-1, -1, 1, 1), w = c(1, 2, 1, 2),
  label = "Caso Ponderado")
```

---

plot-methods

*Grafico de un objeto de la clase loca.p*

---

## Description

Este método proporciona una representación gráfica de un objeto de la clase `loca.p`.

## Usage

```
## S3 method for class 'loca.p':
plot(x, xlab="", ylab="", main="Plot of loca.p object", ...)
```

## Arguments

<code>x</code>	El objeto <code>loca.p</code> a representar.
<code>xlab</code>	La etiqueta para el eje x.
<code>ylab</code>	La etiqueta para el eje y.
<code>main</code>	El título principal del gráfico.
<code>...</code>	Otras opciones gráficas.

## Details

Gráfico de los puntos de demanda con límites de evaluación automáticos.

## Value

La representación gráfica de los puntos de demanda.

## See Also

Véase también [orloca-package](#), [loca.p](#) y [plot.zsum](#).

## Examples

```
# Un objeto de la clase loca.p no ponderado
loca <- new("loca.p", x = c(-1, 1, 1, -1), y = c(-1, -1, 1, 1))

# El gráfico del objeto loca.p
plot(loca)
```

---

`plot.zsum`*Grafica de la funcion objetivo min-sum*

---

**Description**

Proporciona dos representaciones gráficas de la función objetivo del problema min-sum (`zsum`).

**Usage**

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

**Arguments**

<code>x</code>	El objeto <code>loca.p</code> para calcular el objetivo.
<code>xmin</code>	El valor mínimo del eje x.
<code>xmax</code>	El valor máximo del eje x.
<code>ymin</code>	El valor mínimo del eje y.
<code>ymax</code>	El valor máximo del eje y.
<code>n</code>	El número de divisiones para la rejilla.
<code>...</code>	Otras opciones.

**Value**

La función `contour` representa un gráfico de curvas de nivel y la función `persp` un gráfico 3D.

**See Also**

Véase también [orloca-package](#), [plot.loca.p](#) y [loca.p](#).

**Examples**

```
# Un objeto loca.p no ponderado  
loca <- loca.p(x = c(-1, 1, 1, -1), y = c(-1, -1, 1, 1))  
  
# El gráfico de curvas de nivel del objeto loca.p  
contour(loca)  
  
# El gráfico 3D del objeto loca.p  
persp(loca)
```

---

`rloca.p`*Generador de instancias aleatorias de objetos de la clase loca.p*

---

## Description

Devuelve una instancia aleatoria de un objeto de la clase `loca.p` en una determinada región.

## Usage

```
rloca.p(n, xmin=0, xmax=1, ymin=0, ymax=1, groups=numeric(0),
        xmin=xmin, xmax=xmax, ymin=ymin, ymax=ymax)
```

## Arguments

<code>n</code>	El número de puntos de demanda.
<code>xmin</code>	Mínimo valor para la coordenada x de los puntos de demanda.
<code>xmax</code>	Máximo valor para la coordenada x de los puntos de demanda.
<code>ymin</code>	Mínimo valor para la coordenada y de los puntos de demanda.
<code>ymax</code>	Máximo valor para la coordenada y de los puntos de demanda.
<code>groups</code>	El número de grupos (de aproximadamente igual tamaño) o una lista con los tamaños de los grupos a generar.
<code>xgmin</code>	Mínimo valor para la coordenada x de los puntos de demanda respecto del punto referencia del grupo.
<code>xgmax</code>	Máximo valor para la coordenada x de los puntos de demanda respecto del punto referencia del grupo.
<code>ygmin</code>	Mínimo valor para la coordenada y de los puntos de demanda respecto del punto referencia del grupo.
<code>ygmax</code>	Máximo valor para la coordenada y de los puntos de demanda respecto del punto referencia del grupo.

## Details

`n` debe ser al menos 1.

`xmin` debe ser menor o igual que `xmax`.

`ymin` debe ser menor o igual que `ymax`.

Cuando se suministra valor para `groups` los puntos se generan en dos fases, en la primera se genera un punto de referencia, en la segunda se genera un desplazamiento sobre dicho punto de referencia que se suma a éste.

Obsérvese que `groups = 1` no es equivalente a `groups = numeric(0)`, debido a que en el primer caso se genera un punto de referencia en la primera etapa.

**Value**

Si los argumentos son valores válidos, devuelve un nuevo objeto de la clase `loca_p`, en otro caso informa de un error.

**See Also**

Véase también [orloca-package](#) y [loca.p](#)

**Examples**

```
# Un objeto aleatorio loca.p en el cuadrado unidad con 5 puntos de demanda
rloca.p(5)

# En otra región
rloca.p(10, xmin=-2, xmax=2, ymin=-2, ymax=2)

# Cinco grupos
rloca.p(48, groups=5)

# Tres grupos de distinto tamaño
rloca.p(1, groups=c(10, 7, 2))
```

---

zsum

*zsum* y *zsumgra* del paquete *orloca*


---

**Description**

La función objetivo y la función gradiente para el problema de localización min-sum.

**Usage**

```
zsum(o, x=0, y=0)
zsumgra(o, x=0, y=0)
```

**Arguments**

<code>o</code>	Un objeto de la clase <code>loca.p</code> .
<code>x</code>	La coordenada x del punto a ser evaluado.
<code>y</code>	La coordenada y del punto a ser evaluado.

**Value**

`zsum` devuelve la función objetivo para el problema de localización min-sum,  $\sum_{a_i \in o} w_i d(a_i, (x, y))$ , donde  $d(a_i, (x, y))$  es la distancia euclídea entre  $a_i$  y  $(x, y)$ .

`zsumgra` devuelve el vector gradiente de la función `zsum`.

**See Also**

Véase también [orloca-package](#) y [zsummin](#).

**Examples**

```
# Un objeto loca.p no ponderado
loca <- loca.p(x = c(-1, 1, 1, -1), y = c(-1, -1, 1, 1))

# Evaluación de zsum en (0, 0)
zsum(loca)

# Evaluación de zsum en (1, 3)
zsum(loca, 1, 3)

# La función gradiente
zsumgra(loca, 1, 3)
```

---

zsummin

*zsummin en el paquete orloca*


---

**Description**

Reusolve el problema de localización min-sum para un objeto dado de la clase `loca.p`.

**Usage**

```
zsummin(o, x=0, y=0, max.iter=100, eps=1.e-3, verbose=FALSE,
        algorithm="weiszfeld")
```

**Arguments**

<code>o</code>	Un objeto de la clase <code>loca.p</code> .
<code>x</code>	La coordenada x del punto inicial.
<code>y</code>	La coordenada y del punto inicial.
<code>max.iter</code>	Número máximo de iteraciones permitido.
<code>eps</code>	La norma del gradiente en la regla de parada.
<code>verbose</code>	Si es TRUE la función proporciona salida detallada.
<code>algorithm</code>	El método a utilizar. Los valores válidos son: "gradient" o "g" para el método basado en el gradiente, "search" o "s" para método de busca local, y "weiszfeld" o "w" para el método de Weiszfeld. "Weiszfeld" es el valor por defecto.

**Value**

`zsummin` devuelve un vector con las coordenadas del punto solución.

**See Also**

Vea también [orloca-package](#), [loca.p](#) y [zsum](#).

**Examples**

```
# Un objeto loca.p no ponderado
loca <- new("loca.p", x = c(-1, 1, 1, -1), y = c(-1, -1, 1, 1))

# Calcula el mínimo
sol<-zsummin(loca)

# Muestra los resultados
sol

# Evaluación de la función objetivo en el punto solución
zsum(loca, sol[1], sol[2])
```

# Index

- \*Topic **classes**
  - as-methods, 3
  - czsum, 4
  - czsummin, 5
  - loca.p-class, 6
  - plot-methods, 7
  - plot.zsum, 8
  - zsum, 10
  - zsummin, 11
- \*Topic **datagen**
  - rloca.p, 9
- \*Topic **hplot**
  - plot-methods, 7
  - plot.zsum, 8
- \*Topic **methods**
  - as-methods, 3
- \*Topic **optimize**
  - czsum, 4
  - czsummin, 5
  - loca.p-class, 6
  - orloca.es-package, 2
  - zsum, 10
  - zsummin, 11
- \*Topic **package**
  - orloca.es-package, 2
- as, loca.p-method (*as-methods*), 3
- as-methods, 3
- as.data.frame (*as-methods*), 3
- as.data.frame, loca.p-method (*as-methods*), 3
- as.data.frame.loca.p (*as-methods*), 3
- as.loca.p (*as-methods*), 3
- as.loca.p, data.frame-method (*as-methods*), 3
- as.loca.p, matrix-method (*as-methods*), 3
- as.loca.p.data.frame (*as-methods*), 3
- as.loca.p.matrix (*as-methods*), 3
- as.matrix (*as-methods*), 3
- as.matrix, loca.p-method (*as-methods*), 3
- as.matrix.loca.p (*as-methods*), 3
- contour, loca.p-method (*plot.zsum*), 8
- contour.loca.p (*plot.zsum*), 8
- czsum, 4
- czsum, loca.p-method (*czsum*), 4
- czsumgra (*czsum*), 4
- czsumgra, loca.p-method (*czsum*), 4
- czsummin, 5
- czsummin, loca.p-method (*czsummin*), 5
- initialize, loca.p-method (*loca.p-class*), 6
- loca.p, 2, 4, 6–8, 10, 12
- loca.p (*loca.p-class*), 6
- loca.p-class, 6
- orloca (*orloca.es-package*), 2
- orloca-package, 6–8, 10–12
- orloca-package (*orloca.es-package*), 2
- orloca.es (*orloca.es-package*), 2
- orloca.es-package, 2
- persp, loca.p-method (*plot.zsum*), 8
- persp.loca.p (*plot.zsum*), 8
- plot, loca.p-method (*plot-methods*), 7
- plot-methods, 7
- plot.loca.p, 2, 8
- plot.loca.p (*plot-methods*), 7
- plot.zsum, 2, 7, 8
- print, loca.p-method (*loca.p-class*), 6

`print-method(loca.p-class)`, 6  
`print.loca.p(loca.p-class)`, 6

`rloca.p`, 2, 9

`summary`, `loca.p-method`  
    (*loca.p-class*), 6  
`summary-method(loca.p-class)`, 6  
`summary.loca.p(loca.p-class)`, 6

`zsum`, 2, 5, 6, 10, 12  
`zsum`, `loca.p-method(zsum)`, 10  
`zsumgra`, 5  
`zsumgra(zsum)`, 10  
`zsumgra`, `loca.p-method(zsum)`, 10  
`zsummin`, 2, 5, 6, 11, 11  
`zsummin`, `loca.p-method(zsummin)`,  
    11