Package ‘multigraph’

June 23, 2023

Type Package
Title Plot and Manipulate Multigraphs
Version 0.99
Depends R (>= 3.6.0), multiplex (>= 3.0.0)
Imports methods
Date 2023-06-23
Author Antonio Rivero Ostoic [aut, cre]
Maintainer Antonio Rivero Ostoic <multiplex@post.com>
Description Functions to plot and manipulate multigraphs, signed and valued graphs, bipartite graphs, multilevel graphs, and Cayley graphs with various layout options.
URL https://github.com/mplex/multigraph/
BugReports https://github.com/mplex/multigraph/issues/
Repository CRAN
License GPL-3
NeedsCompilation no
Date/Publication 2023-06-23 09:00:08 UTC

R topics documented:

  multigraph-package .................................................. 2
  bmgraph ............................................................. 3
  ccgraph ............................................................. 6
  conc ............................................................... 8
  frcd ............................................................... 9
  mlggraph .......................................................... 11
  multigraph ......................................................... 14
  stsm ............................................................... 18

Index 20
Plot and Manipulate Multigraphs

Description

Functions to create and manipulate multigraphs, bipartite graphs, Cayley graphs, and valued multilevel graphs.

Details

Package: multigraph
Type: Package
Version: 0.99 (devel)
Depends: multiplex (>= 3.0.0)
Date: 23 June 2023
License: GPL-3

This package contains functions to plot diverse types of graphs representing complex network structures. For one-mode data, it is possible to depict signed and valued multigraphs and bipartite graphs for two-mode data as well. Moreover, multilevel graphs that combine one- and two-mode network data are represented with the latest function. Finally, Cayley graphs serve to depict relations among the ties in multiplex networks recorded in the algebraic object semigroup.

Note that this package is still under development.

Author(s)

J. Antonio Rivero Ostoic

Maintainer: Antonio Rivero Ostoic <multiplex@post.com>

References


See Also

multiplex-package, incubs, zbind, transf
**bmgraph**  

**Description**

A function to create and manipulate bipartite multigraphs

**Usage**

```r
bmgraph(net, layout = c("bip", "bip3", "bip3e", "bipc", "force", "rand", "circ",  
                      "stress", "CA", "circ2"), scope, coord, alpha = c(1, 1, 1), showLbs, showAtts,  
                      att = NULL, lbat = "1", main = NULL, cex.main, bg, mar, directed, valued,  
                      collRecip, cex, pos, lwd, lty, col, ecol, vcol, vcol0, asp, seed = NULL,  
                      maxiter = 100, bwd, clu, pch, rot, mirrorX, mirrorY, mirrorV, mirrorH, hds,  
                      vedist, jitter, sort, add, adc, perm, ffamily, fstyle, fsize, fcol, vclu, ...)  
```

**Arguments**

- `net` data frame or array representing the two-mode network (see details)
- `layout` the visualization layout:
  - `bip` (default) bipartite graph
  - `bip3` bipartite graph with three columns
  - `bip3e` bipartite graph with three columns for events
  - `bipc` “clustered” bipartite graph
  - `force` force-directed algorithm
  - `rand` random
  - `circ` circular
  - `stress` stress-majorization algorithm
  - `CA` correspondence analysis
  - `circ2` two semi-circles
- `scope` (optional) scope of the graph (see details)
- `coord` (optional) data frame with the coordinates of the vertices; if coordinates are given then the `layout` option is ignored
- `alpha` vector (vertex, edge, bg) with the alpha color transparency
- `showLbs` (optional and logical) whether or not to show the vertex labels when dimnames available
- `showAtts` (optional and logical) whether or not to show the vertex attribute labels
- `att` (optional) a vector or an array representing the vertex attributes
- `lbat` (optional) labels for the vertex attributes
- `main` (optional) title of the plot
- `cex.main` (optional) size of the plot’s title
- `bg` (optional) background color of the plot
mar (optional) margins of the plot
directed (optional and logical) whether or not the graph is directed or undirected
valued (optional and logical) whether or not the graph is valued or with dichotomous data
collRecip (optional and logical) whether or not collapse reciprocated edges in the undirected graph
cex (optional) size of the vertices
pos (optional) position of the vertices’ labels (Ø means “at the center of the vertex”)
lwd (optional) width of the edges; ignored if valued is set to TRUE
lty (optional) shape of the edges
col (optional) alias for vcol
ecol (optional) color of the edges
vcol (optional) color of the vertices
vcol0 (optional) color of the vertices’ contour (only works for pch 21 through 25
asp (optional) aspect ratio of the plot
seed (optional) random seed number for the vertices’ initial coordinates. Ignored except for force, stress and rand
maxiter (optional) maximum number of iterations in layout algorithms. Ignored except for force, stress and rand
bwd (optional) width of the bundle edges: ranges from 0 (edges collapsed) to the default 1 (depending on the vertices’ size), and for valued a value greater than one is possible
clu (optional) clustering of the vertices (see details)
pch (optional) symbol representing the vertices
rot (optional) clockwise rotation of the graph in degrees
mirrorX (optional) mirror of the X axis
mirrorY (optional) mirror of the Y axis
mirrorV same as mirrorX
mirrorH same as mirrorY
hds (optional and experimental) arcs’ head scale
vedist (optional and experimental) a real number with vertex - edge distance
jitter (optional) jitter in stress or CA
sort (optional and logical) sort the vertex labels
add (optional) add nodes to the graph’s domain
adc (optional) add nodes to the graph’s codomain
perm (optional) a list of vectors for the permutation of network members in both the domain and codomain
ffamily (optional) font family
fstyle (optional) font style
Bipartite graphs are visualization devices for two-mode networks. Although this type of data would typically record as a data frame, it is possible to use even three-dimensional arrays where each level corresponds to a particular type of tie. Thus the bipartite graphs, in this case, will be depicted with parallel edges. Besides, it is possible to obtain a figure of the bipartite network using the binomial approach to two-mode data and plot it with a force-directed algorithm.

Since bipartite graphs have two domains of vertices, the clustering information in vclu, for the colors of vertices for example, is in a list with two vectors; one vector for each domain. It is possible to class all members of the domain or co-domain into a single class by setting the vector to NULL.

Value

A plot of the two-mode network as a bipartite graph or multigraph with a projection

Examples

```r
## two binary relations among three elements
arr <- round( replace( array(runif(18), c(3,3,2)), array(runif(18), c(3,3,2))>.5, 3 ) )

## network as bipartite graph
bmgraph(arr)

## with a force directed algorithm
bmgraph(arr, layout = "force")

## with a Correspondence Analysis method
bmgraph(arr, layout = "CA", asp = NA)
```
ccgraph  

Cayley colour graph

Description

A function to create and manipulate bipartite Cayley colour graphs

Usage

ccgraph(x, main=NULL, seed=0, maxiter=100, alpha=c(1, 1, 1), scope, loops,
collRecip, undRecip, showLbs, cex.main, conc, coord, clu, cex, lwd,
pch, lty, bwd, bwd2, att, bg, mar, pos, asp, ecol, vcol, vcol0, lbs,
col, lbat, swp, swp2, scl, mirrorX, mirrorY, mirrorD, mirrorL, mirrorV,
mirrorH, roth, hds, vedist, ffamily, fstyle, fsize, fcol, nr, gens, ...)

Arguments

x  
an algebraic structure, typically a "Semigroup" object class
main  
(optional) title of the plot
seed  
(optional) random seed number for the vertices' initial coordinates; ignored except for force, stress and rand
maxiter  
(optional) maximum number of iterations in layout algorithms; ignored except for force, stress and rand
alpha  
vector (vertex, edge, bg) with the alpha color transparency
scope  
(optional) scope of the graph (see details)
loops  
(optional, logical, and experimental) plot graph loops?
collRecip  
(optional and logical) whether or not collapse reciprocated edges in the undirected graph
undRecip  
(optional and logical) whether or not plot reciprocated edges as undirected
showLbs  
(optional and logical) whether or not show the vertex labels when dimnames available
cex.main  
(optional) size of the plot's title
conc  
(optional and logical) whether the layout is concentric or not
coord  
(optional) data frame with the coordinates of the vertices; if coordinates are given then the layout option is ignored
clu  
(optional) clustering of the vertices (see details)
cex  
(optional) size of the vertices
lwd  
(optional) width of the edges; ignored if valued is set to TRUE
pch  
(optional) symbol representing the vertices
lty  
(optional) shape of the edges
**ccgraph**

- **bwd** (optional) width of the bundle edges. Ranges from 0 (edges collapsed) to the default 1 (depending on the vertices’ size), and for valued a value greater than one is possible
- **bwd2** (optional) width of the bundle loop edges.
- **att** (optional) a vector or an array representing the vertex attributes
- **bg** (optional) background color of the plot
- **mar** (optional) margins of the plot
- **pos** (optional) position of the vertices’ labels (0 means “at the center of the vertex”)
- **asp** (optional) aspect ratio of the plot
- **ecol** (optional) color of the edges
- **vcol** (optional) color of the vertices
- **vcol0** (optional) color of the vertices’ contour (only works for pch 21 through 25
- **lbs** (optional) vertex labels
- **col** (optional) alias for vcol
- **lbat** (optional) labels for the vertex attributes
- **swp** (optional and logical) whether or not to swap the bundle patterns
- **swp2** (optional and logical) whether or not to swap reciprocals
- **scl** (optional and experimental) numerical scalar \((x, y)\) or vector \((x, y)\) of the graph’s scale
- **mirrorX** (optional) mirror of the \(X\) axis
- **mirrorY** (optional) mirror of the \(Y\) axis
- **mirrorD** (optional) mirror reflection across diagonal \(Y = X\)
- **mirrorL** (optional) mirror reflection across diagonal \(Y = -X\)
- **mirrorV** same as mirrorX
- **mirrorH** same as mirrorY
- **rot** (optional) clockwise rotation of the graph in degrees
- **hds** (optional and experimental) arcs’ head scale
- **vedist** (optional and experimental) a real number with vertex - edge distance
- **ffamily** the font family
- **fstyle** the font style
- **fsize** the font size
- **fcol** the font color
- **nr** for conc layout, number of radii
- **gens** (optional when absent) semigroup generators in \(x\)
- **...** Additional argument items (see e.g. par)

**Details**

The Cayley colour graph is a graphical representation of the relationships among relations in the relational structure of a given multiplex network. Both nodes and directed edges represent string relations, and each shape (and color) corresponds to a specific generator relation of the semigroup structure.
Value

A plot of the semigroup or group structure.

Author(s)

Antonio Rivero Ostoic

See Also

semigroup, multigraph, frcd, conc

Examples

```r
## Create an abstract semigroup from random data
arr <- round(replace(array(runif(18), c(3,3,2)), array(runif(18), c(3,3,2))>.5, 1 ))

S <- semigroup(arr)

## plot semigroup
ccgraph(S)
```

conc

Concentric layout

Description

A function to compute the graph coordinated system with a concentric layout

Usage

```r
conc(net, nr, irot, inv, flip, mirror=c("N","X","Y","D","L"), ...)
```

Arguments

- **net**: an array representing the network relations
- **nr**: a scalar with the number of radii, or a vector with the clustering of the actors.
- **irot**: a scalar or vector with the “internal rotation” for each circle from closer to the center point to further away
- **inv**: (optional and logical) should the circles be with an inverted ordering?
- **flip**: (optional and logical) should the alternating circles be flipped?
- **mirror**: mirror transformation N identity (default)
  - X reflection through the vertical center line
  - Y reflection through the horizontal center line
  - D reflection across diagonal $Y = X$
  - L reflection across diagonal $Y = -X$
- **...**: Additional argument items
Details

In a Euclidean plane computes the coordinated system with a concentric layout with at least two radii (unless $n = 1$). In case $nr$ is not specified, approx. half of the vertices are located at one radius and half in another one.

The clustering of the actors may be used to establish the location of the vertices in different radii as a numerical, character, or factor vector.

Value

A data frame with a coordinated system with two columns representing the abscissa and the ordinate in a two-dimensional rectangular Cartesian coordinate system.

Author(s)

Antonio Rivero Ostoic

See Also

`multigraph`, `bmgraph`, `frcd`, `stsm`

Examples

```r
## Create the data: two binary relations among three elements
arr <- round(replace(array(runif(18), c(3,3,2)), array(runif(18), c(3,3,2))>.5, 3 ))

## Coordinates for the concentric layout with two radii
coord <- conc(arr, nr = 2)

## Plot multigraph with customized coordinates
multigraph(arr, coord = coord)
```

---

**frcd**

*Force directed layout*

Description

A function to compute the graph coordinated system with a force directed layout algorithm

Usage

`frcd(net, seed = seed, maxiter, drp, scl, mov, ...)"
Arguments

- **net**: an array representing the network relations
- **seed**: (mandatory) the seed of the initial layout (see details)
- **maxiter**: (optional) the maximum number of iterations
- **drp**: (optional) for valued networks, drop values less than specified
- **...**: Additional argument items
- **scl**: (optional and experimental) numerical scalar \((x, y)\) or vector \((x, y)\) of the graph’s scale
- **mov**: (optional and experimental) numerical scalar \((x, y)\) or vector \((x, y)\) to move the graph

Details

This function is meant as an internal routine for graph visualization. However, it can be used for the `coord` option both in `multigraph` and in `bmgraph` where `NULL` in `seed` implies a random seed based on the clock watch of the computer.

Value

A data frame with a coordinated system with two columns representing the abscissa and the ordinate in a two-dimensional rectangular Cartesian coordinate system.

Author(s)

Antonio Rivero Ostoic

References


See Also

- `multigraph`, `bmgraph`, `stsm`, `conc`

Examples

```r
## Create the data: two binary relations among three elements
arr <- round( replace( array(runif(18), c(3,3,2)), array(runif(18), c(3,3,2))>.5, 3 ) )

## Coordinates for the force directed layout with random start
coord <- frcd(arr, seed = NULL)

## Plot multigraph with customized coordinates
multigraph(arr, coord = coord)
```
mlgraph

**Multilevel graph**

**Description**

A function to create and manipulate multilevel graphs

**Usage**

```r
mlgraph(net, layout = c("circ", "force", "stress", "rand", "conc", "bip"), main = NULL, seed = NULL, maxiter = 100, directed = TRUE, alpha = c(1, 1, 1), scope, collRecip, undRecip, showLbs, showAtts, cex.main, coord, clu, cex, lwd, pch, lty, bwd, bwd2, att, bg, mar, pos, asp, ecol, vcol, vcol0, col, lb, swap, loops, swap2, mirrorX, mirrorY, mirrorD, mirrorL, lbs, mirrorV, mirrorH, rot, hds, scl, vedist, ffamily, fstyle, fsize, fcol, valued, modes, elv, lng, nr, ...)```

**Arguments**

- `net`: a "Multilevel" class object or a three dimensional array with clustering information
- `layout`: the visualization layout:
  - `circ` circular
  - `force` force-directed
  - `stress` stress-majorization
  - `rand` random
  - `conc` concentric
- `main`: (optional) title of the plot
- `seed`: (optional) random seed number for the vertices’ initial coordinates. Ignored except for `force`, `stress` and `rand`
- `maxiter`: (optional) maximum number of iterations in layout algorithms. Ignored except for `force`, `stress` and `rand`
- `directed`: (logical) whether or not the graph is directed or undirected
- `alpha`: vector (vertex, edge, bg) with the alpha color transparency
- `scope`: (optional) scope of the graph (see details)
- `collRecip`: (optional and logical) whether or not collapse reciprocated edges in the undirected graph
- `undRecip`: (optional and logical) whether or not plot reciprocated edges as undirected
- `showLbs`: (optional and logical) whether or not to show the vertex labels
- `showAtts`: (optional and logical) whether or not to show the vertex attribute labels
- `cex.main`: (optional) size of the plot’s title
coord (optional) data frame with the coordinates of the vertices. If coordinates are given then the layout option is ignored
clu (optional) clustering of the vertices as a list of vectors with integers or NULL (see details)
cex (optional) size of the vertices
lwd (optional) width of the edges; ignored if valued is set to TRUE
pch (optional) symbol representing the vertices
lty (optional) shape of the edges
bwd (optional) width of the bundle edges. Ranges from 0 (edges collapsed) to the default 1 (depending on the vertices’ size), and for valued a value greater than one is possible
bwd2 (optional) width of the bundle loop edges.
att (optional) a vector or an array representing the vertex attributes
bg (optional) background color of the plot
mar (optional) margins of the plot
pos (optional) position of the vertices’ labels (Ø means “at the center of the vertex”)
asp (optional) aspect ratio of the plot
ecol (optional) color of the edges
vcol (optional) color of the vertices
vcol0 (optional) color of the vertices’ contour (only works for pch 21 through 25)
col (optional) alias for vcol
lbat (optional) labels for the vertex attributes
swp (optional and logical) whether or not to swap the bundle patterns
loops (optional, logical, and experimental) plot graph loops?
swp2 (optional and logical) whether or not to swap reciprocals
mirrorX (optional) mirror of the X axis
mirrorY (optional) mirror of the Y axis
mirrorD (optional) mirror reflection across diagonal Y = X
mirrorL (optional) mirror reflection across diagonal Y = −X
lbs (optional) vertex labels
mirrorV same as mirrorX
mirrorH same as mirrorY
rot (optional) clockwise rotation of the graph in degrees
hds (optional and experimental) arcs’ head scale
scl (optional and experimental) numerical scalar (x and y) or vector (x, y) of the graph’s scale
vedist (optional and experimental) a real number with vertex - edge distance
ffamily the font family
mlgraph

fstyle the font style
fsize the font size
fcol the font color
valued (optional and logical) whether the graph is depicted as valued or not
modes (optional) a vector indicating which matrices are domains and which codomains
(work only with a "Multilevel" class object)
elv (experimental) control loops 1
lng (experimental) control loops 2
nr integer or NULL with the number of radii for conc layout (see details)
... Additional argument items (see e.g. par)

Details

Multilevel graphs serve to represent networks with different “levels” such as different domains in
the network structure. A characteristic of multilevel networks is the existence of ties within and
across domains.

Since this function can handle a large number of arguments, these can be stored as a list object that
is passed through the scope option. In this case, a vector made of lists and scalars or combinations
of these is accepted.

The bundle width specified by bwd and bwd2 ranges from 0 (edges collapsed) to the default 1 (de-
dpending on the vertices’ size). For the valued option, a number greater than one is possible.

In a multilevel structure, argument clu is to class network members and it is possible to class all
members of the domain or co-domain into a single class by setting the vector to NULL. Similarly,
NULL in argument nr for the conc layout implies the use of two radii, one for each domain.

Value

A plot of the multilevel graph structure for the network

Note

Multilevel graphs depend on multilevel class objects

Author(s)

Antonio Rivero Ostoic

See Also

mlvl, multigraph, bmgraph, frcd, stsm, conc
Examples

## Not run:

```
# create network data as arrays
arr <- round( replace( array(runif(18), c(3,3,2)), array(runif(18),
c(3,3,2))>.5, 3 ) )
arr2 <- round( replace( array(runif(18), c(3,3,2)), array(runif(18),
c(3,3,2))>.5, 3 ) )

# create multilevel class object and plot multilevel graph
require(multiplex)
mlvl(arr, arr2) |> 
mlgraph()
## End(**Not run**)```

---

**multigraph**

**Multigraphs and valued multigraphs**

Description

A function to create and manipulate multigraphs and valued multigraphs with different layout options.

Usage

```
multigraph(net, layout=c("circ", "force", "stress", "conc", "rand"), scope, directed=TRUE, loops, signed, valued, values, lbs, showLbs, att, lab, showAtts, main=NULL, cex.main, col.main, font.main, coord, collRecip, undRecip, seed=NULL, maxiter=100, clu, cex, cex2, pch, lwd, lty, vcol, vcol0, col, ecol, bwd, bwd2, pos, bg, bg2, asp, drp, add, swp, swp2, alpha=c(1, 1, 1, 1), rot, mirrorX, mirrorY, mirrorD, mirrorL, mirrorV, mirrorH, scl, hds, vedist, mar, ffamily, fstyle, fsize, fsize2, fcol, fcol2, lclu, sel, new, mai, lscl, rm.isol, ...)```

Arguments

- **net**: an array; usually with three dimensions of stacked matrices where the multiple relations are placed.
- **layout**: the visualization layout:
  - circ: circular
  - force: force-directed
  - stress: stress-majorization
  - conc: concentric
  - rand: random
- **scope**: (optional) the scope of the graph (see details)
- **directed**: (logical) whether or not the graph is directed or undirected
- **loops**: (optional, logical, and experimental) plot graph loops?
signed  (optional and logical) whether or not the graph is a signed structure
valued  (optional and logical) whether the graph is depicted as valued or not
values  (optional and logical) print the values of the bonds in edges?
lbs  (optional) the vertices labels
showLbs  (optional and logical) whether or not show the vertex labels
att  (optional) a vector or an array representing the vertex attributes
labat  (optional) the labels for the vertices’ attributes
showAtts  (optional and logical) whether or not show the vertex attribute labels
main  (optional) title of the plot
cex.main  (optional) the size of the plot’s title
col.main  (optional) the color of the plot’s title
font.main  (optional) the font of the plot’s title
coord  (optional) data frame with the coordinates of the vertices. If coordinates are
given then the layout option is ignored
collRecip  (optional and logical) whether or not collapse reciprocated edges in the unidrected graph
undRecip  (optional and logical) whether or not plot reciprocated edges as undirected
seed  (optional) the random seed number for the vertices’ initial coordinates. Ignored for circ and conc
maxiter  (optional) the maximum number of iterations in layout algorithms. Only for force, stress, and rand
clu  (optional) the clustering of the vertices (see details)
cex  (optional) the size of the vertices
cex2  the size of the background for the values with the valued option
pch  (optional) the symbol representing the vertices
lwd  (optional) the width of the edges; ignored if weighted is set to TRUE
lty  (optional) the shape of the edges
vcol  (optional) the color of the vertices
vcol0  (optional) the color of the vertices’ contour (only works for pch 21 through 25
col  (optional) alias for vcol
ecol  (optional) the color of the edges
bwd  (optional) the width of the bundle edges.
bwd2  (optional) the width of the bundle loop edges.
pos  (optional) the position of the vertices’ labels (0 means “in middle of vertex”)
bg  (optional) the background color of the plot
bg2  (optional) the background color for values
asp  (optional) the aspect ratio of the plot
drp  (optional) for valued networks, drop values less than the specified
Multigraphs are graphs having parallel edges depicting different types of relations in a network. By default, a circular layout is applied where each type of tie has a distinctive shape and gray color scale. For better visualization, undirected multigraphs automatically collapse the reciprocal relations, and there is an argument to prevent this from happening. It is possible to combine the symbols and colors of vertices by assigning a class to each network member in the clustering option. Vertices can also have different sizes by specifying the argument with a vector with a length size similar to the network order.

Details

Multigraphs are graphs having parallel edges depicting different types of relations in a network. By default, a circular layout is applied where each type of tie has a distinctive shape and gray color scale. For better visualization, undirected multigraphs automatically collapse the reciprocal relations, and there is an argument to prevent this from happening. It is possible to combine the symbols and colors of vertices by assigning a class to each network member in the clustering option. Vertices can also have different sizes by specifying the argument with a vector with a length size similar to the network order.
Since this function can handle a large number of arguments, these can be stored as a list object that is passed through the scope option. In this case, a vector made of lists and scalars or combinations of these is accepted for describing characteristics.

The bundle width specified by bwd (and bwd2 for loops) ranges from 0 (edges collapsed) to the default 1 (depending on the vertices’ size). For the valued option, numbers higher than one are possible. Use vedist to adjust vertex–edge distance for large and dense networks.

In some cases, such as when working with dynamic networks, it is needed to specify the ordering of the “levels” of the clustering information given in clu, and this is done in argument lclu.

When using new for plotting the graph with a background image, the previous plot(s), however, can require having an equivalent command to graphics::plot.new() (cf. e.g. sdam::plot.map() function).

Value

A plot of the network as a multigraph or a weighted multigraph.

Author(s)

Antonio Rivero Ostoic

See Also

bmggraph, ccgraph, frcd, stsm, conc

Examples

## Create the data: two binary relations among three elements
arr <- round( replace( array(runif(18), c(3,3,2)),
array(runif(18),
c(3,3,2))>.5, 3 ) )

## Plot the multigraph of this network
multigraph(arr)

## Now with a force directed algorithm
multigraph(arr, layout = "force")

## As weighted graph
multigraph(arr, weighted = TRUE)

## As signed graph
multigraph(arr, signed = TRUE)

## With loops and a customized vertex size
multigraph(arr, cex = 3, loops = TRUE)
Description
A function to compute the graph coordinated system with a stress majorization layout algorithm

Usage
stsm(net, seed = seed, maxiter = 40, drp, jitter, method, ...)

Arguments
- net: an array representing the network relations
- seed: (mandatory) the seed of the initial layout (see details)
- maxiter: (optional) the maximum number of iterations
- drp: (optional) for valued networks, drop values less than specified
- jitter: (optional) jitter in the layout
- method: (optional) initial distance method (default binary)
- ...: Additional argument items

Details
This function is meant as an internal routine for graph visualization. However, it can be used with the coord option both in multigraph and in bmgraph where NULL in seed implies a random seed based on the clock watch of the computer.

Value
A data frame with a coordinated system with two columns representing the abscissa and the ordinate in a two-dimensional rectangular Cartesian coordinate system.

Author(s)
Antonio Rivero Ostoic

References

See Also
multigraph, bmgraph, frcd, conc
Examples

```r
## Create the data: two binary relations among three elements
arr <- round( replace( array(runif(18), c(3,3,2)), array(runif(18),
    c(3,3,2))>.5, 3 )
)

## Coordinates for the stress majorization layout with random start
coord <- stsm(arr, seed = NULL)

## Plot multigraph with customized coordinates
multigraph(arr, coord = coord)
```
Index

* IO
  multigraph-package, 2

* data
  multigraph-package, 2

* file
  multigraph-package, 2

* graphics
  bmgraph, 3
  ccgraph, 6
  conc, 8
  frcd, 9
  mlgraph, 11
  multigraph, 14
  stsm, 18

* graphs
  multigraph-package, 2

* manip
  conc, 8
  frcd, 9
  multigraph-package, 2
  stsm, 18

* math
  ccgraph, 6
  bmgraph, 3, 9, 10, 13, 17, 18
  ccgraph, 6, 17
  conc, 5, 8, 8, 10, 13, 17, 18
  frcd, 5, 8, 9, 9, 13, 17, 18
  incubs, 2
  mlgraph, 11
  mlvl, 13
  multigraph, 5, 8–10, 13, 14, 18
  multigraph-package, 2
  par, 5, 7, 13, 16
  semigroup, 8

  stsm, 5, 9, 10, 13, 17, 18
  transf, 2
  zbind, 2