Package ‘cooptrees’

February 19, 2015

Type     Package
Version  1.0
Date     2014-09-01
Title    Cooperative aspects of optimal trees in weighted graphs
Author   Manuel Fontenla
Maintainer Manuel Fontenla <manu.fontenla@gmail.com>
Depends  R (>= 2.7.0), igraph (>= 0.7.1), optrees (>= 1.0)
Imports  gtools
Description Computes several cooperative games and allocation rules associated with minimum cost spanning tree problems and minimum cost arborescence problems.
License  GPL-3
NeedsCompilation no
Repository CRAN
Date/Publication 2014-09-04 19:13:22

R topics documented:

  cooptrees-package .................................................. 2
  maBird ............................................................... 3
  maCooperative ...................................................... 4
  maERO ............................................................... 4
  maGames ............................................................. 5
  maIrreducible ...................................................... 6
  maPessimistic ...................................................... 7
  maRules .............................................................. 7
  mstBird ............................................................. 8
  mstCooperative .................................................... 9
  mstDuttaKar ........................................................ 10
  mstEROKruskal ..................................................... 11
  mstGames ........................................................... 12
  mstIrreducible ..................................................... 13
  mstKar ............................................................... 13
cooptrees-package

mstOptimistic .......................................................... 14
mstPessimistic .......................................................... 15
mstRules ................................................................. 16
shapleyValue ............................................................. 17

Index 18

---

Cooperative aspects of optimal trees in weighted graphs

---

Description

Computes several cooperative games and allocation rules associated with minimum cost spanning tree problems and minimum cost arborescence problems.

Details

Package: cooptrees
Type: Package
Version: 1.0
Date: 2014-09-01
License: GPL-3

The most important functions are mstCooperative and maCooperative. The other functions included in the package are auxiliary ones that can be used independently.

Author(s)

Manuel Fontenla
Maintainer: Manuel Fontenla <manu.fontenla@gmail.com>

References

maBird


Description

Given a graph with a minimum cost arborescence, the maBird function divides the cost of this arborescence among the agents. For that purpose, it uses the Bird rule, where each agent pays the cost of the last arc that connects him to the source.

Usage

maBird(nodes, arcs)

Arguments

nodes vector containing the nodes of the graph, identified by a number that goes from 1 to the order of the graph.

arcs matrix with the list of arcs of the graph. Each row represents one arc. The first two columns contain the two endpoints of each arc and the third column contains their weights.

Value

maBird returns a matrix with the agents and their costs.

See Also

The more general function maRules.

Examples

# Graph
nodes <- 1:4
arcs <- matrix(c(1,2,7, 1,3,6, 1,4,4, 2,3,8, 2,4,6, 3,2,6, 3,4,5, 4,2,5, 4,3,7), ncol = 3, byrow = TRUE)

# Bird
maBird(nodes, arcs)
\textbf{maCooperative} \hspace{1cm} \textit{Cooperation in minimum cost arborescence problems}

\textbf{Description}

Given a graph with at least one minimum cost arborescence, the \texttt{maCooperative} function computes the cooperative and "Bird" and "ERO" rules.

\textbf{Usage}

\texttt{maCooperative(nodes, arcs, show.data = TRUE)}

\textbf{Arguments}

- \texttt{nodes}: vector containing the nodes of the graph, identified by a number that goes from 1 to the order of the graph.
- \texttt{arcs}: matrix with the list of arcs of the graph. Each row represents one arc. The first two columns contain the two endpoints of each arc and the third column contains their weights.
- \texttt{show.data}: logical value indicating if the function displays the console output (TRUE) or not (FALSE). By default its value is TRUE.

\textbf{Value}

\texttt{maCooperative} returns and prints a list with the cooperative games and allocation rules of a minimum cost arborescence problem.

\textbf{Examples}

\begin{verbatim}
# Graph
nodes <- 1:4
arcs <- matrix(c(1,2,2, 1,3,3, 1,4,4, 2,3,3, 2,4,4, 3,2,3,
                 3,4,1, 4,2,1, 4,3,2), ncol = 3, byrow = TRUE)
# Cooperation in minimum cost arborescence problems
maCooperative(nodes, arcs)
\end{verbatim}

\textbf{maERO} \hspace{1cm} \textit{ERO rule for minimum cost arborescence problems}

\textbf{Description}

Given a graph with a minimum cost arborescence, the \texttt{maERO} function divides the cost of the arborescence among the agents according to the ERO rule. For that purpose, the irreducible form of the problem is obtained. The ERO rule is just the Shapley value of the cooperative game associated with the irreducible form.
**maGames**

**Usage**

maERO(nodes, arcs)

**Arguments**

- nodes: vector containing the nodes of the graph, identified by a number that goes from 1 to the order of the graph.
- arcs: matrix with the list of arcs of the graph. Each row represents one arc. The first two columns contain the two endpoints of each arc and the third column contains their weights.

**Value**

maERO returns a matrix with the agents and their costs.

**See Also**

The more general function maRules.

**Examples**

```r
# Graphs
nodes <- 1:4
arcs <- matrix(c(1,2,7, 1,3,6, 1,4,4, 2,3,8, 2,4,6, 3,2,6,
                 3,4,5, 4,2,5, 4,3,7), ncol = 3, byrow = TRUE)

# ERO
maERO(nodes, arcs)
```

---

### maGames

*Cooperative game associated with minimum cost arborescences*

**Description**

Given a graph with at least one minimum cost arborescence the maGames function builds the cooperative game associated with it.

**Usage**

maGames(nodes, arcs, game = "pessimistic", show.data = TRUE)

**Arguments**

- nodes: vector containing the nodes of the graph, identified by a number that goes from 1 to the order of the graph.
- arcs: matrix with the list of arcs of the graph. Each row represents one arc. The first two columns contain the two endpoints of each arc and the third column contains their weights.


game denotes the game to be obtained, known as the "pessimistic" game.

show.data logical value indicating if the function displays the console output (TRUE) or not (FALSE). By default its value is TRUE.

Value

maGames returns a vector with the characteristic function of the cooperative game associated with the graph.

Examples

# Graph
nodes <- 1:4
arcs <- matrix(c(1,2,7, 1,3,6, 1,4,4, 2,3,8, 2,4,6, 3,2,6,
                 3,4,5, 4,2,5, 4,3,7), ncol = 3, byrow = TRUE)
# Cooperative games
maGames(nodes, arcs, game = "pessimistic")

maIrreducible

Irreducible form for a minimum cost arborescence problem

Description

Given a graph with at least one minimum cost arborescence the maIrreducible function obtains the irreducible form.

Usage

maIrreducible(nodes, arcs)

Arguments

nodes vector containing the nodes of the graph, identified by a number that goes from 1 to the order of the graph.

arcs matrix with the list of arcs of the graph. Each row represents one arc. The first two columns contain the two endpoints of each arc and the third column contains their weights.

Value

maIrreducible returns a matrix with the list of arcs of the irreducible form.

Examples

# Graph
nodes <- 1:4
arcs <- matrix(c(1,2,7, 1,3,6, 1,4,4, 2,3,8, 2,4,6, 3,2,6,
                 3,4,5, 4,2,5, 4,3,7), ncol = 3, byrow = TRUE)
# Irreducible form
maIrreducible(nodes, arcs)
**maPessimistic**

Pessimistic game associated with minimum cost arborescences

**Description**

Given a graph with at least one minimum cost arborescence, the `maPessimistic` function builds the pessimistic game.

**Usage**

`maPessimistic(nodes, arcs)`

**Arguments**

- **nodes** vector containing the nodes of the graph, identified by a number that goes from 1 to the order of the graph.
- **arcs** matrix with the list of arcs of the graph. Each row represents one arc. The first two columns contain the two endpoints of each arc and the third column contains their weights.

**Value**

`maPessimistic` returns a vector with the characteristic function of the pessimitic game.

**Examples**

```r
# Graph
nodes <- 1:4
arcs <- matrix(c(1,2,7, 1,3,6, 1,4,4, 2,3,8, 2,4,6, 3,2,6,
                 3,4,5, 4,2,5, 4,3,7), ncol = 3, byrow = TRUE)
# Pessimistic game
maPessimistic(nodes, arcs)
```

**marules**

Allocation rules for minimum cost arborescence problems

**Description**

Given a graph with at least one minimum cost arborescence, the `marules` function divides the cost of the arborescence among the agents according to "Bird" and "ERO" rules.

**Usage**

`marules(nodes, arcs, rule, show.data = TRUE)`
Arguments

nodes vector containing the nodes of the graph, identified by a number that goes from 1 to the order of the graph.

arcs matrix with the list of arcs of the graph. Each row represents one arc. The first two columns contain the two endpoints of each arc and the third column contains their weights.

rule denotes the chosen allocation rule: "Bird" or "ERO".

show.data logical value indicating if the function displays the console output (TRUE) or not (FALSE). By default its value is TRUE.

Value

maRules returns a matrix with the agents and their costs. It also prints the result in console.

Examples

# Graph
nodes <- 1:4
arcs <- matrix(c(1,2,7, 1,3,6, 1,4,4, 2,3,8, 2,4,6, 3,2,6,
                 3,4,5, 4,2,5, 4,3,7), ncol = 3, byrow = TRUE)

# Allocation rules
maRules(nodes, arcs, rule = "Bird")
maRules(nodes, arcs, rule = "ERO")

mstBird

Bird rule for minimum cost spanning tree problems

Description

Given a graph with at least one minimum cost spanning tree, the mstBird function divides the cost of the tree obtained with Prim’s algorithm among the agents. For that purpose it uses the Bird rule, where each agent pays the cost of the arc that connects him to the tree source.

Usage

mstBird(nodes, arcs)

Arguments

nodes vector containing the nodes of the graph, identified by a number that goes from 1 to the order of the graph.

arcs matrix with the list of arcs of the graph. Each row represents one arc. The first two columns contain the two endpoints of each arc and the third column contains their weights.
mstCooperative

Value

mstBird returns a matrix with the agents and their costs.

See Also

The more general function mstRules.

Examples

# Graphs
nodes <- 1:4
arcs <- matrix(c(1,2,6, 1,3,10, 1,4,6, 2,3,4, 2,4,6, 3,4,4),
               byrow = TRUE, ncol = 3)
# Bird
mstBird(nodes, arcs)

mstCooperative

Cooperation in minimum cost spanning tree problems

Description

Given a graph with at least one minimum cost spanning tree, the mstCooperative function computes the pessimistic and optimistic games; and the most known allocation rules: "Bird", "Dutta-Kar", "Kar" and "ERO".

Usage

mstCooperative(nodes, arcs, show.data = TRUE)

Arguments

nodes vector containing the nodes of the graph, identified by a number that goes from 1 to the order of the graph.

arcs matrix with the list of arcs of the graph. Each row represents one arc. The first two columns contain the two endpoints of each arc and the third column contains their weights.

show.data logical value indicating if the function displays the console output (TRUE) or not (FALSE). By default its value is TRUE.

Value

mstCooperative returns and print a list with the cooperative games and the allocation rules of a minimum cost spanning tree problem.
Examples

# Graph
nodes <- 1:4
arcs <- matrix(c(1,2,6, 1,3,10, 1,4,6, 2,3,4, 2,4,6, 3,4,4),
               byrow = TRUE, ncol = 3)
# Cooperation in minimum cost spanning tree problems
mstCooperative(nodes, arcs)

---

Description

Given a graph with at least one minimum cost spanning tree, the mstDuttaKar function divides the cost of the tree obtained with Prim’s algorithm among the agents according to the Dutta-Kar rule. This rule specifies that each agent chooses to pay the minimum cost between the last arc that connects him to the source and the cost that rejects his successor. The order is set by Prim’s algorithm.

Usage

mstDuttaKar(nodes, arcs)

Arguments

- **nodes**: vector containing the nodes of the graph, identified by a number that goes from 1 to the order of the graph.
- **arcs**: matrix with the list of arcs of the graph. Each row represents one arc. The first two columns contain the two endpoints of each arc and the third column contains their weights.

Value

mstDuttaKar returns a matrix with the agents and their costs.

See Also

The more general function mstRules.

Examples

# Graph
nodes <- 1:4
arcs <- matrix(c(1,2,6, 1,3,10, 1,4,6, 2,3,4, 2,4,6, 3,4,4),
               byrow = TRUE, ncol = 3)
# Dutta-Kar
mstDuttaKar(nodes, arcs)
**mstEROKruskal**

**ERO rule for minimum cost spanning tree problems with Kruskal’s algorithm**

**Description**

Given a graph with at least one minimum cost spanning tree, the mstEROKruskal function divides the cost of the tree among the agents according to the ERO rule.

**Usage**

```r
mstEROKruskal(nodes, arcs)
```

**Arguments**

- `nodes` vector containing the nodes of the graph, identified by a number from 1 until the order of the graph.
- `arcs` matrix with the list of arcs of the graph. Each row represents one arc. The first two columns contain the two endpoints of each arc and the third column contains their weights.

**Value**

mstEROKruskal returns a matrix with the agents and their costs.

**See Also**

The more general function mstRules.

**Examples**

```r
# Graph
dodes <- c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
arcs <- matrix(c(1,2,6, 1,3,10, 1,4,6, 2,3,4, 2,4,6, 3,4,4), byrow = TRUE, ncol = 3)

# ERO with Kruskal
mstEROKruskal(nodes, arcs)
```
Description

Given a graph with at least one minimum cost spanning tree, mstGames builds both cooperative games: the pessimistic and the optimistic game.

The pessimistic game associated with a minimum cost spanning tree problem is a cooperative game in which every coalition of agents obtains the minimum cost assuming that the agents outside the coalition are not connected.

The optimistic game associated with a minimum cost spanning tree problem is a cooperative game in which every coalition of agents obtains the minimum cost assuming that the agents outside the coalition are connected. Thus, the agents in the coalition can benefit from their connections to the source.

Usage

mstGames(nodes, arcs, game = "pessimistic", show.data = TRUE)

Arguments

- **nodes**: vector containing the nodes of the graph, identified by a number that goes from 1 to the order of the graph.
- **arcs**: matrix with the list of arcs of the graph. Each row represents one arc. The first two columns contain the two endpoints of each arc and the third column contains their weights.
- **game**: denotes the game that we want to obtain: "pessimistic" or "optimistic".
- **show.data**: logical value indicating if the function displays the console output (TRUE) or not (FALSE). By default its value is TRUE.

Value

mstGames returns a vector with the characteristic function of the selected game associated with the graph and prints the result in console.

Examples

```r
# Graph
nodes <- 1:4
arcs <- matrix(c(1,2,6, 1,3,10, 1,4,6, 2,3,4, 2,4,6, 3,4,4),
              byrow = TRUE, ncol = 3)

# Cooperative games
mstGames(nodes, arcs, game = "pessimistic")
mstGames(nodes, arcs, game = "optimistic")
```
mstIrreducible

Description
Given a graph with at least one minimum cost spanning tree, the mstIrreducible function obtains the irreducible form.

Usage
mstIrreducible(nodes, arcs)

Arguments
- **nodes**: vector containing the nodes of the graph, identified by a number that goes from 1 to the order of the graph.
- **arcs**: matrix with the list of arcs of the graph. Each row represents one arc. The first two columns contain the two endpoints of each arc and the third column contains their weights.

Value
mstIrreducible returns a matrix with the list of arcs of the irreducible form.

Examples
```r
# Graph
nodes <- 1:4
arcs <- matrix(c(1,2,6, 1,3,10, 1,4,6, 2,3,4, 2,4,6, 3,4,4),
               byrow = TRUE, ncol = 3)
# Irreducible form
mstIrreducible(nodes, arcs)
```

mstKar

Kar rule for minimum cost spanning tree problems

Description
Given a graph with at least one minimum cost spanning tree, the mstKar function divides the cost of the tree among the agents according to the Kar rule. That rule is obtained with the Shapley value of the pessimistic game.

Usage
mstKar(nodes, arcs)
mstOptimistic

Arguments

nodes vector containing the nodes of the graph, identified by a number that goes from 1 to the order of the graph.

arcs matrix with the list of arcs of the graph. Each row represents one arc. The first two columns contain the two endpoints of each arc and the third column contains their weights.

Value

mstKar returns a matrix with the agents and their costs.

See Also

The more general function mstRules.

Examples

# Graph
nodes <- 1:4
arcs <- matrix(c(1,2,6, 1,3,1, 1,4,6, 2,3,4, 2,4,6, 3,4,4),
                   byrow = TRUE, ncol = 3)
# Kar
mstKar(nodes, arcs)
mstpessimistic

See Also

The more general function mstGames.

Examples

# Graph
nodes <- 1:4
arcs <- matrix(c(1,2,6, 1,3,10, 1,4,6, 2,3,4, 2,4,6, 3,4,4),
   byrow = TRUE, ncol = 3)
# Optimistic game
mstoptimistic(nodes, arcs)

mstpessimistic

Pessimistic game from a minimum cost spanning tree problem

Description

Given a graph with at least one minimum cost spanning tree, the mstpessimistic function builds
the pessimistic game.

Usage

mstpessimistic(nodes, arcs)

Arguments

nodes vector containing the nodes of the graph, identified by a number that goes from
1 to the order of the graph.

arcs matrix with the list of arcs of the graph. Each row represents one arc. The first
two columns contain the two endpoints of each arc and the third column contains
their weights.

Value

mstpessimistic returns a vector with the characteristic function of the pessimistic game.

See Also

The more general function mstGames.

Examples

# Graph
nodes <- 1:4
arcs <- matrix(c(1,2,6, 1,3,10, 1,4,6, 2,3,4, 2,4,6, 3,4,4),
   byrow = TRUE, ncol = 3)
# Pessimistic game
mstpessimistic(nodes, arcs)
mstRules

Allocation rules for minimum cost spanning tree problem

Description

Given a graph with at least one minimum cost spanning tree, the mstRules function divides the cost of the tree among the agents according to the most known rules: "Bird", "Dutta-Kar", "Kar", "ERO".

Usage

mstRules(nodes, arcs, rule, algorithm = "Kruskal", show.data = TRUE)

Arguments

nodes vector containing the nodes of the graph, identified by a number that goes from 1 to the order of the graph.

arcs matrix with the list of arcs of the graph. Each row represents one arc. The first two columns contain the two endpoints of each arc and the third column contains their weights.

rule denotes the chosen allocation rule: "Bird", "Dutta-Kar", "Kar" or "ERO".

algorithm denotes the algorithm used with the ERO rule: "Kruskal".

show.data logical value indicating if the function displays the console output TRUE or no FALSE. The deafult is TRUE.

Value

mstRules returns a matrix with the agents and the cost that each one of them has to pay. It also prints the result in console.

Examples

# Graph
nodes <- 1:4
arcs <- matrix(c(1,2,6, 1,3,10, 1,4,6, 2,3,4, 2,4,6, 3,4,4),
byrow = TRUE, ncol = 3)
# Allocation Rules
mstRules(nodes, arcs, rule = "Bird")
mstRules(nodes, arcs, rule = "Dutta-Kar")
mstRules(nodes, arcs, rule = "Kar")
mstRules(nodes, arcs, rule = "ERO", algorithm = "Kruskal")
Description

Given a cooperative game, the shapleyValue function computes its Shapley value.

Usage

shapleyValue(n, S = NULL, v)

Arguments

- `n` number of players in the cooperative game.
- `S` vector with all the possible coalitions. If none has been specified the function generates it automatically.
- `v` vector with the characteristic function of the cooperative game.

Details

The Shapley value is a solution concept in cooperative game theory proposed by Lloyd Shapley in 1953. It is obtained as the average of the marginal contributions of the players associated with all the possible orders of the players.

Value

The shapleyValue functions returns a matrix with all the marginal contributions of the players (contributions) and a vector with the Shapley value (value).

Examples

```r
# Cooperative game
n <- 3  # players
v <- c(4, 4, 4, 8, 8, 8, 14)  # characteristic function
# Shapley value
shapleyValue(n, v = v)
```
Index

cooptrees (cooptrees-package), 2
cooptrees-package, 2

maBird, 3
maCooperative, 2, 4
maERO, 4
maGames, 5
maIrreducible, 6
maPessimistic, 7
maRules, 3, 5, 7
mstBird, 8
mstCooperative, 2, 9
mstDuttaKar, 10
mstEROKruskal, 11
mstGames, 12, 15
mstIrreducible, 13
mstKar, 13
mstOptimistic, 14
mstPessimistic, 15
mstRules, 9–11, 14, 16

shapleyValue, 17