Package ‘pcSteiner’

October 14, 2022

Type Package
Version 1.0.0.1
Date 2020-08-21

Title Convenient Tool for Solving the Prize-Collecting Steiner Tree Problem

Description The Prize-Collecting Steiner Tree problem asks to find a subgraph connecting a given set of vertices with the most expensive nodes and least expensive edges. Since it is proven to be NP-hard, exact and efficient algorithm does not exist. This package provides convenient functionality for obtaining an approximate solution to this problem using loopy belief propagation algorithm.

Author Aleksei Krasikov <krasikov.as@phystech.edu>
Maintainer Aleksei Krasikov <krasikov.as@phystech.edu>

URL https://github.com/kraskov/pcSteiner

BugReports https://github.com/krashkov/pcSteiner/issues

Depends R (>= 3.1.0), igraph (>= 0.6.0)
Imports stats
Suggests knitr, rmarkdown
VignetteBuilder knitr

License GPL-3
Repository CRAN
Encoding UTF-8

R topics documented:

pcs.tree ................................................................. 2

Index 3

1
pcs.tree

Solve the Prize-Collecting Steiner Tree problem

Description

Solve the Prize-Collecting Steiner Tree problem.

Usage

```
pcs.tree(graph, terminals, lambda, root, depth, eps, max_iter, terminal_infty=10000)
```

Arguments

- **graph**: an igraph graph.
- **terminals**: a numeric or character vector which contains either ids or names of terminal nodes.
- **lambda**: a numeric parameter which establishes a ratio between edge costs and node prizes (see Sec.1 or Sec.3 in the vignette).
- **root**: a numeric or character scalar which corresponds to either id or name of a root (see Sec.3 in the vignette).
- **depth**: a numeric scalar which sets depth of the resultant tree (see Sec.3 in the vignette).
- **eps**: a numeric scalar which specifies tolerance for termination.
- **max_iter**: a numeric scalar which specifies maximum number of iterations.
- **terminal_infty**: a numeric scalar which corresponds to a prize for each terminal node. This value should be large enough to ensure that all terminals will be presented in a solution.

Value

Returns a list with cost and edges of the final tree.

References


Examples

```
g <- graph('Bull')
E(g)$costs <- c(3, 3, 3, 3)
V(g)$prizes <- c(10, 2, 2, 2)
treeData <- pcs.tree(graph=g, terminals=c(4,5), lambda=1, root=3, depth=5, eps=1e-3, max_iter=10)
```
Index

pcs.tree, 2