Package ‘TSPpackage’

March 1, 2020

Type Package

Title Etude statistique du problème du voyageur de commerce

Version 1.0

Date 2020-02-18

Author Irene Gannaz

Maintainer Irene Gannaz <irene.gannaz@insa-lyon.fr>

Description Paquet pour le TP de statistique. Attention, ce paquet est destiné à une utilisation interne dans les séances de TP et ne doit pas etre diffusé, par respect des droits d'auteurs. Les codes donnant les trajets hamiltoniens exacts du voyageur de commerce par resolution dynamique et par ``branch & bound'' ont été fournis par Christine Solnon et adaptés en C++ pour une intégration dans R par Rcpp.

License GPL (>= 2)

Encoding UTF-8

Imports Rcpp (>= 1.0.3), maps, TSP

LinkingTo Rcpp

RoxygenNote 6.1.1

NeedsCompilation yes

R topics documented:

`calculeLongueur` *calcule la longueur d'un chemin a partir d'une matrice de couts*

Description

calcule la longueur d'un chemin a partir d'une matrice de couts

Usage

```
calculeLongueur(couts, path)
```

Arguments

`couts,`	matrice carree contenant les couts
`path,`	vecteur avec lec sommets visites

Value

longueur du parcours (ferme)

`distance` *Evaluation of the cartesian distance given the (x,y)-coordinates.*

Description

Evaluation of the cartesian distance given the (x,y)-coordinates.

Usage

```
distance(coordonnees)
```

Arguments

`coordonnees`	Bidimensionnal matrix containing the coordinates: abscisse in first column and in second column

Value

Returns the matrix of distances between each points

See Also

distanceGPS for distance between GPS coordinates

Examples

```
nbSommets <- 5
points <- cbind(x=runif(nbSommets),y=runif(nbSommets))
plot(points, pch=paste(1:nbSommets))
distance(points)
```

`distanceGPS` *Evaluation of the geodesic distance given the latitude and longitude coordinates.*

Description

Evaluation of the geodesic distance given the latitude and longitude coordinates.

Usage

```
distanceGPS(coordonneesGPS)
```

Arguments

`coordonneesGPS`	Bidimensionnal matrix containing the coordinates: latitude in first column and longitude in second column

Value

Returns the matrix of distances between each points

See Also

distance for cartesian distance given coordinates

Examples

```
nbSommets <- 5
points <- data.frame(x=runif(nbSommets,-0.5,6),y=runif(nbSommets,44,49))
plotTrace(points, sample.int(5,5))
distance(points)
```

plotTrace *Plot a map of France and a path between coordinates.*

Description

Plot a map of France and a path between coordinates.

Usage

```
plotTrace(coordonnees, path, title = "France")
```

Arguments

`coordonnees`	Bidimensionnal matrix containing the coordinates of the path steps: abscissa in first column and ordinate in second column
`path`	Vector of the order of visited points
`title`	Title of the plot

Examples

```
nbSommets <- 5
points <- data.frame(x=runif(nbSommets,-0.5,6),y=runif(nbSommets,44,49))
path <- sample.int(5,5)
plotTrace(points,path)
```

TSPbranch *Solution du TSP sur un graphe, par branch and bound*

Description

Solution du TSP sur un graphe, par branch and bound

Usage

```
TSPbranch(couts)
```

Arguments

`couts,`	matrice carree contenant les couts

Value

vecteur contenant la liste des noeuds parcourus

Examples

```
nbSommets <- 5
points <- data.frame(x=runif(nbSommets),y=runif(nbSommets))
plot(points, pch=paste(1:nbSommets))
dist <- distance(points)
TSPbranch(dist)
```

`TSPdynamique` *Solution du TSP sur un graphe, methode dynamique*

Description

Solution du TSP sur un graphe, methode dynamique

Usage

```
TSPdynamique(couts)
```

Arguments

`couts`	matrice carree contenant les couts

Value

vecteur contenant la liste des noeuds parcourus

Examples

```
nbSommets <- 5
points <- data.frame(x=runif(nbSommets),y=runif(nbSommets))
plot(points, pch=paste(1:nbSommets))
dist <- distance(points)
TSPdynamique(dist)
```

`TSPnearest` *TSP par plus proches voisins sur un graphe*

Description

TSP par plus proches voisins sur un graphe

Usage

```
TSPnearest(couts)
```

Arguments

`couts,`	matrice carree contenant les couts

Value

vecteur contenant la liste des noeuds parcourus

Examples

```
nbSommets <- 5
points <- data.frame(x=runif(nbSommets),y=runif(nbSommets))
plot(points, pch=paste(1:nbSommets))
dist <- distance(points)
TSPnearest(dist)
```

TSPpackage *Package for the practical session in statistics*

Description

The package deals with the traveller salesman problem. Different solvers of Hamiltonian paths on graphs are implemented. Two algorithms finding optimal solutions were provided by Christine Solnon. They are based on the AAIA practical sessions. Other algorithms are coming from package TSP, available on the cran.

Details

The main function is TSPsolve, which returns the length of the Hamiltonian paths obtained by the methods implemented.

Author(s)

Irène Gannaz

References

TSP package documentation and AAIA documents of Christine Solon and Pierre-Edouard Portier.

See Also

TSP

Examples

```
  nbSommets <- 5
  points <- data.frame(x=runif(nbSommets),y=runif(nbSommets))
  plot(points, pch=paste(1:nbSommets))
  dist <- distance(points)
  TSPsolve(dist,'branch')
```

`TSPsolve` *Evaluation of the length of Hamiltonian paths given a cost matrix*

Description

Evaluation of the length of Hamiltonian paths given a cost matrix

Usage

```
TSPsolve(costs, method)
```

Arguments

`costs`	matrix of the costs
`method`	**'functions from TSP package:'** "nearest_insertion", "cheapest_insertion", "farthest_insertion", "arbitrary_insertion", "nn", "repetitive_nn", "two_opt" see TSP documentation for more details, non optimals hamiltonian path **'nearest'** hamiltonian path by nearest neighbour principle **'dyn'** costly dynamical resolution for optimal hamiltonian path **'branch'** branch & bound algorithm for optimal hamiltonian path

Value

Returns the matrix of distances between each points

Examples

```
nbSommets <- 5
points <- data.frame(x=runif(nbSommets),y=runif(nbSommets))
dist <- distance(points)
# length of the path obtained by nearest neighbours
TSPsolve(dist,'nearest')
# length of the path obtained by branch & bound
TSPsolve(dist,'branch')
```

Index