Package 'IBFS'

May 7, 2026

Type Package

Title Initial Basic Feasible Solution for Transportation Problem

Version 1.0.0

Description The initial basic feasible solution (IBFS) is a significant step to achieve the minimal total cost (optimal solution) of the transportation problem. However, the existing methods of IBFS do not always provide a good feasible solution which can reduce the number of iterations to find the optimal solution. This initial basic feasible solution can be obtained by using any of the following methods.
a) North West Corner Method.
b) Least Cost Method.
c) Row Minimum Method.
d) Column Minimum Method.
e) Vogel's Approximation Method.
etc.
For more technical details about the algorithms please refer below URLs.
<https://theintactone.com/2018/05/24/ds-u2-topic-8-transportation-problems-initial-basic-feasible-solution/>.
<https://www.brainkart.com/article/Methods-of-finding-initial-Basic-Feasible-Solutions_39037/>.
<https://myhomeworkhelp.com/row-minima-method/>.
<https://myhomeworkhelp.com/column-minima-method/>.

License GPL-2

Encoding UTF-8

RoxygenNote 7.2.3

NeedsCompilation no

Author Royson Almeida [aut, cre]

Maintainer Royson Almeida <almeidaroyson09@yahoo.com>

Repository CRAN

Date/Publication 2023-01-09 10:53:14 UTC

Contents

CMM 2

LCM 3
NWCM 3
RMM 4
VAM 5

Index **6**

CMM *Column Minimum Method*

Description

In the column minimum method, the first column that is the lowest cost cell is exhausted. The objective is to allocate the maximum either at the first source or demand at the destinations or to satisfy both. This process must be continued for all the other reduced transportation costs until and unless the supply and demand are satisfied.

Usage

```
CMM(ex_matrix)
```

Arguments

ex_matrix	A cost matrix where last column must be the supply and last row must be the demand. Input matrix should not have any missing values (NA), otherwise function will throw an error. It should be balanced i.e. total demand must be equal to total supply.

Value

A List which contain the allocation matrix and the total optimized cost.

Examples

```
#Input matrix where last row is the Demand and last column is the Supply
ex_matrix=data.frame(D1=c(6,3,4,20),E1=c(4,8,4,95),F1=c(1,7,2,35),
                     Supply=c(50,40,60,150),row.names = c("A1","B1","C1","Demand"))
CMM(ex_matrix)
```

LCM *Least Cost Method*

Description

The main objective is to minimize the total transportation cost, transport as much as possible through those routes (cells) where the unit transportation cost is lowest. This method takes into account the minimum cost of transportation for obtaining the initial solution.

Usage

```
LCM(ex_matrix)
```

Arguments

ex_matrix	A cost matrix where last column must be the supply and last row must be the demand. Input matrix should not have any missing values (NA), otherwise function will throw an error. It should be balanced i.e. total demand must be equal to total supply.

Value

A List which contain the allocation matrix and the total optimized cost.

Examples

```
#Input matrix where last row is the Demand and last column is the Supply
ex_matrix=data.frame(D1=c(6,3,4,20),E1=c(4,8,4,95),F1=c(1,7,2,35),
                     Supply=c(50,40,60,150),row.names = c("A1","B1","C1","Demand"))
LCM(ex_matrix)
```

NWCM *North-West Corner Method*

Description

This method does not take into account the cost of transportation on any route of transportation.

Usage

```
NWCM(ex_matrix)
```

Arguments

ex_matrix	A cost matrix where last column must be the supply and last row must be the demand. Input matrix should not have any missing values (NA), otherwise function will throw an error. It should be balanced i.e. total demand must be equal to total supply.

Value

A List which contain the allocation matrix and the total optimized cost.

Examples

```
#Input matrix where last row is the Demand and last column is the Supply
ex_matrix=data.frame(D1=c(6,3,4,20),E1=c(4,8,4,95),F1=c(1,7,2,35),
                     Supply=c(50,40,60,150),row.names = c("A1","B1","C1","Demand"))
NWCM(ex_matrix)
```

RMM *Row Minimum Method*

Description

In the row minimum method, the first row that is the lowest cost cell is exhausted. The objective is to allocate the maximum either at the first source or demand at the destinations or to satisfy both. This process must be continued for all the other reduced transportation costs until and unless the supply and demand are satisfied.

Usage

```
RMM(ex_matrix)
```

Arguments

ex_matrix	A cost matrix where last column must be the supply and last row must be the demand. Input matrix should not have any missing values (NA), otherwise function will throw an error. It should be balanced i.e. total demand must be equal to total supply.

Value

A List which contain the allocation matrix and the total optimized cost.

Examples

```
#Input matrix where last row is the Demand and last column is the Supply
ex_matrix=data.frame(D1=c(6,3,4,20),E1=c(4,8,4,95),F1=c(1,7,2,35),
                     Supply=c(50,40,60,150),row.names = c("A1","B1","C1","Demand"))
RMM(ex_matrix)
```

VAM *Vogel's Approximation Method*

Description

Vogel's approximation method is preferred over NWCM and LCM methods. In this method allocation is made on the basis of the opportunity cost that would have been incurred if the allocation in certain cells with minimum unit transportation cost were missed. Hence, allocations are made in such a way that the penelty cost is minimized. An initial solution obtained by using this method is nearer to an optimal solution or is the optimal solution itself.

Usage

```
VAM(ex_matrix)
```

Arguments

ex_matrix	A cost matrix where last column must be the supply and last row must be the demand. Input matrix should not have any missing values (NA), otherwise function will throw an error. It should be balanced i.e. total demand must be equal to total supply.

Value

A List which contain the allocation matrix and the total optimized cost.

Examples

```
#Input matrix where last row is the Demand and last column is the Supply
ex_matrix=data.frame(D1=c(6,3,4,20),E1=c(4,8,4,95),F1=c(1,7,2,35),
                     Supply=c(50,40,60,150),row.names = c("A1","B1","C1","Demand"))
VAM(ex_matrix)
```

Index

CMM, 2

LCM, 3

NWCM, 3

RMM, 4

VAM, 5