Package ‘mfpp’

August 21, 2023

Type Package
Title 'Matrix-Based Flexible Project Planning'
Version 0.0.5
Maintainer Zsolt T. Kosztyan <kosztyan.zsolt@gtk.uni-pannon.hu>
Description Matrix-Based Flexible Project Planning. This package models, plans, and schedules flexible, such as agile, extreme, and hybrid project plans. The package contains project planning, scheduling, and risk assessment functions. Kosztyan (2022) <doi:10.1016/j.softx.2022.100973>.
License GPL (>= 2)
Encoding UTF-8
LazyData TRUE
URL https://github.com/kzst/mfpp
Depends R (>= 4.00)
Imports pracma, nsga2R, igraph, genalg, ggplot2, reshape2, Rfast
RoxygenNote 7.2.3
Suggests rmarkdown, knitr, Matrix
NeedsCompilation no
Author Zsolt T. Kosztyan [aut, cre],
Aamir Sagir [aut]
Repository CRAN
Date/Publication 2023-08-21 14:10:16 UTC

R topics documented:

mfpp-package ................................................................. 2
Batselier ................................................................. 3
Boctor ................................................................. 3
generatepdm ............................................................. 4
get.structures .......................................................... 6
is.flexible ............................................................ 7
mfpp-package

Matrix-Based Flexible Project Planning

Description
Matrix-Based Flexible Project Planning. This package model, plan and schedule flexible, such as agile, extreme and hybrid project plans. The package contains project planning, scheduling and risk assessment functions.

Author(s)
Zsolt T. Kosztyan*, Aamir Saghir
*e-mail: kzst@gtk.uni-pannon.hu

References

See Also
generatepdm, get.structures, is.flexible, percent, phase1, phase2, phase3, plot.PDM_matrix, summary.PDM_matrix
**Batselier**  

*Real-life project database by Batselier and Vanhoucke (2015)*

**Description**

Real-life project database of single-mode individual projects by Batselier and Vanhoucke (2015)

**Usage**

```r
data("Batselier")
```

**Format**

A list (Collection_PDM) of PDM_list and PDM_constraint types of lists, where

- **PDM_list** contains PDM_matrix (PDM) and the number of completion modes (w), and number of resources (Rs)
- **PDM_const** a list of (renewable resource) constraints

**References**


**Examples**

```r
data(Batselier)

# Summary of data collection
summary(Batselier)

# Summary of the first project structure
summary(Batselier[[1]]$PDM_list)

# Summary of the first project constraints
summary(Batselier[[1]]$PDM_const)
```

---

**Boctor**  

*Sumulated project database by Boctor (1993)*

**Description**

Sumulated project database of multimode, individual projects by Boctor (1993)

**Usage**

```r
data("Boctor")
```
**generatepdm**

**Format**

A list (Collection_PDM) of PDM_list and PDM_constraint types of lists, where

- **PDM_list** contains PDM_matrix (PDM) and the number of completion modes (w), and number of resources (Rs)
- **PDM_const** a list of (renewable resource) constraints

**References**


**Examples**

```r
data(Boctor)

# Summary of data collection
summary(Boctor)

# Summary of the first project structure
summary(Boctor[[1]]$PDM_list)

# Summary of the first project constraints
summary(Boctor[[1]]$PDM_const)
```

---

**generatepdm**

*Function to generate a PDM matrix for flexible project planning problems.*

**Description**

Function to generate a PDM matrix for flexible project planning problems.

**Usage**

```r
generatepdm(N, ff, cf, mTD, mCD, mRD, w, nR, nW, scale=1.4, QD=FALSE, lst=FALSE)
```

**Arguments**

- **N** Number of tasks (a numeric value)
- **ff** flexibility factor between [0,1] (a numeric value).
- **cf** connectivity factor (a numeric integer).
- **mTD** max value of TD (a numeric positive integer).
- **mCD** max value of CD (a numeric positive integer).
- **mRD** max value of RD (a numeric positive integer).
generatepdm

- **w**: number of modes (a numeric integer).
- **nR**: number of resources (a numeric integer).
- **nW**: number of possible extra tasks (a positive numeric integer).
- **scale**: scale parameter, the default value is 1.4 (a numeric value).
- **QD**: logic parameter, the default value is FALSE, which means, the quality domain is neglected.
- **lst**: logic parameter, the default value is FALSE, which means, the output is a PDM matrix, while TRUE means the output is a PDM list including number of modes (w) and the number of resources (Rs).

**Value**

- **PDM**: [PEM, TD, CD, QD, RD] is a Project Domain Matrix (a numeric matrix), where:
  - **PEM**: An N by N upper triangular matrix of logic domain (a numeric matrix).
  - **TD**: N by w matrix of task durations (a numeric matrix).
  - **CD**: N by w matrix of cost demands (a numeric matrix).
  - **QD**: N by w matrix of quality parameters (a numeric matrix).
  - **RD**: N by w*nR matrix of resource demands (a numeric matrix).
- **w**: Number of completion modes (scalar, if the output is PDM list (lst=TRUE)).
- **Rs**: Number of resources (scalar, if the output is PDM list (lst=TRUE)).

**Author(s)**

Zsolt T. Kosztyan*, Aamir Saghir

E-mail: kzst@gtk.uni-pannon.hu

**References**


**See Also**

tpc, tpq, tpt, tpr, maxscore_PEM, generatepdm.

**Examples**

```r
# Generation of PDM matrix for flexible project planning MFPP package.

# Define number of modes, flexibility factor and connectivity factor of a project scenario.
N=4; ff=0.05; cf=0

# Define maximum value of time domain, cost domain and resource domain of a project scenario.
mTD=3; mCD=4; mRD=3

# Define number of modes, number of resources,
```
# number of possible extra tasks, scale and quality domain of a project scenerio.

w=2; nR=2; nW=2
scale=1.4  #(default value)

# Generation of PDM matrix for TTCP a project scenario using MFPP package.
generatepdm(N, ff, cf, mTD, mCD, mRD, w, nR, nW, scale=1.4)

# Generation of PDM list for TTCP a project scenario using MFPP package.
generatepdml(N, ff, cf, mTD, mCD, mRD, w, nR, nW, scale=1.4, QD=FALSE, lst=TRUE)

# Generation of PDM matrix for TQCTP a project scenario using MFPP package.
generatepdm(N, ff, cf, mTD, mCD, mRD, w, nR, nW, scale=1.4, QD=TRUE)

# Generation of PDM list for TQCTP a project scenario using MFPP package.
generatepdml(N, ff, cf, mTD, mCD, mRD, w, nR, nW, scale=1.4, QD=TRUE, lst=TRUE)

---

get.structures  
*Function to calculate minimal/maximal/most likely project structures.*

**Description**

Specify minimal, maximal, maximin, minimax, and most likely/most desired project structures.

**Usage**

```
get.structures(x, type=c("min", "max", "minimax", "maximin", "most"))
```

**Arguments**

- **x**  
  N by M PDM matrix or a PDM list.

- **type**  
  type of specified project structure. "min" is a minimal structure, where the time, cost demands, and as well as the quality parameters are minimal; "max" where the time, cost demands, and as well as the quality parameters are maximal; "minimax", where the cost and resource demands are minimal; "maximin", where the cost and resource demands are maximal; "most", where the total project score is maximal.

**Value**

- **PDMs**  
  set of PDM_matrix or a set of PDM_list (depending on the input type)

**Author(s)**

Zsolt T. Kosztyan*, Aamir Saghir

e-mail: kzst@gtk.uni-pannon.hu
is.flexible

References


See Also

generatepdm.

Examples

# Generation of PDM matrix for flexible project planning MFPP package.

# Define number of modes, flexibility factor and connectivity factor of a project scenrio.
N=4;ff=0.05;cf=0

# Define maximum value of time domain, Cost domain and Resources domain of a project scenrio.
mTD=3;mCD=4;mRD=3

# Define number of modes, number of resources, number of possible extra tasks, scale and quality domain of a project scenrio.
w=2;nR=2;nW=2
scale=1.4  #(default value)

# Generation of PDM matrix for TCTP a project scenario using MFPP package.
PDM<-generatepdm(N,ff,cf,mTD,mCD,mRD,w,nR,nW,scale=1.4)

# Get main structures.
PDMs<-get.structures(PDM)
PDMs

# Generation of PDM list for TCTP a project scenario using MFPP package.
PDM<-generatepdm(N,ff,cf,mTD,mCD,mRD,w,nR,nW,scale=1.4,QD=FALSE,lst=TRUE)

# Get main structures.
PDMs<-get.structures(PDM)
PDMs


is.flexible Function to check the flexibility of PDM matrix.

Description

Checking flexibility of PDM matrix or a PDM list.
Usage

\texttt{is.flexible(x)}

Arguments

\begin{itemize}
  \item \texttt{x} \hspace{1cm} \text{N by M PDM matrix or a PDM list.}
\end{itemize}

Value

\begin{itemize}
  \item \texttt{is.flexible} \hspace{1cm} \text{Boolean output whether the project is flexible or not.}
\end{itemize}

Author(s)

Zsolt T. Kosztyan*, Aamir Saghir

e-mail: kzst@gtk.uni-pannon.hu

References


See Also

\texttt{generatepdm}.

Examples

\begin{verbatim}
# Generation of PDM matrix for flexible project planning MFPP package.

# Define number of modes, flexibility factor and connectivity factor of a project scenario.
N=4;ff=0.05;cf=0

# Define maximum value of time domain, Cost domain and Resources domain of a project scenario.
mTD=3;mCD=4;mRD=3

# Define number of modes, number of resources,
# number of possible extra tasks, scale and quality domain of a project scenario.
w=2;nR=2;nW=2
scale=1.4  #(default value)

# Generation of PDM matrix for TCTP a project scenario using MFPP package.
PDM<-generatepdm(N,ff,cf,mTD,mCD,mRD,w,nR,nW,scale=1.4)

# Flexibility check.
is.flexible(PDM)

# Flexibility check of rounded values.
is.flexible(round(PDM))
\end{verbatim}
Function to calculate maximal score value (PMAX) of possible project scenarios.

Description
Calculate maximal score value (PMAX) of possible project scenarios.

Usage
maxscore_PEM(PEM, P=PEM, Q=1-PEM)

Arguments
- **PEM**: N by N upper triangular adjacency matrix of logic network (a numeric matrix).
- **P**: N by N score matrix of task/dependency inclusion (a numeric matrix). The default P matrix is P=PEM
- **Q**: N by N score matrix of task/dependency exclusion (a numeric matrix). The default Q matrix is Q=1-PEM

Value
- **score**: The maximal score value of the project scenario (a scalar).

Author(s)
Zsolt T. Kosztyan*, Aamir Saghir
e-mail: kzst@gtk.uni-pannon.hu

References

See Also
- tpc, tpq, tpr, tpt.
Examples

# Calculation of the maximal score value of the project scenario using MFPP package.

# Define a 3 by 3 upper triangular adjacency matrix (PEM) of logic domain of a project.
PEM <- rbind(c(0.8,0.4,0.8),
c(0.0,0.7,0.7),
c(0.0,0.0,0.4))

# Define a 3 by 3 score matrix of task/dependency inclusion.
P <- PEM

# Define a 3 by 3 score matrix of task/dependency exclusion.
Q <- 1-P

# Calculation of the maximal score value of the project using MFPP package.
maxscore_PEM(PEM,P, Q)

minscore_PEM

Function to calculate minimal score value of possible project scenarios.

Description
Calculate minimal score value (PMIN) of possible project scenarios.

Usage

minscore_PEM(PEM,P=PEM, Q=1-PEM)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PEM</td>
<td>An N by N upper triangular adjacency matrix of logic network (a numeric matrix).</td>
</tr>
<tr>
<td>P</td>
<td>N by N score matrix of task/dependency inclusion (a numeric matrix). The default P matrix is P=PEM</td>
</tr>
<tr>
<td>Q</td>
<td>N by N score matrix of task/dependency exclusion (a numeric matrix). The default Q matrix is Q=1-PEM</td>
</tr>
</tbody>
</table>

Value

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>score</td>
<td>minimal score value of a project (a numeric value).</td>
</tr>
</tbody>
</table>

Author(s)
Zsolt T. Kosztyan*, Aamir Saghir

e-mail: kzst@gtk.uni-pannon.hu
References


See Also

tpc, tpq, tpt, tpr, maxscore_PEM, generatepdm, phase3, percent.

Examples

# Calculation of minimal score value of a project scenarios using MFPP package.

# Define a 3 by 3 upper triangular logic domain of a project scenario.

PEM = matrix(c(0.8, 0.4, 0.8, 0.0, 0.7, 0.7, 0.0, 0.0, 0.4), ncol=3, byrow=TRUE)

# Define a score matrix of task/dependency inclusion.
P = PEM

# Define a score matrix of task/dependency exclusion.
Q = 1 - P

# Calculation of minimal score value for the above specification
# of a project scenario using MFPP package.

minscore_PEM(PEM, P, Q)

---

paretores  Calculate Pareto-optimal resource allocation

Description

Calculate Pareto-optimal resource allocation

Usage

paretores(DSM, TD, RD)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSM</td>
<td>An N by N upper triangular adjacency matrix of logic network with fixed dependencies (a binary matrix).</td>
</tr>
<tr>
<td>TD</td>
<td>N by 1 matrix of task durations (a numeric matrix)</td>
</tr>
<tr>
<td>RD</td>
<td>N by r matrix of task resources, where the number of resources is r.</td>
</tr>
</tbody>
</table>
Value

RD  Pareto-optimal TPR (1 by r numeric vector).
SST  Scheduled Start Time (N by N numeric matrix)).

Author(s)

Zsolt T. Kosztyan*, Aamir Saghir
e-mail: kzst@gtk.uni-pannon.hu

References


See Also
tpr, percent.

Examples

# Calculation of Pareto-optimal resource allocation using MFPP package.
# Define a 3 by 3 upper traingular logic domain of a project structure.

# Specification of Logic Domain
DSM<-matrix(c(1,1,0,1,0,0,
              0,1,0,0,0,0,
              0,0,1,0,0,0,
              0,0,0,1,0,0,
              0,0,0,0,1,0,
              0,0,0,0,0,0),nrow=6, byrow=TRUE)

# Specification of Time Domain
TD<-matrix(c(2,
            3,
            1,
            4,
            0,
            0),nrow=6, byrow=TRUE)

# Specification of Resource Domain
RD<-matrix(c(4,2,
             3,2,
             5,1,
             6,4,
             0,0,
             0,0),nrow=6, byrow=TRUE)

RES<-paretores(DSM,TD,RD)
RES
percent

# Verify results
tpr(RES$SST, DSM, TD, RD)

percent Function to calculate desired project completion characteristic of a project structure.

Description

Function to calculate project completion desired characteristic type either cost, or quality and/or resources with or without quality parameters, or score ratio or time duration of a project scenario for the given ratio and completion modes.

Usage

percent(PDM, type=c("c","q","qd","r","s","t"), w=2, Rs=2, ratio=1)

Arguments

PDM is a Project Domain Matrix (a numeric matrix), where PDM=[LD,TD, CD, QD, RD]. LD is an N by N upper triangular matrix of logic domain (a numeric matrix); TD is an N by w matrix of task durations (a numeric matrix); CD is an N by w matrix of cost demands (a numeric matrix); QD is an optional N by w matrix of quality parameters (a numeric optional matrix); RD is an N by w*R matrix of resource demands (a numeric optional matrix).

type Type to calculate the project completion characteristic either cost or quality with or without defined quality parameters, or resources with or without quality parameters, or minimum completion score or minimum task duration of a project scenario (a logical value).

w Number of completion modes (a numeric integer).

Rs Number of resources (a numeric integer).

ratio ratio of desired characteristic of a project between interval [0,1] (a numeric value).

Value

w Number of completion modes (a numeric integer).

Rs Number of resources (a numeric integer).

ratio ratio of desired characteristic of a project between interval [0,1] (a numeric value).

Cc Project completion cost (a numeric value).

Cq Project completion quality when quality parameters are not considered (a numeric value).
Cq  Project completion quality when quality parameters are considered (a numeric value).
CR  Minimum resources for the completion when quality parameters are not considered (a numeric value).
CR  Minimum resources for the completion when quality parameters are considered (a numeric value).
Cs  Minimum completion score ratio of a project (a numeric value).
Ct  Minimum task duration to project a project (a numeric value).

Author(s)
Zsolt T. Kosztyan*, Aamir Saghir
e-mail: kzst@gtk.uni-pannon.hu

References

See Also
tpc, tpq, tpt, tpr, maxscore_PEM, generatepdm.

Examples
```r
## Examples for verification
w<-2 # Number of completion modes
Rs<-2 # Number of resources
ratio<-1.0 # Percentage 0-1

# Specification of Logic Domain
LD<-matrix(c(1,1,0,1,0,0,
             0,0,0,1,0,0,
             0,0,0,0,1,0,
             0,0,0,0,0,1,
             0,0,0,0,0,0),nrow=6, byrow=TRUE)

# Specification of Time Domain
TD<-matrix(c(1.9755,2.3408,
             2.6564,2.7002,
             0.9018,1.1077,
             1.0237,1.2117,
             0,0),nrow=6, byrow=TRUE)

# Specification of Cost Domain
CD<-matrix(c(2.0768,2.3008,
             0.4819,0.5279,
             3.4315,3.7682,
             3.5902,3.8245,
             0,0),
```
# Specification of Cost Domain
QD<-matrix(c(0.8723,0.9102,  
0.9017,0.8215,  
0.7250,0.9911,  
1.0000,0.8245,  
0,0,  
0,0),nrow=6, byrow=TRUE)

# Specification of Resource Domain
RD<-matrix(c(2.2130,2.4636,1.7948,2.0603,  
1.9709,2.1952,0.8216,0.8890,  
1.6255,1.9432,1.8715,2.2341,  
1.1377,1.3528,1.6768,1.8769,  
0,0,0,0,  
0,0,0,0),nrow=6, byrow=TRUE)

# PDM without QD
PDM<-cbind(LD,TD,CD,RD)

# Percentiles without QD
CONST<-percent(PDM,type=c("c","r","s","t"),w,Rs,ratio)  
CONST

# PDM with QD
PDM<-cbind(LD,TD,CD,QD,RD)

# Percentiles with QD
CONST<-percent(PDM,type=c("c","q","r","s","t"),w,Rs,ratio)  
CONST

---

**phase1**

*Function to simulate estimation uncertainty.*

---

**Description**

Simulating the effects of the estimation uncertainty.

**Usage**

`phase1(x,a=-0.1,b=0.3,pdftype="uniform")`

**Arguments**

- **x** is either a Project Domain Matrix (a numeric matrix), where PDM=[LD,TD,CD,QD,RD]. LD is an N by N upper triangular matrix of logic domain (a numeric matrix); TD is an N by w matrix of task durations (a numeric matrix);
CD is an N by w matrix of cost demands (a numeric matrix); QD is an optional N by w matrix of quality parameters (a numeric optional matrix); RD is an N by w*nR matrix of resource demands (a numeric optional matrix), or PDM list, which contain a PDM matrix, and the number of completion modes (w) and the number of resources.

\( a \)

Optimistic duration, the negative percent deviation from the actual duration (a negative value). The default value is \( a=-0.1 \), which means if the actual duration is 1, the minimal possible duration is 0.9.

\( b \)

Pessimistic duration, the positive percent deviation from the actual duration (a positive value). The default value is \( b=0.3 \), which means if the actual duration is 1, the maximal possible duration is 1.3.

\( \text{pdftype} \)

The pdftype is either 'uniform' or 'beta' (the default value is 'uniform'), which means the generated distribution either follows uniform or a beta distribution between interval \([a,b]\)

**Details**

The changed demands are generated by the interval between \([o+a,o+b]\), where \( o \) is the original value. The random generator can follow uniform (=default), or beta distribution.

**Value**

\( PDMout \)

PDM matrix with same structure as the input PDM matrix (a matrix).

**Author(s)**

Zsolt T. Kosztyan*, Aamir Saghir

e-mail: kzst@gtk.uni-pannon.hu

**References**


**See Also**

\( tpc, tpq, tpt, tpr, maxscore_PEM, phase2, phase3, generatepdm \).

**Examples**

```r
# Simulation of project domain matrix based on customer claims
# of a project scenario using MFPP package.

# Generate a project domain matrix based on
# logic domain, task duration, cost demands,
# quality parameters (optional), resource demand (optional) of a project.

PDM=generatepdm(30,0.05,0,20,30,20,2,2,QD=TRUE,lst=TRUE)
PDM
```
# Define negative percentage deviation from actual duration.
a=-0.20

# Define negative percentage deviation from actual duration.
b=0.40

# Simulate the effects of estimation uncertainty of a project scenario using MFPP package
# with default values.
Res<-phase1(PDM)
summary(Res)

# Simulate the estimation uncertainty of customer claims of a project scenario using MFPP package
# with specified values.
Res<-phase1(PDM$PDM,a,b,"beta")
summary(Res,w=2,Rs=2)

---

**phase2**

*Function to simulate shock effects.*

**Description**

Phase 2 simulates the shock effects. Where p percent of task demands can be increased up to s times.

**Usage**

```
phase2 (x,p=0.10, s=5.0)
```

**Arguments**

- **x**
  - is either a Project Domain Matrix (a numeric matrix), where PDM=[LD,TD,CD,QD, RD], LD is an N by N upper triangular matrix of logic domain (a numeric matrix); TD is an N by w matrix of task durations (a numeric matrix); CD is an N by w matrix of cost demands (a numeric matrix); QD is an optional N by w matrix of quality parameters (a numeric optional matrix); RD is an N by w*nR matrix of resource demands (a numeric optional matrix), or PDM list, which contain a PDM matrix, and the number of completion modes (w) and the number of resources.

- **p**
  - Probability factor for task selection (a numeric value between 0 and 1). The default value is p=1.00

- **s**
  - Scale factor: the ratio of the modification (a numeric integer). The default value is s=5.0

**Value**

- **PDMout**
  - PDM matrix with same structure as the input PDM matrix (a matrix).
Author(s)
Zsolt T. Kosztyan*, Aamir Saghir
e-mail: kzst@gtk.uni-pannon.hu

References

See Also
tpc, tpq, tpt, tpr, maxscore_PEM, phase1, phase3, generatepdm.

Examples

# Simulation of project domain matrix based on customer claims
# of a project scenario using MFPP package.

# Generate a project domain matrix based on
# logic domain, task duration, cost demands,
# quality parameters (optional), resource demand (optional) of a project.

PDM=generatepdm(30,0.05,0,20,30,2,2,2,QD=TRUE,lst=TRUE)
PDM

# Define probability factor for task selection of a project.
p=0.10

# Choose a scale factor for the ration of modification.
s=5.0

# Simulate the shock effect of a project scenario using MFPP package
# with default values.
Res<-phase2(PDM)
summary(Res)

# Simulate the effect of a project scenario using MFPP package
# with specified values.
Res<-phase2(PDM,0.1,p=0.1,s=5.0)
summary(Res,w=2,Rs=2)

phase3

Function to simulate the effects of the change of customer claims.

Description
Phase 3 selects P percent of nodes (i.e. tasks) or arcs (i.e. dependencies), where the score value can be changed up to the maximal change effects.
phase3

Usage

phase3 (x,p=0.10, s=0.50)

Arguments

x is either a Project Domain Matrix (a numeric matrix), where PDM=[LD,TD, CD,QD,RD]. LD is an N by N upper triangular matrix of logic domain (a numeric matrix); TD is an N by w matrix of task durations (a numeric matrix); CD is an N by w matrix of cost demands (a numeric matrix); QD is an optional N by w matrix of quality parameters (a numeric optional matrix); RD is an N by w*nR matrix of resource demands (a numeric optional matrix), or PDM list, which contain a PDM matrix, and the number of completion modes (w) and the number of resources.

p Probability of structural change (default value is p=0.10).

s Increase/decrease ratio of priorities (default value is s=0.50).

Value

PDMout PDM matrix with same structure as the input PDM matrix (a matrix).

Author(s)

Zsolt T. Kosztyan*, Aamir Saghir
e-mail: kzst@gtk.uni-pannon.hu

References


See Also

tpc, tpq, tpt, tpr, phase1, phase2, maxscore_PEM, generatepdm.

Examples

# Simulation of project domain matrix based on customer claims
# of a project scenario using MFPP package.

# Generate a project domain matrix based on
# logic domain, task duration, cost demands,
# quality parameters (optional), resource demand (optional) of a project.

PDM=generatepdm(30,0.05,0,20,30,20,2,2,2,QD=TRUE,lst=TRUE)
PDM

# Define probability factor for task selection and dependency selection of a project.
p=0.10

# Choose a scale factor for the ration of modification.
s=0.10

# Simulate the effects of change of customer claims of a project scenario using MFPP package
# with default values.
Res<-phase3(PDM)
summary(Res)

# Simulate the effects of change of customer claims of a project scenario using MFPP package
# with specified values.
Res<-phase3(PDM$PDM,p,s)
summary(Res,w=2,Rs=2)

---

**plot.mfpp**

*Plot function for Matrix-Based Flexible Project Planning*

**Description**

Plot function for the Package of Matrix-Based Flexible Project Planning (mfpp)

**Usage**

```r
## S3 method for class 'PDM_matrix'
plot(x,w=NULL,Rs=NULL,
type=c("orig","max","min","maximin","minimax","most","const"),
main=NULL,col=NULL,...)

## S3 method for class 'PDM_list'
plot(x,
type=c("orig","max","min","maximin","minimax","most","const"),
main=NULL,col=NULL,...)

## S3 method for class 'Set_PDM_matrix'
plot(x,w=NULL,Rs=NULL,
type=c("orig","max","min","maximin","minimax","most","const"),col=NULL,...)

## S3 method for class 'Set_PDM_list'
plot(x,
type=c("orig","max","min","maximin","minimax","most","const"),col=NULL,...)

## S3 method for class 'TPT'
plot(x,
sched="E",...)
```

**Arguments**

- **x** an object of class 'PDM_matrix' or 'PDM_list'.
- **w** number of completion modes (positive integer)
- **Rs** number of resources (positive integer)
- **type** set of plots, where (orig) means original, (min) minimal, (max) maximal, (maximin) maximin, (minimax) minimax structures are plotted, respectively, and as well as the minimal/maximal constraints (const)
sched  character starts are calculated either "E" to early, or "L" to late, or "S" to scheduled start time
main   title of the plot of fixed structure
col    color of barplot
...    other graphical parameters.

Value
No return value, called for side effects

Author(s)
Zsolt T. Kosztyan*, Aamir Saghir
e-mail: kzst@gtk.uni-pannon.hu

References

See Also
summary, generatepdm, percent.

Examples

# Plot PDM lists.

# Define number of modes, flexibility factor and connectivity factor of a project scenario.
N=4;ff=0.3;cf=0

# Define maximum value of time domain, Cost domain and Resources domain of a project scenario.
mTD=3;mCD=4;mRD=3

# Define number of modes, number of resources, number of possible extra tasks, scale and quality domain of a project scenario.
w=2;nR=2;nW=2
scale=1.6

# Generation of PDM list for TCTP a project scenario using MFPP package.
PDM<-generatepdm(N,ff,cf,mTD,mCD,mRD,w,nR,nW, scale, lst=TRUE)

plot(PDM)
Print PDM constraints, matrices, lists, sets, collections.

### Usage

```r
## S3 method for class 'PDM_const'
summary(object, digits = getOption("digits"), ...)  
## S3 method for class 'PDM_matrix'
summary(object, digits = getOption("digits"), w=getOption("w"), Rs=getOption("Rs"), ...)  
## S3 method for class 'PDM_list'
summary(object, digits = getOption("digits"), ...)  
## S3 method for class 'Set_PDM_matrix'
summary(object, digits = getOption("digits"), w=getOption("w"), Rs=getOption("Rs"), ...)  
## S3 method for class 'Set_PDM_list'
summary(object, digits = getOption("digits"), ...)  
## S3 method for class 'Collection_PDM'
summary(object, digits = getOption("digits"), ...)  
## S3 method for class 'TPT'
summary(object, digits = getOption("digits"), ...)  
```

### Arguments

- `object`: an object of class 'PDM_const', 'PDM_matrix', 'PDM_list', 'Set_PDM_matrix', or 'Set_PDM_list'.
- `digits`: the number of significant digits to use when `add.stats = TRUE`.
- `w`: number of completion modes (positive integer)
- `Rs`: number of resources (positive integer)
- `...`: additional arguments affecting the print produced.

### Value

No return value, called for side effects

### Author(s)

Zsolt T. Kosztyan*, Aamir Saghir  
e-mail: kzst@gtk.uni-pannon.hu

### References

See Also

`generatepdm`, `get.structures`, `percent`, `plot`.

Examples

```r
# Print PDM contraints, matrices, lists, and sets.
# Define number of modes, flexibility factor and connectivity factor of a project scenerio.
N=4;ff=0.05;cf=0

# Define maximum value of time domain, Cost domain and Resources domain of a project scenerio.
mTD=3;mCD=4;mRD=3

# Define number of modes, number of resources, number of possible extra tasks, scale and quality domain of a project scenerio.
w=2;nR=2;nW=2
scale=1.4  # (default value)

# Generation of PDM list for TCTP a project scenario using MFPP package.
PDM<-generatepdm(N,ff,cf,mTD,mCD,mRD,w,nR,nW,scale=1.4,lst=TRUE)

# Generate PDM maximal constraints (ratio=1.0)
CONST<-percent(PDM)

# Calculate main structures
PDMs<-get.structures(PDM)

# print of PDM constraint
summary(CONST)

# print of PDM list
summary(PDM)

# print of main structures of PDM matrices
summary(PDMs)

# print of PDM collection
data(Boctor)
collection<-Boctor
summary(collection)

# schedule table of the first completion mode of the first project structure
# get the first project structure
```
PDM<-collection[[1]]$PDM_list$PDM

# get the logic domain
LD<-PDM[1:nrow(PDM),1:nrow(PDM)]

# get the time demands of the first completion mode
TD<-PDM[1:nrow(PDM),nrow(PDM)+1]

# calculate TPT
TPT<-tpt(LD,TD)

# print schedule table
summary(TPT)

---

tpc  Function of Cost demands of a project.

### Description

Calculate cost demands of a project structure.

### Usage

tpc(DSM, CD)

### Arguments

- **DSM**: N by N Upper triangular binary matrix of logic domain (a numeric matrix).
- **CD**: N by 1 vector of cost demands (a numeric vector).

### Value

- **TPC**: Total Project Cost (a scalar).

### Author(s)

Zsolt T. Kosztyan*, Aamir Saghir

e-mail: kzst@gtk.uni-pannon.hu

### References

See Also

tpc, tpt, tpq, tpr, maxscore_PEM.

Examples

# Calculation of TPC for a generated project scenario.
# using MFPP package.

# Generation of 10 by 10 Upper triangular binary matrix (DSM) of logic domain.

library(Matrix)
library(pracma)

DSM<-triu(round(rand(10)*.5+.5)) # Generate DSM

# Generation of 10 by 1 cost vector (cost demands 30)
C<-rand(10,1)*30 # Generate C vector (cost demands)

# Calculation of total project cost using MFPP package.
TPC<-tpc(DSM,C) # Calculate TPC

## tpq

Function to calculate Total Project Quality for a project structure.

Description

Calculate Total Project Quality of a project structure.

Usage

tpq (DSM,PEM,q,QD=NULL)

Arguments

DSM N by N Upper triangular binary matrix of logic domain (a project structure of a PEM matrix) (a numeric matrix).
PEM N by N Upper triangular binary matrix of logic domain (a numeric matrix).
q N by 1 vector of quality parameters (a numeric vector).
QD N by w Quality Domain to calculate relative TPQ. The default value is QD=NULL

Value

TPQ Total Project Quality (a scalar).
Author(s)

Zsolt T. Kosztlyan, Aamir Saghir*
e-mail: kzst@gtk.uni-pannon.hu

References


See Also

tpc, tpr, tpt, maxscore_PEM.

Examples

# Calculation of total project quality of a project using MFPP package.
library(pracma)

# Generation of a 10 by 10 upper triangular binary matrix (PEM) of logic domain of a project.
PEM <- triu(rand(10)*0.5+0.5)

# Generation of a 10 by 10 Upper triangular binary matrix of logic domain
# (a project structure of a PEM matrix).
DSM <- round(PEM)

# Generation of a 10 by 1 vector of quality parameters of a project.
q <- rand(10,1)

# For fixed project plan the TPQ is the geometric mean of quality parameters
tpq(DSM, DSM, q)

# For flexibility project plan the TPQ
# is the weighted geometric mean of
# quality parameters

tpq(DSM, PEM, q)

QD2 <- cbind(q, rand(10,1)) # Generate two completion modes

# Calculate relative quality considering the best choice
tpq(DSM, PEM, q, QD2)
Function to calculate maximum resource demands of a project.

Description

Calculate maximum resource demands of a project scenario.

Usage

tpr (SST, DSM, TD, RD, res.graph=FALSE)

Arguments

SST  N by 1 vector of Scheduled Start Time (a numeric vector).
DSM  N by N binary upper triangular matrix of the logic domain (a numeric matrix).
TD   N by 1 vector of task durations (a numeric vector).
RD   N by nR matrix of resource demands (a numeric matrix).
res.graph  Logic value of whether plot or not the resource graph

Value

rMAX  an nR by 1 vector of maximum resource demands (a scalar vector).

Author(s)

Zsolt T. Kosztyan*, Aamir Saghir
e-mail: kzst@gtk.uni-pannon.hu

References


See Also

tpc, tpq, tpt, maxscore_PEM.

Examples

# Calculation of maximum resource demands of a project scenario using MFPP package.
library(pracma)

# Generate a 10 by 10 upper triangular binary matrix (DSM) of logic domain of a project.
DSM <- triu(round(rand(10)))
# Generate a 10 by 1 vector of task durations of a project.
TD <- rand(10,1)*20

# Define a 10 by 1 vector of Scheduled Start Time using tpt function.
SST <- tpt(DSM,TD)["EST"]

# Generate a 10 by 3 matrix of resource demands of a project.
RD <- rand(10,3)*5

# Calculation of maximum resource demands of a project using MFPP package.
tpr(SST,DSM,TD, RD)

## Plot resources for SST
tpr(SST,DSM,TD,RD,res.graph = TRUE)

tpt

Function to evaluate EST, EFT, LST and LFT times of activity of a project.

Description

Calculate EST, EFT, LST and LFT times of activity of a project.

Usage

tpt(DSM,TD,SST=NULL)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSM</td>
<td>N by N Upper triangular binary matrix of logic domain (a numeric matrix).</td>
</tr>
<tr>
<td>TD</td>
<td>N by 1 vector of time duration (a numeric vector).</td>
</tr>
<tr>
<td>SST</td>
<td>N by 1 vector of scheduled start time (a numeric vector).</td>
</tr>
</tbody>
</table>

Value

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TPT</td>
<td>Total Project Time (a scalar).</td>
</tr>
<tr>
<td>EST</td>
<td>Early Start Time (a vector).</td>
</tr>
<tr>
<td>EFT</td>
<td>Early Finish Time (a vector).</td>
</tr>
<tr>
<td>LST</td>
<td>Latest Start Time (a vector).</td>
</tr>
<tr>
<td>LFT</td>
<td>Latest Finish Time (a vector).</td>
</tr>
<tr>
<td>SST</td>
<td>Scheduled Start Time (a vector).</td>
</tr>
<tr>
<td>SFT</td>
<td>Scheduled Finish Time (a vector).</td>
</tr>
</tbody>
</table>

Author(s)

Zsolt T. Kosztyan*, Aamir Saghir
e-mail: kzst@gtk.uni-pannon.hu
**truncpdm**

Function to drop excluded tasks.

**Description**

Drop excluded tasks, where the diagonal value is 0.

**Usage**

truncpdm(x)
truncpdm

Arguments

- `x`: N by M PDM matrix or a PDM list.

Value

- `PDM`: N by M PDM matrix or a PDM list

Author(s)

Zsolt T. Kosztyan*, Aamir Saghir

e-mail: kzst@gtk.uni-pannon.hu

References


See Also

- `generatepdm`

Examples

```r
# Generation of PDM matrix for flexible project planning MFPP package.

# Define number of modes, flexibility factor and connectivity factor of a project scenario.
N=4;ff=0.05;cf=0

# Define maximum value of time domain, Cost domain and Resources domain of a project scenario.
mTD=3;mCD=4;mRD=3

# Define number of modes, number of resources, number of possible extra tasks, scale and quality domain of a project scenario.
w=2;nR=2;nW=2
scale=1.4  #(default value)

# Generation of PDM matrix for TCTP a project scenario using MFPP package.
PDM<-generatepdm(N,ff,cf,mTD,mCD,mRD,w,nR,nW,scale=1.4)

# Drop excluded tasks.
PDM<-truncpdm(PDM)

# Generation of PDM list for TCTP a project scenario using MFPP package.
PDM<-generatepdm(N,ff,cf,mTD,mCD,mRD,w,nR,nW,scale=1.4,QD=FALSE,lst=TRUE)

# Drop excluded tasks.
PDM<-truncpdm(PDM)
```
Index

* datasets
  Batselier, 3
  Boctor, 3
* package
  mfpp-package, 2
* project management
  plot.mfpp, 20
  summary.mfpp, 22
* project planning
  generatepdm, 4
  get.structures, 6
  is.flexible, 7
  maxscore_PEM, 9
  minscore_PEM, 10
  paretores, 11
  percent, 13
  phase1, 15
  phase2, 17
  phase3, 18
  tpc, 24
  tpq, 25
  tpr, 27
  tpt, 28
  truncpdm, 29

Batselier, 3
Boctor, 3

generatepdm, 2, 4, 5, 7, 8, 11, 14, 16, 18, 19, 21, 23
get.structures, 2, 6, 23

is.flexible, 2, 7

maxscore_PEM, 5, 9, 11, 14, 16, 18, 19, 25–27, 29
mfpp (mfpp-package), 2
mfpp-package, 2
minscore_PEM, 10

paretores, 11

percent, 2, 11, 12, 13, 21, 23
phase1, 2, 15, 18, 19
phase2, 2, 16, 17, 19
phase3, 2, 11, 16, 18, 18
plot, 23
plot.mfpp, 20
plot.PDM_list (plot.mfpp), 20
plot.PDM_matrix, 2
plot.PDM_matrix (plot.mfpp), 20
plot.Set_PDM_list (plot.mfpp), 20
plot.Set_PDM_matrix (plot.mfpp), 20
plot.TPT (plot.mfpp), 20

summary, 21
summary.Collection_PDM (summary.mfpp), 22
summary.mfpp, 22
summary.PDM_const (summary.mfpp), 22
summary.PDM_list (summary.mfpp), 22
summary.PDM_matrix, 2
summary.PDM_matrix (summary.mfpp), 22
summary.Set_PDM_list (summary.mfpp), 22
summary.Set_PDM_matrix (summary.mfpp), 22
summary.TPT (summary.mfpp), 22

tpc, 5, 9, 11, 14, 16, 18, 19, 24, 25–27, 29
tpq, 5, 9, 11, 14, 16, 18, 19, 25, 25, 27, 29
tpr, 5, 9, 11, 12, 14, 16, 18, 19, 25, 26, 27, 29
tpt, 5, 9, 11, 14, 16, 18, 19, 25–27, 28
truncpdm, 29

31