Package: MOODE (via r-universe)

January 9, 2025

Type Package

Title Multi-Objective Optimal Design of Experiments

Version 1.0.1.9000

Description Provides functionality to generate compound optimal designs for targeting the multiple experimental objectives directly, ensuring that the full set of research questions is answered as economically as possible. Designs can be found using point or coordinate exchange algorithms combining estimation, inference and lack-of-fit criteria that account for model inadequacy. Details and examples are given by Koutra et al. (2024) <doi:10.48550/arXiv.2412.17158>.

License GPL (>= 3)

URL https://github.com/vkstats/MOODE

BugReports https://github.com/vkstats/MOODE/issues

Imports cli, far, progressr, Rdpack, rlang

Suggests doFuture, foreach

RdMacros Rdpack

Encoding UTF-8

NeedsCompilation no

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Repository https://vkstats.r-universe.dev

RemoteUrl https://github.com/vkstats/moode

RemoteRef HEAD

RemoteSha 292f8957ef124c00ae88c7bbb5dbc3f315e5d446

Contents

`candidate_set_full` *Forms the full candidate set of treatments for all polynomial terms*

Description

This function forms the full extended candidate set with all polynomial terms, with labels, not orthonormalised.

Usage

```
candidate_set_full(cand.trt, K)
```

Arguments

`cand.trt`	Candidate set of treatments, the first column contains treatment labels. Usually obtained as output from the candidate_trt_set function.
`K`	Number of factors.

Value

The full extended candidate set: the column of treatment labels, then named columns with polynomial terms up to the 4th order. For example, "x12" stands for x_1^2, and "x1x2 stands for $x_1 x_2$, and "x23x4" for $x_2^3 x_4$.

Examples

```
# Full extended candidate set for two 3-level factors

K<-2; Levels <- rep(list(1:3),K);
candidate_set_full(candidate_trt_set(Levels, K), K)
```

`candidate_set_orth` *Forms the orthonormalised full candidate set*

Description

This function forms the full extended orthonormalised candidate set of primary and potential terms, with an intercept column and labels.

Usage

```
candidate_set_orth(cand.full, primary.terms, potential.terms)
```

Arguments

`cand.full`	Candidate set containing terms up to 4th order, with labels in the first column.
`primary.terms`	Character vector identifying primary model terms.
`potential.terms`	Character vector identifying potential model terms.

Value

The orthonormalised full candidate set containing primary and potential terms, with labels.

Examples

```
# Full extended orthonormalised candidate set for two 4-level factors,
# full quadratic polynomial model as primary model and all three-order terms as potential.

K<-2; Levels <- rep(list(1:4),K)
cand.trt <- candidate_trt_set(Levels, K)
cand.full <- candidate_set_full(cand.trt, K)
prime.terms <- colnames(cand.full)[2:7]
poten.terms <- colnames(cand.full)[8:11]
Parameters <- c(1, rep(1,K), rep(1,K), K*(K-1)/2)
candidate_set_orth(cand.full, prime.terms, poten.terms)
```

candidate_trt_set *Forms the labelled candidate set of treatments*

Description

This function forms the candidate set of treatments from the factors' levels, adds labels, with optional spherical transformation of the coordinates.

Usage

```
candidate_trt_set(Levels, K, Hypercube = TRUE)
```

Arguments

Levels	Levels of each factor.
K	Number of factors.
Hypercube	Indicates if the experimental region is a hypercube ('TRUE') or spherical ('FALSE').

Value

Matrix of candidate set of treatments, with treatment labels contained in the first column.

Examples

```
# Candidate treatment set for five 3-level factors

K<-5; Levels <- rep(list(1:3),K);
candidate_trt_set(Levels, K)
```

criteria.GD *Calculates the values of the Generalised Ds-criterion and its components*

Description

This function evaluates the Generalised Ds-criterion (Goos et al. 2005) for given primary and potential model matrices.

Usage

```
criteria.GD(X1, X2, search.object, eps = 1e-23)
```

Arguments

X1	The primary model matrix, with the first column containing the labels of treatments, and the second – the intercept term.
X2	The matrix of potential terms, with the first column containing the labels of treatments.
search.object	Object of class mood() specifying experiment parameters.
eps	Computational tolerance, the default value is 10^-23

Value

A list of values: indicator of whether the evaluation was successful ("eval"), Ds-criterion value – intercept excluded ("Ds"), Lack-of-fit criterion value ("LoF"), the bias component value ("bias"), the number of pure error degrees of freedom ("df") and the value of the compound criterion ("compound").

References

Goos P, Kobilinsky A, O'Brien TE, Vandebroek M (2005). "Model-Robust and Model-Sensitive Designs." *Computational Statistics and Data Analysis*, **49**, 201-216.

Examples

```
#Experiment: one 5-level factor, primary model -- full quadratic, one potential (cubic) term
# setting up the example
ex.mood <- mood(K = 1, Levels = 5, Nruns = 7, criterion.choice = "GDP",
               kappa = list(kappa.Ds = 1./3, kappa.LoF = 1./3, kappa.bias = 1./3),
          model_terms = list(primary.model = "second_order", potential.model = "cubic_terms"))
# Generating candidate set: orthonormalised
K <- ex.mood$K
Levels <- ex.mood$Levels
cand.not.orth <- candidate_set_full(candidate_trt_set(Levels, K), K)
cand.full.orth <- candidate_set_orth(cand.not.orth, ex.mood$primary.terms, ex.mood$potential.terms)
# Choosing a design
index <- c(rep(1, 2), 3, 4, rep(5, 3))
X.primary <- cand.full.orth[index, c(1, match(ex.mood$primary.terms, colnames(cand.full.orth)))]
X.potential <- cand.full.orth[index,
(c(1, match(ex.mood$potential.terms, colnames(cand.full.orth))))]
# Evaluating a compound GD-criterion
criteria.GD(X1 = X.primary, X2 = X.potential, ex.mood)
# Output: eval = 1, Ds = 0.7334291, LoF = 0.7212544, bias = 1.473138, df = 3, compound = 0.9202307
```

---

criteria.GDP *Calculates the values of the Generalised DPs-criterion and its components*

Description

This function evaluates the Generalised DPs-criterion for given primary and potential model matrices. Components: Ds-, DPs-, LoF(DP)- and Bias(D)-optimality.

Usage

```
criteria.GDP(X1, X2, search.object, eps = 10^-23)
```

Arguments

X1	The primary model matrix, with the first column containing the labels of treatments, and the second – the intercept term.
X2	The matrix of potential terms, with the first column containing the labels of treatments.
search.object	Object of class mood() specifying experiment parameters.
eps	Computational tolerance, the default value is 10^-23

Value

A list of values: indicator of whether the evaluation was successful ("eval"), Ds-criterion value – intercept excluded ("Ds"), DPs-criterion value – intercept excluded ("DPs"), Lack-of-fit(DP) criterion value ("LoF"), the bias component value ("bias"), the number of pure error degrees of freedom ("df") and the value of the compound criterion ("compound").

Examples

```
# Experiment: one 5-level factor, primary model -- full quadratic, X^3 and X^4 potential terms.
ex.mood <- mood(K = 1, Levels = 5, Nruns = 8, criterion.choice = "GDP",
           kappa = list(kappa.Ds = .25, kappa.LoF = .25, kappa.bias = .25, kappa.DP = .25),
             model_terms = list(primary.model = "second_order", potential.terms = "x14"))
# Generating candidate sets: primary and full orthonormalised ones
K <- ex.mood$K
Levels <- ex.mood$Levels
cand.not.orth <- candidate_set_full(candidate_trt_set(Levels, K), K)
cand.full.orth <- candidate_set_orth(cand.not.orth, ex.mood$primary.terms, ex.mood$potential.terms)
# Choosing a design
index <- c(rep(1,2),3,rep(4,2),rep(5,3))
X.primary <- cand.full.orth[index, c(1, match(ex.mood$primary.terms, colnames(cand.full.orth)))]
X.potential <- cand.full.orth[index,
(c(1, match(ex.mood$potential.terms, colnames(cand.full.orth))))]
# Evaluating a compound GDP-criterion
criteria.GDP(X1 = X.primary, X2 = X.potential, ex.mood)

# Output: eval = 1, Ds = 0.6884783, DP = 4.4538023, LoF = 3.895182,
# bias = 1.03807, df = 4, compound = 2.465318
```

---

criteria.GL *Calculates the values of the Generalised L-criterion and its components*

Description

This function evaluates the Generalised L-criterion (Goos et al. 2005) for given primary and potential model matrices.

Usage

```
criteria.GL(X1, X2, search.object, eps = 10^-23)
```

Arguments

X1	The primary model matrix, with the first column containing the labels of treatments, and the second – the intercept term.
X2	The matrix of potential terms, with the first column containing the labels of treatments.
search.object	Object of class mood() specifying experiment parameters.
eps	Computational tolerance, the default value is 10^-23

Value

A list of values: indicator of whether the evaluation was successful ("eval"), Ls-criterion value – intercept excluded ("Ls"), Lack-of-fit criterion value ("LoF"), the bias component value ("bias"), the number of pure error degrees of freedom ("df") and the value of the compound criterion ("compound").

References

Goos P, Kobilinsky A, O'Brien TE, Vandebroek M (2005). "Model-Robust and Model-Sensitive Designs." *Computational Statistics and Data Analysis*, **49**, 201-216.

Examples

```
#Experiment: one 5-level factor, primary model -- full quadratic, one potential (cubic) term
# setting up the example
ex.mood <- mood(K = 1, Levels = 5, Nruns = 7, criterion.choice = "GL",
               kappa = list(kappa.L = 1./3, kappa.LoF = 1./3, kappa.bias = 1./3),
          model_terms = list(primary.model = "second_order", potential.model = "cubic_terms"))
# Generating candidate set: orthonormalised
K <- ex.mood$K
Levels <- ex.mood$Levels
cand.not.orth <- candidate_set_full(candidate_trt_set(Levels, K), K)
cand.full.orth <- candidate_set_orth(cand.not.orth, ex.mood$primary.terms, ex.mood$potential.terms)
# Choosing a design
index <- c(rep(1, 2), 3, 4, rep(5, 3))
X.primary <- cand.full.orth[index, c(1, match(ex.mood$primary.terms, colnames(cand.full.orth)))]
X.potential <- cand.full.orth[index,
(c(1, match(ex.mood$potential.terms, colnames(cand.full.orth))))]
# Evaluating a compound GD-criterion
criteria.GL(X1 = X.primary, X2 = X.potential, ex.mood)
# Output: eval = 1, L = 0.3118626, LoF = 0.7212544, bias = 1.473138, df = 3, compound = 0.6919878
```

`criteria.GLP` *Calculates the values of the Generalised LP-criterion and its components*

Description

This function evaluates the Generalised LP-criterion for given primary and potential model matrices. Components: L-, LP-, LoF(LP)- and Bias(L)-optimality.

Usage

```
criteria.GLP(X1, X2, search.object, eps = 1e-23)
```

Arguments

`X1`	The primary model matrix, with the first column containing the labels of treatments, and the second – the intercept term.
`X2`	The matrix of potential terms, with the first column containing the labels of treatments.
`search.object`	Object of class `mood()` specifying experiment parameters.
`eps`	Computational tolerance, the default value is 10^-23

Value

A list of values: indicator of whether the evaluation was successful ("eval"), Ls-criterion value – intercept excluded ("L"), LP-criterion value – intercept excluded ("LP"), Lack-of-fit(LP) criterion value ("LoF"), the bias component value ("bias"), the number of pure error degrees of freedom ("df") and the value of the compound criterion ("compound").

Examples

```
#' # Experiment: one 5-level factor, primary model -- full quadratic, X^3 and X^4 potential terms.
ex.mood <- mood(K = 1, Levels = 5, Nruns = 8, criterion.choice = "GLP",
            kappa = list(kappa.L = .25, kappa.LoF = .25, kappa.bias = .25, kappa.LP = .25),
             model_terms = list(primary.model = "second_order", potential.terms = "x14"))
# Generating candidate sets: primary and full orthonormalised ones
K <- ex.mood$K
Levels <- ex.mood$Levels
cand.not.orth <- candidate_set_full(candidate_trt_set(Levels, K), K)
cand.full.orth <- candidate_set_orth(cand.not.orth, ex.mood$primary.terms, ex.mood$potential.terms)
# Choosing a design
index <- c(rep(1,2),3,rep(4,2),rep(5,3))
X.primary <- cand.full.orth[index, c(1, match(ex.mood$primary.terms, colnames(cand.full.orth)))]
X.potential <- cand.full.orth[index,
(c(1, match(ex.mood$potential.terms, colnames(cand.full.orth))))]
# Evaluating a compound GDP-criterion
criteria.GLP(X1 = X.primary, X2 = X.potential, ex.mood)
# Output: eval = 1, L = 0.2952603, LP = 4.584705, LoF = 3.895182,
```

```
# bias = 1.03807, df = 4, compound = 1.529564
```

`criteria.mseD` *Calculates the values of the MSE DPs-criterion and its components*

Description

This function evaluates the MSE DPs-criterion for given primary and potential model matrices. Candidate full model matrices do not have to be orthonormalised. Components: DPs-, LoF(DP)- and MSE(D)-optimality.

Usage

```
criteria.mseD(X1, X2, search.object, eps = 1e-23)
```

Arguments

`X1`	The primary model matrix, with the first column containing the labels of treatments, and the second – the intercept term.
`X2`	The matrix of potential terms, with the first column containing the labels of treatments.
`search.object`	Object of class `mood()` specifying experiment parameters.
`eps`	Computational tolerance, the default value is 10^-23

Value

A list of values: indicator of whether the evaluation was successful ("eval"), DPs-criterion value – intercept excluded ("DP"), Lack-of-fit(DP) criterion value ("LoF"), the MSE(D) component value ("mse"), the number of pure error degrees of freedom ("df") and the value of the compound criterion ("compound").

Examples

```
# Experiment: one 5-level factor, primary model -- full quadratic, X^3 and X^4 potential terms.
set.seed(20210930)
ex.mood <- mood(K = 1, Levels = 5, Nruns = 8, criterion.choice = "MSE.D",
                kappa = list(kappa.DP = 1./3, kappa.LoF = 1./3, kappa.mse = 1./3),
                control = list(Biter = 1000),
              model_terms = list(primary.model = "second_order", potential.terms = "x14"))
# Generating candidate sets: primary and full orthonormalised ones
K <- ex.mood$K
Levels <- ex.mood$Levels
cand.not.orth <- candidate_set_full(candidate_trt_set(Levels, K), K)
cand.full.orth <- candidate_set_orth(cand.not.orth, ex.mood$primary.terms, ex.mood$potential.terms)
# Choosing a design
index <- c(rep(1,2),3,rep(4,2),rep(5,3))
X.primary <- cand.full.orth[index, c(1, match(ex.mood$primary.terms, colnames(cand.full.orth)))]
```

```
X.potential <- cand.full.orth[index,
(c(1, match(ex.mood$potential.terms, colnames(cand.full.orth))))]
# Evaluating a compound GDP-criterion
criteria.mseD(X.primary, X.potential, ex.mood)
# Output: eval = 1, DP = 4.538023, LoF = 3.895182, mse = 0.6986903, df = 4, compound = 2.310728
```

`criteria.mseL` *Calculates the values of the MSE LP-criterion and its components*

Description

This function evaluates the MSE LP-criterion for given primary and potential model matrices. Candidate full model matrices do not have to be orthonormalised. Components: LP-, LoF(LP)- and MSE(L)-optimality.

Usage

```
criteria.mseL(X1, X2, search.object, eps = 10^-23)
```

Arguments

`X1`	The primary model matrix, with the first column containing the labels of treatments, and the second – the intercept term.
`X2`	The matrix of potential terms, with the first column containing the labels of treatments.
`search.object`	Object of class `mood()` specifying experiment parameters.
`eps`	Computational tolerance, the default value is 10^-23

Value

A list of values: indicator of whether the evaluation was successful ("eval"), LP-criterion value – intercept excluded ("LP"), Lack-of-fit(LP) criterion value ("LoF"), the MSE(L) component value ("mse"), the number of pure error degrees of freedom ("df") and the value of the compound criterion ("compound").

Examples

```
#'# Experiment: one 5-level factor, primary model -- full quadratic, X^3 and X^4 potential terms.
ex.mood <- mood(K = 1, Levels = 5, Nruns = 8, criterion.choice = "MSE.L",
              kappa = list(kappa.LP = 1./3, kappa.LoF = 1./3, kappa.mse = 1./3),
            model_terms = list(primary.model = "second_order", potential.terms = "x14"))
# Generating candidate sets: primary and full orthonormalised ones
K <- ex.mood$K
Levels <- ex.mood$Levels
cand.not.orth <- candidate_set_full(candidate_trt_set(Levels, K), K)
cand.full.orth <- candidate_set_orth(cand.not.orth, ex.mood$primary.terms, ex.mood$potential.terms)
# Choosing a design
index <- c(rep(1,2),3,rep(4,2),rep(5,3))
```

```
X.primary <- cand.full.orth[index, c(1, match(ex.mood$primary.terms, colnames(cand.full.orth)))]
X.potential <- cand.full.orth[index,
(c(1, match(ex.mood$potential.terms, colnames(cand.full.orth))))]
# Evaluating a compound GDP-criterion
criteria.mseL(X.primary, X.potential, ex.mood)
# Output: eval = 1, LP = 4.584705, LoF = 3.895182, mse = 0.3926842, df = 4, compound = 1.914084
```

`criteria.mseP` *Calculates the values of the MSE DPs-criterion using the point prior for the MSE(D)-component estimation*

Description

This function evaluates the MSE DPs-criterion for given primary and potential model matrices, using point MSE(D)-component estimation. Candidate full model matrices do not have to be orthonormalised. Components: DPs-, LoF(DP)- and MSE(D)-optimality.

Usage

```
criteria.mseP(X1, X2, search.object, eps = 10^-23)
```

Arguments

`X1`	The primary model matrix, with the first column containing the labels of treatments, and the second – the intercept term.
`X2`	The matrix of potential terms, with the first column containing the labels of treatments.
`search.object`	Object of class `mood()` specifying experiment parameters.
`eps`	Computational tolerance, the default value is 10^-23

Value

A list of values: indicator of whether the evaluation was successful ("eval"), DPs-criterion value – intercept excluded ("DP"), Lack-of-fit(DP) criterion value ("LoF"), the MSE(D) component value ("mse"), the number of pure error degrees of freedom ("df") and the value of the compound criterion ("compound").

Examples

```
# Experiment: one 5-level factor, primary model -- full quadratic, X^3 and X^4 potential terms.
ex.mood <- mood(K = 1, Levels = 5, Nruns = 8, criterion.choice = "MSE.P",
                kappa = list(kappa.DP = 1./3, kappa.LoF = 1./3, kappa.mse = 1./3),
              model_terms = list(primary.model = "second_order", potential.terms = "x14"))
# Generating candidate sets: primary and full orthonormalised
K <- ex.mood$K
Levels <- ex.mood$Levels
cand.not.orth <- candidate_set_full(candidate_trt_set(Levels, K), K)
cand.full.orth <- candidate_set_orth(cand.not.orth, ex.mood$primary.terms, ex.mood$potential.terms)
```

```
# Choosing a design
index <- c(rep(1,2),3,rep(4,2),rep(5,3))
X.primary <- cand.full.orth[index, c(1, match(ex.mood$primary.terms, colnames(cand.full.orth)))]
X.potential <- cand.full.orth[index,
(c(1, match(ex.mood$potential.terms, colnames(cand.full.orth))))]
# Evaluating a compound GDP-criterion
criteria.mseP(X.primary, X.potential, ex.mood)
# Output: eval = 1, DP = 4.538023, LoF = 3.895182, mse = 0.6992699, df = 4, compound = 2.312135
```

`criteria.values.G` *Evaluating individual criteria of the designs, from the Generalized compound criteria (Goos et al. 2005; Egorova,~O. 2017)*

Description

Calculating values of determinant- and trace-based components of Generalized D-, DP-, L- and LP-criteria for an output of a search object, with model and control parameters set in a mood object.

Usage

```
criteria.values.G(search.obj, mood.obj, eps = 10^-23)
```

Arguments

`search.obj`	Output of the 'Search' function
`mood.obj`	Output of the 'mood' function
`eps`	Computational tolerance, default 10^-20

Value

List of the calculated values:

- `df` pure error degrees of freedom
- `Ds` Ds-criterion value, intercept excluded
- `DP` DPs-criterion value, intercept excluded
- `LoFD` LoF(D)-criterion value from the GD-criterion
- `LoFDP` LoF(DP)-criterion value from the GDP-criterion
- `biasD` bias(D)-criterion value from the GD-criterion
- `Ls` Ls-criterion value, intercept excluded
- `LP` LPs-criterion value, intercept excluded
- `LoFL` LoF(L)-criterion value from the GL-criterion
- `LoFLP` LoF(LP)-criterion value from the GLP-criterion
- `biasL` bias(L)-criterion value from the GL-criterion

References

Egorova,~O. (2017). *Optimal Design of Experiments for Multiple Objectives*. Ph.D. thesis, University of Southampton.

Goos P, Kobilinsky A, O'Brien TE, Vandebroek M (2005). "Model-Robust and Model-Sensitive Designs." *Computational Statistics and Data Analysis*, **49**, 201-216.

`criteria.values.mse` *Evaluating individual criteria of the designs*

Description

Calculating values of determinant- and trace-based components of MSE(D)- and MSE(L)- criteria for an output of a search object, with model and control parameters set in a mood object.

Usage

```
criteria.values.mse(search.obj, mood.obj, eps = 10^-20, Biter = 1000)
```

Arguments

`search.obj`	Output of the 'Search' function
`mood.obj`	Output of the 'mood' function
`eps`	Computational tolerance, default 10^-20
`Biter`	MC sample size for evaluating the mse(D)-component

Value

List of the calculated values:

- `df` pure error degrees of freedom
- `Ds` Ds-criterion value, intercept excluded
- `DP` DPs-criterion value, intercept excluded
- `LoFDP` LoF(DP)-criterion value
- `mseD` mse(D)-criterion value, obtained via MC sampling
- `mseP` mse(D)-criterion value, obtained using point prior
- `L` L-criterion value, intercept excluded
- `LP` LP-criterion value, intercept excluded
- `LoFLP` LoF(LP)-criterion value
- `mseL` mse(L)-criterion value

mood *Setting up the parameters of a factorial experiment to search for multi-objective optimal completely randomised design.*

Description

Creates an object containing the parameters of the experiment, compound optimality criterion with the weights and parameters of the search.

Usage

```
mood(
  K,
  Levels,
  Nruns,
  criterion.choice = c("GD", "GL", "GDP", "GLP", "MSE.D", "MSE.L", "MSE.P"),
  kappa = list(),
  control = list(),
  prob = list(),
  model_terms = list(primary.model = "first_order")
)
```

Arguments

K	Number of factors.
Levels	Either (a) a common number of levels for each factor or (b) a list of length K of the vectors containing levels of each factor.
Nruns	Number of runs of the experiment.
criterion.choice	Compound criterion to be used for the optimal design search or evaluation. Possible values are: • GL, GD for Generalised D- and L-optimality (Goos et al. 2005) • GDP and GLP for Generalised DP- and LP-optimality (Gilmour and Trinca 2012) • MSE.D, MSE.L and MSE.P for compound criteria with MSE-component: determinant-based, trace-based and determinant-based but with point estimates for the MSE(D)-component
kappa	List specifying the weights used in the compound criterion. Each named entry must be between 0 and 1. • kappa.Ds Weight of the Ds-criterion (default = 1 if criterion.choice = GD) • kappa.DP Weight of the DP-criterion (default = 1 if criterion.choice = GDP) • kappa.L Weight of the L-criterion (default = 1 if criterion.choice = GL)

- kappa.LP Weight of the LP-criterion (default = 1 if criterion.choice = GLP)
- kappa.LoF Weight of the lack-of-fit criterion
- kappa.bias Weight of the bias criterion
- kappa.mse Weight of the MSE criterion (default = 1 if 'criterion.choice = MSE.*')

control — Named list specifying control parameters for the design search.

- Nstarts The number of randomly generated start designs of the search algorithm (default = 10).
- Cubic Indicator of whether the experimental region is cubic (TRUE, default) or spherical (FALSE).
- tau2 The variance scaling parameter for the prior distribution of the potential terms (default = 1).
- Biter Number of samples for evaluating the MSE determinant-based component criterion (default = 50).
- MC Indicator of whether to apply a multiple comparison (Bonferroni) correction for trace-based criteria (TRUE, default) or not (FALSE).
- orth Indicator of whether to orthonormalise the potential and primary terms (TRUE', default) or not (FALSE').

prob — Named list specifying confidence levels for DP- (prob.DP), LP- (prob.LP) and Lack-of-fit (prob.LoF) criteria (pre-Bonferroni correction). All default to 0.95.

model_terms — A list specifying the primary (fitted) and potential (biased) models with the following named elements (see Details).

- primary.model The order of the fitted polynomial model. Alternatively polynomial terms can be directly specified through the primary.terms parameter.
- potential.model The order of the potential/extra polynomial terms. Alternatively can be specified through the potential.terms parameter.
- primary.terms Alternative specification of the primary model via character vector of the names of the primary terms.
- potential.terms Alternative specification of the potential model via character vector of the names of the potential terms.

Details

The function provides different ways of specifying the levels of the factors and the models. Although some default options are provided for, e.g., criterion.choice and kappa.* values, specification of these input parameters should be carefully chosen to reflect the aims of the experiment and available prior information.

Specifying the factors and levels

If all K factors have the same number of levels, Levels parameter is used to input that number. Otherwise, Levels is set to be a list of vectors containing the values of the factors, e.g. list(1:3, 1:2, 1:4) for 3 factors with equally spaced $3, 2$ and 4 levels respectively.

Specifying the fitted model and the potential terms

There are two ways to describe the primary and potential sets of model terms via the `model_terms` list. Named elements `primary.model` and `potential.model` can be used to specify the fitted model and the potential terms via a string form. They are used to generate the sets of `primary.terms` and `potential.terms` which alternatively can be input directly. Possible values of `primary.model` are:

- `main_effects` – main effects for all the factors (default for all criteria)
- `first_order` – main effects and linear interactions
- `second_order` – full second order polynomial
- `third_order` – full second order polynomial model and all interactions of degree 3
- `cubic` – third order polynomial model with cubic terms

The intercept is always included as a primary term.

Possible elements of the `potential.model` vector argument:

- `linear_interactions` – linear interactions among the factors (default for MSE criteria)
- `quadratic_terms` – quadratic terms for all the factors
- `third_order_terms` – all interactions of degree 3: linear-by-linear-by-linear and quadratic-by-linear terms
- `cubic_terms` – cubic terms for all the factors
- `fourth_order_terms` – all interactions of degree 4, similar to `third_order_terms`

`primary.terms` and `potential.terms` arguments are used to specify the fitted model and the potential terms explicitly – up to the total power of 4.

- Single factor powers, are coded as a string starting with with "x" and followed by the index of the factor and the power: `"x32"`. For example, x_3^2 is coded as `"x32"`; `"x22"` stands for x_2^2, and `"x4"` stands for the linear term x_4.
- Interaction of factors' powers are coded by merging the individual factors' powers, ordered by the factors' indexes. For example, $x_2^2 \times x_1$ is coded as `"x1x22"`, $x_3 x_1 2 x_4$ – as `"x12x3x4"`.

Value

List of parameters of the experiment, compound criterion of choice, and primary and potential model terms.

- `K` Number of factors.
- `Klev` Number of levels of each factor, if all factors have the same number of levels.
- `Levels` List of length K of the vectors containing values of the factors.
- `Nruns` Number of runs of the experiment.
- `criterion.choice` Compound criterion to be used for the optimal design search or evaluation.
- `Nstarts` The number of randomly generated start designs of the search algorithm.
- `Biter` Number of samples for evaluating the MSE determinant-based component criterion.
- `tau2` The variance scaling parameter for the prior distribution of the potential terms.

- `tau` The square root of `tau2`
- `Cubic` Whether the experimental region is cubic (`TRUE`) or spherical (`FALSE`).
- `MC` Indicator of the multiple comparison (Bonferroni) correction for trace-based criteria.
- `prob.DP, prob.LP, prob.LoF, prob.LoFL` Confidence levels for the DP-, LP-, lack of fit determinant- and trace-based criteria.
- `alpha.DP, alpha.LP, alpha.LoF, alpha.LoFL` Significance levels for the DP-, LP-, lack of fit determinant- and trace-based criteria.
- `orth` Whether the candidate sets are orthonormalised (`TRUE`) or not (`FALSE`).
- `Z0` Z0 matrix.
- `W` Weight matrix for Ls criterion.
- `primary.terms` Fitted (primary) model terms.
- `potential.terms` Potential terms.
- `P` The number of terms in the fitted model (including intercept).
- `Q` The number of potential terms.
- `kappa.Ds, kappa.DP, kappa.L, kappa.LP, kappa.LoF, kappa.bias, kappa.mse` Compound criterion weights.
- `warning.msg` Warning messages.

References

Gilmour SG, Trinca LA (2012). "Optimum Design of Experiments for Statistical Inference (with discussion)." *Journal of the Royal Statistical Society C*, **61**, 345-401.

Goos P, Kobilinsky A, O'Brien TE, Vandebroek M (2005). "Model-Robust and Model-Sensitive Designs." *Computational Statistics and Data Analysis*, **49**, 201-216.

Examples

```
example1 <- mood(K = 5, Levels = 3, Nruns = 40, criterion.choice = "GDP",
kappa = list(kappa.Ds = 1./3, kappa.DP = 1./3, kappa.LoF = 1./3),
control = list(Nstarts = 50, tau2 = 0.1),
model_terms = list(primary.model = "second_order",
potential.terms = c("x12x2", "x22x3", "x32x4", "x42x5")))
example1

example2 <- mood(K = 3, Nruns = 12, Levels = list(1:3, 1:2, 1:2), criterion.choice = "MSE.L",
kappa = list(kappa.LP = 1./2, kappa.LoF = 1./4, kappa.mse = 1./4),
control = list(Nstarts = 50, tau2 = 1),
model_terms = list(primary.terms = "first_order",
potential.terms = c("x12", "x12x2", "x12x3")))
example2
```

point.swap *Swapping points between the current design and candidate set*

Description

Performing point-exchange algorithm, extensive swap of points procedure between the current design and candidate set.

Usage

```
point.swap(X1, X2, cand.full, search.object)
```

Arguments

X1	Current fitted (primary) model matrix
X2	Current potential terms matrix
cand.full	Full candidate matrix
search.object	Object for the search

Details

point.swap is called within the Search function

Value

A list of model matrices, criteria values and whether the search needs to continue

print.settings *S3 print method*

Description

Prints a summary of the mood object, including parameters that define the experiment and the (compound) criterion under which the design will be sought.

Usage

```
## S3 method for class 'settings'
print(x, ...)
```

Arguments

x	mood object
...	further arguments passed to or from other methods

Value

No return value, prints summary of object to output

`Search` *Searching for a multi-objective optimal completely randomised design.*

Description

Performing search for a (nearly) optimum factorial design, optimising with respect to a specified compound criterion.

Usage

```
Search(
  mood.object,
  algorithm = c("ptex", "coordex"),
  parallel = FALSE,
  verbose = TRUE
)
```

Arguments

`mood.object`	The object generated by `mood` function, containing the parameters of the experiment, the compound criterion and search parameters
`algorithm`	Parameter specifying the search algorithm. If `ptex` (default for $K <= 4$), the point-exchange algorithm is used, and if `coordex` (default for $K > 4$), the coordinate-exchange.
`parallel`	If `TRUE` use the `doFuture` package to run independent iterations of the algorithm in parallel using `foreach`. Requires `doFuture` library to be installed and a `Future plan` to be specified. See examples.
`verbose`	If `TRUE`, progress messages through the search iterations are shown.

Details

`Search` takes the mood object as an input with all the parameters of the experiment. Runs a point-exchange or a coordinate-exchange algorithm, returns design and model matrices, computation time and criteria values. See (Koutra et al. 2024) for examples of using `parallel = TRUE`.

Value

List of the outputs generated by the search:

- `X.design` Design matrix.
- `df` The number of pure error degrees of freedom.
- `X1` Primary model matrix for the found (nearly-) optimum design.

- `X2` Model matrix of potential terms for the found (nearly-) optimum design.
- `compound.value` The compound criterion value of the (nearly-) optimum design.
- `criteria.values` Component criteria values of the (nearly-) optimum design.
- `path` The "path" of compound criterion values of the optimum designs obtained after for each random start.
- `time` Computation time.
- `algorithm` Point exchange or coordinate exchange used to find the design?
- `parallel` Were different runs of the algorithm performed across different CPU cores (`TRUE`/`FALSE`)

References

Koutra V, Egorova O, Gilmour SG, Trinca LA (2024). "MOODE: An R Package for Multi-Objective Optimal Design of Experiments." arXiv:2412.17158, https://arxiv.org/abs/2412.17158.

See Also

mood

Examples

```
example1 <- mood(K = 2, Levels = 3, Nruns = 10, criterion.choice = "GDP",
                 kappa = list(kappa.Ds = 1./3, kappa.DP = 1./3, kappa.LoF = 1./3),
                 control = list(tau2 = 0.1),
                 model_terms = list(primary.model = "first_order",
                 potential.terms = c("x12", "x22", "x1x2")))
# Using point exchange
Search_point <- Search(example1, algorithm = 'ptex')
Search_point
# Using coordinate exchange (the default for K>4)
Search_coord <- Search(example1)
Search_coord
```

Index