# Package 'suropt'

January 11, 2019

**Type** Package

**Title** Surrogate-Based Optimization

**Version** 0.1.1

**Description** Multi-Objective optimization based on surrogate models.
Important functions: build_surmodel, train_hego, train_mego, train_sme.

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 6.1.1

**Depends** R (>= 3.4.0)

**Imports** methods, lhs, ggplot2, dplyr, tidyr, purrr, tibble,
DiceKriging, DiceOptim, GPareto, emoa, mco, rgenoud, pso, GenSA

**NeedsCompilation** no

**Author** Adriano Passos [aut, cre],
Marco Luersen [ctb]

**Maintainer** Adriano Passos <adriano.utfpr@gmail.com>

**Repository** CRAN

**Date/Publication** 2019-01-11 12:50:02 UTC

## R topics documented:

---

.SEL *sel_funs: Selection Functions*

---

## Description

Selection functions

## Usage

```
.SEL(x, ...)

.X(x)

## S3 method for class 'data.frame'
.X(x)

## S3 method for class 'list'
.X(x)

## Default S3 method:
.X(x, ...)

.Y(x)

## S3 method for class 'data.frame'
.Y(x)

## S3 method for class 'list'
.Y(x)

## Default S3 method:
.Y(x, ...)

.G(x)

## S3 method for class 'data.frame'
.G(x)

## S3 method for class 'list'
.G(x)
```

```
## Default S3 method:
.G(x, ...)
```

### Arguments

| | |
|---|---|
| x | input data to be subeseted |
| ... | aditional parameters |

---

## .SET *set_funs: Setting Functions*

---

### Description

Seting functions

### Usage

```
.SET(x, value)

.Y(x) <- value

.G(x) <- value
```

### Arguments

| | |
|---|---|
| x | input data to be subeseted |
| value | value to be set to x |
| ... | aditional parameters |

---

## build_surmodel *Build an surmodel object*

---

### Description

Build an surmodel object

### Usage

```
build_surmodel(fn, n_in, d_in, doe_type = "rlhs", sur_type = "mkm",
  pre_process = NULL, post_process = NULL)
```

## Arguments

| | |
|---|---|
| fn | high fidelity function (fun must return a list of vectors with the format list(y = c(y1, y2, y3), g = c(g1, g2))). Alternatively, it can e a data.frame with names such as X.1, X.2, Y.1, G.1, G.2, G3 |
| n_in, d_in | integer number of observations and dimension of the input variables |
| doe_type | string that defines the doe generation scheme (list valid schemes) |
| sur_type | string that defines the surogate type (list valid types) |
| pre_process | string vector defining the pre processing functions |
| post_process | string vector defining the post processing |

## Value

surogate model

## Examples

```
fn <- function(x) list(y = x^2)
model <- build_surmodel(fn, 20, 1)

fn <- function(x) list(y = DiceKriging::branin(x))
model <- build_surmodel(fn, 20, 2)

fn <- function(x) list(y = DiceKriging::branin(x), g = 0.2 - prod(x))
model <- build_surmodel(fn, 20, 2)

fn <- shaffer2
model <- build_surmodel(fn, 20, 1)

fn <- binh
model <- build_surmodel(fn, 20, 2)

data <- data.frame(X.1 = runif(5), X.2 = runif(5), Y.1 = runif(5))
model <- build_surmodel(data)

data <- data.frame(X.1 = runif(5), X.2 = runif(5), Y.1 = runif(5), G.1 = rnorm(5))
model <- build_surmodel(data)
```

---

# get *get: get prediction data of models*

---

## Description

Get prediction dafa of models

### Usage

```
get_feasibility(model, newdata)

get_entropy(model, newdata)

get_stats(model, newdata, slot = "all", stats = c("mean", "sd"))
```

### Arguments

| | |
|---|---|
| model | suropt model |
| newdata | input data where predictions will be done |
| slot | character indicating slot, must be 'y', 'g' or 'all' (default) |
| stats | character indicating statistics to be taken, must be one or some of 'mean', 'sd', 'lower95', 'upper95' or 'trend' |

### Details

Bla

---

## nowacki_beam *Test function: The Nowacki Beam*

---

### Description

This function is a variation of the classic multi-objective optimization problem (NOWACKI, 1980). In this problem the aim is to design a tip loaded cantilever eam for minimum cross-sectional aera and lowest bending stress subject to a number of constraints.

### Usage

```
nowacki_beam(x, g = c(5, 240, 120, 10, 2), l = 1500, F = 5000,
  E = 216620, G = 86650, v = 0.27, box = data.frame(b = c(10, 50),
  h = c(20, 250)))
```

### Arguments

| | |
|---|---|
| x | vector of length 2 correspon the normalized beath and height of the beam |
| g | vector of lenght 5 containing the upper limits of each constraint |
| l | numeric length of the beam |
| F | numeric force applied at the beam tip |
| E | numeric elastic longitudinal moduli |
| G | numeric elastic transversal moduli |
| v | numeric poison ratio |
| box | data.frame structure containing the upper and lower limits for b and h |

## Value

vector of objective and constrain responses

## References

Forrester, A., Sobester, A., & Keane, A. (2008). *Engineering design via surrogate modelling: a practical guide.* John Wiley & Sons.

## Examples

```
nowacki_beam(c(0,0))
nowacki_beam(c(1,1))
```

---

`predict,surmodel-method` *Predictor surogate model*

---

## Description

This functions performs predictions for an surogate model.

## Usage

```
## S4 method for signature 'surmodel'
predict(object, newdata = NULL)
```

## Arguments

| | |
|---|---|
| `object` | An object of class surmodel |
| `newdata` | An opttional vector, matrix or data.frame containing the points where to perfom predictions. If not provided the predicted optima will be outputed. |

---

`pre_process` *Pre-Processing of models*

---

## Description

Pre-Processing of models

## Usage

```
pre_process(data, operations = NULL, control = NULL)
```

### Arguments

| | |
|---|---|
| data | input data.frame |
| operations | string vector with the operation names in order to be applied |
| control | list of control parameters for the operations |

### Examples

```
fn <- binh
model <- build_surmodel(fn, 20, 2)
data <- model@data
```

---

## show.surmodel *Custom method for showing model objects*

---

### Description

Custom method for showing model objects

### Usage

```
show.surmodel(object)
```

### Arguments

| | |
|---|---|
| object | surmodel object |

---

## surmodel-class *The* surmodel *class*

---

### Description

The surmodel class

### Slots

data a data.frame with the model data

sur a list with the surogates

fn a function used as high-fidelity model

suropt *suropt: Surrogate-based optimization*

## Description

suropt is a implementation of some surrogate-based single and multi-objective optimization schmes.

## Details

The idea, is to alleviate the computational cost of optimization exploiting meta-models of the functions.

test_functions *nowacki_beam_tps' Test functions for optimization*

## Description

This page is a collection of test functions commonly used to test optimization algorithms

## Usage

```
shaffer1(x)

shaffer2(x)

fonseca(x)

kursawe(x)

viennet(x)

binh(x)
```

## Arguments

| | |
|---|---|
| x, | numeric value (or vector for multivariable functions) |

## References

https://en.wikipedia.org/wiki/Test_functions_for_optimization

http://www.sfu.ca/~ssurjano/optimization.html

## Examples

```
#function should be evaluated in the -A < x < A interval,
#where A is from 10 to 10^5 and \length(x) = 1
shaffer1(0)

#function should be evaluated in the -5 < x < 10 interval \length(x) = 1
shaffer2(0)

#function should be evaluated in the -20 < x < 20 interval and \length(x) >= 1
fonseca(rep(0,10))

#function should be evaluated in the -5 < x < 5 interval and \length(x) == 3
kursawe(rep(0,3))

#function should be evaluated in the -3 < x < 3 interval and \length(x) == 2
viennet(c(0.5,0.5))

#function should be evaluated in the 0 < x < (5,3) interval and \length(x) == 2
binh(c(0,0))
```

---

`train_hego` *Trainer for a surmodel object based on the HEGO algorithm*

---

## Description

Trainer for a surmodel object based on the HEGO algorithm

## Usage

```
train_hego(model, niter, optimizer = "gen")
```

## Arguments

| | |
|---|---|
| `model` | surmodel object to be trained |
| `niter` | integer indicating number of iterations |
| `optimizer` | string one of: "gen" (genetic, default), "sa" (simulated annealing) or "ps" (particle swarm) |

## Examples

```
fn <- binh
# model <- build_surmodel(fn, 5, 2) %>% train_hego(1, 'sa')
```

## train_mego *Trainer for a surmodel object based on the MEGO algorithm*

### Description

Trainer for a surmodel object based on the MEGO algorithm

### Usage

```
train_mego(model, niter, optimizer = "gen")
```

### Arguments

| | |
|---|---|
| model | surmodel object to be trained |
| niter | integer indicating number of iterations |
| optimizer | string one of: "gen" (genetic, default), "sa" (simulated annealing) or "ps" (particle swarm) |

### Examples

```
fn <- binh
# model <- build_surmodel(fn, 5, 2) %>% train_mego(1, 'sa')
```

## train_sme *Trainer for a surmodel object based on the SME algorithm*

### Description

Trainer for a surmodel object based on the SME algorithm

### Usage

```
train_sme(model, niter, optimizer = "nsga2")
```

### Arguments

| | |
|---|---|
| model | surmodel object to be trained |
| niter | integer indicating number of iterations |
| optimizer | character, only working for nsga2 by now |

## Examples

```
# fn <- shaffer2
# model <- build_surmodel(fn, 10, 1) %>% train_sme(5)

# fn <- binh
# model <- build_surmodel(fn, 10, 2) %>% train_sme(5)
# suropt:::plot_predict(model)

# data <- data.frame(X.1 = runif(5), X.2 = runif(5), Y.1 = runif(5), G.1 = rnorm(5))
# x_star <- build_surmodel(data) %>% train_sme(-1)
```

# Index