Package ‘spotGUI’

October 10, 2018

Type Package

Title Graphical User Interface for the Package ‘SPOT’

Version 0.2.1

Author Frederik Rehbach [aut, cre],
        Martin Zaefferer [aut],
        Thomas Bartz-Beielstein [ctb],
        Andreas Fischbach [ctb],
        Lorenzo Gentile [ctb]

Maintainer Frederik Rehbach <frederik.rehbach@th-koeln.de>

Description A graphical user interface for the Sequential Parameter Optimization Toolbox (package ‘SPOT’).
It includes a quick, graphical setup for spot, interactive 3D plots, export possibilities and more.

License GPL (>= 2)

Encoding UTF-8

LazyData true

Depends R (>= 3.1.0), shinyBS

Imports smoof, shiny, shinydashboard, SPOT (>= 2.0.3), gridExtra,
        shinyjs, rhandsontable, XML, rclipboard, plotly, tools, httpuv,
        methods

Suggests testthat, shiniytest, devtools

RoxygenNote 6.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2018-10-10 09:30:02 UTC

R topics documented:

evaluateMissingCandidateSolutions .................................................. 2
getServer .................................................................................. 3
getTextoutputBestSolution ............................................................... 3
getUIPage .................................................................................. 4
runSpotGUI ............................................................................... 4
**evaluateMissingCandidateSolutions**

---

**Description**

evaluateMissingCandidateSolutions evaluates all non-evaluated candidate solutions in a given data.frame. This function is used as a convenience function for codes that are automatically generated by the SPOT-GUI.

**Usage**

evaluateMissingCandidateSolutions(currX, currY = NULL, fun)

**Arguments**

- **currX**: A matrix containing all candidate solutions. One candidate per row.
- **currY**: A column vector with all known objective function results for the given matrix of candidate solutions. Default = NULL (In this case all candidate solutions will be evaluated). Missing values have to be marked as NA.
- **fun**: The objective function on which the given candidate solutions shall be evaluated.

**Value**

- **y**: An updated column vector with evaluation results for all candidate solutions given in currX

**Examples**

```r
library(SOPT)
spotData <- NULL
# Generating DOE
spotData$x <- designLHD(x = NULL, lower = c(-5, -5), upper = c(5, 5),
    control = list(size = 10,
        types = c("numeric", "numeric")))

# Evaluating Candidate Solutions
spotData$y <- evaluateMissingCandidateSolutions(
    currX = spotData$x, currY = spotData$y, fun = funSphere)

# Build model on evaluated data
spotData$modelFit <- buildKriging(as.matrix(spotData$x), as.matrix(spotData$y))
```
**getServer**

*Generate Server Part of SPOT-GUI*

**Description**
Generates the server part of the SPOT-GUI. This method is used internally in the starting process of the GUI. Manual use of this function is not advised.

**Usage**
```
getServer(input, output, session)
```

**Arguments**
- `input` shiny UI-input
- `output` shiny UI-output
- `session` shiny UI-session

**getTextoutputBestSolution**

*Textoutput Field 'Best Solution'*

**Description**
Generates the outputField to show the best till then found candidate solution

**Usage**
```
getTextoutputBestSolution(input, data)
```

**Arguments**
- `input` shiny input
- `data` data.frame with all candidate solutions

**Value**
```
outputField
```
getUIPage  

Define UI of SPOT-GUI

Description

Generates the UI part of the SPOT-GUI. This method is used internally in the starting process of the GUI. Manual use of this function is not advised.

Usage

getUIPage()

runSpotGUI  

runSpotGUI

Description

Run the starting command of the SPOT-GUI. Opens the graphical shiny application through which the user can access the SPO Toolbox.

Usage

runSpotGUI()
Index

evaluateMissingCandidateSolutions, 2
getServer, 3
getTextoutputBestSolution, 3
getUIPage, 4
runSpotGUI, 4