Package: CPoptim (via r-universe)

September 11, 2024

Type Package

Title Convex Partition Optimisation

Version 0.1.0

Author Erick G.G. de Paz [aut, cre] (<https://orcid.org/0000-0001-7878-8238>), Humberto Vaquera Huerta [aut] (<https://orcid.org/0000-0002-2805-804X>), Francisco Javier Albores Velasco [aut] (<https://orcid.org/0009-0003-1871-7288>), John R. Bauer Mengelberg [aut] (<https://orcid.org/0000-0002-8106-8546>), Juan Manuel Romero Padilla [aut] (<https://orcid.org/0000-0002-2654-5266>)

Maintainer Erick G.G. de Paz <giles.erick@colpos.mx>

Description Convex Partition is a black-box optimisation algorithm for single objective real-parameters functions. The basic principle is to progressively estimate and exploit a regression tree similar to a CART (Classification and Regression Tree) of the objective function. For more details see 'de Paz' (2024) <doi:10.1007/978-3-031-62836-8_3> and 'Loh' (2011) <doi:10.1002/widm.8> .

License GPL-2

Encoding UTF-8

NeedsCompilation yes

Date 2024-07-07

Depends R (>= 2.10.0)

Date/Publication 2024-07-12 11:10:02 UTC

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

RemoteUrl https://github.com/cran/CPoptim

RemoteRef HEAD

RemoteSha 92b8fa6d20e86ed0bbbdda263e9aea6a7da8b456

Contents

`CPoptim` *CPoptim*

Description

Minimise a given objective function in a bounded search space.

Usage

```
CPoptim(rFunction,lower,upper,maxFE,sampleSize)
```

Arguments

`rFunction`	the function to minimise
`lower`	a vector providing the lower bounds of the search space
`upper`	a vector providing the upper bounds of the search space
`maxFE`	number of function evaluations to compute (default 5000*dim)
`sampleSize`	sample size per partition (default 1000)

Details

Convex Partition is a black-box optimisation algorithm for single objective functions with real parameters. The basic principle is to progressively estimate and exploit a regression tree similar to CART (Classification and Regression Tree) of the objective function. This model is computed by recursively partitioning the function domain into subsets and estimating local statistics for each one. The subsets most likely to contain the optimum are selected by the Bayesian inference method proposed in *de Paz et al. (2024).*

The optimisation strategy consists of iterating two phases: 1) Selecting the most promising subset to contain extreme values according to the current model, and 2) Updating the model by sampling over the most promising subsets.

Value

CPoptim returns two lists: `sample` and `subsets`. If the function call is not proper, CPoptim returns `NULL`.

`sample` contains three matrices that summarise information about each evaluated point. The i-th row of these matrices contains:

`sample$x`	the i-th point that was evaluated
`sample$y`	the function evaluation for the i-th point

`sample$subsetID`
: the subset-id to which the i-th point belongs

`subsets` summarises information about each defined subset. The i-th row of each matrix contains

`subsets$lower` the lower bounds of the i-th subset

`subsets$upper` the upper bounds of the i-th subset

`subsets$mean` the mean value of the objective fun. in the i-th subset

`subsets$std` the standard deviation of the objective fun. in the i-th subset

`subsets$aPriori`
: the relative (length, area, volume, etc.) of the i-th subset

`subsets$aPosteriori`
: the posteriori probability that the i-th subset contains the optimum

Author(s)

The design is inspired by the algorithm proposed in *de Paz et al. (2024)*. However, instead of the original regression tree based on simplexes, this implementation is based on hyper-rectangular subsets (a model similar to the continuous Classification and Regression Trees) *Loh (2011)*.

References

de Paz, Erick G.G., et al. (2024). *A Regression Tree as Acquisition Function for Low-dimensional Optimisation.* Pattern Recognition. *MCPR 2024. Lecture Notes in Computer Science, vol* **14755**. Springer, Cham. doi: 10.1007/9783031628368_3

Loh, Wei-Yin (2011). *Classification and regression trees.* WIREs Data Mining Knowl Discov 2011 1 14-23. John Wiley & Sons, Inc. doi: 10.1002/widm.8

Examples

```
## An illustrative function
sphere<-function(X) sum(X^2)^.5
bounds<-rep(5,10)

## Calling the CPoptim function
obj<-CPoptim(sphere, -bounds, +bounds)

## Ploting the convergence curve
plot(obj$sample$y,t='l')

## Selecting the best X evaluated
optimum.x<-obj$sample$x[which.min(obj$sample$y),]
```

Index