Package 'soma'

August 29, 2016

Version 1.1.1

Date 2014-11-24

Title General-Purpose Optimisation With the Self-Organising Migrating
Algorithm

Author Jon Clayden; based on the work of Ivan Zelinka

Maintainer Jon Clayden <code@clayden.org>

Depends R (>= 2.5.0)

Imports reportr

Description This package provides an R implementation of the Self-Organising Migrating Algorithm, a general-purpose, stochastic optimisation algorithm. The approach is similar to that of genetic algorithms, although it is based on the idea of a series of ``migrations'' by a fixed set of individuals, rather than the development of successive generations. It can be applied to any cost-minimisation problem with a bounded parameter space, and is robust to local minima.

License GPL-2

URL https://github.com/jonclayden/soma/,

http://www.ft.utb.cz/people/zelinka/soma/

NeedsCompilation no

Repository CRAN

Date/Publication 2014-11-25 09:48:13

R topics documented:

---

soma   *The Self-Organising Migrating Algorithm*

Description

The Self-Organising Migrating Algorithm is a general-purpose, stochastic optimisation algorithm. The approach is similar to that of genetic algorithms, although it is based on the idea of a series of "migrations" by a fixed set of individuals, rather than the development of successive generations. It can be applied to any cost-minimisation problem with a bounded parameter space, and is robust to local minima.

Usage

```
soma(costFunction, bounds, options = list(), strategy = "all2one", ...)

## S3 method for class 'soma'
plot(x, y = NULL, ...)
```

Arguments

costFunction	A cost function which takes a numeric vector of parameters as its first argument, and returns a numeric scalar representing the associated cost value.
bounds	A list with elements min and max, each a numeric vector giving the upper and lower bounds for each parameter, respectively.
options	A list of options for the SOMA algorithm itself. See Details.
strategy	The strategy type to use. Currently "all2one" is the only supported value.
...	Additional parameters to costFunction, or to [plot]().
x	An object of class "soma".
y	Ignored.

Details

A number of options are available to adjust the behaviour of the optimisation, and its termination criteria. The default values used here are recommended by Zelinka (2004).

pathLength: The distance towards the leader that individuals may migrate. A value of 1 corresponds to the leader's position itself, and values greater than one (recommended) allow for some overshoot. The default is 3.

stepLength: The granularity at which potential steps are evaluated. It is recommended that the pathLength not be a whole multiple of this value. The default is 0.11.

perturbationChance: The probability that individual parameters are changed on any given step. The default is 0.1.

minAbsoluteSep: The smallest absolute difference between the maximum and minimum cost function values. If the difference falls below this minimum, the algorithm will terminate. The default is 0, meaning that this termination criterion will never be met.

minRelativeSep: The smallest relative difference between the maximum and minimum cost function values. If the difference falls below this minimum, the algorithm will terminate. The default is 0.001.

nMigrations: The maximum number of migrations to complete. The default is 20.

populationSize: The number of individuals in the population. It is recommended that this be somewhat larger than the number of parameters being optimised over, and it should not be less than 2. The default is 10.

Value

A list of class `"soma"`, containing the following elements.

leader	The index of the "leader", the individual in the population with the lowest cost.
population	A matrix whose columns give the parameter values for each individual in the population at convergence.
cost	A vector giving the cost function values for each individual at convergence.
history	A vector giving the cost of the leader for each migration during the optimisation. This should be nonincreasing.
migrations	The number of migrations completed.

A [plot](#) method is available for this class, which shows the history of leader cost values during the optimisation.

Author(s)

R implementation by Jon Clayden <jon.clayden@gmail.com>

References

I. Zelinka (2004). SOMA - self-organizing migrating algorithm. In G.C. Onwubolu & B.V. Babu, eds, New optimization techniques in engineering. Volume 141 of "Studies in Fuzziness and Soft Computing", pp. 167-217. Springer.

See Also

[optim](#) implements other general-purpose optimisation methods

Examples

```
# Rastrigin's function, which contains many local minima
rastrigin <- function (a) 20 + a[1]^2 + a[2]^2 - 10*(cos(2*pi*a[1])+cos(2*pi*a[2]))

# Find the global minimum over the range -5 to 5 in each parameter
x <- soma(rastrigin, list(min=c(-5,-5),max=c(5,5)))

# Find the location of the leader - should be near the true minimum of c(0,0)
print(x$population[,x$leader])

# Plot the cost history of the leaders
plot(x)
```

Index