Package 'MetaheuristicFPA'

October 12, 2022

Type Package

Title An Implementation of Flower Pollination Algorithm in R

Version 1.0

Date 2016-08-28

Author Amanda Pratama Putra with contributions from Margaretha Ari Anggorowati

Maintainer Amanda Pratama Putra <12.7012@stis.ac.id>

Description A nature-inspired metaheuristics algorithm based on the pollination process of flowers. This R package makes it easy to implement the standard flower pollination algorithm for every user. The algorithm was first developed by Xin-She Yang in 2012 (<DOI:10.1007/978-3-642-32894-7_27>).

License GPL-2

Imports Rcpp (>= 0.12.6)

LinkingTo Rcpp, RcppArmadillo

NeedsCompilation yes

Repository CRAN

Date/Publication 2016-08-29 08:18:35

R topics documented:

fpa_optim 1

Index **4**

`fpa_optim` *Metaheuristic - Flower Pollination Algorithm*

Description

The function `fpa_optim` implements a nature-inspired metaheuristic algorithm. The algorithm is based on the the pollination process of flowers called Flower Pollination Algorithm (FPA).

Usage

```
fpa_optim(N, p, beta, eta, maxiter, randEta, gloMin, objfit, D,
          Lower, Upper, FUN)
```

Arguments

N	integer: the population size, typically between 10 and 25
p	numeric: the probability switch (0...1) to determine whether to pollinate local pollination or global pollination. default = 0.8
beta	numeric; parameter to determine the step size of levy flight. default = 1.5
eta	numeric; scaling factor to control the step size. default = 0.1
maxiter	integer: the number of maximum generations, or iterations until it find the global minimum. default = 2000
randEta	boolean: if it's true, scaling factor eta will be random. default = FALSE
gloMin	numeric: minimum global value of the benchmark function
objfit	numeric: the value of tolerance for difference between the objective value were found with the actual value of global minimum
D	integer: the dimension of the search variables
Lower	numeric: lower bound of the search variables
Upper	upper bound of the search variables
FUN	the objective function to optimize, must be supplied by a user

Details

fpa_optim implements the standard flower pollination algorithm in three major steps. The first step is initialization of FPA Parameters. it will generate the initial population and determine the current best solution.

Secondly, a population of flowers are doing pollination in a *d*-dimensional search or solution space according to the updating rules of the algorithm: each flower will choose to pollinate a local pollination or global pollination at each iteration in the search space. The location of flowers are the solutions vector, and the objective function value for every solutions is achieved.

Then the current best solution is improved for every iteration. The new solution is evaluated and updated. Finally, the best solution has been reached the minimum global problems. See *References* below for more details.

the advantages of the flower pollination algorithm is it can converge very quickly and can avoid the local minima because it make the long distances movement based of levy flight.

Value

Returns a list of 3 values: minimum fitness, best solution(s), number of iterations

Author(s)

Amanda Pratama Putra, Margaretha Ari Anggorowati

References

[1] Yang, X.-S. "Flower Pollination Algorithm for Global Optimization." 11th International Conference, UCNC 2012, Springer-Verlag Berlin Heidelberg, 2012. 65-74.

Examples

```
# find the x-value that gives the minimum of the dejong benchmark function
# y = sum(x[i]^2), i=1:n, -5.12 <= x[i] <= 5.12

# global minimum is 0 when each x = 0
deJong<-function(x){
  deJong = sum(x^2)
}


# run a simulation using the standard flower pollination algorithm
set.seed(1024)  # for reproducive results
library(MetaheuristicFPA)
fpa_opt <- fpa_optim(N = 25, p =0.8 , beta = 1.5, eta = 0.1,
                 maxiter=5000, randEta=FALSE, gloMin=0, objfit=1e-7,
                 D=4, Lower = -5.12, Upper = 5.12, FUN = deJong)
x <- fpa_opt$best_solution
```

Index

fpa_optim, 1