

Package ‘MetaEEEA’

July 24, 2026

Type Package

Title Metaheuristic Algorithms with Explicit Exploration

Description Solves single-objective optimization problems by using bio-inspired metaheuristic algorithms. The implemented metaheuristics are the Butterfly Optimization Algorithm, the Ladybug Beetle Optimization Algorithm and the Prairie Dog Optimization Algorithm. For all these optimization algorithms, the search of optimal values can be reinforced with the explicit exploration strategy proposed by Salinas-Gutiérrez and Muñoz Zavala (2023) <[doi:10.1016/j.asoc.2023.110230](https://doi.org/10.1016/j.asoc.2023.110230)>.

Version 1.0.0

License GPL-3

Encoding UTF-8

Imports EEEA

Author Rogelio Salinas Gutiérrez [aut, cre, cph] (ORCID: <<https://orcid.org/0000-0002-1669-4460>>),
Cristina Díaz Esquivel [aut, cph] (ORCID: <<https://orcid.org/0009-0003-6990-5326>>),
Angela María Gallegos Martínez [aut, cph] (ORCID: <<https://orcid.org/0009-0005-7032-0493>>),
Agustín Moreno Cruz [aut, cph] (ORCID: <<https://orcid.org/0009-0009-9214-7002>>),
Miguel Angel Moreno Urbina [aut, cph] (ORCID: <<https://orcid.org/0009-0008-0036-8936>>),
Pedro Abraham Montoya Calzada [aut, cph] (ORCID: <<https://orcid.org/0009-0002-3497-210X>>),
Carlos Alberto López Hernández [aut, cph] (ORCID: <<https://orcid.org/0009-0000-9428-1293>>),
Ilse Daniela Saldivar Olvera [aut, cph] (ORCID: <<https://orcid.org/0009-0002-2406-2946>>)

Maintainer Rogelio Salinas Gutiérrez <rogelio.salinas@edu.uaa.mx>

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Config/roxygen2/version 8.0.0

NeedsCompilation no

Repository CRAN
Date/Publication 2026-07-24 10:00:26 UTC

Contents

boa_metaheuristic	2
lbo_metaheuristic	4
pdo_metaheuristic	6
Index	8

boa_metaheuristic	<i>Butterfly Optimization Algorithm (BOA)</i>
-------------------	---

Description

Provides an implementation of the Butterfly Optimization (BOA) for optimization problems.

Usage

```
boa_metaheuristic(  
  obj.fun,  
  pop.size=30,  
  dim=2,  
  lb,  
  ub,  
  gen=100,  
  pb=0,  
  EE=FALSE,  
  p=0.8,  
  a=0.1,  
  c=0.01,  
  ...)
```

Arguments

obj.fun	A function to be minimized, with first argument the vector of parameters over which minimization is to take place.
pop.size	An integer specifying the population size (number of ladybugs).
dim	An integer indicating the dimensionality of the search space.
lb	A numeric vector defining the lower bounds of the search space.
ub	A numeric vector defining the upper bounds of the search space.
gen	An integer specifying the maximum number of generations (iterations).
pb	An integer specifying the patience limit for early stopping (the number of consecutive generations without improvement before the algorithm terminates).

EE	A logical value indicating whether to enable explicit exploration mechanisms during population initialization process.
p	A numeric value in [0, 1] representing the switch probability.
a	A numeric value acting as the power exponent for stimulus intensity.
c	A numeric value representing the initial sensory modality.
...	Additional arguments of the objective function.

Details

The Butterfly Optimization Algorithm is a swarm intelligence metaheuristic inspired by the biological foraging and mating behaviors of butterflies. The algorithm simulates the fragrance emission and perception to create a social network of individuals, utilizing global search toward the current best butterfly (highly fragrant stimulus) and local random walks when sensory modality drops, effectively balancing exploration and exploitation within the feasible search space.

Value

Returns a list with the following entries:

best.fit	A numeric value representing the best fitness value found by the algorithm.
best.sol	A numeric vector representing the best individual (solution) found that achieved the best.fit value.

References

Arora, S., & Singh, S. (2019). Butterfly optimization algorithm: A novel approach for global optimization. *Soft Computing*, **23**(3), 715–734. doi:10.1007/s0050001831024 Okwu, M. O., & Tartibu, L. K. (2021). Butterfly Optimization Algorithm. In *Metaheuristic Optimization: Nature-Inspired Algorithms Swarm and Computational Intelligence, Theory and Applications* (Studies in Computational Intelligence, Vol. 927, pp. 105–114). Springer Nature Switzerland AG. doi:10.1007/9783-030611118_11

Examples

```
obj.fun <- function (x){
  y <- sum(x^2)
  return (y)
}
pop.size <- 30
dim <- 2
lb <- -5.12
ub <- 5.12
gen <- 100
pb <- 20
EE <- TRUE

res.sphere <- boa_metaheuristic(obj.fun, pop.size, dim, lb, ub, gen, pb, EE)

cat("Best value on Sphere: ", paste(round(res.sphere$best.fit, 4),
  collapse=", "), "\n")
```

```
cat("Best solution en Sphere: ", paste(round(res.sphere$best.sol, 4),
collapse=", "), "\n")
```

lbo_metaheuristic	<i>Ladybug Optimization Algorithm (LBO)</i>
-------------------	---

Description

Provides an implementation of the Ladybug Optimization (LBO) algorithm for optimization problems.

Usage

```
lbo_metaheuristic(
  obj.fun,
  pop.size=30,
  dim=2,
  lb,
  ub,
  gen=100,
  pb=0,
  EE=FALSE,
  pa=0.1,
  sigma=0.05,
  beta=8,
  temp.init=100,
  ...)
```

Arguments

obj.fun	A function to be minimized, with first argument the vector of parameters over which minimization is to take place.
pop.size	An integer specifying the population size (number of ladybugs).
dim	An integer indicating the dimensionality of the search space.
lb	A numeric vector defining the lower bounds of the search space.
ub	A numeric vector defining the upper bounds of the search space.
gen	An integer specifying the maximum number of generations (iterations).
pb	An integer specifying the patience limit for early stopping (the number of consecutive generations without improvement before the algorithm terminates).
EE	A logical value indicating whether to enable explicit exploration mechanisms during population initialization process.
pa	A numeric value representing the maximum proportion of the population allowed to die during the freezing phase.
sigma	A numeric value controlling the step size during the foraging/dispersion phase (Lévy Flight).

<code>beta</code>	A numeric value representing the parameter for the survival probability equation.
<code>temp.init</code>	A numeric value specifying the initial environmental temperature.
<code>...</code>	Additional arguments of the objective function.

Details

The Ladybug Optimization algorithm is a swarm intelligence metaheuristic inspired by the biological behaviors of ladybugs. The algorithm simulates phases such as foraging/dispersion using Lévy flights, hibernation (aggregation based on global heat), and survival during freezing temperatures to explore and exploit the feasible search space.

Value

Returns a list with the following entries:

<code>best.fit</code>	A numeric value representing the best fitness value found by the algorithm.
<code>best.sol</code>	A numeric vector representing the best individual (solution) found that achieved the <code>best.fit</code> value.

References

Safiri, S., & Nikoofard, A. (2022). Ladybug Beetle Optimization algorithm: application for real-world problems. *The Journal of Supercomputing*, 79, 3511-3560. doi:10.1007/s11227022047552

Examples

```
obj.fun <- function (x){
  y <- sum(x^2)
  return (y)
}
pop.size <- 30
dim <- 2
lb <- -5.12
ub <- 5.12
gen <- 100
pb <- 20
EE <- TRUE

res.sphere <- lbo_metaheuristic(obj.fun, pop.size, dim, lb, ub, gen, pb, EE)

cat("Best value on Sphere: ", paste(round(res.sphere$best.fit, 4),
  collapse=", "), "\n")
cat("Best solution en Sphere: ", paste(round(res.sphere$best.sol, 4),
  collapse=", "), "\n")
```

pdo_metaheuristic *Prairie Dog Optimization Algorithm (PDO)*

Description

Provides an implementation of the Prairie Dog Optimization (PDO) algorithm for optimization problems.

Usage

```
pdo_metaheuristic(
    obj.fun,
    pop.size=30,
    dim=2,
    lb,
    ub,
    gen=100,
    pb=0,
    EE=FALSE,
    ...)
```

Arguments

obj.fun	A function to be minimized, with first argument the vector of parameters over which minimization is to take place.
pop.size	An integer specifying the population size (number of prairie dogs).
dim	An integer indicating the dimensionality of the search space.
lb	A numeric vector defining the lower bounds of the search space.
ub	A numeric vector defining the upper bounds of the search space.
gen	An integer specifying the maximum number of generations (iterations).
pb	An integer specifying the patience limit for early stopping (the number of consecutive generations without improvement before the algorithm terminates).
EE	A logical value indicating whether to enable explicit exploration mechanisms during population initialization process.
...	Additional arguments of the objective function.

Details

The Prairie Dog Optimization algorithm is a swarm intelligence metaheuristic inspired by the biological behaviors of prairie dogs. The algorithm simulates four distinct time-based phases: foraging/exploration using Lévy flights, burrow construction/exploitation via cooperative digging, food alarm communication using vocal alerts, and randomized flight for predator escape to explore and exploit the feasible search space.

Value

Returns a list with the following entries:

<code>best.fit</code>	A numeric value representing the best fitness value found by the algorithm.
<code>best.sol</code>	A numeric vector representing the best individual (solution) found that achieved the <code>best.fit</code> value.

References

Ezugwu, A. E., Agushaka, J. O., Abualigah, L., Mirjalili, S., & Gandomi, A. H. (2022). Prairie Dog Optimization Algorithm. *Neural Computing and Applications*, vol 34, no. 22, pp. 20017-20065. Springer, London. doi:[10.1007/s00521022075309](https://doi.org/10.1007/s00521022075309)

Examples

```
obj.fun <- function (x){
  y <- sum(x^2)
  return (y)
}
pop.size <- 30
dim <- 2
lb <- -5.12
ub <- 5.12
gen <- 100
pb <- 20
EE <- TRUE

res.sphere <- pdo_metaheuristic(obj.fun, pop.size, dim, lb, ub, gen, pb, EE)

cat("Best value on Sphere: ", paste(round(res.sphere$best.fit, 4),
  collapse=" ", "\n")
cat("Best solution en Sphere: ", paste(round(res.sphere$best.sol, 4),
  collapse=" ", "\n")
```

Index

boa_metaheuristic, [2](#)

lbo_metaheuristic, [4](#)

pdo_metaheuristic, [6](#)