Package ‘ExpDE’

January 10, 2018

Type Package
Title Modular Differential Evolution for Experimenting with Operators
Version 0.1.4
Depends R (>= 3.4.0)
Imports assertthat (>= 0.2.0)
Suggests smoof, cec2013
Date 2018-01-09
URL http://github.com/fcampelo/ExpDE
Maintainer Felipe Campelo <fcampelo@ufmg.br>
Description Modular implementation of the Differential Evolution algorithm for experimenting with different types of operators.
License GPL-2
RoxygenNote 6.0.1
NeedsCompilation no
Author Felipe Campelo [aut, cre], Moises Botelho [aut]
Repository CRAN
Date/Publication 2018-01-10 10:45:05 UTC

R topics documented:

  check_stop_criteria ............................................. 2
  create_population ................................................ 3
  evaluate_population ............................................. 3
  ExpDE .............................................................. 4
  mutation_best ..................................................... 8
  mutation_current_to_pbest ...................................... 9
  mutation_mean ..................................................... 10
  mutation_none ................................................... 11
  mutation_operators ............................................. 11
check_stop_criteria

Stop criteria for DE

Description

Implements different stop criteria for the ExpDE framework

Usage

check_stop_criteria()

Value

logical flag indicating whether any stop condition has been reached.

Warning

This routine accesses the parent environment used in the main function ExpDE(), which means that changes made in the variables contained in env WILL change the original values. DO NOT change anything unless you’re absolutely sure of what you’re doing.
**create_population**

Create population

**Description**

Creates a new population for the ExpDE framework.

**Usage**

`create_population(popsize, probpars)`

**Arguments**

- `popsize` population size
- `probpars` list of named problem parameters (see `expde`).

**Value**

A matrix containing the population for the ExpDE

---

**evaluate_population**

Evaluate DE population

**Description**

Evaluates the DE population on a given objective function.

**Usage**

`evaluate_population(probpars, Pop)`

**Arguments**

- `probpars` problem parameters (see `ExpDE` for details).
- `Pop` population matrix (each row is a candidate solution, normalized to the [0, 1] interval.)

**Value**

numeric vector (with length `nrow(Pop)`) containing the function values of each point in the population.
**ExpDE**

**Experimental Differential Evolution - ExpDE**

---

### Description

Modular implementation of the Differential Evolution Algorithm for the experimental investigation of the effects of different operators on the performance of the algorithm.

### Usage

```r
ExpDE(popsize, mutpars = list(name = "mutation_rand", f = 0.2),
      recpars = list(name = "recombination_bin", cr = 0.8, nvecs = 1),
      selpars = list(name = "standard"), stopcrit, probpars, seed = NULL,
      showpars = list(show.iters = "none"))
```

### Arguments

- **popsize**
  population size

- **mutpars**
  list of named mutation parameters. See [Mutation parameters](#) for details.

- **recpars**
  list of named recombination parameters. See [Recombination parameters](#) for details.

- **selpars**
  list of named selection parameters. See [Selection parameters](#) for details.

- **stopcrit**
  list of named stop criteria parameters. See [Stop criteria](#) for details.

- **probpars**
  list of named problem parameters. See [Problem Description](#) for details.

- **seed**
  seed for the random number generator. See [Random Seed](#) for details.

- **showpars**
  parameters that regulate the echoing of progress indicators. See [Showpars](#) for details.

### Details

This routine is used to launch a differential evolution algorithm for the **minimization** of a given problem instance using different variants of the recombination, mutation and selection operators. The input parameters that describe those operators receive list objects describing the operator variants to be used in a given optimization procedure.

### Value

A list object containing the final population (sorted by performance), the performance vector, and some run statistics.
Mutation Parameters

mutpars is used to inform the routine the type of differential mutation to use, as well as any mutation-related parameter values. The current version accepts the following options:

- \texttt{mutation\_best}
- \texttt{mutation\_rand}
- \texttt{mutation\_mean}
- \texttt{mutation\_none}
- \texttt{mutation\_current\_to\_pbest} (incl. special case \texttt{current\_to\_best})
- \texttt{mutation\_wgi}

mutpars receives a list object with name field mutpars$name (containing the name of the function to be called, e.g., name = "mutation\_rand") as well as whatever parameters that function may require/accept (e.g., mutpars$f = 0.7, mutpars$nvecs = 2, etc.). See the specific documentation of each function for details.

Some examples are provided in the Examples section below.

Recombination parameters

As with the mutation parameters, recpars is used to define the desired recombination strategy. The current version accepts the following options:

- \texttt{recombination\_arith}
- \texttt{recombination\_bin}
- \texttt{recombination\_blxAlphaBeta} (incl. special cases \texttt{blxAlpha} and \texttt{flat})
- \texttt{recombination\_eigen}
- \texttt{recombination\_exp}
- \texttt{recombination\_geo}
- \texttt{recombination\_lbga}
- \texttt{recombination\_linear}
- \texttt{recombination\_mmmax}
- \texttt{recombination\_npoint}
- \texttt{recombination\_none}
- \texttt{recombination\_onepoint}
- \texttt{recombination\_pbest}
- \texttt{recombination\_sbx}
- \texttt{recombination\_wright}

recpars receives a list object with name field recpars$name (containing the name of the function to be called, e.g., name = "recombination\_bin") as well as whatever parameters that function may require/accept (e.g., recpars$cr = 0.8, recpars$minchange = TRUE, etc.). See the specific documentation of each function for details.

Some examples are provided in the Examples section below.
Selection parameters

selpars follows the same idea as mutpars and recpars, and is used to define the selection operators. Currently, only the standard DE selection, selection_standard, is implemented.

Stop criteria

stopcrit is similar to recpar and the other list arguments, but with the difference that multiple stop criteria can be defined for the algorithm. The names of the stop criteria to be used are passed in the stopcrit$names field, which must contain a character vector. Other parameters to be used for stopping the algorithm (e.g., the maximum number of iterations stopcrit$maxiter) can also be included as stopcrit fields. Currently implemented criteria are:

- "stop_maxiter" (requires additional field stopcrit$maxiter = ? with the maximum number of iterations).
- "stop_maxeval" (requires additional field stopcrit$maxevals = ? with the maximum number of function calls).

See check_stop_criteria for details.

Problem description

The probpars parameter receives a list with all definitions related to the problem instance to be optimized. There are three required fields in this parameter:

- probpars$name, the name of the function that represents the problem to be solved.
- probpars$xmin, a vector containing the lower bounds of all optimization variables (i.e., a vector of length M, where M is the dimension of the problem).
- probparsxmax, a vector containing the upper bounds of all optimization variables.

This list can also contain the following optional arguments

- probpars$matrixEval, indicates what kind of input is expected by the function provided in probpars$name. Valid entries are "vector", "colMatrix" and "rowMatrix". Defaults to probpars$matrixEval = "rowMatrix"

Important: the objective function routine must receive either a vector or a matrix of vectors to be evaluated in the form of an input parameter named either "x" or "X" or "Pop" (any one of the three is allowed).

Random Seed

The seed argument receives the desired seed for the PRNG. This value can be set for reproducibility purposes. The value of this parameter defaults to NULL, in which case the seed is arbitrarily set using as.numeric(Sys.time()).
Showpars

showpars is a list containing parameters that control the printed output of ExpDE. Parameter showpars can have the following fields:

- `showpars$show.iter = c("dots", "numbers", "none")`: type of output. Defaults to "numbers".
- `showpars$show.every`: positive integer that determines how frequently the routine echoes something to the terminal. Defaults to 1.

Author(s)

Felipe Campelo (<fcampelo@ufmg.br>) and Moises Botelho (<moisesufop@gmail.com>)

References


Examples

```r
# DE/rand/1/bin with population 40, F = 0.8 and CR = 0.5
popsize <- 100
mutpars <- list(name = "mutation_rand", f = 0.8)
recpars <- list(name = "recombination_bin", cr = 0.5, minchange = TRUE)
selpars <- list(name = "selection_standard")
stopcrit <- list(names = "stop_maxiter", maxiter = 100)
probpars <- list(name = "sphere",
                 xmin = rep(-5.12,10), xmax = rep(5.12,10))
seed <- NULL
showpars <- list(show.iter = "numbers", show.every = 1)
ExpDE(popsize, mutpars, recpars, selpars, stopcrit, probpars, seed, showpars)

# DE/wgi/1/blxAlpha
recpars <- list(name = "recombination_blxAlphaBeta", alpha = 0.1, beta = 0.1)
mutpars <- list(name = "mutation_wgi", f = 0.8)
ExpDE(popsize, mutpars, recpars, selpars, stopcrit, probpars)

# DE/best/1/sbx
recpars <- list(name = "recombination_sbx", eta = 10)
mutpars <- list(name = "mutation_best", f = 0.6, nvecs = 1)
ExpDE(popsize, mutpars, recpars, selpars, stopcrit, probpars)

# DE/best/1/eigen/bin
recpars <- list(name = "recombination_eigen",
                othername = "recombination_bin",
                cr = 0.5, minchange = TRUE)
showpars <- list(show.iter = "dots", show.every = 10)
stopcrit <- list(names = "stop_maxeval", maxevals = 10000)
ExpDE(popsize, mutpars, recpars, selpars, stopcrit, probpars, seed = 1234)
```
**mutation_best**  
/\best mutation for DE

**Description**

Implements the "/best/nvecs" mutation for the ExpDE framework

**Usage**

mutation_best(X, mutpars)

**Arguments**

- **X**  
  population matrix

- **mutpars**  
  mutation parameters (see Mutation parameters for details)

**Value**

Matrix M containing the mutated population

**Mutation Parameters**

The mutpars parameter contains all parameters required to define the mutation. mutation_best() understands the following fields in mutpars:

- **f**: scaling factor for difference vector(s). Accepts numeric vectors of size 1 or nvecs.
- **nvecs**: number of difference vectors to use. Accepts $1 \leq nvecs \leq (\text{nrow}(X)/2 - 2)$. Defaults to 1.

**Warning**

This routine will search for the performance vector of population X (fJ) in the parent environment (using parent.frame()). This variable must be defined for mutation_best() to work.

**References**


**Author(s)**

Felipe Campelo (<fcampelo@ufmg.br>)
Description

Implements the "current-to-pbest" mutation for the ExpDE framework

Usage

mutation_current_to_pbest(xL mutpars)

Arguments

  x          population matrix
  mutpars    mutation parameters (see Mutation parameters for details)

Details

This routine also implements one special case:

• current-to-best mutation (current_to_best), by setting mutpars$P = 1);
• Flat recombination (flat), by setting recpars$alpha = recpars$beta = 0)

Value

Matrix M containing the mutated population

Mutation Parameters

The mutpars parameter contains all parameters required to define the mutation. mutation_current_to_pbest() understands the following fields in mutpars:

• f : scaling factor for difference vector(s).
  Accepts numeric vectors of size 1 or nvecs.
• p : either the number of "best" vectors to use (if given as a positive integer) or the proportion of the population to use as "best" vectors (if 0 < p < 1).

Warning

This routine will search for the performance vector of population X (J) in the parent environment (using parent.frame()). This variable must be defined for mutation_current_to_pbest() to work.

References

**mutation_mean**

**Author(s)**
Felipe Campelo (<fcampelo@ufmg.br>)

---

**mutation_mean**    **/mean mutation for DE**

**Description**
Implements the "/mean/nvecs" mutation for the ExpDE framework

**Usage**
```
mutation_mean(X, mutpars)
```

**Arguments**
- `X` population matrix
- `mutpars` mutation parameters (see Mutation parameters for details)

**Value**
Matrix M containing the mutated population

**Mutation Parameters**
The `mutpars` parameter contains all parameters required to define the mutation. `mutation_mean()` understands the following fields in `mutpars`:

- `f`: scaling factor for difference vector(s).
  Accepts numeric vectors of size 1 or `nvecs`.
- `nvecs`: number of difference vectors to use.
  Accepts `1 <= nvecs <= (nrow(X)/2 - 2)`
  Defaults to 1.

**References**

**Author(s)**
Felipe Campelo (<fcampelo@ufmg.br>)
**mutation_none**

NULL mutation for DE

**Description**

Implements the "none" mutation (i.e., no mutation performed) for the ExpDE framework

**Usage**

```
mutation_none(xL, mutpars)
```

**Arguments**

- `x` population matrix
- `mutpars` mutation parameters (see Mutation parameters for details)

**Value**

@return The same matrix `x` used as an input.

**Mutation Parameters**

The `mutpars` parameter contains all parameters required to define the mutation. `mutation_none()` requires no fields in this parameter.

---

**mutation_operators**

Mutation operators available

**Description**

List all available mutation operators in the ExpDE package

**Usage**

```
mutation_operators()
```

**Value**

Character vector with the names of all mutation operators
mutation_rand  /rand mutation for DE

Description

Implements the "/rand/nvecs" mutation for the ExpDE framework

Usage

mutation_rand(X, mutpars)

Arguments

X population matrix  
mutpars mutation parameters (see Mutation parameters for details)

Value

Matrix M containing the mutated population

Mutation Parameters

The mutpars parameter contains all parameters required to define the mutation. mutation_rand() understands the following fields in mutpars:

- f : scaling factor for difference vector(s).  
  Accepts numeric vectors of size 1 or nvecs.
- nvecs : number of difference vectors to use.  
  Accepts 1 <= nvecs <= (nrow(X)/2 - 2)  
  Defaults to 1.

References


Author(s)

Felipe Campelo (<fcampelo@ufmg.br>)
Description

Implements the "wgi/nvecs" mutation (weighted global intermediate) for the ExpDE framework. This variant is based on a recombination strategy known as "weighted global intermediate recombination" (see the References section for details)

Usage

```
motion_wgi(xL mutpars)
```

Arguments

- `x`: population matrix
- `mutpars`: mutation parameters (see Mutation parameters for details)

Value

Matrix `m` containing the mutated population

Mutation Parameters

The `mutpars` parameter contains all parameters required to define the mutation. `motion_wgi()` understands the following fields in `mutpars`:

- `f`: scaling factor for difference vector(s).
  Accepts numeric vectors of size 1 or `nvecs`.
- `nvecs`: number of difference vectors to use.
  Accepts `1 <= nvecs <= (nrow(X)/2 - 2)`
  Defaults to 1.

References


Author(s)

Felipe Campelo (<fcampelo@ufmg.br>)
print_progress  
*Print progress of DE*

**Description**

Echoes the progress of DE to the terminal.

**Usage**

```plaintext
print_progress()
```

**Parameters**

This routine accesses all variables defined in the calling environment using `parent.frame()`, so it does not require any explicit input parameters. However, the calling environment must contain:

- `showpars` list containing parameters that control the printed output of `moead()`. Parameter `showpars` can have the following fields:
  - `$show.its = c("dots", "numbers", "none")`: type of output. Defaults to "numbers".
  - `$show.every`: positive integer that determines how frequently the routine echoes something to the terminal. Defaults to 1.
- `iters()`: counter function that registers the iteration number.

recombination_arith  
*Arithmetic recombination for DE*

**Description**

Implements the "/arith" (arithmetic) recombination for the ExpDE framework.

**Usage**

```plaintext
recombination_arith(X, M, ...)
```

**Arguments**

- `X`: population matrix (original)
- `M`: population matrix (mutated)
- `...`: optional parameters (unused)

**Value**

Matrix U containing the recombined population.
recombination_bin

References


---

recombination_bin  
/bin recombination for DE

Description

Implements the "/bin" (binomial) recombination for the ExpDE framework

Usage

recombination_bin(X, M, recpars)

Arguments

X  population matrix (original)
M  population matrix (mutated)
recpars  recombination parameters (see Recombination parameters for details)

Value

Matrix U containing the recombined population

Recombination Parameters

The recpars parameter contains all parameters required to define the recombination. recombination_bin() understands the following fields in recpars:

- **cr** : component-wise probability of using the value in M.
  Accepts numeric value $0 < cr <= 1$.
- **minchange** : logical flag to force each new candidate solution to inherit at least one component from its mutated 'parent'.
  Defaults to TRUE

References

recombination_blxAlphaBeta

Blend Alpha Beta recombination for DE

Description

Implements the "/blxAlphaBeta" (Blend Alpha Beta) recombination for the ExpDE framework.

Usage

recombination_blxAlphaBeta(X, M, recpars)

Arguments

X population matrix (original)
M population matrix (mutated)
recpars recombination parameters (see Recombination parameters for details)

Details

This routine also implements two special cases:

• BLX-alpha recombination (blxAlpha), by setting recpars$alpha = recpars$beta;
• Flat recombination (flat), by setting recpars$alpha = recpars$beta = 0)

Value

Matrix U containing the recombined population

Recombination Parameters

The recpars parameter contains all parameters required to define the recombination. recombination_blxAlpha() understands the following fields in recpars:

• alpha: extrapolation parameter for 'best' parent vector.
  Accepts real value 0 <= alpha <= 0.5.
• beta: extrapolation parameter for 'worst' parent vector.
  Accepts real value 0 <= beta <= 0.5.

@section Warning: This recombination operator evaluates the candidate solutions in M, which adds an extra popsize evaluations per iteration.

References

**recombination_eigen**

/eigen recombination for DE

---

**Description**

Implements the "/eigen" (eigenvector-based) recombination for the ExpDE framework

**Usage**

recombination_eigen(xL, mL, recpars)

**Arguments**

- **X**: population matrix (original)
- **M**: population matrix (mutated)
- **recpars**: recombination parameters (see Recombination parameters for details)

**Value**

Matrix U containing the recombined population

**Recombination Parameters**

The **recpars** parameter contains all parameters required to define the recombination. `recombination_eigen()` understands the following fields in recpars:

- **othername**: name of the recombination operator to be applied after the projection in the eigenvector basis
- **...**: parameters required (or optional) to the operator defined by recpars$othername

**References**


---

**recombination_exp**

Exponential recombination for DE

---

**Description**

Implements the "/exp" (exponential) recombination for the ExpDE framework

**Usage**

recombination_exp(xL, M, recpars)
recombination_geo

Arguments

X  population matrix (original)
M  population matrix (mutated)
recpars  recombination parameters (see Recombination parameters for details)

Value

Matrix U containing the recombined population

Recombination Parameters

The recpars parameter contains all parameters required to define the recombination. recombination_exp() understands the following fields in recpars:

• cr : component-wise probability of selection as a cut-point.
  Accepts numeric value $0 < cr \leq 1$.

References


recombination_geo  

Description

Implements the "/geo" (geometric) recombination for the ExpDE framework

Usage

recombination_geo(X, M, recpars)

Arguments

X  population matrix (original)
M  population matrix (mutated)
recpars  recombination parameters (see Recombination parameters for details)

Value

Matrix U containing the recombined population
Recombination Parameters

The recpars parameter contains all parameters required to define the recombination. recombination_geo() understands the following fields in recpars:

- alpha: exponent for geometrical recombination.
  Accepts numeric value 0 <= alpha <= 1 or NULL (in which case a random value is chosen for each recombination).

References


Description

Implements the "/lbga" (Linear Breeder Genetic Algorithm) recombination for the ExpDE framework

Usage

recombination_lbga(X, M, ...)

Arguments

X population matrix (original)
M population matrix (mutated)
... optional parameters (unused)

Value

Matrix U containing the recombined population

Warning

This recombination operator evaluates the candidate solutions in M, which adds an extra popsize evaluations per iteration.

References

**recombination_linear**  
*Linear recombination for DE*

### Description
Implements the "/linear" recombination for the ExpDE framework

### Usage

```r
recombination_linear(X, M, ...)
```

### Arguments

- **X**: population matrix (original)
- **M**: population matrix (mutated)
- **...**: optional parameters (unused)

### Value
Matrix $U$ containing the recombined population

### Warning
This recombination operator evaluates $3 \times \text{popsize}$ candidate solutions per iteration of the algorithm. The value of the \textit{nfe} counter and the vector of performance values $G$ are updated in the calling environment.

### References

---

**recombination_mmax**  
*Min Max Arithmetical recombination for DE*

### Description
Implements the "/mmax" (min-max-arithmetical) recombination for the ExpDE framework

### Usage

```r
recombination_mmax(X, M, recpars = list(lambda = NULL))
```
Arguments

\texttt{X} \quad \text{population matrix (original)}

\texttt{M} \quad \text{population matrix (mutated)}

\texttt{recpars} \quad \text{recombination parameters (see Recombination parameters for details)}

Value

Matrix \( U \) containing the recombined population

Warning

This recombination operator evaluates \( 4 \times \text{popsize} \) candidate solutions per iteration of the algorithm. The value of the \( \text{nfe} \) counter and the vector of performance values \( G \) are updated in the calling environment.

Recombination Parameters

The \texttt{recpars} parameter contains all parameters required to define the recombination. \texttt{recombination\_pbest()} understands the following fields in \texttt{recpars}:

- \texttt{\lambda} \text{a} \text{m} \text{ba} \text{d}a : Recombination multiplier.
  
  Optional. Defaults to \texttt{NULL}. Accepts numeric \( 0 < \text{\lambda} < 1 \) or \texttt{NULL} (in which case a random value is independently used for each variable of each recombination pair).

References


\texttt{recombination\_none} \quad \text{NULL recombination for DE}

Description

Implements the "\texttt{/none}" recombination (i.e., no recombination performed) for the ExpDE framework

Usage

\texttt{recombination\_none(X, M, ...)}
Arguments

\( X \)  
population matrix (original)

\( M \)  
population matrix (mutated)

\( \ldots \)  
optional parameters (unused)

Value

The same matrix \( M \) used as an input.

---

**Recombination Parameters**

The \( \text{recpars} \) parameter contains all parameters required to define the recombination. \( \text{recombination_npoint()} \) understands the following fields in \( \text{recpars} \):

- \( N \): cut number points for crossover.
  
  Accepts integer value \( 0 \leq N < n \), where \( n \) is the dimension of the problem; Use \( N = 0 \) or \( N = \text{NULL} \) for randomly choosing a number of cut points. Defaults to \( \text{NULL} \).

References

recombination_onepoint

One-point recombination for DE

Description

Implements the one-point recombination (as used in the Simple GA).

Usage

recombination_onepoint(X, M, recpars = list(K = NULL))

Arguments

- X: population matrix (original)
- M: population matrix (mutated)
- recpars: recombination parameters (see Recombination parameters for details)

Value

Matrix U containing the recombined population

Recombination Parameters

The recpars parameter contains all parameters required to define the recombination. recombination_onepoint() understands the following fields in recpars:

- K: cut point for crossover.
  Accepts integer value $0 \leq K < n$, where $n$ is the dimension of the problem; Use $K = 0$ or $K = \text{NULL}$ for randomly choosing a position for each pair of points.
  Defaults to NULL.

References

recombination_operators

Recombination operators available

Description

List all available recombination operators in the ExpDE package

Usage

recombination_operators()

Value

Character vector with the names of all recombination operator routines

recombination_pbest  p-Best recombination for DE

Description

Implements the "/pbest" (p-Best) recombination for the ExpDE framework

Usage

recombination_pbest(X, M, recpars)

Arguments

X population matrix (original)
M population matrix (mutated)
recpars recombination parameters (see Recombination parameters for details)

Value

Matrix U containing the recombined population

Recombination Parameters

The recpars parameter contains all parameters required to define the recombination. recombination_pbest() understands the following fields in recpars:

- cr: component-wise probability of using the value in M.
  Accepts numeric value \(0 < cr <= 1\).
Recycling sbx

Warning
This routine will search for the iterations counter \( t \), the maximum number of iterations \( \text{stopcrit} \rightarrow \text{maxiter} \), and the performance vector of population \( X(\chi) \) in the parent environment (using parent.frame()). These variables must be defined for \( \text{recombination}_{\text{pbest}}() \) to work.

References

recombination_sbx /sbx recombination for DE

Description
Implements the "\( /\text{sbx} \)" (Simulated Binary) recombination for the ExpDE framework

Usage
recombination_sbx(X, M, recpars)

Arguments
\( X \) population matrix (original)
\( M \) population matrix (mutated)
recpars recombination parameters (see Recombination parameters for details)

Value
Matrix \( u \) containing the recombined population

Recombination Parameters
The \( \text{recompars} \) parameter contains all parameters required to define the recombination. \( \text{recombination}_{\text{sbx}}() \) understands the following field in \( \text{recompars} \):

- \( \eta \) spread factor.
  Accepts numeric value \( \eta > 0 \).

References
recombination_wright    Heuristic Wright recombination for DE

Description

Implements the "/wright" (Heuristic Wright) recombination for the ExpDE framework.

Usage

recombination_wright(X, M, ...)

Arguments

X    population matrix (original)
M    population matrix (mutated)
...  optional parameters (unused)

Value

Matrix U containing the recombined population

Warning

This recombination operator evaluates the candidate solutions in M, which adds an extra popsize evaluations per iteration.

References


selection_standard    Standard selection for DE

Description

Implements the standard selection (greedy) for the ExpDE framework

Usage

selection_standard(X, U, J, G)
Arguments

- **X**: population matrix (original)
- **U**: population matrix (recombined)
- **J**: performance vector for population X
- **G**: performance vector for population U

Value

- list object containing the selected population (Xsel) and its corresponding performance values (Jsel).
## Index

<table>
<thead>
<tr>
<th>Check Stop Criteria</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>create_population</td>
<td>3</td>
</tr>
<tr>
<td>evaluate_population</td>
<td>3</td>
</tr>
<tr>
<td>ExpDE</td>
<td>3, 4</td>
</tr>
<tr>
<td>Mutation Best</td>
<td>5, 8</td>
</tr>
<tr>
<td>Mutation Current To Pbest</td>
<td>5, 9</td>
</tr>
<tr>
<td>Mutation Mean</td>
<td>5, 10</td>
</tr>
<tr>
<td>Mutation None</td>
<td>5, 11</td>
</tr>
<tr>
<td>Mutation Operators</td>
<td>11</td>
</tr>
<tr>
<td>Mutation Rand</td>
<td>5, 12</td>
</tr>
<tr>
<td>Mutation WGI</td>
<td>5, 13</td>
</tr>
<tr>
<td>Print Progress</td>
<td>14</td>
</tr>
<tr>
<td>Recombination Arith</td>
<td>5, 14</td>
</tr>
<tr>
<td>Recombination Bin</td>
<td>5, 15</td>
</tr>
<tr>
<td>Recombination BLXalphaBeta</td>
<td>5, 16</td>
</tr>
<tr>
<td>Recombination Eigen</td>
<td>5, 17</td>
</tr>
<tr>
<td>Recombination Exp</td>
<td>5, 17</td>
</tr>
<tr>
<td>Recombination Geo</td>
<td>5, 18</td>
</tr>
<tr>
<td>Recombination LBGA</td>
<td>5, 19</td>
</tr>
<tr>
<td>Recombination Linear</td>
<td>5, 20</td>
</tr>
<tr>
<td>Recombination MMAX</td>
<td>5, 20</td>
</tr>
<tr>
<td>Recombination None</td>
<td>5, 21</td>
</tr>
<tr>
<td>Recombination Npoint</td>
<td>5, 22</td>
</tr>
<tr>
<td>Recombination Onepoint</td>
<td>5, 23</td>
</tr>
<tr>
<td>Recombination Operators</td>
<td>24</td>
</tr>
<tr>
<td>Recombination Pbest</td>
<td>5, 24</td>
</tr>
<tr>
<td>Recombination SBX</td>
<td>5, 25</td>
</tr>
<tr>
<td>Recombination Wright</td>
<td>5, 26</td>
</tr>
<tr>
<td>Selection Standard</td>
<td>6, 26</td>
</tr>
</tbody>
</table>