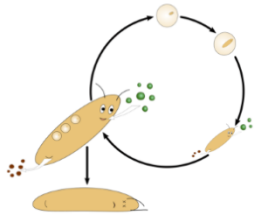


DEB2021 17-28 May 2021
Seventh International Symposium and Thematic School on DEB theory for metabolic organization

MultiCalib4DEB

A toolbox bringing multimodal optimisation for DEB parameters calibration

Juan Francisco Robles [1] Antonio Agüera [2] Manuel Chica [1] Sergio Damas [1]
jfrobles@ugr.es

[1]Department of Computer Science and Artificial Intelligence, Andalusian Research Institute in Data Science and Computational Intelligence, DaSCI, University of Granada

[2]Department of Benthic resources, Norwegian Institute of Marine Research

May 25, 2021

Table of Contents

Table of Contents

Table of Contents

MultiCalib 4DEB

Table of Contents

MultiCalib 4DEB

Table of Contents

Table of Contents

MultiCalib 4DEB

Table of Contents

MultiCalib 4DEB

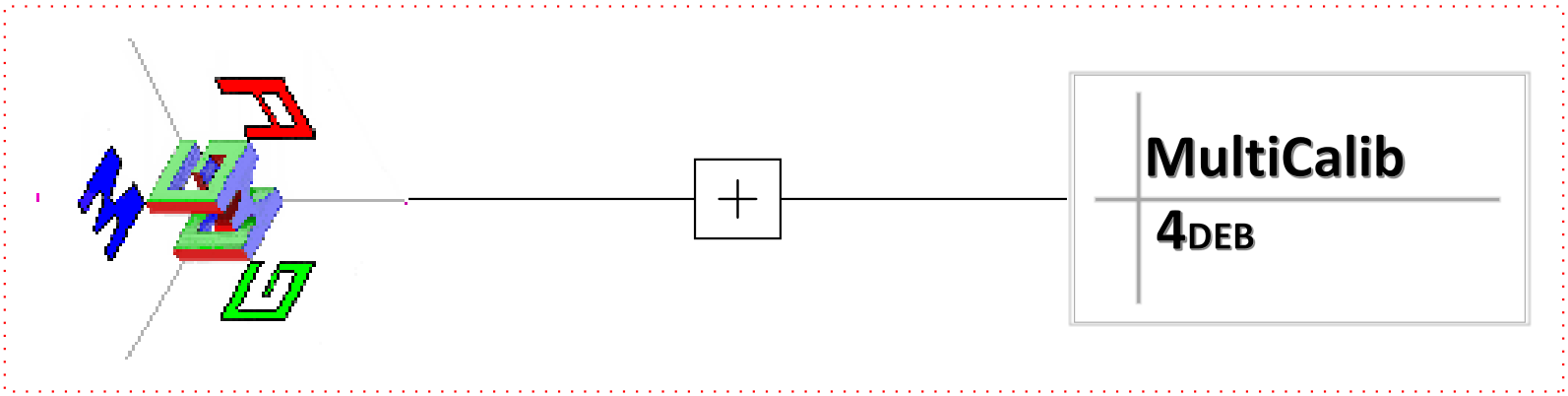
Table of Contents

MultiCalib 4DEB

Table of Contents

MultiCalib 4DEB

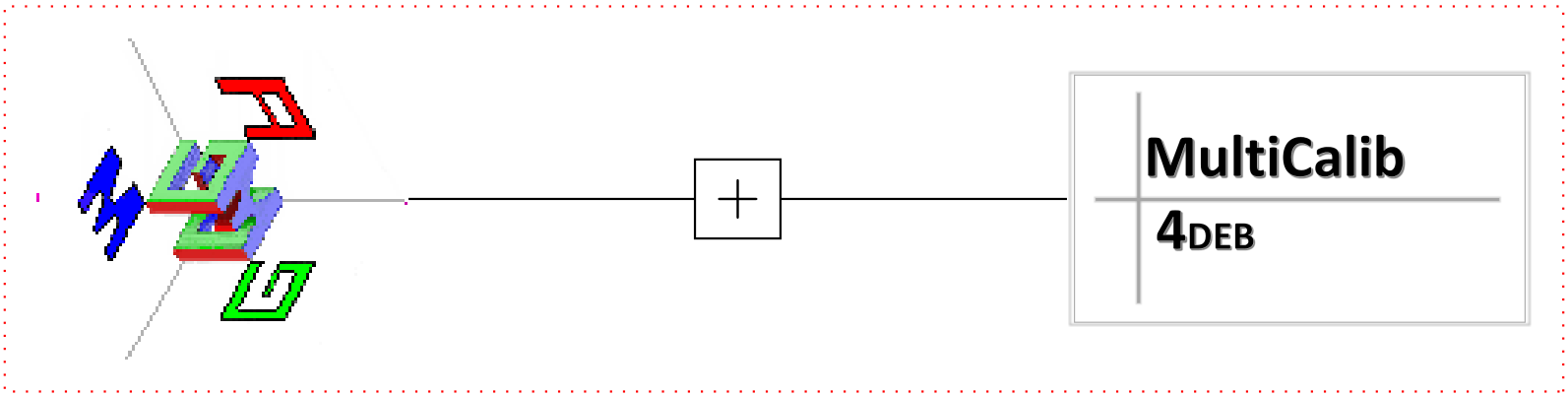
- A MATLAB-based toolbox for the calibration of DEB species



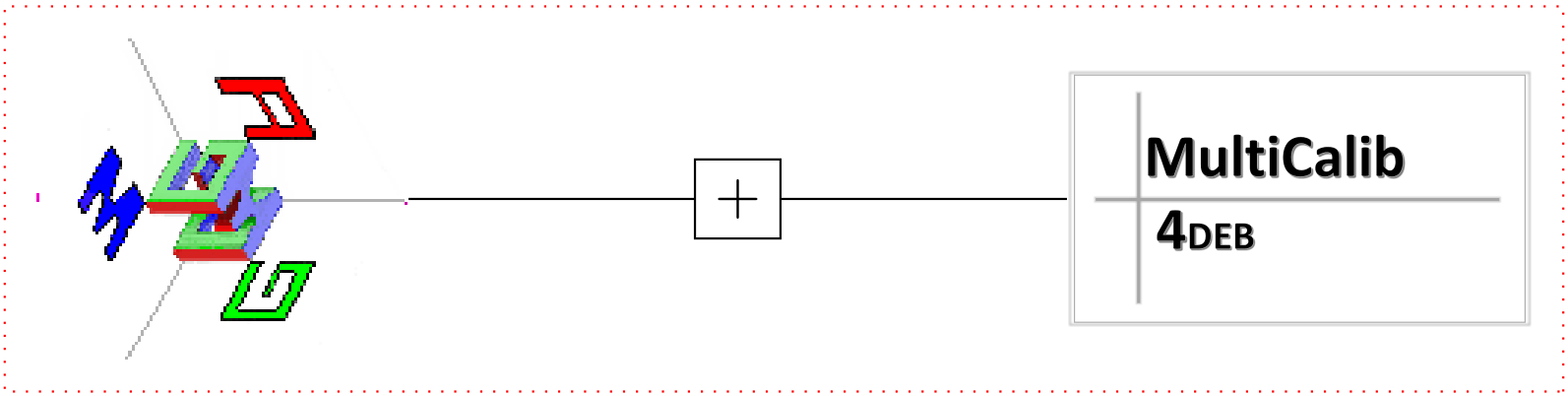
About MultiCalib4DEB

What is MultiCalib4DEB?

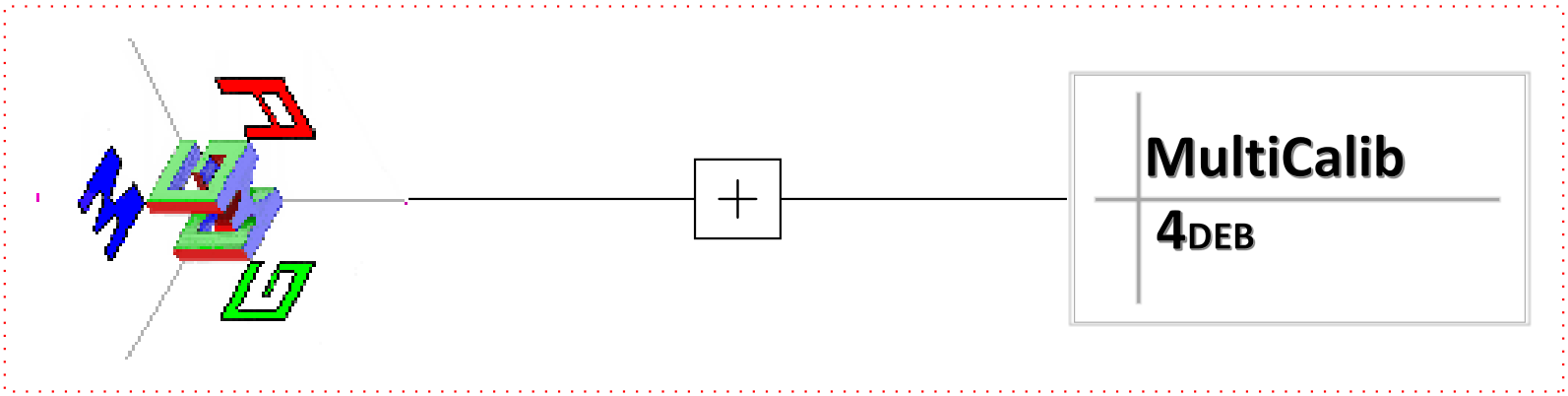
- A MATLAB-based toolbox for the calibration of DEB species
- Easily integrated with the existing DEBtool toolbox



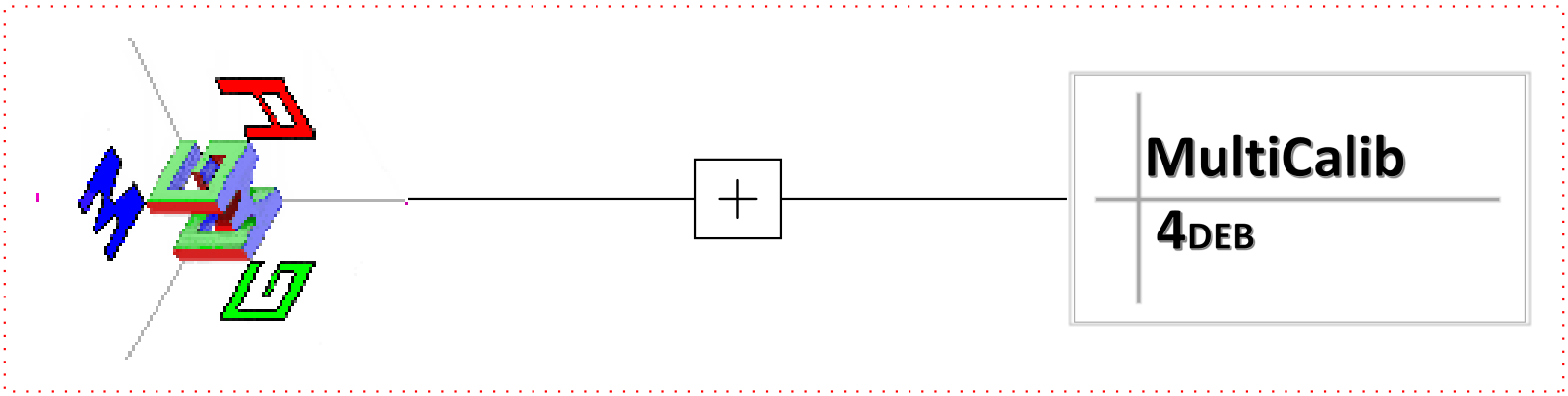
- A MATLAB-based toolbox for the calibration of DEB species
- Easily integrated with the existing DEBtool toolbox
- Works together with the DEBtool filters, loss functions, and other methods



- A MATLAB-based toolbox for the calibration of DEB species
- Easily integrated with the existing DEBtool toolbox
- Works together with the DEBtool filters, loss functions, and other methods
- A calibration framework based on multimodal evolutionary optimisation



- A MATLAB-based toolbox for the calibration of DEB species
- Easily integrated with the existing DEBtool toolbox
- Works together with the DEBtool filters, loss functions, and other methods
- A calibration framework based on multimodal evolutionary optimisation
- A validation tool for DEB modellers



About MultiCalib4DEB

What is the objective of MultiCalib4DEB?

- To enhance the DEBtool's calibration framework

About MultiCalib4DEB

What is the objective of MultiCalib4DEB?

MultiCalib 4DEB

- To enhance the DEBtool's calibration framework
- To return different calibration options both optimal and diverse

About MultiCalib4DEB

What is the objective of MultiCalib4DEB?

MultiCalib 4DEB

- To enhance the DEBtool's calibration framework
- To return different calibration options both optimal and diverse
- To ease the calibration process and the model validation

About MultiCalib4DEB

What is the objective of MultiCalib4DEB?

MultiCalib 4DEB

- To enhance the DEBtool's calibration framework
- To return different calibration options both optimal and diverse
- To ease the calibration process and the model validation
- To increase the tools DEB users have for the calibration/validation of their DEB models

Module name	Description
charts	Multiple plots for the validation of the MultiCalib4DEB calibration results such as heatmaps and scatter plots
configuration	Contains different calibration options
examples	Calibration examples for some DEB species and examples for the visualisation and statistical report modules.
functions	Auxiliary functions for the algorithms, plots, and statistics
methods	The algorithms considered for calibration in the MultiCalib4DEB toolbox.
results	Functionality related to the export of results
statistics	Functions which handle the generation of statistical reports from the calibration results
utils	Utilities for saving and generating reports from calibration results

Table of Contents

MultiCalib 4DEB

Multimodal evolutionary optimisation algorithms [1, 2] (MMEAs) combine Evolutionary Algoritms [3, 4] (EAs) and Niche-preserving techniques [2] to find multiple global and local optima (rather than a single solution) of a function simultaneously. From this point:

Multimodal evolutionary optimisation algorithms [1, 2] (MMEAs) combine Evolutionary Algoritms [3, 4] (EAs) and Niche-preserving techniques [2] to find multiple global and local optima (rather than a single solution) of a function simultaneously. From this point:

- The user can better understand the different optimal solutions in the search space

Multimodal evolutionary optimisation algorithms [1, 2] (MMEAs) combine Evolutionary Algoritms [3, 4] (EAs) and Niche-preserving techniques [2] to find multiple global and local optima (rather than a single solution) of a function simultaneously. From this point:

- The user can better understand the different optimal solutions in the search space
- The current solution may be switched to another one maintaining the optimal system performance

Multimodal evolutionary optimisation algorithms [1, 2] (MMEAs) combine Evolutionary Algoritms [3, 4] (EAs) and Niche-preserving techniques [2] to find multiple global and local optima (rather than a single solution) of a function simultaneously. From this point:

- The user can better understand the different optimal solutions in the search space
- The current solution may be switched to another one maintaining the optimal system performance

The MMEAs family has grown through the years and is widely used in a large number of optimisation problems

Evolutionary computation [3] have its origin in evolution theories and Darwinian natural selection. It provides computational models for search and optimisation problems.

Evolutionary computation [3] have its origin in evolution theories and Darwinian natural selection. It provides computational models for search and optimisation problems. EAs generally:

Evolutionary computation [3] have its origin in evolution theories and Darwinian natural selection. It provides computational models for search and optimisation problems. EAs generally:

- Adapt a population of candidate solutions to a problem

Evolutionary computation [3] have its origin in evolution theories and Darwinian natural selection. It provides computational models for search and optimisation problems. EAs generally:

- Adapt a population of candidate solutions to a problem
- Apply a random selective process based on the quality of the generated solutions (measured according to a fitness function)

Evolutionary computation [3] have its origin in evolution theories and Darwinian natural selection. It provides computational models for search and optimisation problems. EAs generally:

- Adapt a population of candidate solutions to a problem
- Apply a random selective process based on the quality of the generated solutions (measured according to a fitness function)
- Alter the selected solutions using crossover and/or mutation operators

Evolutionary computation [3] have its origin in evolution theories and Darwinian natural selection. It provides computational models for search and optimisation problems. EAs generally:

- Adapt a population of candidate solutions to a problem
- Apply a random selective process based on the quality of the generated solutions (measured according to a fitness function)
- Alter the selected solutions using crossover and/or mutation operators
- Use the new solutions generated to replace those of the current population

Evolutionary computation [3] have its origin in evolution theories and Darwinian natural selection. It provides computational models for search and optimisation problems. EAs generally:

- Adapt a population of candidate solutions to a problem
- Apply a random selective process based on the quality of the generated solutions (measured according to a fitness function)
- Alter the selected solutions using crossover and/or mutation operators
- Use the new solutions generated to replace those of the current population

Due to their population-based approaches, EAs are able to detect multiple solutions within a population in a single run

Multimodal optimization for DEB III

Example of the evolutionary algorithms search

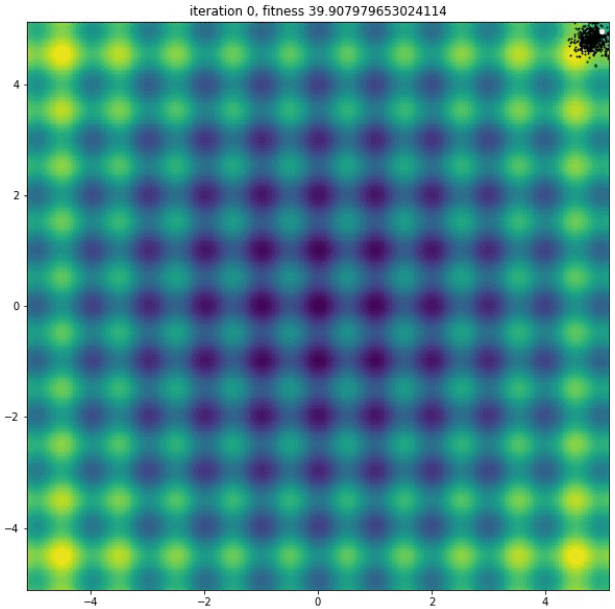


Figure: Example of EA exploration through search space.
Source: https://bit.ly/2SbENI0

For their part, niche-preserving techniques:

For their part, niche-preserving techniques:

- Are division mechanisms to produce different sub-populations exploring different search space regions (niches) according to the similarity of the individuals [1].

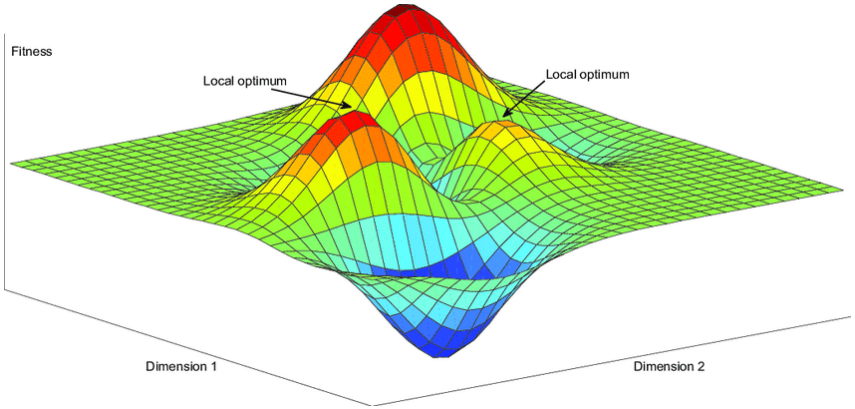


Figure: Example of a multimodal fitness landscape.
Source: https://bit.ly/3u1zCl6

For their part, niche-preserving techniques:

- Are division mechanisms to produce different sub-populations exploring different search space regions (niches) according to the similarity of the individuals [1].
- Promote and maintain the formation of multiple stable sub-populations within a single population to locate multiple globally optimal or sub-optimal solutions

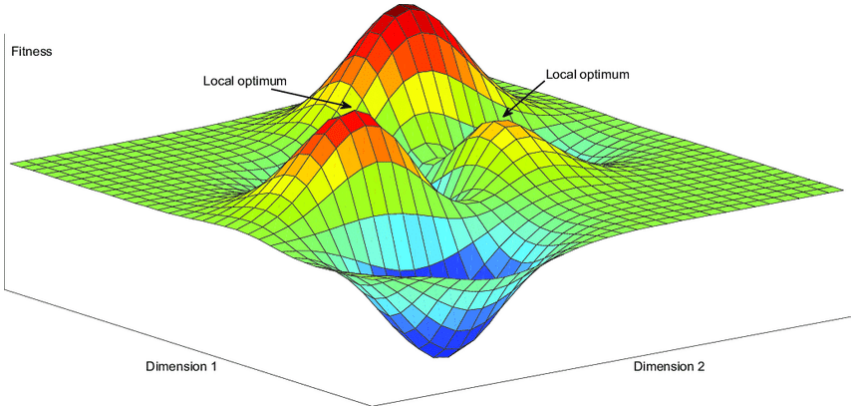


Figure: Example of a multimodal fitness landscape.
Source: https://bit.ly/3u1zCl6

Two MMEAs has been selected from the vast MMEA family as the core calibration algorithms of MultiCalib4DEB:

Two MMEAs has been selected from the vast MMEA family as the core calibration algorithms of MultiCalib4DEB:

- Success-history based adaptive differential evolution (SHADE) [5]

Two MMEAs has been selected from the vast MMEA family as the core calibration algorithms of MultiCalib4DEB:

- Success-history based adaptive differential evolution (SHADE) [5]
- Linear Population Size Reduction Success-history based adaptive differential evolution (L-SHADE) [6]

Two MMEAs has been selected from the vast MMEA family as the core calibration algorithms of MultiCalib4DEB:

- Success-history based adaptive differential evolution (SHADE) [5]
- Linear Population Size Reduction Success-history based adaptive differential evolution (L-SHADE) [6]

SHADE and L-SHADE are multimodal extensions of the differential evolution (DE) algorithm [7], which has been widely used in practical and theoretical optimization problems [8].

Two MMEAs has been selected from the vast MMEA family as the core calibration algorithms of MultiCalib4DEB:

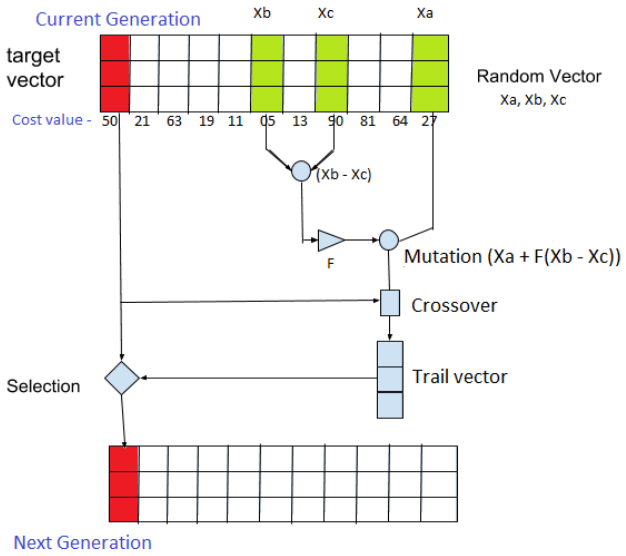
- Success-history based adaptive differential evolution (SHADE) [5]
- Linear Population Size Reduction Success-history based adaptive differential evolution (L-SHADE) [6]

SHADE and L-SHADE are multimodal extensions of the differential evolution (DE) algorithm [7], which has been widely used in practical and theoretical optimization problems [8].

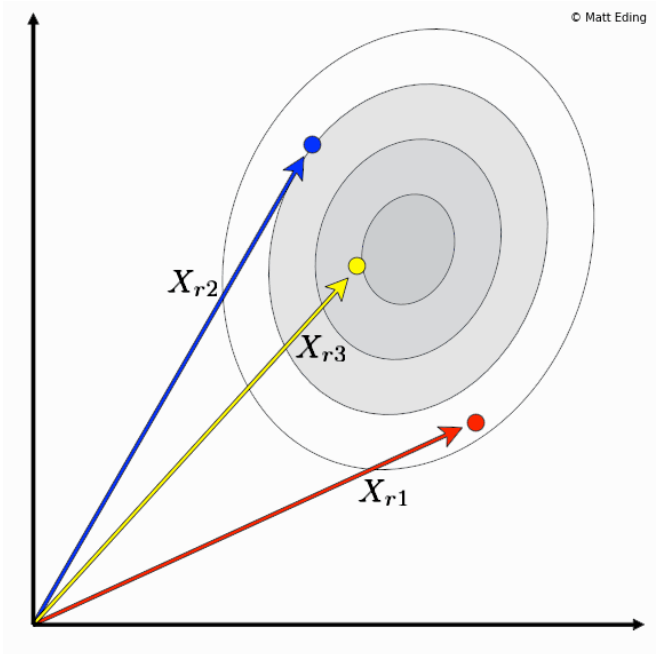
DE, SHADE, and L-SHADE have been successfully applied in areas such as economy [9], energy [10, 11, 12], mathematics [13], and pharmacy [14] and their use increases through the years [15].

The algorithms behind MultiCalib4DEB II

Example of differential evolution functioning



(a) Example of differential evolution combination schema. Source: https://bit.ly/3uYhdxa



(b) Example of differential evolution search. Source: https://bit.ly/3hwytWf

Table of Contents

MultiCalib 4DEB

MultiCalib4DEB is designed to be an easy to use tool. In this way, the step which are necessary to start working with the toolbox are the following:

1. Dowload the latest version of the MultiCalib4DEB from here
2. Decompress MultiCalib4DEB and then copy and paste the entire folder into the DEBtool toolbox "lib" directory
3. Load the MultiCalib4DEB modules in the MATLAB environment

MultiCalib4DEB mimics the DEBtool calibration process to reduce the time a user needs to start working with it. In this way, launching a calibration process involves:

MultiCalib4DEB mimics the DEBtool calibration process to reduce the time a user needs to start working with it. In this way, launching a calibration process involves:

- Loading an species to calibrate from the DEBtool species database

MultiCalib4DEB mimics the DEBtool calibration process to reduce the time a user needs to start working with it. In this way, launching a calibration process involves:

- Loading an species to calibrate from the DEBtool species database
- Selecting the parameters to calibrate and their initialization

MultiCalib4DEB mimics the DEBtool calibration process to reduce the time a user needs to start working with it. In this way, launching a calibration process involves:

- Loading an species to calibrate from the DEBtool species database
- Selecting the parameters to calibrate and their initialization
- Configuring some calibration options (for the calibration algorithm, the calibration results, and the final calibration report)

MultiCalib4DEB is flexible and allows the user to set different calibration settings such as:

The MultiCalib4DEB options I

Calibration options

MultiCalib4DEB is flexible and allows the user to set different calibration settings such as:

- 'method': Select the calibration method

The MultiCalib4DEB options I

Calibration options

MultiCalib4DEB is flexible and allows the user to set different calibration settings such as:

- 'method': Select the calibration method
- 'num_results': Define the number of result to generate

The MultiCalib4DEB options I

Calibration options

MultiCalib4DEB is flexible and allows the user to set different calibration settings such as:

- 'method': Select the calibration method
- 'num_results': Define the number of result to generate
- 'gen_factor': Generate the initial population applying a factor to the species values

The MultiCalib4DEB options I

Calibration options

MultiCalib4DEB is flexible and allows the user to set different calibration settings such as:

- 'method': Select the calibration method
- 'num_results': Define the number of result to generate
- 'gen_factor': Generate the initial population applying a factor to the species values
- 'bounds_from_ind': Take the initial values from the pars_init file or from the pseudodata

MultiCalib4DEB is flexible and allows the user to set different calibration settings such as:

- 'method': Select the calibration method
- 'num_results': Define the number of result to generate
- 'gen_factor': Generate the initial population applying a factor to the species values
- 'bounds_from_ind': Take the initial values from the pars_init file or from the pseudodata
- 'add_initial': To add the initial parameter values into the algorithm's initialisation

MultiCalib4DEB is flexible and allows the user to set different calibration settings such as:

- 'method': Select the calibration method
- 'num_results': Define the number of result to generate
- 'gen_factor': Generate the initial population applying a factor to the species values
- 'bounds_from_ind': Take the initial values from the pars_init file or from the pseudodata
- 'add_initial': To add the initial parameter values into the algorithm's initialisation
- 'refine_best': To refine the best individual

- 'max_fun_evals': To define a maximum number of loss function evaluation as the stop criterion

- 'max_fun_evals': To define a maximum number of loss function evaluation as the stop criterion
- 'max_calibration_time': To stop the calibration process with a given time

- 'max_fun_evals': To define a maximum number of loss function evaluation as the stop criterion
- 'max_calibration_time': To stop the calibration process with a given time
- 'num_runs': To perform more than a single optimisation (for stochastic tests over the calibration engine)

- 'max_fun_evals': To define a maximum number of loss function evaluation as the stop criterion
- 'max_calibration_time': To stop the calibration process with a given time
- 'num_runs': To perform more than a single optimisation (for stochastic tests over the calibration engine)
- 'ranges': To define maximum and minimum values for the parameters selected for calibration

The MultiCalib4DEB options III

Runtime and report options

- Runtime options

The MultiCalib4DEB options III

Runtime and report options

- Runtime options
  - 'verbose': To display information about the calibration time (time accomplished, function evaluations performed, best results found, ...) on screen

- Runtime options
  - 'verbose': To display information about the calibration time (time accomplished, function evaluations performed, best results found, ...) on screen
  - 'verbose_options': Some options for the 'verbose' option such as the number of solutions to display on screen

The MultiCalib4DEB options III

Runtime and report options

- Runtime options
  - 'verbose': To display information about the calibration time (time accomplished, function evaluations performed, best results found, ...) on screen
  - 'verbose_options': Some options for the 'verbose' option such as the number of solutions to display on screen
- Report options

The MultiCalib4DEB options III

Runtime and report options

- Runtime options
  - 'verbose': To display information about the calibration time (time accomplished, function evaluations performed, best results found, ...) on screen
  - 'verbose_options': Some options for the 'verbose' option such as the number of solutions to display on screen
- Report options
  - 'results_outputs': To generate a report from the calibration results

The MultiCalib4DEB options III

Runtime and report options

- Runtime options
  - 'verbose': To display information about the calibration time (time accomplished, function evaluations performed, best results found, ...) on screen
  - 'verbose_options': Some options for the 'verbose' option such as the number of solutions to display on screen
- Report options
  - 'results_outputs': To generate a report from the calibration results
  - 'results_filename': To define the results filename

The MultiCalib4DEB options III

Runtime and report options

- Runtime options
  - 'verbose': To display information about the calibration time (time accomplished, function evaluations performed, best results found, ...) on screen
  - 'verbose_options': Some options for the 'verbose' option such as the number of solutions to display on screen
- Report options
  - 'results_outputs': To generate a report from the calibration results
  - 'results_filename': To define the results filename

A complete list with the MultiCalib4DEB options is available here

The MultiCalib4DEB options VI

A calibration file example

```
close all;
global pets

pets = {'Clarias_gariepinus'}; % The species to calibrate
check_my_pet(pets); % Check species consistence

% Default DEBtool_M options (filters, loss function, ...)
estim_options('default');

% Setting default calibration options
calibration_options('default');
calibration_options('method', 'mm1'); % Calibrate with SHADE
% Stop in 10,000 evaluations
calibration_options('max_fun_evals', 10000);
% Take the parameters from pseudo data
calibration_options('bounds_from_ind', 1);
% Add the initial values into SHADE initialization
calibration_options('add_initial', 1);
calibration_options('verbose', 0); % Deactivate verbose
[best, info, out, best_fvalues] = calibrate; % Calibrate!
```

When the calibration process finishes, MultiCalib4DEB saves the results in a file which contains the information below:

Field name	**Description**
NP	The number of solutions returned after the calibration process
pop	The set of solutions returned after the calibration process. It contains *NP* solutions, each with a different set of calibration parameters
funvalues	The loss function values for each solution
parnames	The names of the calibration parameters. Useful for working with the statistics and visualisation modules
results	MatLab struct which contains information about all the calibration solutions such as the "pair", "metaPar", or "data" files in DEBtool format to work with one or a set of results after the calibration process or to generate reports by using the **results** module of MultiCalib4DEB

Table of Contents

From the MultiCalib4DEB results it is possible to perform statistical by using the functions in the statistics module. The different statistical methods which are available in MultiCalib4DEB are:

- Statistical functions over loss function values
  - Cardinality
  - average
  - standard deviation
  - Average distance
- Statistical functions over parameter values
  - Average
  - Standard deviation
  - Spread
  - Minimum and maximum
  - Kurtosis, skewness, and bimodal coefficient

After the calibration process II

Example of an statistical report

This is an example of the output which is obtained by using the statistical report method:

statistics_report

1x1 struct with 2 fields

Field	Value
fitness	*1x1 struct*
parameters	*1x1 struct*

statistics_report.fitness

Field	Value
cardinality	300
mean	4.4599
std	1.9406
average_dista...	0.2807

statistics_report.parameters

Field	Value
mean	*1x11 double*
std	*1x11 double*
spread	*1x11 double*
minimum	*1x11 double*
maximum	*1x11 double*
kurtosis	*1x11 double*
skewness	*1x11 double*
bimodal_coeffi...	*1x11 double*

statistics_report.parameters.kurtosis

	1	2	3	4	5	6	7	8	9	10	11
1	2.1378	4.7315	7.4109	4.7227	8.2446	2.1144	2.5224	2.4693	2.4291	2.4461	2.6414

After the calibration process III

Visual analyses with MultiCalib4DEB

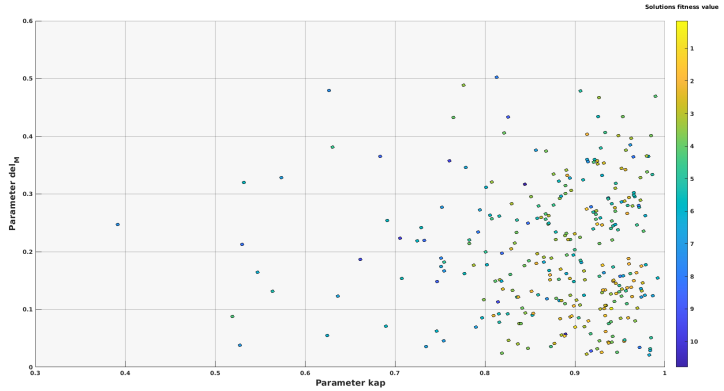
Visual analyses are also available by using the charts module. The charts which are available in the current version of MultiCalib4DEB are:

- Visualisation reports
  - Heat maps
  - Scatter plots
  - Prediction plots

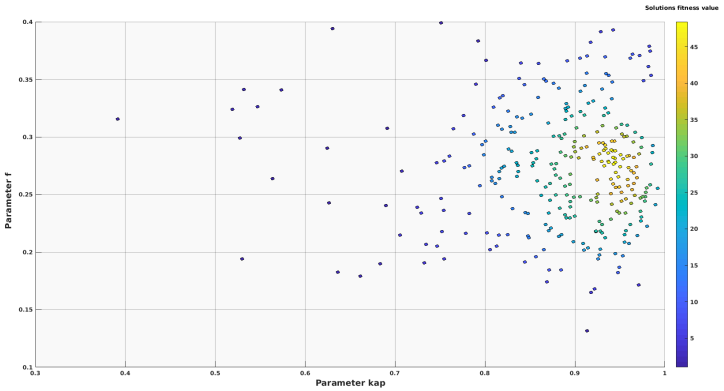
After the calibration process IV

Example of a visual report I

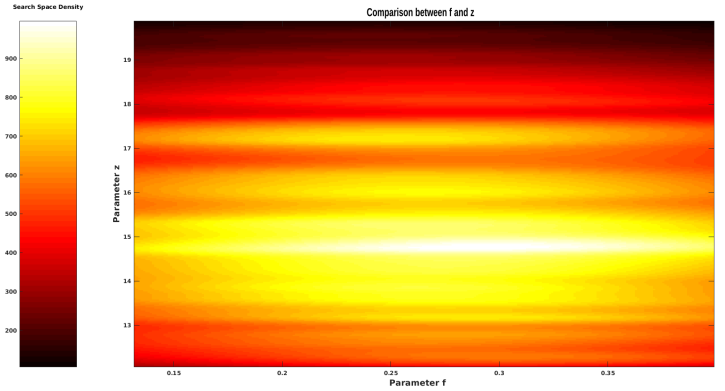
MultiCalib 4DEB



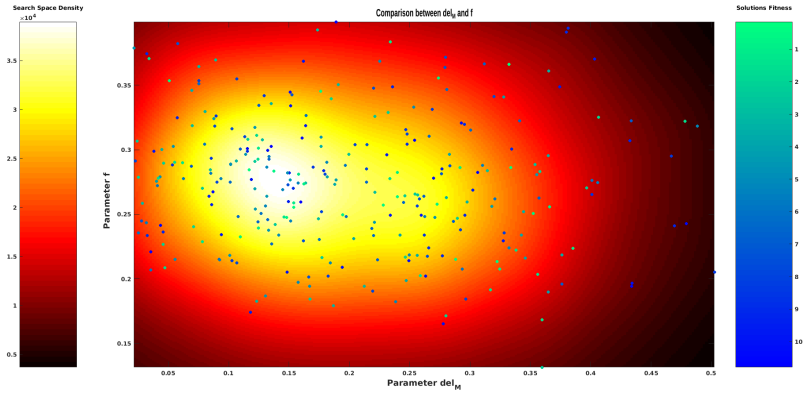
(a) Scatter plot



(b) Density scatter plot



(c) Heatmap



(d) Scatter and heatmap

After the calibration process V

Example of a visual report II

MultiCalib 4DEB

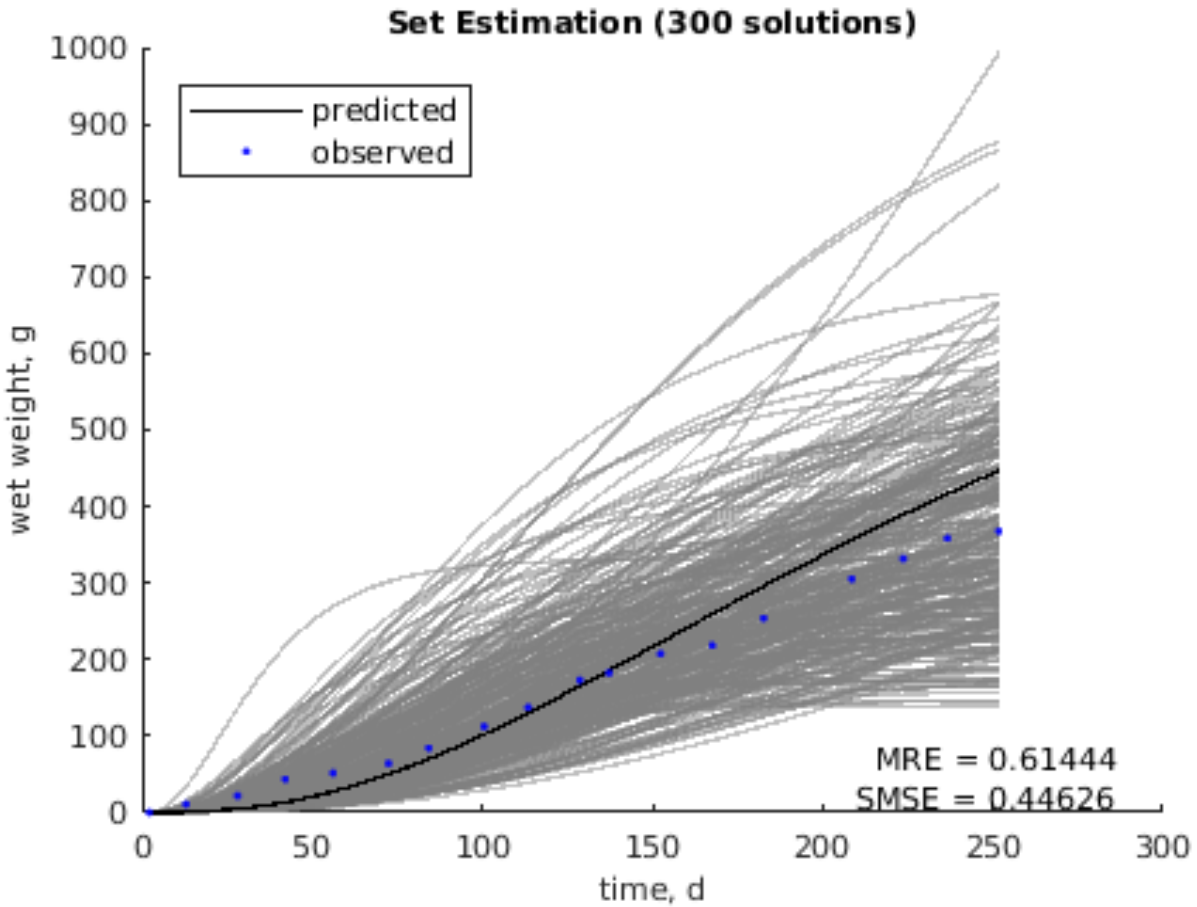


Figure: Prediction plot

Table of Contents

MultiCalib 4DEB

Comparing the DEBtool and the MultiCalib4DEB calibration results

What we compare and what we obtain

species	Type	Num. Params.	Start loss function	DEBtool			MultiCalib4DEB		
				Final loss function	SMSE	MRE	Final loss function	SMSE	MRE
Clarias gariepinus	abj	11	0.15307	0.15293	**0.013**	**0.019**	**0.15278**	0.014	0.020
Heterobranchus longifilis	abj	11	0.12205	**0.11832**	0.013	0.016	0.11833	0.013	0.016
Dipodomys deserti	stx	10	0.2326	0.23258	0.018	0.021	**0.22345**	**0.017**	**0.019**
Dipodomys herrmanni	stx	10	0.32892	0.3289	0.024	**0.042**	**0.32374**	**0.023**	0.044
Dipodomys merriami	stx	10	0.31671	0.31354	0.017	0.018	**0.31018**	**0.015**	**0.016**
Lepus timidus	stx	10	0.30677	0.30663	0.013	0.013	**0.30285**	**0.007**	**0.008**
Cyclops vicinus	abp	9	6.3819	0.48484	0.198	0.171	0.48484	0.198	0.171
Lobatus gigas	abj	9	0.062286	0.062286	0.036	0.038	0.062286	0.036	0.038
Porcellio scaber	std	12	0.18872	0.098775	0.052	0.047	0.098775	0.052	0.047
Magallana gigas	asj	18	16.7519	15.527	0.405	0.463	**14.88372**	**0.397**	**0.455**
Pleurobrachia bachei	abj	12	1.3614	1.3503	**0.182**	0.24	**1.3142**	0.184	**0.236**
Homo sapiens	stx	12	0.28001	0.27999	0.051	0.057	**0.27944**	**0.05**	**0.055**
Asterias rubens	abj	15	0.98047	0.84429	0.143	0.143	**0.84428**	0.143	0.143

Table: Comparison between DEBtool and MultiCali4DEB calibration results for different species.

Table of Contents

The main conclusion coming from the experiments and the analyses we have done by using the MultiCalib4DEB as the calibration engine of DEBtools are:

The main conclusion coming from the experiments and the analyses we have done by using the MultiCalib4DEB as the calibration engine of DEBtools are:

- Completely integrated with the current DEBtool toolbox

The main conclusion coming from the experiments and the analyses we have done by using the MultiCalib4DEB as the calibration engine of DEBtools are:

- Completely integrated with the current DEBtool toolbox
- Customisable and easy to use

The main conclusion coming from the experiments and the analyses we have done by using the MultiCalib4DEB as the calibration engine of DEBtools are:

- Completely integrated with the current DEBtool toolbox
- Customisable and easy to use
- Efficient and powerful calibration engine based on multimodal evolutionary optimisation algorithms

The main conclusion coming from the experiments and the analyses we have done by using the MultiCalib4DEB as the calibration engine of DEBtools are:

- Completely integrated with the current DEBtool toolbox
- Customisable and easy to use
- Efficient and powerful calibration engine based on multimodal evolutionary optimisation algorithms
- Returns both diverse and optimal calibration solutions from where to better explore the species parameters

The main conclusion coming from the experiments and the analyses we have done by using the MultiCalib4DEB as the calibration engine of DEBtools are:

- Completely integrated with the current DEBtool toolbox
- Customisable and easy to use
- Efficient and powerful calibration engine based on multimodal evolutionary optimisation algorithms
- Returns both diverse and optimal calibration solutions from where to better explore the species parameters
- Allows to perform statistical and visual analyses easily from the calibration results

The main conclusion coming from the experiments and the analyses we have done by using the MultiCalib4DEB as the calibration engine of DEBtools are:

- Completely integrated with the current DEBtool toolbox
- Customisable and easy to use
- Efficient and powerful calibration engine based on multimodal evolutionary optimisation algorithms
- Returns both diverse and optimal calibration solutions from where to better explore the species parameters
- Allows to perform statistical and visual analyses easily from the calibration results
- Between a 1% and a 5% better results both for loss function, SMSE, and MRE than DEBtools

Table of Contents

MultiCalib 4DEB

Questions

Table of Contents

MultiCalib 4DEB

References I

[1] David E Goldberg, Jon Richardson, et al. Genetic algorithms with sharing for multimodal function optimization. pages 41–49, 1987.

[2] Swagatam Das, Sayan Maity, Bo-Yang Qu, and P.N. Suganthan. Real-parameter evolutionary multimodal optimization — a survey of the state-of-the-art. *Swarm and Evolutionary Computation*, 1(2): 71–88, 2011. ISSN 2210-6502. doi: https://doi.org/10.1016/j.swevo.2011.05.005. URL https://www.sciencedirect.com/science/article/pii/S221065021100023X.

[3] Thomas Back. *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford university press, 1996.

References II

[4] Holger R Maier, Saman Razavi, Zoran Kapelan, L Shawn Matott, J Kasprzyk, and Bryan A Tolson. Introductory overview: Optimization using evolutionary algorithms and other metaheuristics. *Environmental modelling & software*, 114:195–213, 2019.

[5] Ryoji Tanabe and Alex Fukunaga. Success-history based parameter adaptation for differential evolution. pages 71–78, 2013.

[6] Ryoji Tanabe and Alex S Fukunaga. Improving the search performance of shade using linear population size reduction. pages 1658–1665, 2014.

[7] Rainer Storn and Kenneth Price. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4):341–359, 1997.

References III

[8] Swagatam Das and Ponnuthurai Nagaratnam Suganthan. Differential evolution: A survey of the state-of-the-art. *IEEE transactions on evolutionary computation*, 15(1):4–31, 2010.

[9] RK Jana, Indranil Ghosh, and Debojyoti Das. A differential evolution-based regression framework for forecasting bitcoin price. *Annals of Operations Research*, pages 1–26, 2021.

[10] Partha P Biswas, Ponnuthurai N Suganthan, and Gehan AJ Amaratunga. Optimal placement of wind turbines in a windfarm using l-shade algorithm. In *2017 IEEE Congress on Evolutionary Computation (CEC)*, pages 83–88. IEEE, 2017.

References IV

[11] Shangce Gao, Kaiyu Wang, Sichen Tao, Ting Jin, Hongwei Dai, and Jiujun Cheng. A state-of-the-art differential evolution algorithm for parameter estimation of solar photovoltaic models. *Energy Conversion and Management*, 230:113784, 2021. ISSN 0196-8904. doi: https://doi.org/10.1016/j.enconman.2020.113784. URL https://www.sciencedirect.com/science/article/pii/S0196890420313078.

[12] Viet-Phu Tran, Giang T.T. Phan, Van-Khanh Hoang, Haidang Phan, Nhat-Duc Hoang, and Hoai-Nam Tran. Success-history based adaptive differential evolution method for optimizing fuel loading pattern of vver-1000 reactor. *Nuclear Engineering and Design*, 377: 111125, 2021. ISSN 0029-5493. doi: https://doi.org/10.1016/j.nucengdes.2021.111125. URL https://www.sciencedirect.com/science/article/pii/S0029549321000777.

References V

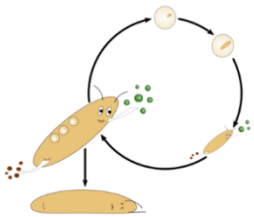
[13] K.B. Cantún-Avila, D. González-Sánchez, S. Díaz-Infante, and F. Peñuñuri. Optimizing functionals using differential evolution. *Engineering Applications of Artificial Intelligence*, 97:104086, 2021. ISSN 0952-1976. doi: https://doi.org/10.1016/j.engappai.2020.104086. URL https://www.sciencedirect.com/science/article/pii/S0952197620303407.

[14] Manjit Kaur, Dilbag Singh, and Vijay Kumar. Drug synergy prediction using dynamic mutation based differential evolution. *Current Pharmaceutical Design*, 2020.

[15] Swagatam Das, Sankha Subhra Mullick, and Ponnuthurai N Suganthan. Recent advances in differential evolution–an updated survey. *Swarm and Evolutionary Computation*, 27:1–30, 2016.

Thank you for your atention

UNIVERSIDAD DE GRANADA



DEB2021 17-28 May 2021

Seventh International Symposium and Thematic School on DEB theory for metabolic organization