

A scalable, black-box hybrid genetic algorithm for continuous multimodal optimization in moderate dimensions

K. Johannsen · N. Goris · B. Jensen · J. Tjiputra

Received: date / Accepted: date

Abstract Optimization problems can be found in many areas of science and technology. Often, not only the global optimum, but also a (larger) number of near-optima are of interest. This gives rise to so-called multimodal optimization problems. In most of the cases, the number and quality of the optima is unknown and assumptions on the objective functions cannot be made. In this paper, we focus on continuous, unconstrained optimization in moderately high dimensional continuous spaces ($d \leq 10$). We present a scalable algorithm with virtually no parameters, which performs well for general objective functions (non-convex, discontinuous). It is based on two well-established algorithms (CMA-ES, deterministic crowding). Novel elements of the algorithm are the detection of seed points for local searches and collision avoidance, both based on nearest neighbors, and a strategy for semi-sequential optimization to realize scalability. The performance of the proposed algorithm is numerically evaluated on the CEC2013 niching benchmark suite for 1 – 20 dimensional functions and a 9 dimensional real-world problem from constraint optimization in climate research. The algorithm shows good performance on the CEC2013 benchmarks and

falls only short on higher dimensional and strongly isotropic problems. In case of the climate related problem, the algorithm is able to find a high number (> 150) of optima, which are of relevance to climate research. The proposed algorithm does not require special configuration for the optimization problems considered in this paper, i.e. it shows good black-box behavior.

Keywords Black-box · continuous optimization · hybrid genetic algorithms · multimodal · niching · parameter-free

1 Introduction

Most of real-world optimization problems have more than one optimal or near optimal solution. Such problems are called multimodal optimization (MMO) problems. They are of great importance in practice, for engineers to understand problems better, for decision makers to make appropriate decisions. In most of the practical cases, the number and quality of optima or near-optima are not known. There is therefore a need for black-box algorithms, which do not require any a priori information about the optimization problem (Audit and Hare, 2017). Furthermore, in order to avoid repeated numerical investigations, it is also highly desirable to have algorithms which do not require (complicated) configuration, ideally none.

Many different strategies have been designed to solve multimodal problems. We will not give an overview, the field of possible strategies is simply too wide. For an overview, we refer to (Li et al., 2016; Preuss, 2015). In this work, we focus on evolutionary algorithms (EA), which are among the well-suited strategies for MMO (Preuss, 2015). Within the EAs, our choice are genetic algorithms (GA). In the context of MMO, GAs usually

✉ K. Johannsen
E-mail: kljo@norceresearch.no

N. Goris
E-mail: nago@norceresearch.no

B. Jensen
E-mail: bjje@norceresearch.no

J. Tjiputra
E-mail: jetj@norceresearch.no

NORCE Norwegian Research Centre AS, Nygårdsgaten
112, 5008 Bergen, Norway

employ diversity preserving methods, so-called niching algorithms. Among them, the most famous ones are crowding (Jong, 1975) and fitness sharing (Goldberg and Richardson, 1987). Later, these strategies have been modified and refined and new ones have been invented, see (Mahfoud, 1995; Mengshoel and Goldberg, 2008; Sareni and Krahenbuhl, 1998; Smith et al., 1993). Similar strategies for other EAs, like particle swarm optimization (PSO) or differential evolution (DE) have been developed.

In order to take advantage of advanced convergence performance of especially designed core algorithms (local solvers), often a hybrid strategy is adopted (Goldberg and Vössner, 1999). In a first stage, a niching algorithm detects the niches and in a second stage, a core algorithm approximates the local or global optimum. We adopt this strategy. In order to scale to in principle any number of optima (Kronfeld and Zell, 2010), we use the niching algorithm in semi-sequential mode with a linearly increasing population.

The aim of our development is the design of a virtually parameter-free algorithm. We therefore choose the components of our algorithm such that good generic choices are possible for them.

The remainder of this paper is organized as follows. In Section 2 we present the overall design of the algorithm as well as its parts. For well-know algorithms we discuss their configuration, for new developments more details are given. In Section 3 we report on numerical experiments. We present results on the CEC2013 niching benchmark problems (Li et al., 2013) and of an optimization problem from climate research (constraint optimization) (Goris et al., 2021). A discussion of the results is given in Section 4 and conclusions are presented in Section 5.

For the remainder of the paper, we will assume that the optimization problem is posed as a maximization problem, and will consequently talk about maxima instead of optima and the maximization problem instead of the optimization problem.

2 The proposed algorithm

In this work we present an optimization algorithm for real-valued multimodal optimization problems in $d \geq 1$ dimensions, which is essentially parameter-free. Only the maximum number of function evaluations needs to be specified.

Let $X \in \mathbb{R}^d$, $d \geq 1$ be the real-valued search space and $f : X \rightarrow \mathbb{R}$ a real-valued objective function to be maximized. The solution to the multimodal maximiza-

tion problem is to find

$$S = \{x \in X \mid x \text{ is a local maximum of } f\}. \quad (1)$$

In many of the practical cases the algorithm will find only a subset $\tilde{S} \subset S$ of all maxima. The algorithm presented here is invariant under strictly monotonous linear transformation of the objective function. It can be expected that maxima are detected in sequence of decreasing size of the corresponding domains of attraction, see Section 2.2.

2.1 Overall strategy

The proposed algorithm has a hybrid structure. To approximate niches, a deterministic crowding genetic algorithm (Mahfoud, 1995) is used as niching algorithm. After a fixed number of generations for this genetic algorithm, a nearest neighbor algorithm is employed to determine starting points and covariance matrices as input for the local solver, a covariance matrix adaptation evolution strategy (CMA-ES) (Hansen et al., 2001, 2019). An outer loop repeatedly calling the niching algorithm and the local solver is successively increasing the population of the niching algorithm. In this way we are realizing a semi-sequential behavior. We call the overall algorithm the *Scaling Hybrid Genetic Algorithm (SHGA)*, see Algorithm 1 below.

Algorithm 1: *Scaling Hybrid Genetic Algorithm (SHGA)*

Function: SHGA

Input: objective function f , max # function evaluations N , population increment n_p , # niching generations n_g , size of neighborhood n_{nb}

Output: set of maxima X

```

1  $p \leftarrow \emptyset$ 
2 while True do
3   //increase population
4    $p \leftarrow p \cup$  [random population of size  $n_p$ ]
5   //approximate niches
6    $p \leftarrow$  Niching Algorithm( $f, p, n_g$ )
7   if # fct evals  $> N$  then return ( $X$ )
8   //get seeds points for local solves
9    $S \leftarrow$  Seeds( $f, p, n_{nb}, X$ ),  $S = \{(s_i, \delta_i)\}_{i=1, \dots, |S|}$ 
10  for  $k = 1, \dots, |S|$  do
11    //local solve
12     $x \leftarrow$  Local Solver( $f, s_k, \delta_k$ )
13    if # fct evals  $> N$  then return ( $X$ )
14     $X \leftarrow X \cup \{x\}$ 
15  end
16 end

```

In Algorithm 1, the outer loop (line 2) increases the size of the population at the beginning of each iteration

by randomly generated n_p individuals (line 4). These are added to the previously existing individuals, which themselves remain unchanged. Thereafter the niching algorithm is executed for the constant number of n_g generations on the increased population (line 6). After that, potential new niches are identified by a nearest neighbor based algorithm as seed points $\{s_i\}$ with associated radii $\{\delta_i\}$ of the niches (line 9). This algorithm is described below (Algorithm 2). Already determined maxima, stored in X , are used by the algorithm to exclude non-promising seed points. We call the algorithm to detect new seed points (and the associated radii) the seeds algorithm. For each seed point the local solver is called (line 12) and the (local) maximum is added to the set of maximizers (line 14). The while loop is ended and the result is returned when the maximum number of function evaluations is exceeded (line 7 and 13).

The algorithm depends on a number of parameters. In addition to internal configuration parameters of the niching algorithm (such as mutation and crossover probabilities, etc.) and the local solver (tolerances etc.), Algorithm 1 requires the specification of the four essential parameters N , n_p , n_g and n_{nb} . N is the maximum number of function evaluations and serves as the stopping criterion. Typical values are ranging from 10^4 for simple problems to 10^6 or more for complex problems. The value must be provided by the user. For the other three parameters, we can make good, problem-independent choices. Together with problem-independent choices for the internal parameters of the niching algorithm and the local solver, this is what we mean when claiming that we can formulate a parameter-free algorithm. The choices for the three parameters n_p , n_g and n_{nb} are discussed in section 2.5 below.

2.2 Niching algorithm

The niching algorithm is used to simultaneously approximate niches. To this end we use a deterministic crowding genetic algorithm (Mahfoud, 1995). Per generation, it performs the standard operations of parent selection, recombination, mutation and survivor selection. As parents, all members of the population are chosen and mate with uniform probability (panmictic mixing). The recombination is performed with 30% probability using bounded simulated binary crossover with crowding degree $\eta = 10$. Afterwards, mutation is performed with 30% probability using bounded polynomial mutation with crowding degree $\eta = 10$. For details, see (Deb and Kumar, 1995; Fortin et al., 2012). At the end of each generation, deterministic crowding is used for the competition of the parent and the offspring population.

In Algorithm 1, the niching algorithm is executed for a fixed number of n_g generations. This leads to a sufficient convergence of individuals towards different niches, where the number of individuals in the niches corresponds approximately linearly to the size of the niches. The number of individuals in the niches is not expected to depend significantly on the value of the associated maximum, see e.g. (Butenko et al., 2017; Eiben and Smith, 2015). Regarding the choice of n_g , we refer to the discussion in Section 2.5.

2.3 Seeds algorithm

In Algorithm 1, line 9, the seeds algorithm is used to detect new niches. It returns a number of so-called seed points, one for each potentially new niche. They are found as local maxima in the current population of the niching genetic algorithm. An approximate niche size is determined associated with each seed point. The algorithm is based on nearest neighbor detection and depends on essentially one parameter, the neighborhood size n_{nb} . The algorithm is given below as Algorithm 2.

Algorithm 2: Seeds

Function: Seeds
Input: objective function f , population p , neighborhood size n_{nb} , maxima X
Output: set $S = \{(s_i, \delta_i)\}$ of seed points s_i with niche radius δ_i

```

1 //select potential seeds as local maxima
2  $S_0 \leftarrow \emptyset$ 
3 for  $k = 1, \dots, |p|$  do
4    $p_k = p[k]$ 
5    $p_{nb} \leftarrow \{n_{nb} \text{ nearest neighbors of } p_k \text{ in the}$ 
   Euclidean norm} //this includes  $p_k$  itself
6    $p_{nb} \leftarrow p_{nb} \cup (2p_k - p_{nb})$ 
7   if  $\min(f(p_{nb})) = f(p_k)$  then
8      $\delta_k \leftarrow \max(\{\|p_k - p_{nb}\|\})$ 
9      $S_0 \leftarrow S_0 \cup \{(p_k, \delta_k)\}$ 
10  end
11 end
12 //exclude seeds in neighborhood of known solutions
13  $S_1 \leftarrow \emptyset$ 
14 for  $k = 1, \dots, |S_0|$  do
15    $(s_k, \delta_k) = S_0[k]$ 
16   if  $\min(\|s_k - X\|) > \delta_k$  then
17      $S_1 \leftarrow S_1 \cup \{(s_k, \delta_k)\}$ 
18   end
19 end
20 return ( $S_1$ )
```

The first loop of the algorithm (line 3) identifies individuals, which are maxima in a local neighborhood (line 7). The local neighborhood is formed by n_{nb} neighbors and the 'mirrored' points $2p_k - p_{nb}$ (line 5 and 6).

Note that this neighborhood in general consists of the central point under consideration (p_k) and additionally $2n_{nb} - 2$ points. The maximum distance from p_k to one of the points serves as an estimate for the associated niche size. In the second loop (line 14), the potential seed points are filtered. Points, which lie too close to already known maxima (X) are excluded (line 16). The closeness criterion is related to the estimated niche size.

For keeping the description simple, we have used a number of intuitive notations in Algorithm 2: in line 4 $p[k]$ refers to the k -th element of set p in an arbitrary enumeration, in line 6 ($2p_k - p_{nb}$) denotes the set $\{2p_k - q \mid q \in p_{nb}\}$, in line 7 $f(p_{nb})$ denotes the set of function values of f for each point in p_{nb} . Correspondingly to be understood are the expressions in line 15 and line 16. Note also, that the meta-code given in Algorithm 2 is for illustration purpose. An efficient implementation in terms of number of function evaluations, would be implemented differently.

2.4 Local solver

We use the covariance matrix adaptation evolution strategy (CMA-ES) as a local solver. The local solve for the niche identified by (s_i, δ_i) is initialized with the seed point s_i and a spheric centroid of radius $\delta_i/3$. We use the Python implementation by N. Hansen et. al. (Hansen et al., 2019) with default configuration.

2.5 Critical parameters

In this section we discuss the critical parameters n_p , n_g and n_{nb} and how to choose them for good performance of the overall algorithm. A summary of the parameters and its values is given at the end of this section in Table 1.

Population increase n_p . The parameter n_p sets the initial population of the niching genetic algorithm and its further increment in every iteration of the `while`-loop in Algorithm 1, line 2. The choice of n_p should allow for the detection of a reasonable number of new maxima in every iteration of the `while`-loop. As a number of n_{nb} individuals is necessary to establish one seed point (see Algorithm 2, line 5), a small multiple of this number is a good choice. From our experience a factor five or higher leads to good results, where the exact choice is not critical. In this paper, we make the following generic choice

$$n_p = 10 n_{nb}. \quad (2)$$

Number of generations n_g for the niching algorithm.

The aim of the niching algorithm is to approximate niches. To this end a significant contraction of individuals towards the maximum of each niche needs to be accomplished. Numerical investigations have shown that 20 generations lead to approximately a contraction of 0.3–0.8, dependent on the problem, see Section 3. The contraction rate is measured in a statistical sense. With $n_g = 20$, Algorithm 2 yields good seed points in terms of their number versus duplicate solutions obtained by the local solver. Experiments with $n_g \leq 10$ lead to rather poor seed points, $n_g \geq 40$ does not improve the quality of the seed points significantly as to balance the additional effort. We therefore take the generic choice

$$n_g = 20. \quad (3)$$

Size of neighborhood n_{nb} in the seeds algorithm. The choice of n_{nb} is crucial for the performance of the entire algorithm. A too small size will lead to too many seed points, initiating too many local solves which either often do not converge or converge to already known maxima. A too large neighborhood size will lead to few seeds. This will cause few local searches and correspondingly to detection of only few maxima. To compensate for that, the population size of the niching genetic algorithm would need to increase much more, leading to an inefficient algorithm.

A smooth objective function f can locally be approximated by a quadratic function, which in turn gives information about the location of the local maximum (if it exists). It is therefore reasonable to relate the neighborhood size to the amount of information needed to approximate f by a quadratic polynomial. The minimum required number is therefore given by the degrees of freedom a quadratic polynomial in d dimensions

$$n_{nb} = 1 + 1.5d + 0.5d^2, \quad (4)$$

which is our generic choice. For the detection, we add the 'mirrored' points $2p_k - p_{nb}$ (Algorithm 2, line 6). This is to avoid generating seeds lying at the border of a cloud of individuals, likely located at the slope of a niche and potentially far away from its maximum.

3 Experiments

We evaluate the proposed algorithm on the CEC2013 niching benchmark suite (Li et al., 2013) and a multimodal optimization problem arising from constraint optimization in climate research. For the CEC2013 niching benchmark suite the locations of the absolute maxima are known. For these problems, we also investigate the convergence of the niching algorithm (crowding GA), justifying the choice of (3).

Table 1 Critical parameters of the niching and the seeds algorithm (d is dimension of the optimization problem).

parameter	value	description
n_p	$10n_{nb}$	population size, see Section 2.1
n_g	20	# of generations, see Section 2.1
n_{nb}	$1 + 1.5d + 0.5d^2$	# of neighbors, see Section 2.3

3.1 CEC 2013 niching benchmark suite

The CEC2013 niching benchmark suite consists of 20 problems with dimensions ranging from 1 to 20. The problems are listed in Table 2. They are to be solved within the given budget of function evaluations. For each problem the solution procedure is called 50 times and the so-called average peak ratio is reported, the peak ratio being the fraction of the global maxima detected. For details, see (Li et al., 2013).

The performance of Algorithm 1 on the benchmark suite are given in Table 3. The average peak ratio, the average number of function evaluation and an average contraction number (see below) is given for each of the benchmark problems. The performance of the algorithm (average peak ratio) is good for all problems, with the exception of the Vincent problems (problem 7 and 9) and the 10D and 20D problems (problem 18 - 20).

A comparison of the proposed algorithm with the algorithms from the most recent international competition on this benchmark suite (GECCO 2020) is given in Table 4. We compare the results according to Scenario I. For details, see (Li et al., 2020). The proposed algorithm would rank number 4 compared to the competing other 7 algorithms.

The poor performance of our algorithm on the Vincent problems and on the higher dimensional problems stems from the small basins of attraction of a number of maxima.

Convergence of the niching algorithm. For the CEC-2013 benchmark suite all global maxima are known. With this knowledge, we can define a contraction number $\xi \in [0, \infty[$ measuring the speed of convergence of the niching genetic algorithm. In essence, it describes the average movement of individuals in the population towards their nearest global maximum (in the Euclidean norm) over a number of generations, in our case n_g . A contraction number $\xi < 1$ corresponds to convergence, a contraction number $\xi > 1$ to divergence. The contraction number is best described algorithmically; we give a pseudo code with Algorithm 3. As earlier, we use a simplified, intuitive notation to keep the description

Table 2 Niching benchmark suite from the CEC2013 special session on multi-modal optimization (Li et al., 2013). For each problem function name, problem dimension d , number of maxima #max (global/local), and budget (maximum # function evaluations) are given.

#	function name	d	#max	budget
1	Five-Uneven-Peak Trap	1	2/3	50k
2	Equal Maxima	1	5/0	50k
3	Uneven Decreasing Max.	1	1/4	50k
4	Himmelblau	2	4/0	50k
5	Six-Hump Camel Back	2	2/2	50k
6	Shubert	2	18/many	200k
7	Vincent	2	36/0	200k
8	Shubert	3	81/many	200k
9	Vincent	3	216/0	200k
10	Modified Rastrigin	2	12/0	200k
11	Composition Function 1	2	6/many	200k
12	Composition Function 2	2	8/many	200k
13	Composition Function 3	2	6/many	200k
14	Composition Function 3	3	6/many	400k
15	Composition Function 4	3	8/many	400k
16	Composition Function 3	5	6/many	400k
17	Composition Function 4	5	8/many	400k
18	Composition Function 3	10	6/many	400k
19	Composition Function 4	10	8/many	400k
20	Composition Function 4	20	8/many	400k

Table 3 Performance of Algorithm 1 on the CEC2013 niching benchmark suite (Li et al., 2013). For each problem is given the average peak ratio (fraction of global maxima detected), the average # of function evaluations and the geometric average of the contraction number ξ , as defined by Algorithm 3. The averages are taken over 50 runs.

#	avg. peak ratio	avg. # fct. eval.	avg. ξ
1	1.0000	885	0.4080
2	0.9960	7,320	0.2880
3	1.0000	1,446	0.6572
4	1.0000	7,021	0.4162
5	1.0000	5,346	0.3929
6	1.0000	35,846	0.6025
7	0.7544	191,377	0.6604
8	1.0000	302,761	0.7648
9	0.2904	389,671	0.6941
10	1.0000	13,461	0.4316
11	1.0000	9,716	0.4755
12	0.9975	54,678	0.6078
13	0.9933	53,649	0.5908
14	0.8300	391,121	0.7785
15	0.7500	396,966	0.7581
16	0.6900	389,671	0.8119
17	0.7500	391,949	0.8166
18	0.3933	381,209	0.8498
19	0.5000	371,067	0.8132
20	0.2500	366,589	0.8554

Table 4 Comparison of the proposed Algorithm 1 with the algorithms of the GECCO 2020 Competition on Niching Methods. The proposed algorithm ranks on position 4 wrt. to the score defined by scenario I. For details see (Li et al., 2020).

algorithm	score	rank
RS-CMSA-ESII	0.9063	1
GaMeDE	0.8923	2
CMSA-ES-DIPS	0.8335	3
SHGA	0.8097	
CMSA-ES	0.8004	4
EMSO-MMO	0.6977	5
DErand-H	0.5819	6
FastNichingEP	0.5549	7

simple. The sets S , P_0 , P_1 in Algorithm 3 have an order, where the order of S and the order of the partitioning subsets of P_0 , P_1 correspond. We use indexes to denote access to the element in this order. Finally, the order within the partitioning subsets correspond to the increasing distance to the corresponding elements in S .

As we use a clustering with respect to the Euclidean distance, the contraction number is strictly speaking only meaningful for multimodal problems with isotropic, equally sized basins of attraction. We measured contraction numbers for the benchmark problems, which vary from roughly 0.3 to 0.85. With a contraction of $\xi \leq 2/3$, a good clustering around the associated maxima is achieved, sufficient for a clustering based seed detection. This is true for 11 the benchmark problems. For another six benchmark problems, this level of contraction is reached after a second execution of n_g iterations, and for the remaining three it is reached after a third execution of n_g iterations. Though this argument is rather simplistic, we believe that it in essence captures the important feature of the niching algorithm. We consider this as the first explanation, why a generic choice of $n_g = 20$ leads to successful overall behavior of the proposed algorithm.

3.2 Constraint optimization in climate research

At the heart of current investigations about possible pathways of expected future global warming is the Coupled Model Intercomparison Project (CMIP). It started in 1996 with a handful global coupled climate models (CMIP1) and has in the current phase (CMIP6) more than 50 models producing future climate projections. Despite a lot of progress, inter-model uncertainty of these future projections is still relatively high, depending on the variable and region considered. The evaluation of these models is challenging due to their complexity, sparsity of observations and unknown future

Algorithm 3: Contraction

Function: Contraction

Input: set of locations of global maxima S , set of generations $G = \{g_i\}_{i=0, \dots, |G|}$, where each generation $g_i = \{x_{i,j}\}_{j=1, \dots, |g_i|}$ is a set of individuals (generation may have different numbers of individuals)

Output: contraction number $\xi \in [0, \infty[$

```

1  $Q = \emptyset$ 
2 for  $k = 1, \dots, |G|$  do
3   //partitioning of  $g_{k-1}$ 
4    $P_0 = \{P_0[1], \dots, P_0[|P_0|]\}$  partition of  $g_{k-1}$  in
   clusters of nearest neighbors  $\forall s_i \in S$ 
5   //partitioning of  $g_k$ 
6    $P_1 = \{P_1[1], \dots, P_1[|P_1|]\}$  partition of  $g_k$  in
   clusters of nearest neighbors  $\forall s_i \in S$ 
7   for  $i = 1, \dots, |S|$  do
8     //number of individuals to compare
9      $n = \min(|P_0[i]|, |P_1[i]|)$ 
10    for  $j = 1, \dots, n$  do
11      //contraction per compared individuals
12       $Q = Q \cup \{|P_1[i][j] - S[i]|_2 / |P_0[i][j] - S[i]|_2\}$ 
13    end
14  end
15 end
16 //geometric mean over all generations
17  $\xi = \text{geometric mean}(Q)^{|G|}$ 
18 return ( $\xi$ )
```

dynamics. The emergent constraint approach, see e.g. (Hall et al., 2019), allows the use of present-day measurements to constrain future spread in the model ensemble. At the core of this concept lies the idea that certain present-day dynamics will influence the future projection of a model. Collectively, the systematic bias in the present-day dynamic among the models can be used to constrain the spread in the future projections through cross correlation across the model ensemble. Climate scientists have spend a lot of efforts into finding such emergent constraints for certain areas and processes. However, in previous attempts, the investigated areas are often pre-defined and the effort of the investigation is put into finding highly correlated dynamics. Here, we follow the opposite approach, where the correlated dynamics are already determined, and we try to optimise the area of the present-day dynamics used in the emergent constraint. The correlated dynamics that we consider are the spatially-averaged modelled scalar value of the anthropogenically altered future CO_2 -uptake over the North Atlantic (here denoted as Y) and the present-day modelled percentage of deep ocean CO_2 -storage (here denoted as X). The details of this are beyond the scope of this paper and we refer to (Goris et al., 2021) for further details.

We describe here the related mathematical optimization problem. The aim is to identify regions e ,

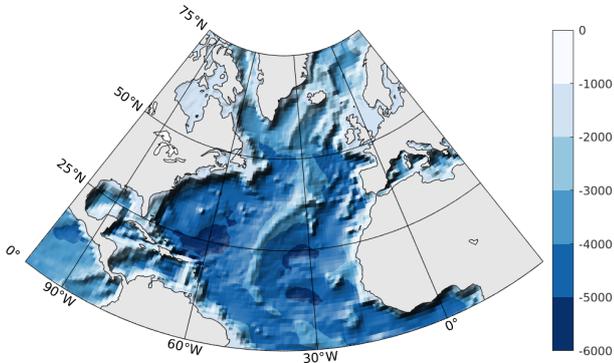


Fig. 1 The area under consideration lies in the North Atlantic ocean between America to the west, and Europe/Africa in the east, restricted to latitudes between 0° and 75°N , from the sea surface to the sea bottom.

which optimize the cross-correlation between Y and $X(e)$. These regions are supposed to lay within a 3D-domain, the North Atlantic ocean between America to the west, and Europe/Africa in the east, restricted to latitudes between 0° and 75°N , from the sea surface to the sea bottom, see Figure 1. Here, the data of X is available on a longitude-latitude-depth axis-parallel grid with grid spacing $1^\circ \times 1^\circ \times 100\text{m}$. Only cells with a model-variance larger than the spatially-averaged model-ensemble standard deviation of $4.84e - 08$ are considered, so that the optimal region can indeed be used to constrain model spread. We denote this set of cells by \mathcal{G} and seek the set of cells (a subset of \mathcal{G}) which maximizes the correlation (in terms of the Pearson-correlation coefficient) of the randomly picked 11 model's average X and their Y . In order to regularize this ill-posed problem, only areas resembling ellipsoids are considered. The set of these discrete ellipsoids $\mathcal{E} \subset 2^{\mathcal{G}}$ is defined as the image of $\gamma : E \rightarrow \mathcal{E}$, $e \rightarrow \{c \in \mathcal{G} \mid \text{midpoint}(c) \in e\}$, where E denotes the set of all ellipsoids (subsets of \mathbb{R}^3). We can now define the objective function of our multimodal optimization problem. Let

$$f : \mathcal{E} \rightarrow \mathbb{R}, e \rightarrow \rho_{X(e), Y}, \quad (5)$$

where $\rho_{X,Y}$ denotes the Pearson correlation coefficient and $X, Y \in \mathbb{R}^{11}$ is the present percentage of deep ocean CO_2 storage and future average anthropogenically altered CO_2 -uptake

$$X = X(e) = (X_1(e), \dots, X_{11}(e))$$

with

$$X_m(e) = \sum_{t,c \in e} \% \text{CO}_2\text{-storage}(t,c) \text{ of model } m|_{m=1\dots 11},$$

where t is running in monthly steps over the time period [1999, 2009] (the decade surrounding the available observations normalised for the year 2004) and $m = 1 \dots 11$ is an arbitrary enumeration of the climate models here under consideration. Correspondingly, Y is defined by its components

$$Y_m = \sum_{t,c \in \mathcal{G}} \text{CO}_2\text{-uptake}(t,c) \text{ of model } m|_{m=1\dots 11},$$

where t runs in monthly steps over the time period [2090, 2100]. When equipped with Steinhaus distance

$$d(e_1, e_2) := \frac{|e_1 \Delta e_2|}{|e_1 \cup e_2|}, \forall e_1, e_2 \in \mathcal{E},$$

the set of discrete ellipsoids $\mathcal{E} \subset 2^{\mathcal{G}}$ becomes a finite metric space (Δ denotes the symmetric difference and $|\cdot|$ the counting measure).

In order to use Algorithm 1 to find the maxima of objective function f (5), we choose to describe an ellipsoid $e \in \mathcal{E}$ by a real-valued genome. Ellipsoids in \mathbb{R}^3 have 9 degrees of freedom. We choose the following representation

$$e(x, S, D) := \{y \in \mathbb{R}^3 \mid ((y - x), e^{\pi S} D^\dagger e^{-\pi S} (y - x)) \leq 1\},$$

where $x \in \mathbb{R}^3$ is the center of the ellipsoid, $S \in \mathbb{R}^{3 \times 3}$ is an antisymmetric matrix, $D \in \mathbb{R}^{3 \times 3}$ is a diagonal matrix and \dagger denotes the Moore-Penrose inverse. We encode (x, S, D) as $\xi \in \mathbb{R}^9$ and use the above defined map γ to define

$$\tilde{f}(\xi) : \mathbb{R}^9 \rightarrow \mathbb{R}, \xi \rightarrow f \circ \gamma \circ e(\xi) \quad (6)$$

Due to the nature of γ , the objective function \tilde{f} is piece-wise constant. The length scale of variation of \tilde{f} is typically 10^{-4} , so that the function assumes likely $\geq 10^{30}$ different function values in $[0, 1]^9$. On larger scales ($\delta x \approx 10^{-2}$) the function resembles a smooth, continuous function. It turns out that it has many (local) maxima, many of them clustering together. To find all or a major part of them therefore requires an efficient multimodal optimization algorithm, as the one presented with Algorithm 1. (Strictly speaking, for piecewise constant functions, such as \tilde{f} the notion of local or global maxima is problematic. Normally, the set of local maxima equals the entire domain with the exception of a set with measure 0 and the set of strict local or global maxima is empty. Only the set of (non-strict) global maxima is non-trivial. We disregard this aspect as \tilde{f}

Table 5 Statistics of the multimodal optimization of \tilde{f} (6). Each row corresponds to an iteration of the `while`-loop (Algorithm 1, line 2). The population is increasing linearly by 550 each iteration, the number of maxima (local or global) found increases from 1 after the first iteration to 187 after the ninth iteration. In addition the cumulative number of function evaluations is shown. The last column shows the efficiency, the percentage of function evaluations used in local solves as part of the overall number, a measure for the efficiency of the multimodal algorithm.

Iteration	Pop. size	# Sol.	# Fct eval	Efficiency
1	550	1	10,858	41.8%
2	1,100	11	61,377	69.6%
3	1,650	25	144,715	74.4%
4	2,200	39	225,501	73.2%
5	2,750	59	335,384	73.0%
6	3,300	88	492,141	74.3%
7	3,850	115	654,946	74.4%
8	4,400	147	821,009	73.8%
9	4,950	187	997,194	73.1%

resembles a smooth, continuous function on a larger scale.)

We use Algorithm 1 with the standard configuration for a 9-dimensional search space: $n_p = 550$, $n_g = 20$ and $n_{nb} = 55$ and allow for $N = 10^6$ function evaluations. The optimization executed nine iterations of a successively increasing population (Algorithm 1, line 2) with an initial population of 550 and a final population of 4950. A total of 187 local maxima have been found. About 25% of the function evaluations were used to find the starting point for local solves (seed points) and 75% were used in local solves by the CMA-ES algorithm. We call this the *efficiency* of the multimodal optimization. Table 5 gives the statistics of the optimization run.

Because \tilde{f} is piece-wise constant, the local solver (CMA-ES) identifies a number of saddle points as local maxima. A random testing of 1000 points in a 10^{-1} -Euclidean neighborhood is used to detect the saddle points (the search space is normalized to $[0, 1]^9$). After eliminating spurious maxima a number of 170 maxima remain. The 10 best solutions have objective function values

$$\begin{aligned} \eta_0 &= 0.9819, & \eta_1 &= 0.9809, & \eta_2 &= 0.9792, & \eta_3 &= 0.9778, \\ \eta_4 &= 0.9769, & \eta_5 &= 0.9768, & \eta_6 &= 0.9756, & \eta_7 &= 0.9749, \\ \eta_8 &= 0.9744, & \eta_9 &= 0.9705 \end{aligned}$$

corresponding to $\xi_i, i = 0, \dots, 9 \in \mathbb{R}^9$ ellipsoids. Since the representation of the ellipsoids in \mathbb{R}^9 and its metric is not natural (e.g. the map $\gamma \circ e$ is not injective, two different ξ can represent the same discrete ellipsoid), it is appropriate to investigate the maxima in the metric space (\mathcal{E}, d) of discrete ellipsoids as defined above. In order to visualize the first 10 ellipsoids in terms of their distances, a 9 dimensional Euclidean space would be necessary. If we allow for a controlled uncertainty

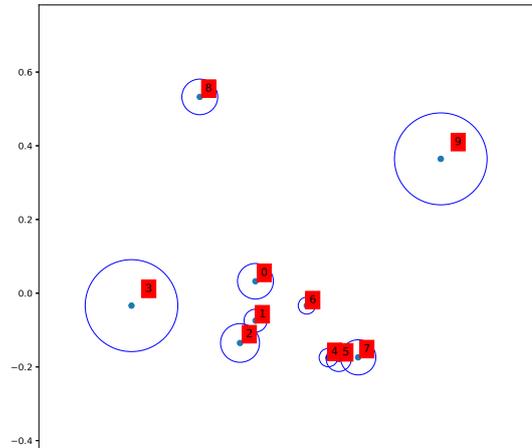


Fig. 2 The 10 best solutions (0 best), represented by circles. Distances of solutions lie in the range of possible distances between the points in the circles. The solutions 0, 1 and 2 (and possibly 6) form one cluster, the solutions 4, 5 and 7 form another cluster, whereas the solutions 3, 8 and 9 are clearly separated.

in the display, we can display the metric relationship between the first 10 ellipsoids in the Euclidean plane, see Figure 2. In the figure, solutions are represented as circles and their corresponding distances in (\mathcal{E}, d) lie in the interval of minimum and maximum distant points in corresponding circles. It can be seen that the solutions 0, 1 and 2 (and possibly 6) form one cluster, the solutions 4, 5 and 7 form another cluster, and the solutions 3, 8 and 9 are clearly separated.

A discussion of the corresponding ellipsoids in terms of climate related constraint optimization problem is beyond the scope of this paper. For further details, we refer to (Goris et al., 2021).

4 Discussion

The aim of this work was to design a widely applicable black-box algorithm for continuous, multimodal optimization problems of moderate dimension. We chose a hybrid algorithmic approach of (1) identifying basins of attraction and (2) carrying out local solves. For (1), we chose a standard panmictic genetic algorithm (deterministic crowding) with nearest neighbor based detection of basins of attraction in order to avoid a priori assumptions on size and shape of the basins. In order to cope with black-box objective functions, which may have an arbitrary number of maxima, the genetic algorithm is used with a successive increasing population. Local solves (2) are carried out with the standard CMA-ES solver. The key to a successful optimization

algorithm is the proper balance of the efforts of global exploration and local solves.

It turns out that a proper balance can be achieved by carefully choosing three parameters, two configuring the panmictic deterministic crowding algorithm (population size, number of generations) and one specifying the identification of the basins of attraction (number of individuals required per basin). These parameters can be chosen dependent on the problem dimensionality. The algorithm is efficient for a large variety of problems from low to moderately high dimensions. Exceptions from good performance are (a) anisotropic problems and (b) higher-dimensional problems with large amounts of local maxima. We believe that this can be attributed to the large differences in size of the basins of attraction in the case (a) of the anisotropic problems. In the case of the higher-dimensional problems (b), these differences play certainly a role as well, together with the very large number of local maxima for the high dimensional problems.

In summary, we state that our design choice is successful and that the algorithm can be valuable for a larger number of applications.

5 Conclusion

In this work we presented two-phase multimodal optimization scheme built upon the well-established deterministic crowding GA and CMA-ES. It identifies niches through a simple use of nearest neighbors. The scheme is flexible enough to capture niches of various shapes and sizes. Robustness is achieved through a sensible balancing of efforts of exploration and exploitation. The settings can be chosen such that the algorithm becomes essentially a black box algorithm without a need for tuning parameters.

Declarations

Funding, Support This work was supported by the Research Council of Norway through the project COLUMBIA (grant number 275268). The simulations were performed on resources provided by NORCE Norwegian Research Centre AS.

Conflict of interest The authors declare that they have no conflict of interest.

Human and animal rights This article does not contain any studies with human participants or animals performed by any of the authors.

Availability of code, data and material In preparation.

References

- Audit C, Hare W (2017) Derivative-Free and Blackbox Optimization. Springer, Berlin, DOI <https://doi.org/10.1007/978-3-319-68913-5>
- Butenko S, Pardalos PM, Shylo V (2017) Optimization Methods and Applications. Springer, Berlin, DOI <https://doi.org/10.1007/978-3-319-68640-0>
- Deb K, Kumar A (1995) Real-coded genetic algorithms with simulated binary crossover: Studies on multimodal and multiobjective problems. *Complex Systems* 9:431–454
- Eiben AE, Smith JE (2015) Introduction to Evolutionary Computing. Springer, Berlin, DOI <https://doi.org/10.1007/978-3-662-05094-1>
- Fortin FA, Rainville FMD, Gardner MA, Parizeau M, Gagné C (2012) DEAP: Evolutionary algorithms made easy. *Journal of Machine Learning Research* 13:2171–2175
- Goldberg D, Richardson J (1987) Genetic algorithms with sharing for multimodal function optimization. In: *Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on Genetic Algorithms*, p 4149
- Goldberg D, Vössner S (1999) Optimizing global-local search hybrids. In: *GECCO*
- Goris N, Johannsen K, Tjiputra J (2021) Application of a genetic algorithm to regionally optimize emergent constraints of the future north atlantic carbon uptake, unpublished paper
- Hall A, Cox P, Huntingford C, Klein S (2019) Progressing emergent constraints on future climate change. *Nat Clim Change* 9:269278, DOI <https://doi.org/10.1038/s41558-019-0436-6>
- Hansen N, Ostermeier A, Bar F (2001) Completely derandomized self-adaptation in evolution strategies. *IEEE Computational Intelligence Magazine* 9:159195
- Hansen N, Akimoto Y, Baudis P (2019) Pycma - a python implementation of cma-es. <https://github.com/CMA-ES/pycma>
- Jong KAD (1975) The analysis of the behaviour of a class of genetic adaptive systems. PhD thesis, University of Michigan, Ann Arbor
- Kronfeld M, Zell A (2010) Towards scalability in niching methods. In: *Proceedings of the 2010 IEEE Congress on Evolutionary Computation (CEC10)*, pp 1–8
- Li X, Engelbrecht A, Epitropakis M (2013) Benchmark functions for cec 2013 special session and competition on niching methods for multimodal function optimization. Technical report. evolutionary computation and machine learning group, RMIT University, Australia

- Li X, Epitropakis MG, Deb K, Engelbrecht A (2016) Seeking multiple solutions: An updated survey on niching methods and their applications. *IEEE Transactions on Evolutionary Computation* 21(4):1–21, DOI <https://doi.org/10.1109/TEVC.2016.2638437>
- Li X, Engelbrecht A, Epitropakis M (2020) Gecco 2020 competition on niching methods for multimodal optimization. <https://github.com/mikeagn/CEC2013>
- Mahfoud SW (1995) Niching methods for genetic algorithms. PhD thesis, University of Illinois at Urbana-Champaign, Champaign, IL, USA
- Mengshoel OJ, Goldberg D (2008) The crowding approach to niching in genetic algorithms. *Evolutionary Computation* 16(3):315354, DOI <https://doi.org/10.1162/evco.2008.16.3.315>
- Preuss M (2015) *Multimodal Optimization by Means of Evolutionary Algorithms*. Springer, Berlin, DOI <https://doi.org/10.1007/978-3-319-07407-8>
- Sareni B, Krahenbuhl L (1998) Fitness sharing and niching methods revisited. *IEEE Transactions on Evolutionary Computation* 2(3):97106
- Smith R, Forrest S, Perelson A (1993) Searching for diverse, cooperative populations with genetic algorithms. *Evolutionary Computation* 1(2):127–149, DOI <https://doi.org/10.1162/evco.1993.1.2.127>