



HAL
open science

Nevergrad: black-box optimization platform

Pauline Bennet, Carole Doerr, Antoine Moreau, Jeremy Rapin, Fabien Teytaud, Olivier Teytaud

► **To cite this version:**

Pauline Bennet, Carole Doerr, Antoine Moreau, Jeremy Rapin, Fabien Teytaud, et al..
Nevergrad: black-box optimization platform. ACM SIGEVOlution, 2021, 14 (1), pp.8-15.
10.1145/3460310.3460312 . hal-03464208

HAL Id: hal-03464208

<https://uca.hal.science/hal-03464208v1>

Submitted on 3 Dec 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License



Nevergrad: Black-Box Optimization Platform

Nevergrad is an open source platform for black-box optimization.

Join the [user group](#)!

and if you like Nevergrad, please support us by adding a star on GitHub (click on “star” here: <https://github.com/facebookresearch/nevergrad>).

Authors: Pauline Bennet, Carola Doerr, Antoine Moreau, Jeremy Rapin, Fabien Teytaud, Olivier Teytaud

[What is Black-Box Optimization ?](#)

[The Science of Black-Box Optimization](#)

[Black-box optimization in the presence of noise](#)

[Fully parallel black-box optimization](#)

[Structured optimization](#)

[Optimization Wizards](#)

[Discrete Optimization: choosing the mutation rates](#)

[Multiobjective Optimization](#)

[Optimization with a Neural Quality Assessment in the Loop](#)

[What can we do with Nevergrad ?](#)

[What's New?](#)

[Who uses Nevergrad & Black-Box Optimization ?](#)

[Electricity, Photonics, and Other Real-World Problems](#)

[Visibility in conferences](#)

[Frameworks using Nevergrad](#)

[Open Optimization Competition](#)

[Selected publications using Nevergrad](#)

[Future work](#)

[Nevergrad Contributors \(random order\)](#)

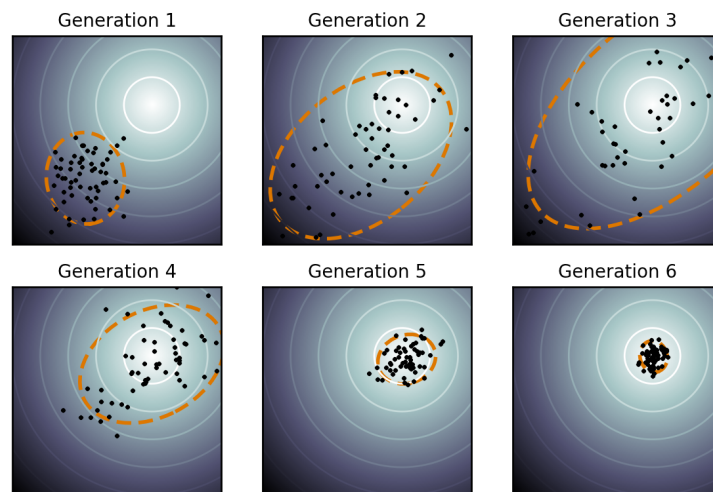
What is Black-Box Optimization ?

Black-box optimization deals with the solution of problems for which we can assess the quality of its solution candidates, but for which we do not have (or do not want to use) gradients or other useful a priori information. Structural engineering or the design of neural networks are classical examples for black-box optimization, where the evaluation of a potential design returns the quality of this particular solution candidate, but typically does not reveal much information about other design alternatives. Information about the problem must hence be collected through the evaluation of several solution candidates. Black-box optimization algorithms are often sequential, iterating between the evaluation of one or more

solution candidates and adjusting the strategy by which the next candidates are generated. Black-box optimization problems can be subject to constraints or to noise. It is not uncommon to have two or more objective functions, for which one aims to find good trade-offs. Decision spaces can be purely numerical, combinatorial, or a mixture of both.

Many different approaches to solve black-box optimization problems exist, and one of the biggest challenges in applying these is in selecting the most suitable technique for a given problem.

Nevergrad aims at supporting its users in this selection task by providing very broad sets of benchmark problems that the algorithms can be compared upon, by making available state-of-the-art black-box optimization algorithms, powerful algorithm selection wizards which support users in selecting an algorithm from our portfolio, and by providing a [frequently updated dashboard](#) of experimental results to support researchers in the analysis and design of efficient black-box optimization techniques.



Evolution strategies (wikipedia), one of the methods included in Nevergrad.

The Science of Black-Box Optimization

Black-Box Optimization in the Presence of Noise

In a [ground-breaking paper](#), population-control was proposed as a simple solution for fast noisy optimization: this combines parallelizability, ability to converge with simple regret $1/n$, and small constants making the algorithm reliable in low dimension. In Nevergrad, the population control algorithm [TBPSA](#), fixes a bias in previous population control methods; it is quite robust for noisy optimization of continuous variables and has been used for an application to Stockfish.

Fully Parallel Black-Box Optimization

In particular for fully parallel hyperparameter search, Various fundamental studies analyzed one-shot black-box optimization methods:

- In high-dimension we should focus closer to the center: this is [quasi-opposite sampling](#); a first [theoretical analysis](#) was followed by [an optimal solution](#).

- Averaging: we should use the average of the k best points rather than just the empirically best one [in one-shot optimization](#); this is a partial solution to the problem of the [benefit of sex](#).

Structured Optimization

In particular for real-world problems, optimization takes into account some high-level information on the structure of the problem ([groups of more inter-related variables](#): they typically use [collaborative coevolution](#)). For example, many variants of differential evolution win [competitions based on the LSGO benchmarks](#). [Scientific reports](#) published some applications to physical structures at the nanometric scale.

Optimization Wizards

Automatic algorithm selection is central in combinatorial optimization and planning: it consists in selecting automatically the probably best algorithm in a wide range of possibilities. Under the name “wizard”, such combined methods routinely [win competitions in SAT planning](#) and combinatorial optimization. We apply it for all forms of black-box optimization. [Some optimization wizards](#) use essentially the budget, the dimension, the type of variables for choosing an algorithm; [improved forms](#) also carefully use chaining (running algorithms the ones after the others, in particular for combining fast local search and robust global search as in [memetic algorithms](#)) and [meta-models](#) (fast learnt approximations of the objective function). Dynamically choosing an algorithm using results is also part of the picture, with [“bet and run”](#) as a classical solution.

Discrete Optimization: Choosing the Mutation Rates

After the initial enthusiasm on simple rules for choosing optimal fixed mutation rates such as [the RLS and the \(1+1\)-evolutionary algorithm](#), new variants used [random mutation rates](#) and then adaptive mutation rates; there is now a whole body of work, including [self-adjusting](#) mutation rates or [mutation rates embedded in the individuals](#) and [coordinate-wise mutation rates](#).

Multiobjective Optimization

Multiobjective optimization consists in looking for trade-offs between several objective functions. Variants of differential evolution ([PDE](#) and [DEMO](#)) are well-known for this, though now all single-objective optimization methods can be adapted to the multiobjective setting using [hypervolume indicators](#), or using [NSGA-II](#) selection methods. A key challenge is to build principled comparisons between those different methods. The [dashboard](#) provides extensive results on multiobjective optimization using different evaluation methods -- including classical ones like hypervolume, epsilon indicators, etc., but also user-centric comparisons using quality assessment tools trained on real human data.

Optimization with a Neural Quality Assessment in the Loop

Evolutionary optimization is convenient for adding a user in the loop, as it does not need gradients (humans typically answer “I prefer this” rather than “the gradient of my favorability is 0.4 wrt the 7th axis of the latent variable”), so one can use evolutionary optimization for combining preferences (provided by humans or by hard to differentiate deep IQA) as in

[Evolutionary GAN](#) (see a demo [here](#)) or [Evolutionary super-resolution](#) or for [interactive GANs](#).

What can we do with Nevergrad ?

Nevergrad is a benchmarking platform that is designed to help researchers gain insight into the strengths and weaknesses of different black-box optimization techniques. For practitioners, Nevergrad provides powerful state-of-the-art optimization techniques, conveniently accessible through a user-friendly python environment.

The key distinguishing feature of Nevergrad is the breadth of algorithms and problem suites which it covers and its [publicly available dashboard](#), which provides convenient access and visualization of our rich data sets.

Algorithm Portfolio: Nevergrad provides implementations of Covariance-Matrix-Adaptation, Particle Swarm Optimization, evolutionary algorithms, Differential Evolution, Bayesian optimization, HyperOpt, Powell, Cobyta, LHS, quasi-random point constructions, NSGA-II, ... Building on our rich benchmark data, our algorithm selector NGOpt combines these algorithms by automatically selecting a solver based on high-level problem information, by sequentially executing two or more algorithms from the list, or by leveraging parallel resources to actively select the best-performing approach.

Benchmark Problems: Nevergrad provides interfaces to several collections of benchmark problems, either home-built (YABBOB, LSGO) or external (MuJoCo, MLDA, PBO). Together, these problem suites range from classical optimization problems over Machine Learning tasks to real-world optimization challenges.

We still include many algorithms (CMA, PSO, DE, Bayesian Optimization, HyperOpt, LHS, Quasi-Random, NSGA-II, ...) and benchmarks (YABBOB, LSGO, LMDA, Mujoco, and many others including real-world stuff e.g. in power systems).

For example we can do hyperparameter optimization, optimizing policies for model-based RL, operations research, multiobjective optimization, and user-controlled GAN or [quality-controlled GAN](#) (see [beautiful cats and horses](#)).



Yes we can generate cats and horses. (collab. Univ. Littoral Cote d'Opale & Univ. Konstanz)

Compared to traditional GANs, the latent variables are slightly mutated (not too much, for preserving diversity) in order to improve the quality of images. For difficult datasets (such as horses) we get rid of weird artifacts such as balls of horse skin floating in the air or horses with multiple heads. A side remark about GANs that the ``abstract'' ``high-level'' quality of images (no weird additional limb) is actually correlated with image quality as estimated from low-level features.

What's New?

Recent progress includes improvement of the multiobjective setting and the constraint management. [MuJoCo](#) (a robotic benchmark for which results are notoriously influenced by implementation details) was added so that you can easily run MuJoCo without suffering for the interfacing. [HyperOpt](#) is added. A competence map automatically selects an algorithm for your problem, if you have used the instrumentation for describing your problem. We also now run a [dashboard](#), which maintains a list of problems and the performance of many algorithms on it. We already had [MLDA](#) as a classical benchmark and YABBOB as our own variant of BBOB, we now also include [LSGO](#) -- all with the same interface. Compared to most existing frameworks (BBOB/COCO, LSGO), we have: real-world problems, realistic ML, correct implementation of noisy optimization, larger scale, and we include algorithms (not only benchmarks). Compared to optimization platforms, we have a wide range of algorithms with the same interface. To the best of our knowledge, Nevergrad is the only platform which periodically [reruns all benchmarks](#).

Who uses Nevergrad & Black-Box Optimization ?

Electricity, Photonics, and Other Real-World Problems

Madagascar has 75% of its population without any access to electricity and even the rest of the population. We collaborate with Univ. Antananarivo for modeling the key “what if” questions regarding the electrification of Madagascar.

Other models for electricity are under development.

We include problems close to [waveguides](#).



Antireflective coating. Right: silicium. Left: silicium + antireflective coating.

AI-designed AR coating typically reduce the part of light which is lost.

(collaboration Univ. Clermont-Ferrand)

Visibility in conferences

Nevergrad is mainly a thing in optimization conferences (GECCO, PPSN, Dagstuhl/optim), but it becomes known in ML conferences: 6 PDFs on OpenReview ICLR mention Nevergrad, 1 at ICML-Proceedings, 71 on Arxiv.

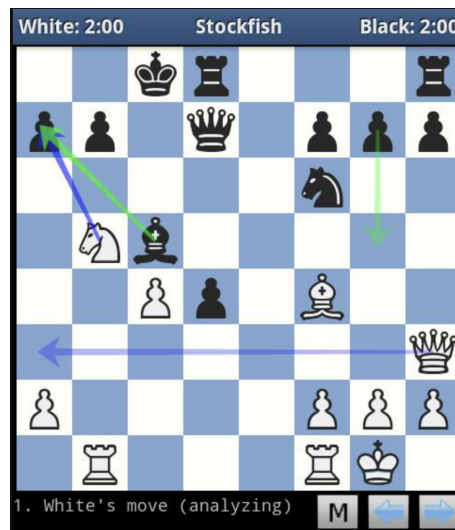
Several workshops and competitions were based on Nevergrad.

Frameworks using Nevergrad

Hydra and Ray use Nevergrad for optimizing hyperparameters. Nevergrad also interfaces easily with submitit (which is a spinoff of nevergrad). [IOHprofiler](#) profiler is interfaced with Nevergrad, so that problems can be accessed both ways. IOHprofiler's data analysis and visualization tool [IOHanalyzer](#) can easily read Nevergrad's performance files.

Nevergrad is used [for tuning StockFish](#), a very strong chess program still winning top competitions in spite of the zero-style programs. Nevergrad's contribution was not negligible, through optimization of the output layer of the value function.

Nevergrad is used for tuning StockFish, a very strong chess program still [winning top competitions in spite of the zero-style programs](#). Nevergrad's has been used for optimizing the weights of the [neural value function](#).



Stockfish used Nevergrad and still wins top competitions in spite of the zero-learning era (Wikipedia).

Nevergrad was also used for other games (<https://github.com/fsmosca/Lakas>). Also for locating Wifi sources (<https://github.com/ericjster/wifilocator>), and various projects with machine learning (in particular GAN, or research around MuZero <https://github.com/andrei-ars/muzero-test3>), for research in black-box optimization (<https://github.com/lazyoracle/optim-benchmark>), for improving the way deep nets are mapped on accelerators (<https://github.com/maestro-project/gamma>), in digital medicine (<https://github.com/tl32rodan/Digital-Medicine-Smoke-Status-Detection> or https://github.com/hello0630/sepsis_framework), AI planning for financial applications (<https://github.com/gordoni/aiplanner>).

Open Optimization Competition

Together with the [IOHprofiler](#) team we are organizing the [Open Optimization Competition 2021](#). The first one was organized in 2020 ([OOC 2020](#)), and rewarded contributions to the benchmarking experience using Nevergrad or IOHprofiler. The 2021 edition also hosts a classical, performance-oriented track, in which participants are invited to submit their algorithms to compete with the state-of-the-art black-box optimization techniques. Accepted submissions algorithms are automatically run on our platform, and results are made public on our regularly updated [dashboard](#).

Selected publications using Nevergrad

- [Fully parallel hyperparameter search](#) (ICML)
- Evolution of photonics designs: <https://www.nature.com/articles/s41598-020-68719-3> (SREP)
- [Automatic algorithm selection](#) (GECCO)
- [Evolutionary GAN](#) (Oral ACCV)
- [Evolutionary super-resolution](#) (ICPR)
- [Averaging in evolution](#) (PPSN, nominated for best paper award)
- [Scaling the variance in one-shot optimization](#) (PPSN)
- [Consistent population control](#) (FOGA, nominated for best paper award)

Future work

Further improvements of our algorithm wizard is an ongoing task, just as we always strive to include more benchmark problems on which we can test the algorithms. Priority in terms of future extensions are an improved constraint management and the inclusion of real-world applications.

Nevergrad Contributors (random order)

Jeremy Rapin, Antoine Moreau, Fabien Teytaud, Carola Doerr, Baptiste Roziere, Laurent Meunier, Herilalaina Rakotoarison, Olivier Teytaud, Pauline Bennet, Diederick Vermetten, Julien Dehos, Pak Kan Wong, Rahamefy Solofohanitra Andriamalala, Toky Axel Andriamizakason, Andry Rasoanaivo, Vlad Hosu and so many others.