OPT-GAN: A Broad-Spectrum Global Optimizer for Black-Box Problems by Learning Distribution

Minfang Lu[1,5,*], **Shuai Ning**[1,2,*], **Shuangrong Liu**[3], **Fengyang Sun**[4], **Bo Zhang**[1,2], **Bo Yang**[2], **Lin Wang**[1,†]

[1] Shandong Provincial Key Laboratory of Network Based Intelligent Computing, University of Jinan, Jinan 250022, China
[2] Quan Cheng Laboratory, Jinan 250100, China
[3] Department of Computer Science, The University of Suwon, Hwaseong 18323, South Korea
[4] Victoria University of Wellington, Wellington 6140, New Zealand
[5] Cainiao Network, Hangzhou, China
luminfang.lmf@alibaba-inc.com, hardyning@gmail.com, liusr.constant@gmail.com, fengyang.sun@vuw.ac.nz, boz.constant@gmail.com, QCL-yangbo@qcl.edu.cn, wangplanet@gmail.com

Abstract

Black-box optimization (BBO) algorithms are concerned with finding the best solutions for problems with missing analytical details. Most classical methods for such problems are based on strong and fixed *a priori* assumptions, such as Gaussianity. However, the complex real-world problems, especially when the global optimum is desired, could be very far from the *a priori* assumptions because of their diversities, causing unexpected obstacles. In this study, we propose a generative adversarial net-based broad-spectrum global optimizer (OPT-GAN) which estimates the distribution of optimum gradually, with strategies to balance exploration-exploitation trade-off. It has potential to better adapt to the regularity and structure of diversified landscapes than other methods with fixed prior, e.g., Gaussian assumption or separability. Experiments on diverse BBO benchmarks and high dimensional real world applications exhibit that OPT-GAN outperforms other traditional and neural net-based BBO algorithms. The code and Appendix are available at https://github.com/NBICLAB/OPT-GAN

Introduction

Optimization is a study of finding the best solutions to a given problem. In this field, global optimization searches for the globally best solution among all possible ones. We are interested in BBO problems:

$$\arg\min f(\boldsymbol{x}) : \Omega \to \mathbb{R}, \forall \boldsymbol{x} \in \Omega \subset \mathbb{R}^n, \qquad (1)$$

where n is the dimensionality of the searching domain Ω. The "black box" means we have little information about f but the function value $f(\boldsymbol{x})$ indicating the quality of the candidate solution $\boldsymbol{x}$.

In terms of stochastic black-box optimization, we define $p(\boldsymbol{x})$ as a distribution representing the estimated chance that the global optimum $\boldsymbol{x}^*$ can be found at $\boldsymbol{x}$. Thus, the optimization can be viewed as progressively reshaping explicit/implicit $p(\boldsymbol{x})$ based on sampled $\boldsymbol{x}$ and corresponding $f(\boldsymbol{x})$, toward the Dirac delta distribution $\delta_{x^*}(\boldsymbol{x})$ with a spike at $\boldsymbol{x}^*$. Note that different optimizer reshapes $p(\boldsymbol{x})$ in different way (see Appendix A for the general framework).

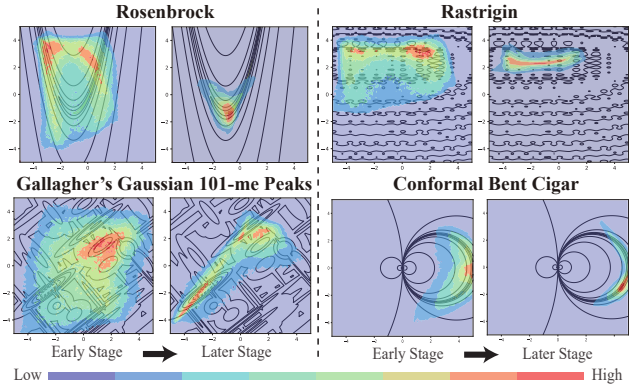

Figure 1: Estimated $p(\boldsymbol{x})$ using OPT-GAN on four problems (see Appendix E for the plotting details of $p(\boldsymbol{x})$). It reshapes $p(\boldsymbol{x})$ progressively according to properties of a landscape.

The No Free Lunch Theorem (NFLT) (Wolpert and Macready 1997) indicates the significance of prior knowledge to a global optimizer, as the performance difference between optimizers depends on how their *a priori* assumptions match with the inherent regularities of problem. Although there are some specialized *narrow-spectrum*[1] optimizers designed for specific problems by utilizing prior knowledge, e.g., Iterative Closest Point algorithm (Chetverikov et al. 2002) focuses on point-set matching related problems, the difficulty for obtaining sufficient problem-related information in real world leads to prosperity of *broad-spectrum* optimizers.

To deal with the challenge of information deficiency, the broad-spectrum optimizers maximize performance with lit-

[*] These authors contributed equally.
[†] Corresponding author: Lin Wang

[1] Inspired by the term broad-spectrum antibiotics, spectrum implies the range of applicable problems of an optimizer. The broad-spectrum ones expect to achieve acceptable performance at a wide range via strong configurability. However, a narrow-spectrum one aspires to the best performance at specific families of problems.

tle *a priori* knowledge on target black-box optimization problems. They expect to achieve acceptable performance at various problems via strong configurability, i.e., balancing exploration-exploitation (E-E) trade-off using configurable parameters or modules.

There are two branches of broad-spectrum optimizers: *model-building* and *model-free* algorithm, e.g., Genetic Algorithms (GA) (Holland 1975) and Particle Swarm Optimization (PSO) (Eberhart and Kennedy 1995), directly manipulate solutions in an implicit way. Despite wide application, they are weak in capturing complex problem structures (Katoch, Chauhan, and Kumar 2021; Liu and Liu 2017). The model-building ones search with explicit models, such as modeling landscape (i.e., f) or gradient (i.e., ∇f). Among them, modeling $p(\boldsymbol{x})$ directly is one of primary branches, e.g., Estimation of Distribution Algorithms (EDAs) (Hauschild and Pelikan 2011), which have achieved considerable advances (Liang et al. 2020; De Bonet, Isbell, and Viola 1996; Chen et al. 2017). They have potentials to capture complex structures and learn accurate distribution of promising solutions without much attention on landscape details (Cheng et al. 2018; Antoniou and Papa 2021).

However, the diversity of real-world problems with massive unknown properties makes choosing proper *a priori* assumption for EDAs difficult. For instance, as a widely used EDA method, Evolutionary Strategy with Covariance Matrix Adaptation (CMA-ES) fails to reshape its distribution type for problem whose landscape is distinct from a Gaussian surface or mixture of Gaussians (Liu et al. 2020a).

Without violating the NFLT, a "shapeshifter" broad-spectrum optimizer with less dependency on prior of target problems is desired, especially a flexible $p(\boldsymbol{x})$ EDA method, which tunes distribution to adapt to diversified problems progressively. As an universal distribution learner (Lu and Lu 2020), the Generative Adversarial Networks (GANs) (Goodfellow et al. 2014) is one of the strongest candidates for capturing arbitrary distribution. In fact, their effectiveness for optimization have been preliminarily verified by GAN-based Solver (GBS) (Gillhofer et al. 2019) for *local optimization* and MOE-GAN (He et al. 2021) for *multiobjective optimization*.

Motivation Unfortunately, the *global optimization* by estimating the distribution of optima progressively is still hard to reach, as we have to carefully balance the exploration-exploitation trade-off during optimization to gradually expose global optimum. This problem is even more notable for black-box tasks because we have no information on which side is better.

A question arises here: *can we design a broad-spectrum global optimizer by modeling a flexible $p(\boldsymbol{x})$ estimator, enabling balancing E-E trade-off and adapting to arbitrary black-box problem?*

Contribution We propose a generative adversarial net-based broad-spectrum optimizer for global optimization, named OPT-GAN, which estimates the distribution of optimum gradually by balancing E-E trade-off (see Fig. 1). OPT-GAN shows potential to better adapt to the structures of different landscapes than other EDA methods with fixed prior, e.g., Gaussian distribution assumption in CMA-ES. Detailed contributions are listed as follows:

- A novel GAN-based broad-spectrum global optimizer for black-box problems is proposed to estimate the distribution of global optimum gradually.
- As a global optimizer, OPT-GAN adopts a bi-discriminators framework to guide the generator to learn balancing E-E trade-off during optimization.
- A continually updating and shrinking optimal set provides data about regularities, encouraging exploration at the early stage and exploitation at the late stage.
- To avoid biased initialization and premature convergence, a generator pre-training method is used to ensure full domain initialization.
- Experiments manifest that OPT-GAN achieves the best results on diversified problems compared with different types of optimizers, including neural network-based state-of-the-art optimizers.

Related Works

Narrow-Spectrum Optimizers Narrow-spectrum optimizers can solve specific problems in an efficient way by exploiting the problem-related knowledge (Serofino 2014), e.g., Backpropagation (LeCun et al. 1988), FBGAN (Gupta and Zou 2019), CbAs (Brookes, Park, and Listgarten 2019), DbAs (Brookes and Listgarten 2020), MetricGAN (Fu et al. 2019), and PGATS (Zhou et al. 2014). However, such prior knowledge or analytical details are often not available for many real-world optimization problems (Dulac-Arnold et al. 2021; Thor, Kulvicius, and Manoonpong 2021).

Model-Free Optimizers As a typical family of broad-spectrum optimizers, model-free algorithms manipulate solutions directly. This family involves, e.g., BFGS (Liu and Nocedal 1989), Nelder-Mead method (NM) (Nelder and Mead 1965), Generalised Pattern Search (Torczon 1997), Simulated Annealing (Kirkpatrick, Gelatt, and Vecchi 1983), GA, and PSO. They are popular for their approachable and easily implementable mechanism, yet suffering from weak mathematical structure and low efficiency (Audet and Hare 2017).

Model-Building Optimizers Model-building optimizers rely on adjusting an explicit model to search the BBO problems. **(1) Surrogate Optimization Algorithms** simulate the landscape to reduce practical evaluations, e.g., Response Surface Methods (Jones 2001) and Support Vector Machine-based Surrogate Models (Ciccazzo, Di Pillo, and Latorre 2016). Bayesian Optimization adopts Gaussian processing to accurately regress the landscape, and has been successfully applied to many expensive but low dimensional problems (Balandat et al. 2020; Shahriari et al. 2015; Eriksson et al. 2019). However, when solving high-dimensional problems, sample size that accurate modeling requires increases exponentially, leading to degraded sampling and inefficient search (Raponi et al. 2020; Luo et al. 2012). In addition, its time consumption increases drastically when reconstructing the model with increasing samples. Although some approaches such as incremental learning could alleviate this

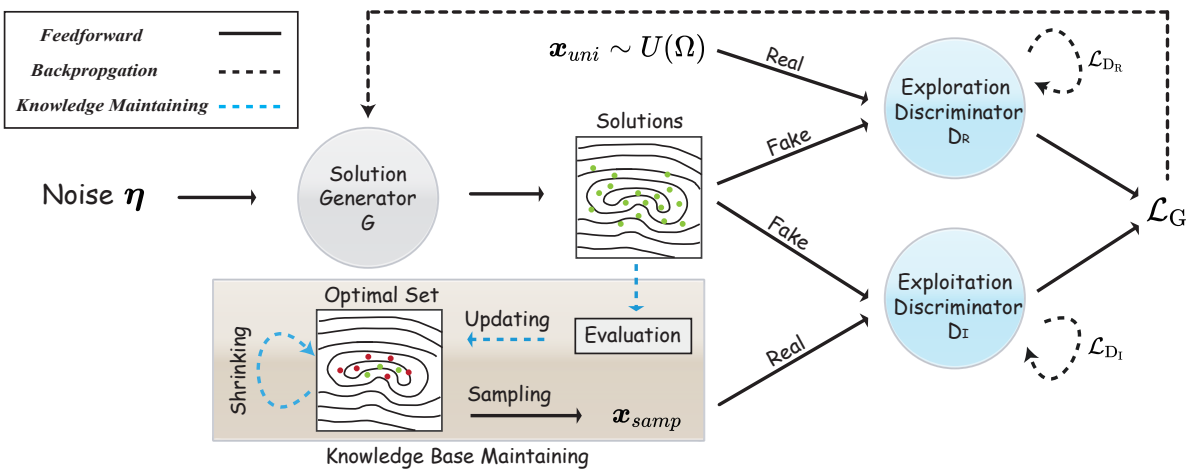

Figure 2: The framework of the proposed OPT-GAN. The $U(\Omega)$ is a uniform distribution on Ω.

issue (Jenatton et al. 2017; Klein et al. 2017), extra flaw of catastrophic forgetting limits its learning ability. textcolor-blueNote that the discussions are limited to navie Bayesian Optimization. **(2) Gradient Estimators**, such as OPEN-ES (Salimans et al. 2017), and Adaption Directional Gaussian Smoothing (Tran and Zhang 2020), perform well on quasi-convex problems due to local learning on gradient but poorly on non-convex problems.

(3) The last family **EDAs** estimate the distribution of optima by progressively sampling solutions, such as Cross Entropy Method (Rubinstein and Kroese 2004) and Latent Space-based EDA (Dong, Wang, and Zhou 2019). As one of the most prominent EDAs, CMA-ES estimates the better region by adaptively reshaping the Gaussian model and has received lots of successful stories (Kämpf and Robinson 2009; Loshchilov and Hutter 2016). EDAs have presented remarkable performance on the problems compatible with their distribution assumption, e.g., Gaussian distribution. Nevertheless, EDAs with strong *a priori* are weak in adapting to diversified real-world problems.

Neural Network-based Optimizers The neural network(NN)-based black-box optimizers have recently gained growing attention due to their powerful approximation ability, e.g., RandomRL (Mania, Guy, and Recht 2018), GradientLess Descent (Golovin et al. 2020), and Indirect Gradient Learning (IGL) (Lillicrap et al. 2015). Explicit Gradient Learning (EGL) (Sarafian et al. 2020) directly estimates the gradient ∇f by learning the parametric weights. In addition, Weighted Retraining (WR) (Tripp, Daxberger, and Hernández-Lobato 2020) adopts latent manifold learning to convert a problem landscape into a latent space via generative models and search by traditional optimizer in the new space (Faury, Calauzènes, and Fercoq 2019; Faury et al. 2019; Sakamoto et al. 2020).

Given the universal distribution learning capability of GANs, the GAN-based optimizer may herald a fruitful direction by modeling $p(\boldsymbol{x})$. In 2019, focusing on inverse problem, a GAN-based Solver (Gillhofer et al. 2019) was proposed. Although it's a preliminary work and only tested on topology optimization, the potential of GANs have been exhibited. If we temporally move our attention to another related field *multiobjective optimization*, there is also a GAN-based optimizer proposed recently (He et al. 2021).

Challenge Although landscape or gradient estimating methods, such as IGL and EGL, take advantage of the universal approximation ability and present encouraging performance on some BBO problems, their inherent natures with local gradient learning determine that they are more capable of solving quasi-convex problems (Sarafian et al. 2020; Liu et al. 2020b). Latent manifold learning methods could result in the loss of landscape information owing to inaccurate conversion, though they require weak *a priori* assumption.

The GAN-based solver is a pioneer of GAN-based optimization, but is almost a *local optimizer*. It only pays attention to the exploitation by gradually dividing the search space, but loses the ability of jumping out of local optimum, to say nothing of balancing E-E trade-off. It also has many other weaknesses, e.g., non-smoothly reshaping, premature convergence, and dependency on initial state (see Appendix B for more details). Thus, a broad-spectrum *global optimizer* reshaping distribution flexibly and progressively with strategies of balancing E-E trade-off, is highly desired.

Methodology

Formulation

Generally, proper definition of $p(\boldsymbol{x})$ guarantees searching efficiency (see Appendix A). Since there is no meaningful information about the black box f, we can only use historical $\boldsymbol{x}$ and $f(\boldsymbol{x})$ recorded to construct $p(\boldsymbol{x})$. We firstly assume the optimum $\boldsymbol{x}^*$ is drawn from $h(\boldsymbol{x})$, i.e, the distribution of *historical best* solutions $\boldsymbol{x}_{\text{opt}}$. Refining $h(\boldsymbol{x})$ toward the delta distribution $\delta_{x^*}(\boldsymbol{x})$ for *exploitation* can be achieved by updating $\boldsymbol{x}_{\text{opt}}$ with samples from previous $h(\boldsymbol{x})$.

However, if $\boldsymbol{x}^*$ is out of distribution $h(\boldsymbol{x})$, the optimizer would fail in locating it by exploitation solely, which leads to the necessity of additional *exploration* on the global domain Ω. As the uniformly random search is often viewed as an extreme exploration, we use the mixture distribution

Notation	Description	Notation	Description
n	Dimensionality of a benchmark.	G	The solution generator.
fitness	Goodness of a solution, related to the value of f.	D	Bi-discriminators.
$U(\Omega)$	Uniform distribution on domain Ω.	D_I	Exploitation discriminator.
$\boldsymbol{x}_{\text{uni}}$	Solution set sampled from $U(\Omega)$.	D_R	Exploration discriminator.
$\boldsymbol{x}_{\text{samp}}$	Solution set bootstrap sampled from $\boldsymbol{x}_{\text{opt}}$.	$\boldsymbol{\omega_G}, \boldsymbol{\omega_I}, \boldsymbol{\omega_R}$	Network parameter of G , D_I, D_R.
$\boldsymbol{x}_{\text{opt}}$	Historical best solution set.	$PreIter$	Number of generator pre-training iterations.
$\boldsymbol{x}_{\text{G}}$	Solution set generated by G.	$GANIter$	Number of training iterations of GAN.
M	Population size of $\boldsymbol{x}_{\text{G}}$ when updating.	$DIter$	Number of training iterations of D.
K	Initial size of $\boldsymbol{x}_{\text{opt}}$.	β	Gradient penalty factor.
a	Shrinking rate.	λ	Adjustment factor for E-E trade-off.
FEs	Number of function evaluations.	S	Batch size when training GAN.
$MAXFes$	Maximum of FEs.	$\hat{\boldsymbol{x_I}}$	Samples uniformly sampled between $\boldsymbol{x}_{\text{G}}$ and $\boldsymbol{x}_{\text{samp}}$.
$\boldsymbol{\eta}$	Random noise.	$\hat{\boldsymbol{x_R}}$	Samples uniformly sampled between $\boldsymbol{x}_{\text{G}}$ and $\boldsymbol{x}_{\text{uni}}$.

Table 1: Description of notations.

of $h(\boldsymbol{x})$ and a multivariate uniform distribution $u(\boldsymbol{x})$ as the final $p(\boldsymbol{x})$, which is defined as

$$p(\boldsymbol{x}) = (1/(1+\lambda)) \cdot h(\boldsymbol{x}) + (\lambda/(1+\lambda)) \cdot u(\boldsymbol{x}), \quad (2)$$

where the E-E trade-off is tuned by adjustment factor λ. The $u(\boldsymbol{x})$ is defined as

$$u(\boldsymbol{x}) = \begin{cases} \frac{1}{v(\Omega)} & \boldsymbol{x} \in \Omega \\ 0 & \boldsymbol{x} \notin \Omega \end{cases}, \quad (3)$$

where $v(\Omega)$ denotes for the volume (area) of Ω. When $\lambda = 0$, $p(\boldsymbol{x})$ degenerates to $h(\boldsymbol{x})$. On the contrary, $\lambda = +\infty$ means all positions in Ω have the same chance to be $\boldsymbol{x}^*$. The $u(\boldsymbol{x})$ can flatten $p(\boldsymbol{x})$ and increase the structural diversity of candidate solutions. Such a design facilitates the recovery of solution diversity lost caused by the refinement of $h(\boldsymbol{x})$ and explores new solutions.

Framework of OPT-GAN

How to achieve $p(\boldsymbol{x})$ in practice? Although reshaping $p(\boldsymbol{x})$ has the potential to approach optimum $\boldsymbol{x}^*$, this distribution could be arbitrary shape in high-dimensional space during optimization. Although there are many candidate models for constructing a distribution, the GAN is considered an effective universal distribution learner (Lu and Lu 2020; Grover, Dhar, and Ermon 2018) for multi-dimensional and arbitrary distributions. Thus, in this work, we design a GAN-based framework to carry out the formulation of Eq. 2.

The OPT-GAN consists of three components: solution generator G, bi-discriminators $D = \{D_I, D_R\}$, and historical best solution set $\boldsymbol{x}_{\text{opt}}$ (see Fig. 2). G learns the mixture distribution $p(\boldsymbol{x})$ by adversarially learning against D_I and D_R at the same time. $\boldsymbol{x}_{\textbf{opt}}$ serves as a knowledge base related to $h(\boldsymbol{x})$ and is continually maintained during optimization. The loss of G exquisitely contains the correction to $h(\boldsymbol{x})$ by D_I and correction to $u(\boldsymbol{x})$ by D_R, gradually enforcing G to approach their mixture distribution $p(\boldsymbol{x})$, provided proper training of D_I and D_R.

In addition, the mode collapse problem (Li et al. 2021) may lead to optimization failure. Although refining distribution toward $\boldsymbol{x}^*$ can be viewed as a "positive" mode collapse, the abnormal collapse during optimization still results in premature loss of solution diversity. Therefore, Wasserstein GAN with gradient penalty (WGAN-GP) (Gulrajani et al. 2017) is adopted to prevent premature convergence to local optima.

Exploitation Discriminator D_I is for exploitation by differentiating generated solutions by G from $\boldsymbol{x}_{\text{opt}}$, i.e., $h(\boldsymbol{x})$. It is fed a solution and outputs a value indicating which distribution the solution comes from.Its loss $\mathcal{L}_{\text{D}_\text{I}}$ is defined as

$$\begin{aligned} \mathcal{L}_{\text{D}_\text{I}} =& \mathbb{E}_\eta[D_I(G(\boldsymbol{\eta}))] - \mathbb{E}_{\boldsymbol{x}_{\text{samp}}}[D_I(\boldsymbol{x}_{\text{samp}})] \\ &+\beta\, \mathbb{E}_{\hat{x_I}}[(\|\nabla_{\hat{\boldsymbol{x_I}}} D_I(\hat{\boldsymbol{x_I}})\|_2 - 1)^2], \end{aligned} \quad (4)$$

where $\boldsymbol{x}_{\text{samp}}$ is sampled from $\boldsymbol{x}_{\text{opt}}$ (bootstrapping method). $\boldsymbol{\eta}$ is the random noise, and β is gradient penalty factor. $\hat{\boldsymbol{x_I}}$ is sampled uniformly along straight lines between $G(\boldsymbol{\eta})$ and $\boldsymbol{x}_{\text{samp}}$ (Gulrajani et al. 2017).

Exploration Discriminator D_R aims at differentiating sampled solutions by G, from samples of $u(\boldsymbol{x})$. It is associated with OPT-GAN's exploration ability. Similar to D_I, it outputs a value indicating which distribution a solution draws from. Its loss $\mathcal{L}_{\text{D}_\text{R}}$ is defined as

$$\begin{aligned} \mathcal{L}_{\text{D}_\text{R}} =& \mathbb{E}_\eta[D_R(G(\boldsymbol{\eta}))] - \mathbb{E}_{\boldsymbol{x}_{\text{uni}}}[D_R(\boldsymbol{x}_{\text{uni}})] \\ &+\beta\, \mathbb{E}_{\hat{x_R}}[(\|\nabla_{\hat{\boldsymbol{x_R}}} D_R(\hat{\boldsymbol{x_R}})\|_2 - 1)^2], \end{aligned} \quad (5)$$

where $\boldsymbol{x}_{\text{uni}}$ is from $u(\boldsymbol{x})$. $\hat{\boldsymbol{x_R}}$ is sampled along straight lines between $G(\boldsymbol{\eta})$ and $\boldsymbol{x}_{\text{uni}}$ (Gulrajani et al. 2017).

Solution Generator As a sampler, G generates candidate solutions $\boldsymbol{x}_{\text{G}}$ from the intrinsic distribution $p(\boldsymbol{x})$ by injecting the noises $\boldsymbol{\eta}$. According to Eq. 2, G mixes distribution $h(\boldsymbol{x})$ with $u(\boldsymbol{x})$ to balance the E-E trade-off. Thus, the generated solutions not only exploit the discovered regions, but also explore the whole searching domain from a global point of view. The loss function $\mathcal{L}_{\text{G}}$ consists of the adversarial loss against D_I and D_R, which is defined as

$$\mathcal{L}_{\text{G}} = \mathbb{E}_\eta[\frac{1}{1+\lambda} D_I(G(\boldsymbol{\eta})) + \frac{\lambda}{1+\lambda} D_R(G(\boldsymbol{\eta}))]. \quad (6)$$

[1]Table 1 lists the notations used in the paper.

Algorithm 1: OPT-GAN (see Appendix Algorithm 2 for the full version)

Input: The size of optimal set K; the batch size S; the population size for updating optimal set M; the maximum number of FEs $MAXFes$; the searching domain Ω; the number of iterations $GANIter$ and $DIter$ for GAN and discriminators, respectively.
Init: Initialize $\boldsymbol{x}_{\text{opt}} = \{(\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(K)}) | \boldsymbol{x}^{(s)} \sim U(\Omega),\ s = 1, 2, \cdots, K\}$; $MaxEpoch = \lceil (MAXFes - K)/M \rceil$.
Generator Pre-training:
Training solution generator G and discriminator D_R by Eq. 8 and Eq. 5, with $U(\Omega)$.
Distribution Reshaping:
for $epoch = 1$ **to** $MaxEpoch$ **do**
  for $iter_G = 1$ **to** $GANIter$ **do**
    for $iter_D = 1$ **to** $DIter$ **do**
      Training discriminator D_I by $\mathcal{L}_{\text{D}_\text{I}}$ in Eq. 4, with $\boldsymbol{x}_{\text{opt}}$.
      Training discriminator D_R by $\mathcal{L}_{\text{D}_\text{R}}$ in Eq. 5, with $U(\Omega)$.
    end for
    Training solution generator G by $\mathcal{L}_{\text{G}}$ in Eq. 6.
  end for
  Updating:
  $\boldsymbol{x}_{\text{G}} = G(\boldsymbol{\eta})$; $\boldsymbol{B} = \boldsymbol{x}_{\text{opt}} \cup \boldsymbol{x}_{\text{G}}$.
  Sort $\boldsymbol{B}$ according to $f(\boldsymbol{B})$ in ascending pattern.
  $\boldsymbol{x}_{\text{opt}} = \boldsymbol{B}^{(1:K)}$.
  Shrinking $\boldsymbol{x}_{\text{opt}}$ with K in Eq. 7.
end for
Return: $\arg\min f(\boldsymbol{x}_{\text{opt}})$

Knowledge Base Maintaining

How can the optimizer be guided toward the global optimum in the landscape? Functioning as a knowledge base, $\boldsymbol{x}_{\mathbf{opt}}$ is a critical component guiding the optimization direction. Solutions in $\boldsymbol{x}_{\text{opt}}$ are viewed as samples drawn from $h(\boldsymbol{x})$. According to Subsection Formulation, refining $h(\boldsymbol{x})$ is related to exploitation, and important in reshaping $p(\boldsymbol{x})$ toward the delta distribution $\delta_{x^*}(\boldsymbol{x})$. In OPT-GAN, $h(\boldsymbol{x})$ is iteratively refined by two strategies: updating and shrinking $\boldsymbol{x}_{\text{opt}}$.

Updating Since the information on landscape is stored in historical solutions, updating $\boldsymbol{x}_{\text{opt}}$ by incoming ones according to their goodness is indispensable to promote $h(\boldsymbol{x})$ at better regions. The generated solutions in $\boldsymbol{x}_{\text{G}}$ are firstly evaluated on f. Then, $\boldsymbol{x}_{\text{opt}}$ is replaced by the best K solutions in $\boldsymbol{x}_{\text{opt}} \cup \boldsymbol{x}_{\text{G}}$. This strategy ensures the superiority of $\boldsymbol{x}_{\text{opt}}$ in the historical solutions, and enables new knowledge flows into $\boldsymbol{x}_{\text{opt}}$ to refine $h(\boldsymbol{x})$. Actually, the updating strategy enables the model to be attracted by better regions.

Shrinking Ideally, a size-fixing or even size-increasing $\boldsymbol{x}_{\text{opt}}$ could save more knowledge about the landscape. Especially at the early stage, a large $\boldsymbol{x}_{\text{opt}}$ improves the diversity of solutions from a global point of view. However, as optimization goes on, refining $h(\boldsymbol{x})$ with a large $\boldsymbol{x}_{\text{opt}}$ could be trapped into a tediously long "tug-of-war", especially when the problem is multi-modal. Thus, considering the search budget, we design a shrinking strategy to balance E-E trade-off over time. This strategy gradually and smoothly reduces the size of $\boldsymbol{x}_{\text{opt}}$ to push OPT-GAN to focus on the interest region. The size K of $\boldsymbol{x}_{\text{opt}}$ at the t^{th} number of function evaluations (FEs) is defined as follows

$$K^{(t)} = \lceil K^{(0)^{(1-a\cdot(t/MAXFes))}} \rceil, \tag{7}$$

where $K^{(0)}$ denotes the initial size of $\boldsymbol{x}_{\text{opt}}$, $MAXFes$ is the maximum of FEs, and a is the shrinking rate. Specifically, $a = 0$ means a size-fixing $\boldsymbol{x}_{\text{opt}}$, and $a \geq 1$ means only the best solution stays at the end of optimization (Appendix C includes visualization of the effects of a).

Optimization

How do we use OPT-GAN to optimize a black-box problem? The optimization process consists of two successive stages: generator pretraining and distribution reshaping (see Algorithm 1 for pseudocode).

Generator Pre-training Facing a black-box problem, an optimizer should start with an unbiased $p(\boldsymbol{x})$ over the Ω to have a global field-of-view for avoiding premature convergence, i.e., $p(\boldsymbol{x})$ needs to be $u(\boldsymbol{x})$ initially. However, the intrinsic $p(\boldsymbol{x})$ in a randomly initialized generator hardly meets this requirement. Thus, G is pre-trained only with D_R for initialization (without consuming FEs). During this stage, the loss of D_R is still Eq. 5, but the loss of G is redefined as

$$\mathcal{L}_{\text{G}} = \mathbb{E}_{\eta}[D_R(G(\boldsymbol{\eta}))]. \tag{8}$$

Distribution Reshaping After generator pre-training, we start to search for the global optimum in Ω. The solution generator G with intrinsic $p(\boldsymbol{x})$ generates candidate solutions at each epoch (epoch denotes a single period in which OPT-GAN reshapes $p(\boldsymbol{x})$ and generates new candidates). It is for updating $\boldsymbol{x}_{\text{opt}}$ that will be learnt by G at next step. Ideally, the initial $p(\boldsymbol{x})$ is expected to be reshaped gradually towards the delta distribution $\delta_{x^*}(\boldsymbol{x})$.

In each epoch, G adversarially learns current $p(\boldsymbol{x})$ against D_I and D_R, under guidance of $\boldsymbol{x}_{\text{opt}}$ and samples from $u(\boldsymbol{x})$. Then, $\boldsymbol{x}_{\text{opt}}$ is maintained by updating and shrinking. This process is performed epoch by epoch to reshape $p(\boldsymbol{x})$ progressively. The found best solution is in the last $\boldsymbol{x}_{\text{opt}}$. Note that training GAN is usually commented as a difficult task. Thus, to stabilize the estimation of current $p(\boldsymbol{x})$, G, D_I, and D_R are sufficiently trained at each epoch.

Experiments

We examine the efficacy of OPT-GAN in terms of its broad-spectrum global optimization ability. The baseline methods are tested on the challenging black-box benchmarks from the COCO platform (Hansen et al. 2021), CEC'19 Benchmark Suite (Price et al. 2018), Conformal Bent Cigar (Liu et al. 2020a), and Simulationlib (Surjanovic and Bingham 2021). We also validate the efficacy on two real-world problems. Considering this study focuses on NN-based broad-spectrum optimizer, two groups of optimizers are adopted. The first group includes various broad-spectrum optimizers,

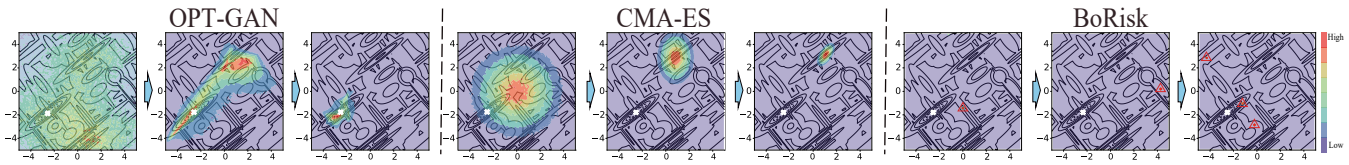

Figure 3: Comparison of visualized $p(\boldsymbol{x})$ during optimization between OPT-GAN, CMA-ES, and BoRisk (see Appendix E for the plotting details of $p(\boldsymbol{x})$). The white $\times$ denotes the x^*. The red $\triangle$ marks the peaks of $p(\boldsymbol{x})$ for the best view.

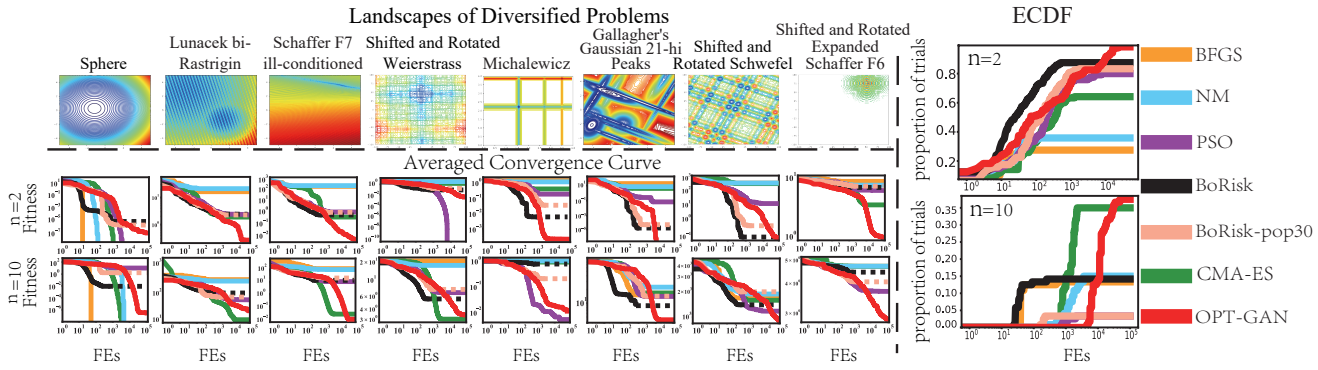

Figure 4: Average convergence curves and ECDF results of OPT-GAN and traditional optimizers on 8 benchmarks in 2 and 10 dimensions. The dotted line indicates that the optimizer stops because its elapsed time exceeds 3 hours. The PT for an optimizer equals the number of trials whose fitness reaches the precision divided by the total number of trials (defined in Appendix D.1).

including BFGS, CMA-ES, Nelder-Mead method (NM), PSO, Bayesian Optimization by Density-Ratio Estimation (BORE) (Tiao et al. 2021), Bayesian Optimization of Risk Measures (BoRisk) (Cakmak et al. 2020), and BoRisk with 30 candidates (BoRisk-pop30; see Appendix D.1). The second group includes state-of-the-art NN-based optimizers, Explicit Gradient Learning (EGL), GAN-based Solver (GBS), and Weighted Retraining (WR). In order to verify the framework's effectiveness and eliminate the interference of architectural tricks to evaluation results, we only use compact architectures (see Appendix D.1) to implement the prototypes of the generator and discriminators.

The detailed configurations about model specifications, benchmarks, and experimental settings are introduced in Appendix D.1, respectively. We also conducted the ablation analysis, convergence analysis, and hyperparameter analysis for OPT-GAN (see Appendix D.2, D.3, and D.4).

Visualization of the Learned Distribution To evaluate the ability of distribution estimation, Fig. 3 visualizes the learnt $p(\boldsymbol{x})$ of OPT-GAN, CMA-ES, and BoRisk (see Appendix E) during optimization on Gallagher's Gaussian 101-me Peaks function. CMA-ES builds the distribution by Gaussian prior directly, but easily misses the global optima at the later stage of optimization. The internal $p(\boldsymbol{x})$ in BoRisk presents a distribution with spikes because of the maximization of acquisition function. Although it tries to sample few thoughtful solutions at each epoch to save FEs, unfortunately, due to the cumulative error effect, BoRisk sampling may miss the optimal regions in the complex landscape. By contrast, OPT-GAN provides redundancy on multiple potential optimal regions via arbitrary shaped distribution, to obtain the holistic view of the problem. Benefiting from a delicate balance between exploration and exploitation on learning $p(\boldsymbol{x})$, OPT-GAN gradually samples some promising solutions and converges to global optima.

Comparison with Traditional Optimizers To verify the adaptability to diverse problems, the convergence curves and empirical cumulative distribution function (ECDF) (Hansen et al. 2021) of OPT-GAN are compared with various traditional optimizers (see Fig. 4). Considering an optimizer may have significantly different performance between different benchmark suits, we choose diversified benchmarks from different suits to display the influence of *a priori* assumption between different optimizers. As the available dimensions between benchmark suites differ, for consistency and fairness, the dimensionality of benchmarks is set to the shared available dimensions, i.e., 2 and 10. The Sphere has significant Gaussian characteristics suitable to BFGS, NM (Schraudolph, Yu, and Günter 2007; Gao and Han 2012), and CMA-ES. Unsurprisingly, their performance on the Sphere outperforms OPT-GAN. CMA-ES also shows outstanding performance on the 10-dimensional (10D) Lunacek bi-Rastrigin (local Gaussian-type), though it is worse than OPT-GAN on the 2-dimensional (2D) case. These imply that a successful matching between landscape characteristics and model prior can accelerate optimization.

However, for benchmarks without significant Gaussian shape, OPT-GAN gains ground on competitors. Although PSO performs better on the 2D Shifted and Rotated Weierstrass and 10D Michalewicz, its overall performance is inferior to OPT-GAN. BoRisk and BORE gradually lags behind OPT-GAN as the dimensionality increases. Overall, OPT-

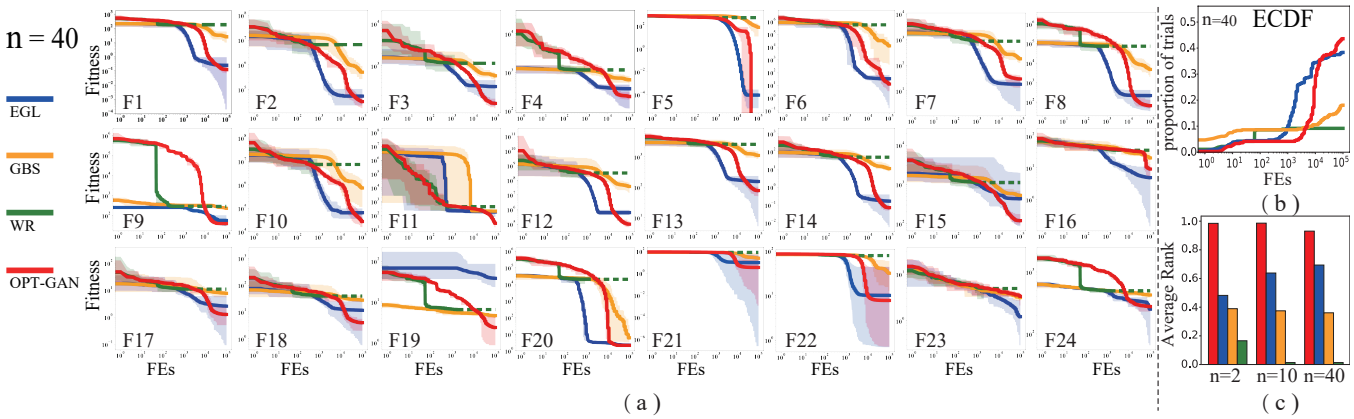

Figure 5: Performance results about (a) average convergence curves, (b) ECDF, and (c) average ranks of compared optimizers on the COCO test problems. The dotted line indicates that the optimizer stops due to elapsed time exceeding 3 hours. The shaded area reflects the "Range". The results on benchmarks with n=2 and n=10 are shown in Appendix D.6.

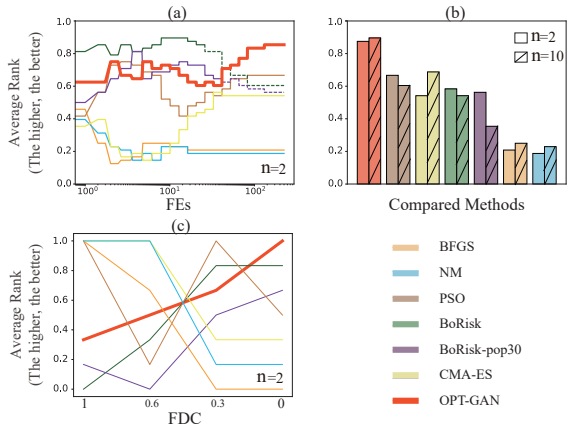

Figure 6: Average rank results versus FEs, dimensionality, and FDC. The ranks of optimizers are normalized into [0, 1], where the best rank of optimizers is 1; the worst is 0.

GAN shows the best or second-best performance in most benchmarks, reflecting its *broad-spectrum* adaptability. It can adapt to diversified landscapes, benefiting from GAN's universal distribution learning ability (Huster et al. 2021).

In terms of ECDF, although the proportion of trials (PT) of OPT-GAN increases slowly at the early stage, it shows the highest PT at the later stage, indicating it can find the global optimum with low FEs. The t-test results (see Appendix D.5) also reveal the superiority of OPT-GAN over competitors in terms of statistical significance.

Fig. 6 shows the results of average ranks versus FEs, dimensionality, and fitness distance correlation (FDC), separately. In Fig. 6 (a), OPT-GAN illustrates a better average rank than its opponents. Although BoRisk is better than OPT-GAN at the early stage, it is surpassed with the increase of FEs, because OPT-GAN searches different basins of attraction by balancing E-E trade-off. In Fig. 6 (b), OPT-GAN possesses the best average rank for 2D and 10D cases. The performance of BoRisks decreases as problems' dimensionality rises. In short, OPT-GAN presents the best average rank as FEs or dimension increases.

We also visualize the performance vs. global trend of landscape via fitness distance correlation (FDC) (Jones, Forrest et al. 1995), where a higher FDC means the stronger global trend (i.e., Gaussian property). Optimizers are tested on COCO benchmarks with different FDC values (around 1, 0.6, 0.3, 0). Fig. 6 (c) shows the average rank of CMA-ES decreases as FDC decreases due to its Gaussian assumptions. This phenomenon is also found on other benchmarks in the COCO platform. CMA-ES performs better on benchmarks with high FDC values (the first 14 benchmarks in the COCO platform). In contrast, OPT-GAN and BoRisks increase, which means they can adapt to diversified problems. Overall, OPT-GAN is less sensitive to the Gaussian trend of the problem.

Comparison with NN-based Optimizers Considering NN-based optimizers are commonly tested on the well-known COCO platform (Sarafian et al. 2020), the performance results of OPT-GAN and other NN-based optimizers are also summarized in Fig.5. For the convergence curves in Fig.5 (a), OPT-GAN shows the best performance compared with other NN-based optimizers on the majority of benchmarks. For separable problems (F1-F5) and the problems with conditioning (F6-F14), although the convergence speed of OPT-GAN is slower than those of EGL and GBS, it obtains the best fitness in the most cases because OPT-GAN is encouraged to explore different regions at the early stage and to focus on exploiting the basin of attraction at the later stage. In addition, EGL and GBS are easily trapped into local optimum owing to the lack of exploration for the basin of attraction. Thus, for multi-modal problems with many local optima (F15, F17-F22), OPT-GAN still displays the best results. The above phenomena reflect that OPT-GAN has better global optimization ability by balancing E-E trade-off during optimization. WR is difficult to find the best fitness under the limited resources due to its high time consumption. Thus, it performs poorly on most benchmarks. Moreover, the range of average convergence curves of OPT-GAN is smaller than its opponents on most benchmarks, indicat-

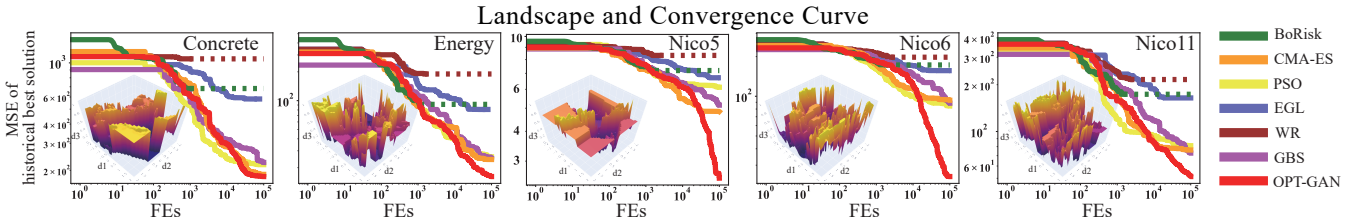

Figure 7: The 2D projected landscapes(Li et al. 2018) of optimizing NEEP and the average convergence curves of optimizers. Given the huge difference in mean square error (MSE) between different solutions in symbolic regression, the normalized rank of MSE (normalized into [0, 1], where the lowest MSE is 0 and the highest is 1) is used as the d3-axis to best view the landscapes. The dotted line indicates that the optimizer stops due to elapsed time exceeding 10 hours.

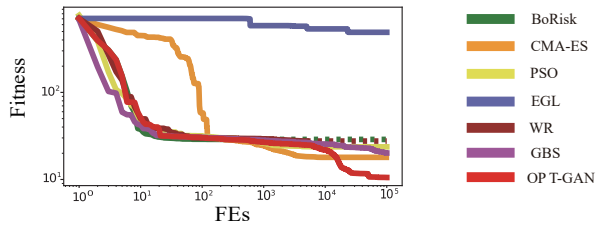

Figure 8: The average convergence curves of different optimizers in frequency modulation sound parameter identification problem. The dotted line indicates that the optimizer stops due to elapsed time exceeding 10 hours.

ing the more stable convergence process of OPT-GAN.

The results from Fig.5 (b) also show the superiority of OPT-GAN over other NN-based optimizers, especially with the increase of dimensions. Although the PT of OPT-GAN lags behind other competitors at the early stages, it boosts rapidly as the optimization proceeds. This is because the strategies to balance E-E trade-off take time to form a global perspective, and OPT-GAN pinpoints the global optimum gradually along the basin of attraction. In Fig.5 (c), the rank of EGL increases at higher dimensionality, whereas those of GBS and WR decrease, as EGL is mainly a local optimizer, avoiding time-consuming deliberative searching in high-dimensional landscapes. Nevertheless, OPT-GAN still exhibits the highest performance. Moreover, OPT-GAN is significantly better than other models on most COCO benchmarks by observing the t-test results (see Appendix D.6).

Performance on Real-world Problems We validate the performance of OPT-GAN by two real-world problems. One of them is the optimization of neural network-based symbolic regresser: NEEP (Anjum et al. 2019), which is a high dimensional optimization problem with very complex landscape. Another is Frequency Modulated Sounds Parameter Identification (FMSPI) problem (Herrera and Lozano 2000), which plays an essential role in modern music, e.g., emulation of acoustic musical instruments.

Fig. 7 shows the average convergence curves of different optimizers on NEEP with five datasets and their corresponding 2D projected landscapes (Li et al. 2018). The dimension of NEEP optimization problem is 400 for nico5, nico6, and nico11, and 640 for concrete and energy (see Appendix D.7 for more information of NEEP problem). It can be seen

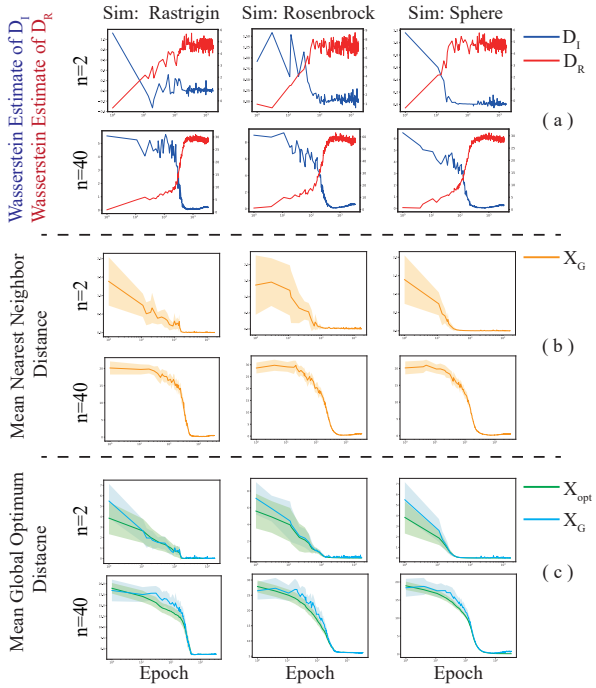

Figure 9: The convergence characteristics of OPT-GAN. The shaded area of curves reflects the standard deviation.

that each landscape is highly complex and multi-modal, posing a challenge to optimizers on balancing the E-E trade-off. EGL shows premature convergence on most datasets, as it relies on the gradient. Moreover, BoRisk and WR fail to find the optimum within the limited elapsed time because of their high time consumption. Although CMA-ES, PSO, and GBS display a similar performance with OPT-GAN on the Concrete, OPT-GAN exceeds them on the rest datasets. Actually, OPT-GAN exhibits the best overall performance on all datasets. These phenomena illustrate that OPT-GAN can balance E-E trade-off to search for global optimum on complex real-world landscapes.

Fig. 8 shows the average convergence curves of different optimizers in the FMSPI problem. This problem is a six-dimensional real-world problem with a complex multi-modal structure and strong variable interdependency. OPT-GAN presents the best performance because it can balance the exploration-exploitation trade-off. WR, BoRisk, PSO,

and GBS present similar worse performances. CMA-ES performs worst at the early stage. EGL presents the worst performance since it is easy to premature convergence due to a complex multi-modal landscape.

Convergence Analysis The convergence of OPT-GAN can be analyzed from three perspectives. The GAN framework is used to make sure the new solutions are sampled from the mixture distribution $p(\boldsymbol{x})$. The Knowledge Base Maintaining strategy ensure the solutions in $\boldsymbol{x}_{\text{opt}}$ could cluster to an optimum. The uniform distribution $u(\boldsymbol{x})$ promise the global optimum could be sampled out even if it is not covered by the $h(\boldsymbol{x})$.

To verify them, a series of experiments were carried out via several benchmarks. Fig. 9 (a) shows the Wasserstein estimate curves related to G and $D = \{D_I, D_R\}$. The blue curve gradually fluctuates down and converges to zero gradually; the red curve fluctuates up and converges to a stable value. That means the distribution represented by G can converge to a stable state over time. Fig. 9 (b) displays the mean nearest neighbor distance (MNND) between generated solutions. The orange curves gradually converge to zero from the large value as the iteration continues, illustrating that the generated solutions spread in different regions at the early stage and gradually converge together at the later stage. Fig. 9 (c) show shows the mean global optimum distances (MGOD) based on $\boldsymbol{x}_{\text{G}}$ and $\boldsymbol{x}_{\text{opt}}$, respectively. Both curves start with a large value; then, they fluctuate down and stabilize to a small value. That means the generated solutions could converge toward the region with global optimum. For more information, please refer to Appendix D.3.

Hyperparameter Analysis To analyze the effect of the hyperparameters of OPT-GAN and to explain how to tune them in practice, the main hyperparameters are analyzed and reported in Fig. 10. Several benchmarks with different preferences for exploration and exploitation ability are used. As a regulator in E-E trade-off, the large λ stresses exploring complex landscapes, e.g. multi-modal ones, whereas smaller value promotes exploiting the current attraction basin. The K and a balance the E-E trade off along the timeline. Small K and large a accelerate the transition from exploration to exploitation. See Appendix D.4 for more details.

Limitations We admit that OPT-GAN shares the same disadvantage with other NN-based models: the learning process is time-consuming. It may not be the best to efficiently solve a Gaussian-shape or unimodal problem due to its weak prior knowledge. However, to deal with massive types of real problem features (Leo Lopes 2013; Osaba et al. 2021), the optimizers with less prior such as OPT-GAN, are highly desirable to a broad extent. In addition, OPT-GAN also presents comparable time consumption with other NN-based optimizers (see Appendix D.8).

Conclusion

This study proposes a *broad-spectrum* global optimizer, named OPT-GAN, for diversified BBO problems. It consists of three collaborative components: solution generator, bi-discriminators, and knowledge base. Adversarial learning between the generator and bi-discriminators enables shaping arbitrary distributions, capturing diversified features in a relatively broad domain. It balances E-E trade-off by three organic and indispensable strategies: supervision of bi-discriminators, updating and shrinking of knowledge base, and pre-training of generator. Experiments show that it outperforms various BBO methods, including neural network-based ones, on diversified problems.

In the future, as we only used the most compact architecture as a prototype of OPT-GAN in the experiment, the network structure requires further attention to take full advantage of GAN's distribution learning ability. In addition, the landscape/problem-related computation methods or adaptive techniques for hyperparameters need to be studied. The extension of the application domain to various real-world problems will also be a concern.

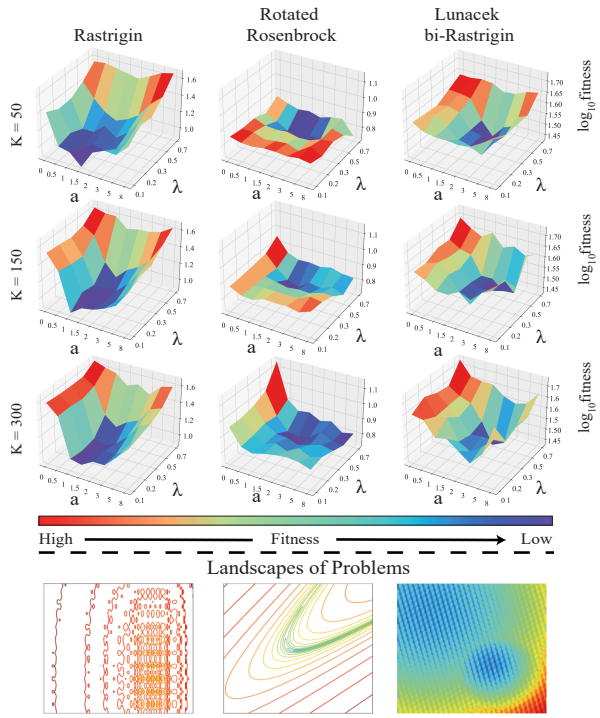

Figure 10: Hyperparameter analysis of $\boldsymbol{x}_{\text{opt}}$'s initial size K, shrinking rate a, and adjustment factor λ.

Acknowledgements

This work was supported by National Natural Science Foundation of China under Grant No. 61872419, No. 62072213, No. 61873324, No. 61903156. Shandong Provincial Natural Science Foundation No. ZR2022JQ30, No. ZR2022ZD01, No. ZR2020KF006. Taishan Scholars Program of Shandong Province, China, under Grant No. tsqn201812077. "New 20 Rules for University" Program of Jinan City under Grant No. 2021GXRC077.

References

Anjum, A.; Sun, F.; Wang, L.; and Orchard, J. 2019. A novel neural network-based symbolic regression method: neuro-encoded ex-

pression programming. In *International Conference on Artificial Neural Networks*, 373–386.

Antoniou, M.; and Papa, G. 2021. Differential evolution with estimation of distribution for worst-case scenario optimization. *Mathematics*, 9(17): 2137.

Audet, C.; and Hare, W. 2017. *Derivative-Free and Blackbox Optimization*. Springer Series in Operations Research and Financial Engineering.

Balandat, M.; Karrer, B.; Jiang, D.; Daulton, S.; Letham, B.; Wilson, A. G.; and Bakshy, E. 2020. BoTorch: a framework for efficient monte-carlo bayesian optimization. In *Advances in Neural Information Processing Systems*, volume 33, 21524–21538.

Brookes, D.; Park, H.; and Listgarten, J. 2019. Conditioning by adaptive sampling for robust design. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, 773–782.

Brookes, D. H.; and Listgarten, J. 2020. Design by adaptive sampling. arXiv:1810.03714.

Cakmak, S.; Astudillo Marban, R.; Frazier, P.; and Zhou, E. 2020. Bayesian optimization of risk measures. *Advances in Neural Information Processing Systems*, 33: 20130–20141.

Chen, Y.; Sun, X.; Gong, D.; Zhang, Y.; Choi, J.; and Klasky, S. 2017. Personalized search inspired fast interactive estimation of distribution algorithm and its application. *IEEE transactions on evolutionary computation*, 21(4): 588–600.

Cheng, R.; He, C.; Jin, Y.; and Yao, X. 2018. Model-based evolutionary algorithms: a short survey. *Complex & Intelligent Systems*, 4(4): 283–292.

Chetverikov, D.; Svirko, D.; Stepanov, D.; and Krsek, P. 2002. The trimmed iterative closest point algorithm. In *2002 International Conference on Pattern Recognition*, volume 3, 545–548.

Ciccazzo, A.; Di Pillo, G.; and Latorre, V. 2016. A SVM Surrogate Model-Based Method for Parametric Yield Optimization. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 35(7): 1224–1228.

De Bonet, J.; Isbell, C.; and Viola, P. 1996. MIMIC: finding optima by estimating probability densities. In *Advances in Neural Information Processing Systems*, volume 9.

Dong, W.; Wang, Y.; and Zhou, M. 2019. A latent space-based estimation of distribution algorithm for large-scale global optimization. *Soft Computing*, 23(13): 4593–4615.

Dulac-Arnold, G.; Levine, N.; Mankowitz, D. J.; Li, J.; Paduraru, C.; Gowal, S.; and Hester, T. 2021. Challenges of real-world reinforcement learning: definitions, benchmarks and analysis. *Machine Learning*, 110(9): 2419–2468.

Eberhart, R.; and Kennedy, J. 1995. A new optimizer using particle swarm theory. In *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, 39–43.

Eriksson, D.; Pearce, M.; Gardner, J.; Turner, R. D.; and Poloczek, M. 2019. Scalable global optimization via local bayesian optimization. In *Advances in Neural Information Processing Systems*, volume 32, 5496–5507.

Faury, L.; Calauzènes, C.; and Fercoq, O. 2019. Benchmarking GNN-CMA-ES on the BBOB Noiseless Testbed. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, 1928–1936.

Faury, L.; Calauzenes, C.; Fercoq, O.; and Krichen, S. 2019. Improving evolutionary strategies with generative neural networks. arXiv:1901.11271.

Fu, S.-W.; Liao, C.-F.; Tsao, Y.; and Lin, S.-D. 2019. MetricGAN: generative adversarial networks based black-box metric scores optimization for speech enhancement. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, 2031–2041.

Gao, F.; and Han, L. 2012. Implementing the Nelder-Mead simplex algorithm with adaptive parameters. *Computational Optimization and Applications*, 51(1): 259–277.

Gillhofer, M.; Ramsauer, H.; Brandstetter, J.; Schäfl, B.; and Hochreiter, S. 2019. A GAN based solver of black-box inverse problems. In *NeurIPS 2019 Workshop on Solving Inverse Problems with Deep Networks*.

Golovin, D.; Karro, J.; Kochanski, G.; Lee, C.; Song, X.; and Zhang, Q. 2020. Gradientless descent: high-dimensional zeroth-order optimization. In *International Conference on Learning Representations*.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. In *Advances in Neural Information Processing Systems*, volume 27.

Grover, A.; Dhar, M.; and Ermon, S. 2018. Flow-GAN: Combining Maximum Likelihood and Adversarial Learning in Generative Models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).

Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; and Courville, A. 2017. Improved training of wasserstein GANs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 5769–5779.

Gupta, A.; and Zou, J. 2019. Feedback GAN for DNA optimizes protein functions. *Nature Machine Intelligence*, 1(2): 105–111.

Hansen, N.; Auger, A.; Ros, R.; Mersmann, O.; Tušar, T.; and Brockhoff, D. 2021. COCO: a platform for comparing continuous optimizers in a black-box setting. *Optimization Methods and Software*, 36(1): 114–144.

Hauschild, M.; and Pelikan, M. 2011. An introduction and survey of estimation of distribution algorithms. *Swarm and Evolutionary Computation*, 1(3): 111–128.

He, C.; Huang, S.; Cheng, R.; Tan, K. C.; and Jin, Y. 2021. Evolutionary multiobjective optimization driven by generative adversarial networks. *IEEE Transactions on Cybernetics*, 51(6): 3129–3142.

Herrera, F.; and Lozano, M. 2000. Gradual distributed real-coded genetic algorithms. *IEEE Transactions on Evolutionary Computation*, 4(1): 43–63.

Holland, J. H. 1975. *Adaptation in natural and artificial systems*. The University of Michigan Press.

Huster, T.; Cohen, J. E. J.; Lin, Z.; Chan, K.; Kamhoua, C.; Leslie, N.; Chiang, C.-Y. J.; and Sekar, V. 2021. Pareto GAN: extending the representational power of GANs to heavy-tailed distributions. arXiv:2101.09113.

Jenatton, R.; Archambeau, C.; González, J.; and Seeger, M. 2017. Bayesian optimization with tree-structured dependencies. In *International Conference on Machine Learning*, 1655–1664.

Jones, D. R. 2001. A Taxonomy of Global Optimization Methods Based on Response Surfaces. *Journal of Global Optimization*, 21(4): 345–383.

Jones, T.; Forrest, S.; et al. 1995. Fitness distance correlation as a measure of problem difficulty for genetic algorithms. In *ICGA*, volume 95, 184–192.

Katoch, S.; Chauhan, S. S.; and Kumar, V. 2021. A review on genetic algorithm: past, present, and future. *Multimedia Tools and Applications*, 80(5): 8091–8126.

Kirkpatrick, S.; Gelatt, C. D.; and Vecchi, M. P. 1983. Optimization by Simulated Annealing. *Science*, 220(4598): 671–680.

Klein, A.; Falkner, S.; Bartels, S.; Hennig, P.; and Hutter, F. 2017. Fast bayesian optimization of machine learning hyperparameters on large datasets. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54, 528–536.

Kämpf, J. H.; and Robinson, D. 2009. A hybrid CMA-ES and HDE optimisation algorithm with application to solar energy potential. *Applied Soft Computing*, 9(2): 738–745.

LeCun, Y.; Touresky, D.; Hinton, G.; and Sejnowski, T. 1988. A theoretical framework for back-propagation. In *Proceedings of the 1988 connectionist models summer school*, volume 1, 21–28.

Leo Lopes, K. S.-M. 2013. Generating applicable synthetic instances for branch problems. *Operations Research*, 61(3): 563–577.

Li, H.; Xu, Z.; Taylor, G.; Studer, C.; and Goldstein, T. 2018. Visualizing the Loss Landscape of Neural Nets. In *Advances in Neural Information Processing Systems*, volume 31.

Li, W.; Fan, L.; Wang, Z.; Ma, C.; and Cui, X. 2021. Tackling mode collapse in multi-generator GANs with orthogonal vectors. *Pattern Recognition*, 110: 107646.

Liang, Y.; Ren, Z.; Yao, X.; Feng, Z.; Chen, A.; and Guo, W. 2020. Enhancing Gaussian Estimation of Distribution Algorithm by Exploiting Evolution Direction With Archive. *IEEE Transactions on Cybernetics*, 50(1): 140–152.

Lillicrap, T. P.; Hunt, J. J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; and Wierstra, D. 2015. Continuous control with deep reinforcement learning. arXiv:1509.02971.

Liu, C.; Sun, F.; Ni, Q.; Wang, L.; and Yang, B. 2020a. A Novel Graphic Bending Transformation on Benchmark. In *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 1538–1543.

Liu, D. C.; and Nocedal, J. 1989. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45(1): 503–528.

Liu, P.; and Liu, J. 2017. Multi-leader PSO (MLPSO): a new PSO variant for solving global optimization problems. *Applied Soft Computing*, 61: 256–263.

Liu, Y.; Zheng, Y.; Lu, J.; Cao, J.; and Rutkowski, L. 2020b. Constrained quaternion-variable convex optimization: a quaternion-valued recurrent neural network approach. *IEEE Transactions on Neural Networks and Learning Systems*, 31(3): 1022–1035.

Loshchilov, I.; and Hutter, F. 2016. CMA-ES for Hyperparameter Optimization of Deep Neural Networks. arXiv:1604.07269.

Lu, Y.; and Lu, J. 2020. A Universal Approximation Theorem of Deep Neural Networks for Expressing Probability Distributions. volume 33, 3094–3105.

Luo, N.; Qian, F.; Zhao, L.; and Zhong, W.-m. 2012. Gaussian process assisted coevolutionary estimation of distribution algorithm for computationally expensive problems. *Journal of Central South University*, 19(2): 443–452.

Mania, H.; Guy, A.; and Recht, B. 2018. Simple random search of static linear policies is competitive for reinforcement learning. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, volume 31, 1805–1814.

Nelder, J. A.; and Mead, R. 1965. A Simplex Method for Function Minimization. *The Computer Journal*, 7(4): 308–313.

Osaba, E.; Villar-Rodriguez, E.; Del Ser, J.; Nebro, A. J.; Molina, D.; LaTorre, A.; Suganthan, P. N.; Coello Coello, C. A.; and Herrera, F. 2021. A Tutorial On the design, experimentation and application of metaheuristic algorithms to real-World optimization problems. *Swarm and Evolutionary Computation*, 64: 100888.

Price, K. V.; Awad, N. H.; Ali, M. Z.; and Suganthan, P. N. 2018. Problem definitions and evaluation criteria for the 100-Digit challenge special session and competition on single objective numerical optimization. Technical report, Nanyang Technological University.

Raponi, E.; Wang, H.; Bujny, M.; Boria, S.; and Doerr, C. 2020. High dimensional Bayesian optimization assisted by principal component analysis. In *International Conference on Parallel Problem Solving from Nature*, 169–183.

Rubinstein, R. Y.; and Kroese, D. P. 2004. *The cross-entropy method: a unified approach to combinatorial optimization, Monte-Carlo simulation, and machine learning*, volume 133 of *Information Science and Statistics*. Springer.

Sakamoto, N.; Semmatsu, E.; Fukuchi, K.; Sakuma, J.; and Akimoto, Y. 2020. Deep generative model for non-convex constraint handling. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*, 636–644.

Salimans, T.; Ho, J.; Chen, X.; Sidor, S.; and Sutskever, I. 2017. Evolution strategies as a scalable alternative to reinforcement learning. arXiv:1703.03864.

Sarafian, E.; Sinay, M.; Louzoun, Y.; Agmon, N.; and Kraus, S. 2020. Explicit gradient learning for black-box optimization. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, 8480–8490.

Schraudolph, N. N.; Yu, J.; and Günter, S. 2007. A stochastic Quasi-Newton Method for online convex optimization. In *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics*, volume 2, 436–443.

Serofino, L. 2014. Optimizing without derivatives: what does the no free lunch theorem actually say? *Notices of the American Mathematical Society*, 61(07): 750.

Shahriari, B.; Swersky, K.; Wang, Z.; Adams, R. P.; and De Freitas, N. 2015. Taking the human out of the loop: a review of Bayesian optimization. *Proceedings of the IEEE*, 104(1): 148–175.

Surjanovic, S.; and Bingham, D. 2021. Virtual library of simulation experiments: test functions and datasets. http://www.sfu.ca/~ssurjano. Accessed: 2021-12-16.

Thor, M.; Kulvicius, T.; and Manoonpong, P. 2021. Generic neural locomotion control framework for legged robots. *IEEE Transactions on Neural Networks and Learning Systems*, 32(9): 4013–4025.

Tiao, L. C.; Klein, A.; Seeger, M. W.; Bonilla, E. V.; Archambeau, C.; and Ramos, F. 2021. BORE: Bayesian Optimization by Density-Ratio Estimation. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, 10289–10300.

Torczon, V. 1997. On the convergence of pattern search algorithms. *SIAM Journal on optimization*, 7(1): 1–25.

Tran, H.; and Zhang, G. 2020. AdaDGS: an adaptive black-box optimization method with a nonlocal directional Gaussian smoothing gradient. arXiv:2011.02009.

Tripp, A.; Daxberger, E.; and Hernández-Lobato, J. M. 2020. Sample-efficient optimization in the latent space of deep generative models via weighted retraining. In *Advances in Neural Information Processing Systems*, volume 33, 11259–11272.

Wolpert, D.; and Macready, W. 1997. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1): 67–82.

Zhou, C.; Hou, C.; Wei, X.; and Zhang, Q. 2014. Improved hybrid optimization algorithm for 3D protein structure prediction. *Journal of Molecular Modeling*, 20(7): 2289.