NONPARAMETRIC DISTRIBUTIONAL BLACK-BOX OPTIMIZATION VIA DIFFUSION PROCESS

Yueming Lyu [1,2*] **Atsushi Nitanda**[1,2,3] **Ivor W. Tsang**[1,2,3]
[1]Centre for Frontier AI Research, Agency for Science, Technology and Research, Singapore
[2]Institute of High Performance Computing, Agency for Science, Technology and Research, Singapore
[3]College of Computing and Data Science, Nanyang Technological University, Singapore

{Lyu_Yueming,Atsushi_Nitanda,Ivor_Tsang}@cfar.a-star.edu.sg

ABSTRACT

Sampling-based black-box optimization, e.g., zeroth-order optimization and Evolution strategy, is important for the material design, molecular design and etc. However, existing sampling-based black-box optimization methods only employ simple parametric distribution, typically Gaussian distribution, as the sampling distribution to generate queries. This limits the capabilities of modeling complex distribution to generate good candidates and influence the query efficiency. In this work, we propose a novel nonparametric black-box optimization method that performs proximal distributional update for sampling. Particularly, we derive the closed-form update rule based on the diffusion process (e.g., Ornstein–Uhlenbeck process). Our sampling and updating method supports black-box target function $f(\cdot)$ without accessing the ∇f, which is critical for our nonparametric distributional black-box optimization.

1 INTRODUCTION

Black-box optimization has demonstrated its success in many recent applications, such as prompt fine-tuning for large language models (Sun et al., 2022b;a), policy search for robot control and reinforcement learning (Choromanski et al., 2019; Lizotte et al., 2007; Barsce et al., 2017; Salimans et al., 2017), automatic hyper-parameters tuning in machine learning problems (Snoek et al., 2012), black-box architecture search in engineering design (Wang & Shan, 2007), drug discovery (Negoescu et al., 2011) and accelerated simulation for scientific discovery (Maddox et al., 2021; Hernández-Lobato et al., 2017), etc. Many efforts have been made for black-box optimization in the literature, including Bayesian optimization (BO) methods (Srinivas et al., 2010; Gardner et al., 2017; Nayebi et al., 2019), stochastic optimization methods like evolution strategies (ES) (Back et al., 1991; Hansen, 2006; Wierstra et al., 2014; Lyu & Tsang, 2021; Lyu, 2023) and genetic algorithms (Srinivas & Patnaik, 1994; Mirjalili & Mirjalili, 2019).

Stochastic optimization methods, e.g., ES (Rechenberg & Eigen, 1973; Nesterov & Spokoiny, 2017), natural evolution strategies (NES) (Wierstra et al., 2014), CMAES (Hansen, 2006), and implicit natural gradient optimizer (INGO) (Lyu & Tsang, 2021), typically sampling form Gaussian distribution and approximate the (natural) gradient for the update of the Gaussian distribution parameters for continuous optimization. However, the requirement of parametric sampling distributions, e.g., Gaussian distribution, may suffer from handling complex distribution. As a result, this may limit the exploration ability and lead to the shallow local optimum.

Recently, diffusion models have shown a great success to generate samples from complex distributions Song et al. (2020). Futhermore, many works investigate the theoretical properties of diffusion model alignment (Kawata et al., 2025), and diffusion model target generation (Lyu et al., 2024; Tan et al., 2025). This inspires us to take advantage of the diffusion model to handle the complex distributions in black-box optimization. Recently, Yang et al. (2020) proposed a particale-based ditributional optimization algorithm under variational transport form. However, it cannot directly handle black-box optimization.

[*]Corresponding author: Yueming Lyu (Lyu_Yueming@cfar.a-star.edu.sg)

In this paper, we propose a novel nonparametric distributional black-box optimization algorithm. Our method performs proximal distributional update and takes advantage of the diffusion SDE to sample from the nonparametric distribution. We derive the closed-form sampling and update rule that supports black-box target function $f(\cdot)$ without accessing the gradient ∇f.

2 OUR METHODS

2.1 PROXIMAL DISTRIBUTIONAL UPDATE

Note that the optimization problem of $f(\boldsymbol{x})$ w.r.t. $\boldsymbol{x}$ is equivalent to the optimization problem w.r.t. the Dirac distribution as below:

$$\min_{\boldsymbol{x}\in\mathbb{R}^d} f(\boldsymbol{x}) \Leftrightarrow \min_{\delta_{\boldsymbol{x}}} \mathbb{E}_{X\sim\delta_{\boldsymbol{x}}}[f(X)]. \tag{1}$$

Thus, to minimize the target function $f(\cdot)$, we can optimize an augmented problem as Eq.(2)

$$\min_{p\in\mathcal{P}} \mathbb{E}_{X\sim p}[f(X)]. \tag{2}$$

To optimize the above problem, we propose a proximal distributional update by iteratively solving

$$p^{k+1} = \arg\min_{p\in\mathcal{P}} \left\{\mathbb{E}_p[f(X)] + \lambda\mathbf{KL}(p||p^k)\right\}, \tag{3}$$

where p^k denotes the distribution at k^{th} iteration.

The optimal distribution that minimizes the problem in Eq.(3) has the following form:

$$p^{k+1}(\boldsymbol{x}) \propto p^k(\boldsymbol{x})e^{-\frac{f(\boldsymbol{x})}{\lambda}} \tag{4}$$

We now show the property of our proximal distributional update. From the update rule of p^{k+1} for each k, we know that

$$p^{k+1}(\boldsymbol{x}) \propto p^k(\boldsymbol{x})e^{-\frac{f(\boldsymbol{x})}{\lambda}} \propto p^{k-1}(\boldsymbol{x})e^{-\frac{2f(\boldsymbol{x})}{\lambda}} \propto \cdots \propto p^0(\boldsymbol{x})e^{-\frac{(k+1)f(\boldsymbol{x})}{\lambda}} \tag{5}$$

Thus, we know the distribution p^k has the following form:

$$p^k(\boldsymbol{x}) \propto p^0(\boldsymbol{x})e^{-\frac{f(\boldsymbol{x})}{\lambda/k}} \tag{6}$$

This shows that p^k is a solution of the problem Eq.(7)

$$p^k = \arg\min_{p\in\mathcal{P}} \left\{\mathbb{E}_p[f(X)] + \lambda_k\mathbf{KL}(p||p^0)\right\}, \tag{7}$$

where $\lambda_k = \frac{\lambda}{k}$ and p^0 is the prior distribution.

We can see that the parameter λ_k that controls the regularziation decays in a $O(\frac{1}{k})$ rate. For $f(\cdot)$ with uniqe optimal solution, the optimal distribution of problem (7) gradually converges to the Dirac distribution $\delta_{\boldsymbol{x}^*}$ at the minimum solution $\boldsymbol{x}^*$ of $f(\cdot)$.

2.2 SAMPLING VIA DIFFUSION PROCESS

There are two challenge points for our proximal distributional update: one is that p_k is a complex non-parametric distribution, which is not easy to generate samples. The other challenge is that the target function $f(\boldsymbol{x})$ is black-box, where the gradient ∇f is not accessible. As a result, the standard Langevin Monte Carlo sampling cannot be used due to the lack of gradient ∇f.

To address these two challenges, we employ the diffusion process (e.g., Ornstein–Uhlenbeck process) to sample from p^{k+1} as each iteration. Consider the forward stochastic differential equation (SDE) of the diffusion model (Song et al., 2020) as below

$$d\boldsymbol{x} = u(\boldsymbol{x}, t)dt + g(t)d\boldsymbol{w} \tag{8}$$

where $\boldsymbol{w}$ is the standard Wiener process (Brown motion). The related backward SDE to recover p^{k+1} is given as follows:

$$d\boldsymbol{x} = [u(\boldsymbol{x}, t) - g(t)^2 \nabla \log(p_t^{k+1}(\boldsymbol{x}))]dt + g(t)d\bar{\boldsymbol{w}} \tag{9}$$

where $\bar{\boldsymbol{w}}$ denotes a standard Wiener process when time flows backwards from T to 0.

The score function $\nabla \log(p_t^{k+1}(\boldsymbol{x}_t))$ can be derived as:

$$\nabla \log(p_t^{k+1}(\boldsymbol{x}_t)) = \nabla \log(\int p^k(\boldsymbol{x}_0) p(\boldsymbol{x}_t|\boldsymbol{x}_0) e^{-\frac{f(\boldsymbol{x}_0)}{\lambda}} \, \mathrm{d}\boldsymbol{x}_0) \tag{10}$$

Particularly, we employ the VP diffusion process (Song et al., 2020). The forward SDE can be written as

$$d\boldsymbol{x} = -\frac{1}{2}\beta(t)\boldsymbol{x}dt + \sqrt{\beta(t)}d\boldsymbol{w} \tag{11}$$

The backward SDE is given as follows:

$$d\boldsymbol{x} = [-\frac{1}{2}\beta(t)\boldsymbol{x} - \beta(t)\nabla \log(p_t^{k+1}(\boldsymbol{x}))]dt + \sqrt{\beta(t)}d\bar{\boldsymbol{w}} \tag{12}$$

The conditional probability $p(\boldsymbol{x}_t|\boldsymbol{x}_0)$ is a Gaussian distribution $\mathcal{N}(\alpha_t \boldsymbol{x}_0, \sigma_t^2 \boldsymbol{I})$, where $\alpha_t = e^{-\frac{1}{2}\int_0^t \beta(s)ds}$ and $\sigma_t = \sqrt{1-\alpha_t^2}$. Then, we can achieve the score function

$$\nabla \log(p_t^{k+1}(\boldsymbol{x}_t)) = \mathbb{E}_{p^k(\boldsymbol{x}_0|\boldsymbol{x}_t)}\left[\frac{\alpha_t \boldsymbol{x}_0 - \boldsymbol{x}_t}{\sigma_t^2} e^{-\frac{f(\boldsymbol{x}_0)}{\lambda}} / C_t^k\right] \tag{13}$$

where $p^k(\boldsymbol{x}_0|\boldsymbol{x}_t)$ denotes the posterior distribution at time t in the k^{th} iteration, and $C_t^k = \int p^k(\boldsymbol{x}_0|\boldsymbol{x}_t) e^{-\frac{f(\boldsymbol{x}_0)}{\lambda}} \mathrm{d}\boldsymbol{x}_0$. More detailed derivation can be found in the Appendix.

Discrete Time: In practice, we need to discrete the time of SDE in Eq.(12) to take samples. In particular, we employ the DPM++ SDE solver in (Lu et al., 2022) to take samples. The concrete sampling rule is given as follows:

$$\boldsymbol{x}_{t-1} = \frac{\sigma_{t-1}}{\sigma_t} e^{-h} \boldsymbol{x}_t + \alpha_{t-1}(1 - e^{-2h})\tilde{\boldsymbol{x}}_0(t) + \sigma_{t-1}\sqrt{1 - e^{-2h}} \, \boldsymbol{z} \tag{14}$$

where $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$, and $h = \log(\frac{\alpha_{t-1}}{\sigma_{t-1}}) - \log(\frac{\alpha_t}{\sigma_t})$, and $\tilde{\boldsymbol{x}}_0(t)$ is given as follows:

$$\tilde{\boldsymbol{x}}_0(t) = \mathbb{E}_{p^k(\boldsymbol{x}_0|\boldsymbol{x}_t)}\left[\boldsymbol{x}_0 \cdot e^{-\frac{f(\boldsymbol{x}_0)}{\lambda}} / C_t^k\right] \tag{15}$$

Finite Samples: We maintain a set of finite particles and employ Gaussian smooth estimation (Gaussian kernel density estimation) to approximate $p^k(\boldsymbol{x}_0)$ at each iteration k. More specifically, given N particles $\{\boldsymbol{x}_{(1)}^k, \cdots, \boldsymbol{x}_{(N)}^k\}$ at iteration k, we employ Eq.(16) to approximate $p^k(\boldsymbol{x}_0)$

$$p^k(\boldsymbol{x}_0) \approx \frac{1}{N}\sum_{i=1}^{N} \frac{1}{(2\pi\sigma^2)^{d/2}} e^{-\frac{1}{2\sigma^2}\|\boldsymbol{x}_0 - \boldsymbol{x}_{(i)}^k\|_2^2} \tag{16}$$

where σ is a bandwidth parameter to control the smoothness.

Because the conditional probability $p(\boldsymbol{x}_t|\boldsymbol{x}_0)$ is a Gaussian distribution $\mathcal{N}(\alpha_t \boldsymbol{x}_0, \sigma_t^2 \boldsymbol{I})$, by the Bayes rule, we know the posterior distribution $p^k(\boldsymbol{x}_0|\boldsymbol{x}_t)$ is a Gaussian mixture distribution:

$$p^k(\boldsymbol{x}_0|\boldsymbol{x}_t) \approx \sum_{i=1}^{N} w_i \mathcal{N}(\boldsymbol{x}_0; \mu_i, \bar{\sigma}^2 \boldsymbol{I}) \tag{17}$$

where the posterior mean is given as below:

$$\mu_i = \left(\frac{\alpha_t^2}{\sigma_t^2} + \sigma^{-2}\right)^{-1} \left(\frac{\alpha_t}{\sigma_t^2}\boldsymbol{x}_t + \sigma^{-2}\boldsymbol{x}_{(i)}^k\right) \tag{18}$$

and posterior variance is given as below:

$$\bar{\sigma}^2 = (\alpha_t^2/\sigma_t^2 + \sigma^{-2})^{-1} \tag{19}$$

Algorithm 1: Diffusion BO

Input: Number of iterations K, number of particles N, batch size M, number of diffusion sampling steps T, diffusion scheme $\{\alpha_t\}_0^T$, bandwidth σ.

Output: The particle set $\{\boldsymbol{x}_{(1)}^K, \cdots, \boldsymbol{x}_{(N)}^K\}$. And $\hat{\boldsymbol{x}}^*$ such that $f(\hat{\boldsymbol{x}}^*)$ achieves the minimum during the sampling process.

Initialize N particles $\{\boldsymbol{x}_{(1)}^0, \cdots, \boldsymbol{x}_{(N)}^0\}$
for $k \leftarrow 0$ **to** $K-1$ **do**
  do parallel for N **particles update**
    Sample $\boldsymbol{x}_T \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$.
    for $t \leftarrow T$ **to** 1 **do**
      Take M i.i.d. samples $\{\boldsymbol{x}_0^{(1)}, \cdots, \boldsymbol{x}_0^{(M)}\}$ from $p^k(\boldsymbol{x}_0|\boldsymbol{x}_t)$ in Eq.(17).
      Perform batch query to achieve black-box scores $\{f(\boldsymbol{x}_0^{(1)}), \cdots, f(\boldsymbol{x}_0^{(M)})\}$
      Compute $\hat{\boldsymbol{x}}_0(t) = \sum_{j=1}^{M} c_j \boldsymbol{x}_0^{(j)}$ to approximate $\tilde{\boldsymbol{x}}_0(t)$.
      Perform diffusion sampling update $\boldsymbol{x}_{t-1}$ by Eq.(14) with the approximation $\hat{\boldsymbol{x}}_0(t)$.
    Set particle $\boldsymbol{x}_{(i)}^{k+1} = \boldsymbol{x}_0$ with process ID i.
  Collect particle set $\{\boldsymbol{x}_{(1)}^{k+1}, \cdots, \boldsymbol{x}_{(N)}^{k+1}\}$

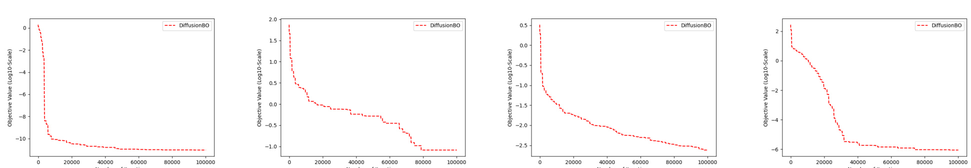
And the weight w_i is given as follows:

$$w_i = \frac{\exp\left(-\frac{\|\boldsymbol{x}_t - \alpha_t \boldsymbol{x}_{(i)}^k\|_2^2}{2(\alpha_t^2\sigma^2 + \sigma_t^2)}\right)}{\sum_{i=1}^{N} \exp\left(-\frac{\|\boldsymbol{x}_t - \alpha_t \boldsymbol{x}_{(i)}^k\|_2^2}{2(\alpha_t^2\sigma^2 + \sigma_t^2)}\right)} \tag{20}$$

Then, we can take M samples $\{\boldsymbol{x}_0^{(1)}, \cdots, \boldsymbol{x}_0^{(M)}\}$ from the posterior $p^k(\boldsymbol{x}_0|\boldsymbol{x}_t)$ to approximate $\tilde{\boldsymbol{x}}_0(t)$ by $\hat{\boldsymbol{x}}_0(t)$ as $\hat{\boldsymbol{x}}_0(t) = \sum_{j=1}^{M} c_j \boldsymbol{x}_0^{(j)}$, where the weight c_j is given as $c_j = \frac{\exp(-f(\boldsymbol{x}_0^{(j)})/\lambda)}{\sum_{j=1}^{M}\exp(-f(\boldsymbol{x}_0^{(j)})/\lambda)}$. Our detailed algorithm for distributional black-box optimization is presented in Algorithm 1.

3 Experiments

We evaluate our Diffusion-BO on challenging benchmark test functions: Levy, Rastrigin, Nesterov, and Rosenbrock. Rastrigin, Levy and Rosenbrock are smooth multi-mode functions, and Nesterov is a non-smooth function. These functions are very challenging benchmarks for black-box optimization. The problems are listed in Table 1 in Appendix. All the problems have minimum $f(\boldsymbol{x}^*) = 0$. The mean objective value (in log10-scale) over 10 trials is presented in Fig 1. It shows that our Diffusion-BO can decrease the objective to a low value, especially on Levy and Rosenbrock, even to a near optimal.



(a) Levy (b) Rasterigin10 (c) Nesterov(non-smooth) (d) Rosenbrock

Figure 1: Mean objective values (in log10 scale) v.s. number of iterations over 10 independent trials

4 Conclusions and Future Work

We proposed a novel nonparametric black-box optimization method that performs proximal distributional update for sampling. We derived the closed-form update rule based on the diffusion process that supports black-box target function $f(\cdot)$ without accessing the ∇f. We leave the convergence analysis of our method as one of our future work.

ACKNOWLEDGMENTS

This work is supported by Career Development Fund (CDF) of the Agency for Science, Technology and Research (A*STAR) (No: C243512014). This research is also supported by the National Research Foundation, Singapore, Infocomm Media Development Authority under its Trust Tech Funding Initiative, and the Ministry of Digital Development and Information under the AI Visiting Professorship Programme (award number AIVP-2024-004). Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore, Infocomm Media Development Authority, and the Ministry of Digital Development and Information.

REFERENCES

Thomas Back, Frank Hoffmeister, and Hans-Paul Schwefel. A survey of evolution strategies. In *Proceedings of the fourth international conference on genetic algorithms*, volume 2. Morgan Kaufmann Publishers San Mateo, CA, 1991.

Juan Cruz Barsce, Jorge A Palombarini, and Ernesto C Martínez. Towards autonomous reinforcement learning: Automatic setting of hyper-parameters using bayesian optimization. In *Computer Conference (CLEI), 2017 XLIII Latin American*, pp. 1–9. IEEE, 2017.

Krzysztof Choromanski, Aldo Pacchiano, Jack Parker-Holder, and Yunhao Tang. From complexity to simplicity: Adaptive es-active subspaces for blackbox optimization. *arXiv:1903.04268*, 2019.

Jacob Gardner, Chuan Guo, Kilian Weinberger, Roman Garnett, and Roger Grosse. Discovering and exploiting additive structure for bayesian optimization. In *Artificial Intelligence and Statistics*, pp. 1311–1319. PMLR, 2017.

Nikolaus Hansen. The cma evolution strategy: a comparing review. In *Towards a new evolutionary computation*, pp. 75–102. Springer, 2006.

José Miguel Hernández-Lobato, James Requeima, Edward O Pyzer-Knapp, and Alán Aspuru-Guzik. Parallel and distributed thompson sampling for large-scale accelerated exploration of chemical space. In *International conference on machine learning*, pp. 1470–1479. PMLR, 2017.

Ryotaro Kawata, Kazusato Oko, Atsushi Nitanda, and Taiji Suzuki. Direct distributional optimization for provable alignment of diffusion models, 2025. URL `https://arxiv.org/abs/2502.02954`.

Daniel J Lizotte, Tao Wang, Michael H Bowling, and Dale Schuurmans. Automatic gait optimization with gaussian process regression. In *IJCAI*, volume 7, pp. 944–949, 2007.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *arXiv preprint arXiv:2211.01095*, 2022.

Yueming Lyu. Fast rank-1 lattice targeted sampling for black-box optimization. In *Neural Information Processing Systems*, 2023. URL `https://api.semanticscholar.org/CorpusID:268030728`.

Yueming Lyu and Ivor W Tsang. Black-box optimizer with stochastic implicit natural gradient. In *Machine Learning and Knowledge Discovery in Databases. Research Track: European Conference, ECML PKDD 2021, Bilbao, Spain, September 13–17, 2021, Proceedings, Part III 21*, pp. 217–232. Springer, 2021.

Yueming Lyu, Kim Yong Tan, Yew Soon Ong, and Ivor W. Tsang. Covariance-adaptive sequential black-box optimization for diffusion targeted generation, 2024. URL `https://arxiv.org/abs/2406.00812`.

Wesley Maddox, Qing Feng, and Max Balandat. Optimizing high-dimensional physics simulations via composite bayesian optimization. *arXiv preprint arXiv:2111.14911*, 2021.

Seyedali Mirjalili and Seyedali Mirjalili. Genetic algorithm. *Evolutionary Algorithms and Neural Networks: Theory and Applications*, pp. 43–55, 2019.

Amin Nayebi, Alexander Munteanu, and Matthias Poloczek. A framework for bayesian optimization in embedded subspaces. In *International Conference on Machine Learning*, pp. 4752–4761. PMLR, 2019.

Diana M Negoescu, Peter I Frazier, and Warren B Powell. The knowledge-gradient algorithm for sequencing experiments in drug discovery. *INFORMS Journal on Computing*, 23(3):346–363, 2011.

Yurii Nesterov and Vladimir Spokoiny. Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics*, 17(2):527–566, 2017.

Ingo Rechenberg and M. Eigen. *Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. PhD thesis, 1973.

Tim Salimans, Jonathan Ho, Xi Chen, Szymon Sidor, and Ilya Sutskever. Evolution strategies as a scalable alternative to reinforcement learning. *arXiv preprint arXiv:1703.03864*, 2017.

Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. In *NeurIPS*, pp. 2951–2959, 2012.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Mandavilli Srinivas and Lalit M Patnaik. Genetic algorithms: A survey. *computer*, 27(6):17–26, 1994.

Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *ICML*, 2010.

Tianxiang Sun, Zhengfu He, Hong Qian, Yunhua Zhou, Xuanjing Huang, and Xipeng Qiu. Bbtv2: Towards a gradient-free future with large language models. In *Proceedings of EMNLP*, 2022a.

Tianxiang Sun, Yunfan Shao, Hong Qian, Xuanjing Huang, and Xipeng Qiu. Black-box tuning for language-model-as-a-service. In *Proceedings of ICML*, 2022b.

Kim Yong Tan, Yueming Lyu, Ivor Tsang, and Yew-Soon Ong. Fast direct: Query-efficient online black-box guidance for diffusion-model target generation, 2025. URL `https://arxiv.org/abs/2502.01692`.

G Gary Wang and Songqing Shan. Review of metamodeling techniques in support of engineering design optimization. *Journal of Mechanical design*, 129(4):370–380, 2007.

Daan Wierstra, Tom Schaul, Tobias Glasmachers, Yi Sun, Jan Peters, and Jürgen Schmidhuber. Natural evolution strategies. *The Journal of Machine Learning Research*, 15(1):949–980, 2014.

Zhuoran Yang, Yufeng Zhang, Yongxin Chen, and Zhaoran Wang. Variational transport: A convergent particle-basedalgorithm for distributional optimization. *arXiv preprint arXiv:2012.11554*, 2020.

Appendix

A Derivation of Eq.(13)

$$\nabla \log(p_t^{k+1}(\boldsymbol{x}_t)) = \nabla \log(\int p^k(\boldsymbol{x}_0)p(\boldsymbol{x}_t|\boldsymbol{x}_0)e^{-\frac{f(\boldsymbol{x}_0)}{\lambda}} \, \mathrm{d}\boldsymbol{x}_0) \tag{21}$$

$$= \frac{\int p^k(\boldsymbol{x}_0)\nabla p(\boldsymbol{x}_t|\boldsymbol{x}_0)e^{-\frac{f(\boldsymbol{x}_0)}{\lambda}} \, \mathrm{d}\boldsymbol{x}_0}{\int p^k(\boldsymbol{x}_0)p(\boldsymbol{x}_t|\boldsymbol{x}_0)e^{-\frac{f(\boldsymbol{x}_0)}{\lambda}} \, \mathrm{d}\boldsymbol{x}_0} \tag{22}$$

Note that $p(\boldsymbol{x}_t|\boldsymbol{x}_0) = \frac{1}{(2\pi\sigma_t^2)^{d/2}} \exp\left(-\frac{1}{2\sigma_t^2}\|\boldsymbol{x}_t - \alpha_t \boldsymbol{x}_0\|_2^2\right)$, we have $\nabla p(\boldsymbol{x}_t|\boldsymbol{x}_0) = p(\boldsymbol{x}_t|\boldsymbol{x}_0)\frac{\alpha_t \boldsymbol{x}_0 - \boldsymbol{x}_t}{\sigma_t^2}$. Then, we have

$$\nabla \log(p_t^{k+1}(\boldsymbol{x}_t)) = \frac{\int p^k(\boldsymbol{x}_0)p(\boldsymbol{x}_t|\boldsymbol{x}_0)\frac{\alpha_t \boldsymbol{x}_0 - \boldsymbol{x}_t}{\sigma_t^2} e^{-\frac{f(\boldsymbol{x}_0)}{\lambda}} \, \mathrm{d}\boldsymbol{x}_0}{\int p^k(\boldsymbol{x}_0)p(\boldsymbol{x}_t|\boldsymbol{x}_0)e^{-\frac{f(\boldsymbol{x}_0)}{\lambda}} \, \mathrm{d}\boldsymbol{x}_0} \tag{23}$$

$$= \frac{\int p^k(\boldsymbol{x}_0|\boldsymbol{x}_t)\frac{\alpha_t \boldsymbol{x}_0 - \boldsymbol{x}_t}{\sigma_t^2} e^{-\frac{f(\boldsymbol{x}_0)}{\lambda}} \, \mathrm{d}\boldsymbol{x}_0}{\int p^k(\boldsymbol{x}_0|\boldsymbol{x}_t)e^{-\frac{f(\boldsymbol{x}_0)}{\lambda}} \, \mathrm{d}\boldsymbol{x}_0} \tag{24}$$

Thus, we know that

$$\nabla \log(p_t^{k+1}(\boldsymbol{x}_t)) = \mathbb{E}_{p^k(\boldsymbol{x}_0|\boldsymbol{x}_t)}\left[\frac{\alpha_t \boldsymbol{x}_0 - \boldsymbol{x}_t}{\sigma_t^2} e^{-\frac{f(\boldsymbol{x}_0)}{\lambda}} / C_t^k\right] \tag{25}$$

where $C_t^k = \int p^k(\boldsymbol{x}_0|\boldsymbol{x}_t)e^{-\frac{f(\boldsymbol{x}_0)}{\lambda}} \mathrm{d}\boldsymbol{x}_0$

B Test Functions

Table 1: Test functions

name	function
Levy	$f(\boldsymbol{x}) := \sin^2(\pi w_1) + \sum_{i=1}^{d-1} (w_i - 1)^2(1 + 10\sin^2(\pi w_i + 1)) + (w_d - 1)^2(1 + \sin^2(2\pi w_d))$ where $w_i = 1 + (x_i - 1)/4, \ i \in \{1, ..., d\}$
Rastrigin10	$f(\boldsymbol{x}) := 10d + \sum_{i=1}^{d} (10^{\frac{i-1}{d-1}} x_i)^2 - 10\cos\left(2\pi 10^{\frac{i-1}{d-1}} x_i\right)$
Nesterov (nonsmooth)	$f(\boldsymbol{x}) := \frac{1}{4}\lvert x_1 - 1\rvert + \sum_{i=1}^{d-1} \lvert x_{i+1} - 2\lvert x_i\rvert + 1\rvert$
Rosenbrock	$f(\boldsymbol{x}) := \sum_{i=1}^{d-1}[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$