Proceedings of the 2025 Winter Simulation Conference
E. Azar, A. Djanatliev, A. Harper, C. Kogler, V. Ramamohan, A. Anagnostou, and S. J. E. Taylor, eds.

NESTED DENOISING DIFFUSION SAMPLING FOR GLOBAL OPTIMIZATION

Yuhao Wang[1], Haowei Wang[2], Enlu Zhou[1], Szu Hui Ng[2]

[1] School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, USA
[2] Dept. of Industrial Systems Engineering and Management, National University of Singapore,
SINGAPORE

ABSTRACT

We propose a novel algorithm, *Nested Denoising Diffusion Sampling (NDDS)*, for solving deterministic global optimization problems where the objective function is a black box—unknown, possibly non-differentiable, and expensive to evaluate. NDDS addresses this challenge by leveraging conditional diffusion models to efficiently approximate the evolving solution distribution without incurring the cost of extensive function evaluations. Unlike existing diffusion-based optimization methods that operate in offline settings and rely on manually specified conditioning variables, NDDS systematically generates these conditioning variables through a statistically principled mechanism. In addition, we introduce a data reweighting strategy to address the distribution mismatch between the training data and the target sampling distribution. Numerical experiments demonstrate that NDDS consistently outperforms the Extended Cross-Entropy (CE) method under the same function evaluation budget, particularly in high-dimensional settings.

1 INTRODUCTION

We consider the following deterministic global optimization problem:

$$x^* \in \arg\max_{x\in\mathscr{X}} H(x),$$

where $H(x)$ is an unknown deterministic objective function that lacks an explicit form and is expensive to evaluate. Such problems commonly arise in domains such as hyperparameter tuning in machine learning, engineering design, and simulation optimization. This class of problems is often referred to as black-box optimization (BBO). Traditional gradient-based methods are typically infeasible in these settings due to the absence of analytical derivatives, necessitating the use of sampling-based strategies with well-designed exploration and exploitation trade-off mechanism for efficient optimization.

A vast body of literature exists on global optimization methods, including but not limited to metamodel-based approaches such as Bayesian optimization (Mockus 2005; Frazier 2018) and radial basis function methods (Buhmann 2000); metaheuristic methods such as genetic algorithms (Reed, Minsker, and Goldberg 2000), simulated annealing (Chibante 2010; Zhou and Chen 2013), and COMPASS (Xu et al. 2010); and randomized algorithms such as estimation of distribution algorithms (EDAs) in evolutionary computing (Lozano 2006), the cross-entropy (CE) method (Kroese et al. 2006), model reference adaptive search (MRAS) (Hu et al. 2007), gradient-based stochastic adaptive search (GASS) (Zhou and Hu 2014), and particle filtering for optimization (PFO) (Zhou et al. 2013). Most of these methods rely on resampling or distribution estimation techniques. This motivates us to adopt diffusion models, which offer powerful data generation capabilities that can enhance optimization efficiency.

As an illustrative example, in this work we focus on a class of randomized algorithms—including EDAs, CE, MRAS, GASS, and PFO, which iteratively update a probability distribution over potential solutions, gradually guiding it towards the optimal solution. This updating process is typically expressed

979-8-3315-8726-0/25/$31.00 ©2025 IEEE

as:

$$p_{k+1}(x) = \frac{S_k(H(x))p_k(x)}{\int_{x' \in \mathscr{X}} S_k(H(x'))p_k(x')\mathrm{d}x'},$$

where $p_k(x)$ is the estimated distribution of the optimal solution x^* at iteration k, and $S_k(\cdot)$ is a positive, increasing function ensuring the positivity of p_{k+1}. By progressively increasing the weight of solutions with higher objective values, the sequence $\{p_k(x)\}$ converges to a Dirac measure centered at the optimal solution, assuming uniqueness of a global optimal solution. A key challenge in distribution-based optimization methods is that p_{k+1} cannot be computed directly. A common approach to this challenge is Monte Carlo estimation, which requires evaluating multiple samples drawn from the current distribution estimate. However, as k increases, the number of required evaluations also grows to mitigate Monte Carlo estimation error, making these methods computationally expensive due to the high cost of evaluating the black-box objective function. In this work, we propose a novel method without expensive function evaluations by leveraging on conditional diffusion models.

Diffusion models, well known as a GenAI (generative artificial intelligence) method, was first introduced by (Sohl-Dickstein et al. 2015). They are a class of generative models that iteratively refine noisy samples to generate data, through a so-called denoising diffusion process learned from the data. Conditional diffusion models (Dhariwal and Nichol 2021; Ho and Salimans 2022) extend this approach by incorporating auxiliary information to guide the generation process. Diffusion models are commonly-used in image processing, i.e. generate image samples. Not many works have explored utilizing diffusion models in generating data samples to improve sample efficiency. In our optimization setting, we utilize a conditional diffusion model to generate solutions corresponding to a specified objective value, where the conditional diffusion model implicitly learns the inverse mapping from the objective value to the solutions. Instead of sampling directly from the current estimated distribution, we first draw an intermediate label y from a learned distribution and then generate solutions via the conditional diffusion model using y as a conditioning variable. Since the denoising diffusion is nested within the randomly generated label, we call this sampling procedure as nested denoising diffusion sampling (NDDS). Notably, both the label distribution and the diffusion model are trained on previously evaluated solutions, this approach avoids costly extensive additional function evaluations when estimating the updated sampling distribution p_{k+1}.

Recent works (Krishnamoorthy et al. 2023; Li et al. 2024) have explored diffusion models for offline black-box optimization, where optimization is performed on a fixed dataset without further function evaluations. In contrast, we consider an online setting where the optimizer iteratively interacts with the objective function to refine the solution, enabling more adaptive and efficient exploration. Moreover, to the best of our knowledge, the proposed NDDS method is the first to systematically generate the conditioning variable y for the denoising process, as opposed to the aforementioned existing methods (Krishnamoorthy et al. 2023; Li et al. 2024), where y is manually chosen or heuristically determined.

2 PRELIMINARIES

2.1 Problem Formulation

In this section, we present the formal mathematical formulation of the problem. We aim to solve the following problem:

$$\max_{x \in \mathscr{X}} H(x), \tag{1}$$

where $H(x)$ is a deterministic function and $\mathscr{X} \subset \mathbb{R}^d$ is the set of solutions. We make the following assumptions for problem (1).

Assumption 1

1. $\mathscr{X}$ is a compact set.
2. There exists a unique $x^* \in \arg\max_{x \in \mathscr{X}} H(x)$.

Assumption 1.1 implies the feasible domain is a bounded set, which generally holds in practice. Assumption 1.2 requires the existence of a unique global optimal solution for the ease of demonstration of the algorithm. Note that, our algorithm is also applicable for multi-modal cases. The framework presented in this work can be applied when multiple optimal solutions exist. Here we do not assume any structure of $H(x)$ such as convexity, and $H(x)$ may contain multiple local optimal solutions.

2.2 Sequential Distribution Estimate for Optimal Solution

As discussed in Section 1, a class of algorithms such as EDAs, CE, MRAS and PFO iteratively update a probability distribution, which gradually concentrates on the optimal solution. Let $\{p_k(x)\}_{k=0}^{\infty}$ denote a sequence of distributions where $p_k(x)$ is the distribution estimate for optimal solution at iteration k. We rewrite the update rule for $p_{k+1}(x)$ as follows:

$$p_{k+1}(x) = \frac{S_k(H(x))p_k(x)}{\int_{x' \in \mathscr{X}} S_k(H(x'))p_k(x')\mathrm{d}x'}. \tag{2}$$

where $S_k(y)$ is a positive and strictly increasing function in $\mathbb{R}$. Notably, since $p_{k+1}(x) \propto S_k(H(x))p_k(x)$, for $x', x \in \mathscr{X}$ such that $H(x') > H(x)$, we have $\frac{p_{k+1}(x')}{p_{k+1}(x)} = \frac{S_k(H(x'))}{S_k(H(x))} \cdot \frac{p_k(x')}{p_k(x)} > \frac{p_k(x')}{p_k(x)}$. That is, promising solutions with a higher function value will be assigned a (relatively) larger density in the update process. Rigorous study of convergence of such distribution update (2) has been studied in (Hu et al. 2007; Zhou et al. 2013). It was proved that under mild conditions, the sequence $\{p_k(x)\}$ converge to the Dirac measure $\delta(x-x^*)$.

However, (2) cannot be directly calculated. Previous approaches such as CE, MRAS, PFO, etc, used Monte Carlo estimation plus some density estimation technique to ease the computation. One possibly simplest estimator is

$$\widehat{p}_{k+1}(x) := \frac{\sum_{i=1}^{n_k} S_k(H(x_k^i))\delta(x-x_k^i)}{n_k},$$

where $x_k^i, i = 1, \ldots, n_k$ is drawn from p_k and $\delta(\cdot)$ is the Dirac measure centered at origin. Notably, the function value $H(x_k^i)$ for each sampled solution x_k^i needs to be evaluated, which is very costly. Moreover, to ensure convergence of the algorithm, it is required in (Hu et al. 2007) that the sample size n_k goes to infinity as $k \to \infty$, further reducing sample efficiency and making such methods impractical in high-cost settings. To address this issue, we leverage advances in conditional diffusion models, which allow us to estimate the updated distribution without excessive function evaluations. We introduce the relevant background in the following section.

2.3 Denoising Diffusion Probabilistic Models (DDPM) and Conditional Diffusion Models

In this section, we introduce some basic knowledge about Denoising Diffusion Probabilistic Models (DDPM), which we will later utilize to develop an efficient way of estimating p_{k+1} in (2). DDPMs ((Sohl-Dickstein et al. 2015; Ho et al. 2020)) are a class of generative models that learn to synthesize complex data distributions through a two-step process: a **forward diffusion process** that gradually corrupts data into noise and a **backward process** that reconstructs the original data distribution by iteratively denoising samples.

Forward Diffusion Process: Given a data sample $x_0 \sim q(x_0)$, the forward process iteratively adds Gaussian noise over T time steps:

$$q(x_t|x_{t-1}) = \mathscr{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I), \tag{3}$$

where β_t is a variance schedule controlling the noise level at each step t. After sufficient steps, x_T approximates an isotropic Gaussian distribution.

Backward Denoising Process: To generate new samples, DDPM learns a parameterized denoising function $p_\psi(x_{t-1}|x_t)$ that reverses the diffusion process:

$$\tilde{q}_\psi(x_{t-1}|x_t) = \mathscr{N}(x_{t-1}; \mu_\psi(x_t,t), \Sigma_\psi(x_t,t)). \tag{4}$$

The model is trained to predict the noise component using a neural network, typically a U-Net architecture. **Conditional Diffusion Models:** Conditional diffusion models (Dhariwal and Nichol 2021; Ho and Salimans 2022) extend DDPM by incorporating auxiliary information y (e.g., class labels) to guide the generative process. The conditional backward process is formulated as:

$$\tilde{q}_{\psi}(x_{t-1}|x_t,y) = \mathscr{N}(x_{t-1};\mu_{\psi}(x_t,y,t),\Sigma_{\psi}(x_t,y,t)), \tag{5}$$

We can see from (4) to (5) the only difference from DDPM to conditional diffusion model is the inclusion of y, where y influences the learned denoising function. Such models enable controlled generation by guiding the sampling process toward desired outputs. In this work, we propose a nested denoising diffusion sampling method (NDDS), which leverages conditional diffusion models to generate promising candidate solutions for optimization by conditioning on the objective value drawn from a learned distribution.

3 NESTED DENOISING DIFFUSION SAMPLING

In this section, we present the NDDS method, which we will utilize to directly draw samples that follow the post-update distribution p_{k+1}. For notational simplicity, in the remainder of the section, we simply assume $S_k(H(x)) = H(x)$ and $H(x) > 0$. Notably, this does not affect the generalization of NDDS.

To start with, suppose we are given a labeled data set $D = \{(x_i,y_i) : y_i = H(x_i)\}_{i=1}^N$ in hand, which will be used to train a conditional DDPM. We make the following assumption.

Assumption 2 $\{(x_i,y_i)\}_{i=1}^N$ are independent and identically distributed ($i.i.d.$) and follow a joint distribution $q_{X,Y}(\cdot,\cdot)$.

Furthermore, let $q_X(\cdot)$ and $q_Y(\cdot)$ denote the marginal distribution for x_i and y_i, respectively. Also, let $q(x|y)$ denote the conditional distribution of x given y, and $q(y|x) = \delta(y-H(x))$ denote the conditional distribution of y given x.

Let $\mathscr{M}^{\psi}$ denote the conditional DDPM with parameter ψ trained from data set D, and $\tilde{q}_{\psi}(x|y)$ denote the distribution of the generated samples x by $\mathscr{M}^{\psi}$ conditioning on a specific y. Recall $\mathscr{M}^{\psi}$ is trained such that $q_{\psi}(x|y) \approx q(x|y)$. NDDS generate a sample by taking the following two steps.

- **Step 1.** Draw a sample y with density proportional to $y \cdot q_Y(y)$.
- **Step 2.** Generate $x \sim q_{\psi}(x|y)$ via the denoising process of $\mathscr{M}^{\psi}$, conditioning on y from Step 1.

Compared to the classic conditional denoising, NDDS uses a randomly generated conditioning variable y. The reason is summarized in the following theorem.

Theorem 1 Suppose conditional DDPM $\mathscr{M}$ is perfectly trained, i.e., $q_{\psi}(x|y) = q(x|y), \forall(x,y)$. Then, the sample x generated by NDDS follows a distribution whose density is proportional to $H(x)q_X(x)$.

Proof. Let $f(x)$ denote the density of x generated by NDDS. We have

$$\begin{aligned} f(x) &\propto \int_y y \cdot q_Y(y) q(x|y) \mathrm{d}y \\ &= \int_y y \cdot q_{X,Y}(x,y) \mathrm{d}y \\ &= \int_y y \cdot q_X(x) q(y|x) \mathrm{d}y \\ &= q_X(x) \int_y y \cdot \delta(y-H(x)) \mathrm{d}y \\ &= q_X(x)H(x). \end{aligned}$$

□

Recall the goal is to estimate $p_{k+1}(x) \propto H(x)p_k(x)$. If $q_X(x) \approx p_k(x)$, then NDDS generates samples that approximately follows $p_{k+1}(x)$. However, q_X is the marginal distribution of the data set D_k, which is the data set that contains the evaluated solutions so far and is used to train the conditional DDPM. In an iterative framework where one sequentially chooses candidate solution to evaluate, the data set D_k, i.e., the set that contains all previously evaluated solutions, is also iteratively updated and hence, the data distribution q_X of D_k varies over iterations and q_X is generally different from p_k. To address this distribution mismatch between q_X and p_k, we adopt a data reweighting approach. We will introduce this approach in Section 4, along with the presentation of the full algorithm.

Remark: For general function $S_k(\cdot)$, the derivation directly follows with $y = H(x)$ replaced by $y = S_k(H(x))$, and q_Y denote the marginal distribution of $S_k(H(x))$ with $x \sim q_X$.

4 NDDS FOR GLOBAL OPTIMIZATION

In this section, we present an iterative framework for solving global optimization, utilizing NDDS introduced in previous section. Generally speaking, our iterative framework contains three main steps in each iteration k, summarized as follows.

1. **Evaluation of new solutions:** Draw and evaluate some new solutions $\{(x_k^i, H(x_k^i))\}_{i=1}^{n_k}\}$, where $x_k^i \sim p_k$, and add them to dataset D_{k-1} to get D_k.
2. **Training of conditional DDPM:** Train a conditional DDPM $\mathscr{M}^{\psi_k}$ using data set D_k.
3. **Estimate of p_{k+1} using NDDS:** Generate sufficiently many solutions $\{\tilde{x}^i\}_{i=1}^{n_k'}$ by NDDS to get an estimate of p_{k+1}.

In Step 2 and 3, we update the distribution estimate using NDDS. Notably here no function evaluations are conducted, and hence, the sample size n_k' for NDDS can be selected to be a sufficiently large number. In Step 1, draw some new solutions to evaluate to enlarge the data set D_k, which is used to train a conditional DDPM. Due to the costly evaluation of the objective function H, n_k is much smaller than n_k' and is usually constrained by the practical budget.

4.1 Parametric Estimation for p_{k+1}

In Step 3, we generate n_k' samples that approximate p_{k+1}, where n_k' is chosen to be a sufficiently large number. To estimate the distribution p_{k+1} from these samples, we adopt a parametric model $\{f(x,\theta) : \theta \in \Theta\}$, where $f(\cdot,\theta)$ denotes a family of density functions parameterized by $\theta \in \Theta$. The rationale for using a parametric model is twofold. First, it allows us to efficiently estimate p_{k+1}, for example via maximum likelihood estimation (MLE), using the generated samples $\{\tilde{x}^i\}_{i=1}^{n_k'}$. The estimated distribution is then represented by $f(x,\theta_{k+1})$, where θ_{k+1} is the learned parameter. Second, this approach enables us to evaluate the likelihood $f(x,\theta_k)$ for any given x, which we use to reweight the data in training the conditional DDPM, as discussed in Section 3. Additionally, the parametric form $f(x,\theta_{k+1})$ facilitates exploration in Step 1 of the algorithm. Specifically, we draw new candidate solutions according to a mixture distribution: $x_k^i \sim (1-\lambda_k)f(x,\theta_k) + \lambda_k f(x,\theta_0)$, where $\lambda_k \in (0,1)$ denotes a small exploration probability, and $f(x,\theta_0)$ is the initial distribution used to ensure global exploration of the domain. In the remaining of the paper, let $f_k(x) := f(x,\theta_k)$ for notational simplicity.

4.2 Data Reweighting

Recall in Section 3, in order for NDDS to generate solutions that well approximate p_{k+1}, the underlying data distribution $q_{X,Y}$ of the data set D_k should possess the property that the marginal distribution q_X is a good approximation of p_k. However, note D_k contains all previously evaluated data $(x_{k'}^i, H(x_{k'}^i)), k' \leq k, i = 1,\ldots,n_{k'}$, where $x_{k'}^i \sim (1-\lambda_{k'})f_{k'}(x) + \lambda_{k'} f_0(x)$. Therefore, the marginal distribution q_X is not necessarily a good approximation of p_k.

To account for the fact that past samples $x^i_{k'}$ were drawn from a mixture distribution, rather than the current target distribution $f_k(x)$, we apply a likelihood-ratio-based reweighting to adjust the empirical data distribution. Specifically, let $\bar{x}^i_{k'} = (x^i_{k'}, H(x^i_{k'}))$ denote the i-th solution-objective pair evaluated at iteration $k' \le k$. According to our mixture sampling strategy, the sampling distribution at iteration k' is

$$(1-\lambda_{k'})f_{k'}(x) + \lambda_{k'} f_0(x).$$

To correct for the mismatch between this mixture and the current distribution $f_k(x)$, we assign each sample $\bar{x}^i_{k'}$ a weight:

$$w^i_{k,k'} := \frac{f_k(x^i_{k'})}{(1-\lambda_{k'})f_{k'}(x^i_{k'}) + \lambda_{k'} f_0(x^i_{k'})}.$$

We then construct a weighted empirical distribution over the augmented data:

$$\bar{p}_k(\bar{x}) := \frac{\sum_{k'=0}^{k}\sum_{i=1}^{n_{k'}} w^i_{k,k'}\,\delta(\bar{x}-\bar{x}^i_{k'})}{\sum_{k'=0}^{k}\sum_{i=1}^{n_{k'}} w^i_{k,k'}}. \tag{6}$$

This weighted distribution $\bar{p}_k$ serves as a refinement of the raw empirical distribution toward the current target joint distribution of $(x, H(x))$ induced by $f(x,\theta_k)$, via the likelihood ratios $\{w^i_{k,k'}\}$. Rather than training the conditional DDPM directly on the raw dataset D_k, we draw $N_k = \sum_{k'=0}^{k} n_{k'}$ samples from $\bar{p}_k$ to form a reweighted dataset $\bar{D}_k$, which is then used to train the conditional DDPM. As a result, the effective data distribution becomes $q_{X,Y} = \bar{p}_k$, and its marginal $q_X(x) = q_{X,Y}(x, H(x)) = \bar{p}_k(x, H(x))$ closely approximates the desired distribution $f(x,\theta_k) \approx p_k(x)$.

Remark: We want to emphasize that here the reweighting is done for the augmented solution $\bar{x} = (x, H(x))$ that already contain the function evaluation. What's more, the discrete empirical distribution (6) can also be replaced by some continuous approximation.

4.3 Algorithm

Equipped with the parametric estimation and data reweighting, we are ready to provide the full algorithm of Nested Denoising Diffusion Sampling for Global Optimization (NDDS-GO) in Algorithm 1.

Please note that while Algorithm 1 supports a general choice of the transformation function S_k, in line 13, we continue to train the conditional DDPM using pairs of the form $(x, H(x))$. This choice facilitates more stable and efficient model updates across iterations, particularly when S_k varies across different iterations k. To ensure the validity of NDDS under this setup, the denoising step (line 14) is modified to condition on $S_k^{-1}(y_j)$, where the inverse is well-defined due to the strict monotonicity of S_k. Moreover, the NDDS step (lines 12–15) can be executed in parallel to improve computational efficiency.

4.4 Consistency

In this section, we present the consistency of the proposed algorithm, which refers to the asymptotic global convergence. Recall we need to generate samples from the mixture distribution (line 4 in Algorithm 1). Specifically, we run the following steps to generate x^i_k.

1. Generate $U^i_k \sim \text{Uniform}(0,1)$ independently from all previous random variables.
2. If $U^i_k \le \lambda_k$, generate $x^i_{k,1} \sim f_0(x)$ independent of all previous random variables and set $x^i_k \leftarrow x^i_{k,1}$; otherwise generate $x^i_{k,2} \sim f_k(x)$ independnet of all previous random variables conditioned on θ_k, and set $x^i_k \leftarrow x^i_{k,2}$.

We make the following regularity condition on $f_0(x)$.

Assumption 3

Algorithm 1 Nested Denoising Diffusion Sampling for Global Optimization (NDDS-GO)

```
**Input:** Sample size for function evaluation $\{n_k\}_{k \geq 0}$ and for NDDS $\{n_k'\}_{k \geq 0}$. Parametric density family $\{f(\cdot, \theta) : \theta \in \Theta\}$ and initial parameter $\theta_0 \in \Theta$. Exploration probability $\{\lambda_k\}_{k \geq 1}$. Conditional DDPM family $\{\mathscr{M}^{\psi} : \psi \in \Psi\}$. Stopping criterion **STOP** until evaluation budget is exhausted.
**Initialization:** $k \leftarrow 0$, $D_{-1} \leftarrow \emptyset$.
**while** Budget left **do**
    Draw and evaluate $x_k^i \sim (1-\lambda_k) f_k(x) + \lambda_k f_0(x)$ to get $\bar{x}_k^i = (x_k^i, H(x_k^i))$,    for $1 \leq i \leq n_k$.
    Update dataset: $D_k \leftarrow D_{k-1} \cup \{\bar{x}_k^i\}_{i=1}^{n_k}$.
    Initialize weights: $w_{k,k}^i \leftarrow \frac{f_k(x_k^i)}{(1-\lambda_k) f_k(x_k^i) + \lambda_k f_0(x_k^i)}$, for $1 \leq i \leq n_k$.
    Update previous weights:
        $w_{k,k'}^i \leftarrow w_{k-1,k'}^i \cdot \frac{f_k(x_{k'}^i)}{f_{k'}(x_{k'}^i)}, \quad 0 \leq k' \leq k-1, \quad 1 \leq i \leq n_{k'}.$
                                                                ▷ Reweighting
    Compute $\bar{p}_k$ as defined in Equation (6).
    Draw $N_k = \sum_{k'=0}^{k} n_{k'}$ samples $\{(x_i', h_i')\}_{i=1}^{N_k}$ from $\bar{p}_k$ to obtain dataset $\bar{D}_k$.    ▷ Resampling
    Compute a marginal empirical distribution $q_Y$ with data $\{y_i' = S_k(h_i')\}_{i=1}^{N_k}$.
    Train a conditional DDPM $\mathscr{M}^{\psi_k}$ with dataset $\bar{D}_k$.
    **for** $j = 1 : n_k'$ **do**                                      ▷ NDDS
        Draw $y_j$ with density proportional to $y q_Y(y)$.
        Generate $\tilde{x}^j$ via the denoising process of $\mathscr{M}^{\psi_k}$, conditioning on $S_k^{-1}(y_j)$.
    **end for**
    Compute $\theta_{k+1}$ with data $\{\tilde{x}^j\}_{j=1}^{n_k'}$ using MLE:
        $\theta_{k+1} \leftarrow \arg\max_{\theta} \frac{1}{n_k'} \sum_{j=1}^{n_k'} \ln f(\tilde{x}^j, \theta).$
    $k \leftarrow k+1$.
**end while**
```

For all $\varepsilon > 0$, $\int_{A_\varepsilon} f_0(x) \mathrm{d}x > 0$, where $A_\varepsilon := \{x \in \mathscr{X} : H(x) \geq H(x^*) - \varepsilon\}$.

Proposition 1 Suppose Assumption 3 holds. Let $\{\lambda_k\}$ be a deterministic, non-increasing sequence. If $\{\lambda_k\}_{k \geq 0}$ and $\{n_k\}_{k \geq 0}$ satisfy $\sum_{k=0}^{\infty} \lambda_k = \infty$ and $\sum_{k=0}^{\infty} n_k = \infty$, respectively. Then, $\max_{0 \leq j \leq k, 1 \leq i \leq n_j} H(x_j^i) \to H(x^*)$ as $k \to \infty$ almost surely.

Proof. Let k_j, i_j denote the index pair such that $j = \sum_{k=0}^{k_j - 1} n_k + i_j$, that is, $x_{k_j}^{i_j}$ is the jth sample evaluated by Algorithm 1. Define

$$z_j = \begin{cases} x_{k_j}^{i_j} & \text{if } x_{k_j}^{i_j} = x_{k_j,1}^{i_j} \\ \bar{z} & \text{o.w.} \end{cases},$$

where $\bar{z}$ is any fixed point in $\mathscr{X}$ other than x^*. Then, $z_j, j \geq 1$ are independent since $U_k^i, x_{k,1}^i, \forall k \geq 0, 1 \leq i \leq n_k$ are independent. For any $\varepsilon > 0$, by Assumption 3, $\mathbb{P}(z_j \in A_\varepsilon) \geq \lambda_{k_j} \int_{A_\varepsilon} f(x, \theta_0) \mathrm{d}x > 0$. Hence,

$$\sum_{j=1}^{\infty} \mathbb{P}(z_j \in A_\varepsilon) \geq \int_{A_\varepsilon} f_0(x) \mathrm{d}x \sum_{j=1}^{\infty} \lambda_{k_j} \geq \int_{A_\varepsilon} f_0(x) \mathrm{d}x \sum_{k=0}^{\infty} \lambda_k = \infty.$$

The last inequality holds because$\{\lambda_k\}_{k\geq 0}$ is non-increasing and the last equality holds since $\sum_{k=0}^{\infty}\lambda_k = \infty$. Furthermore, since $\{z_j\}_{j\geq 1}$ is an independent sequence, we obtain $\mathbb{P}(z_j \in A_\varepsilon$ infinitely often$) = 1$ by the second Borel-Cantelli Lemma. Moreover, for $\varepsilon < H(x^*) - H(\bar{z})$, we know $\{z_j \in A_\varepsilon\} \subseteq \{x_{k_j}^{i_j} \in A_\varepsilon\}$. This implies $\mathbb{P}\left(x_{k_j}^{i_j} \in A_\varepsilon \text{ infinitely often}\right) = 1$. Since ε is arbitrary, we prove the desired result. □

5 EXPERIMENT

5.1 Diffusion Model

In this section, we would like to show that the conditional diffusion model could successfully learn the one-to-many mapping $x = H^{-1}(y)$, which maps a given function value y to the design points. We use StybTang (Styblinski-Tang) 2D as an illustration example and its mathematical form is:

$$H(x) = -\frac{1}{2}\sum_{i=1}^{d}(x_i^4 - 16x_i^2 + 5x_i), \tag{7}$$

where d=2. The StybTang 2D function has only one optimal solution $[-2.903534, -2.903534]$ with a maximum value of 78.33. Figure 1 shows its surface and contour plots. The function is multi-modal with three local maximums and a global maximum.

We trained a conditional diffusion model based on 5,000 design and value pairs $(x_i, H(x_i)), i = 1, \ldots, 5000$, obtained by Latin hypercube sampling (LHS). Figure 2 shows the contours of $H^{-1}(y)$ (red dots) and diffusion samples (blue dots) for different values of y. We observe that the diffusion samples closely match the target contours for different conditional values of y, indicating good learning of the one-to-many mapping $x = H^{-1}(y)$ using the diffusion model.

5.1.1 Optimization

We evaluated the performance of the NDDS-GO algorithm on StybTang 2D ($d = 2$) and StybTang 10D ($d = 10$) (see Equation (7) for the mathematical form). We compare NDDS-GO with the Extended Cross-Entropy (CE) algorithm (i.e., CE with S_k to be an exponential function). For both algorithms, we set $S_k(y) = e^y$. For StybTang 2D, both algorithms start with 500 initial designs generated via LHS, followed by 20 iterations that each adds $n_k = 50$ new evaluation points; for StybTang10D, both algorithms start with

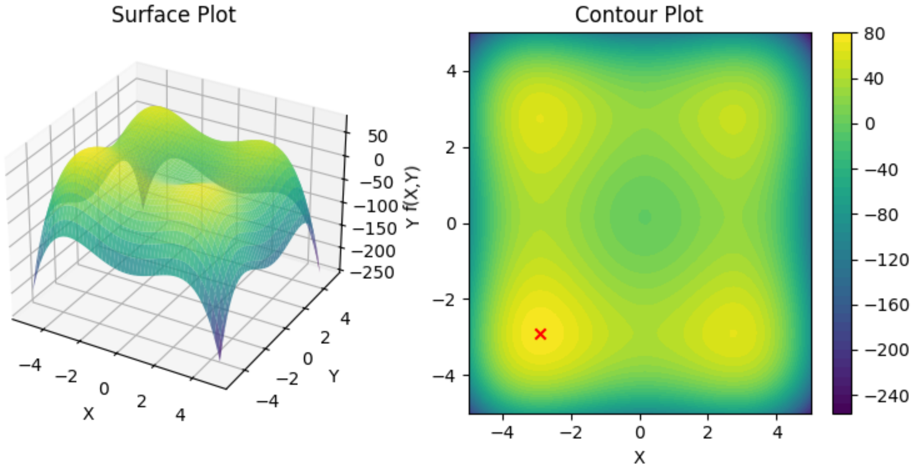
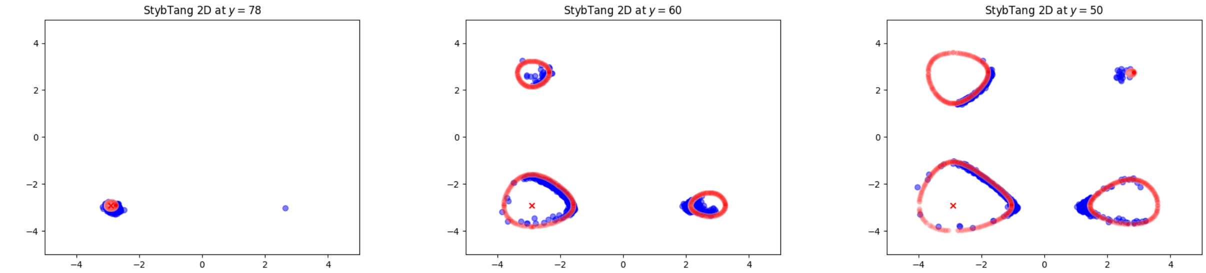


Figure 1: The surface and contour plot of StybTang 2D. The cross mark represents the best design with the maximum value.

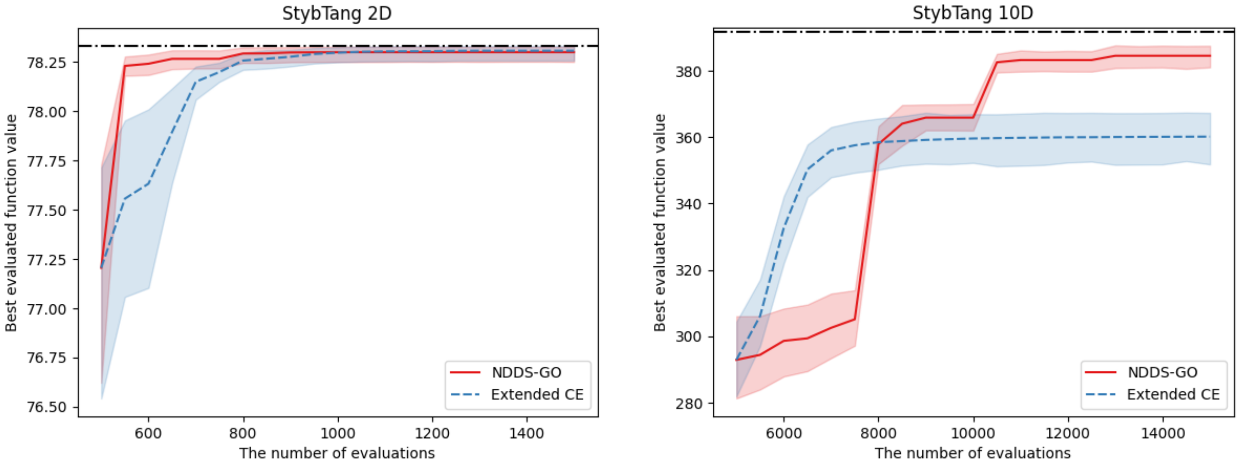


(a) Conditioning on $y = 78$ (b) Conditioning on $y = 60$ (c) Conditioning on $y = 50$

Figure 2: Conditional diffusion model for StybTang 2D under different values of y. The blue dots are the samples generated by the diffusion model, and the red dots show the ground truth inverse contours found using grid search. The cross mark represents the best design with the maximum value.

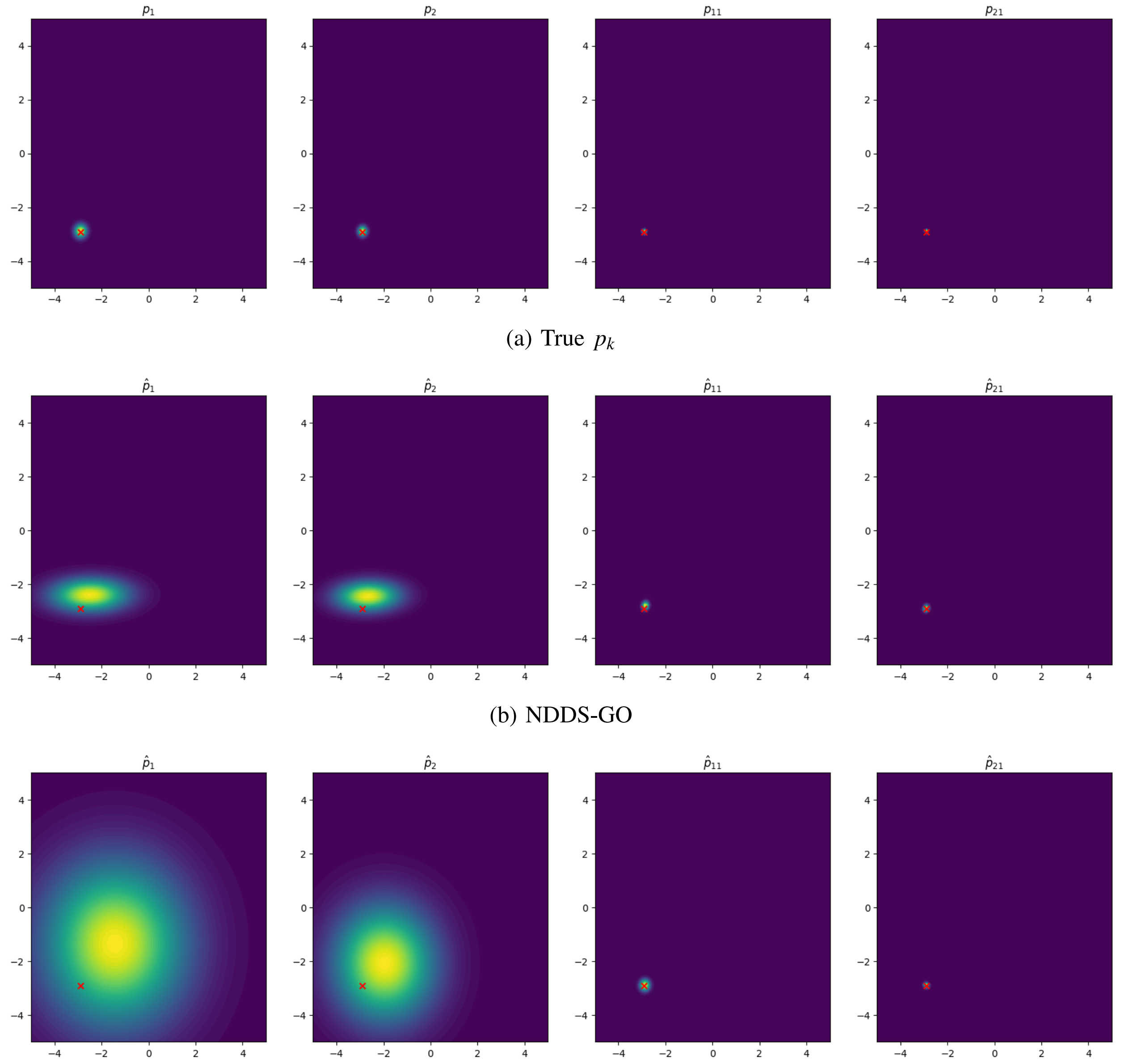
5,000 initial designs obtained using LHS and then run for 20 iterations with $n_k = 500$. The number of diffusion samples for each iteration is chosen to be 2,000.

Figure 3 shows the best evaluated function value vs the number of evaluations for the NDDS-GO and Extended CE algorithms over 10 macro-replications. For StybTang2D, both algorithms successfully converge to the true optimal solution. However, NDDS-GO achieves this more efficiently than Extended CE. Figure 4 provides insight into the distribution estimate of optimal solution for different algorithms. It shows that the estimated distribution based on the diffusion model is much closer to the true distribution (p_k in Equation (2)) in earlier iterations, and it demonstrates superior accuracy and concentration early in the optimization process, resulting in faster convergence compared to the Extended CE. For higher dimension problems like StybTang 10D, the Extended CE algorithm can get stuck in local maximum, whereas the NDDS-GO more easily escapes the local maximum, leading to a more accurate estimation of the maximum value.



(a) StybTang 2D with a maximum value of 78.33 (b) StybTang 10D with a maximum value of 391.65

Figure 3: The best evaluated function value against the number of evaluations for NDDS-GO and Extended CE for StybTang 2D and StybTang 10D over 10 macro-replications.



(a) True p_k

(b) NDDS-GO

(c) Extended CE

Figure 4: The density plot of the true distribution of the optimal solution (p_k from (2)) and the estimated distribution of the optimal solution for NDDS-GO and Extended CE for StybTang 2D. The cross mark represents the true optimal solution with the maximum value.

6 CONCLUSION

We proposed Nested Denoising Diffusion Sampling (NDDS), a novel approach for global optimization of expensive black-box functions. NDDS leverages conditional diffusion models to approximate the evolving solution distribution while systematically generating conditioning variables through a statistically principled mechanism. To improve sample efficiency, we introduced a data reweighting strategy that aligns the training data with the target distribution. Numerical experiments demonstrate that NDDS achieves better performance with the same number of function evaluations, particularly in high-dimensional settings.

This work highlights the promise of combining generative modeling with adaptive sampling for efficient black-box optimization.

ACKNOWLEDGMENT

The authors are grateful for the support by Air Force Office of Scientific Research (AFOSR) under Grant FA9550-22-1-0244, National Science Foundation (NSF) under Grant ECCS-2419562, Artificial Intelligence Institute for Advances in Optimization (AI4OPT) under Grant NSF-2112533, and the Singapore Ministry of Education (MOE) Academic Research Fund (AcRF) under Grant R-266-149–114.

REFERENCES

Buhmann, M. D. 2000. “Radial Basis Functions”. *Acta Numerica* 9:1–38.

Chibante, R. 2010. *Simulated Annealing: Theory with Applications*. BoD–Books on Demand.

Dhariwal, P., and A. Nichol. 2021. “Diffusion Models Beat GANs on Image Synthesis”. *Advances in Neural Information Processing Systems* 34:8780–8794.

Frazier, P. I. 2018. “A Tutorial on Bayesian Optimization”. *arXiv Preprint arXiv:1807.02811*.

Ho, J., A. N. Jain, and P. Abbeel. 2020. “Denoising Diffusion Probabilistic Models”. In *Advances in Neural Information Processing Systems*, 6840–6851.

Ho, J., and T. Salimans. 2022. “Classifier-Free Diffusion Guidance”. *arXiv Preprint arXiv:2207.12598*.

Hu, J., M. C. Fu, and S. I. Marcus. 2007. “A Model Reference Adaptive Search Method for Global Optimization”. *Operations Research* 55(3):549–568.

Krishnamoorthy, S., S. M. Mashkaria, and A. Grover. 2023. “Diffusion Models for Black-Box Optimization”. In *International Conference on Machine Learning*, 17842–17857. PMLR.

Kroese, D. P., S. Porotsky, and R. Y. Rubinstein. 2006. “The Cross-Entropy Method for Continuous Multi-Extremal Optimization”. *Methodology and Computing in Applied Probability* 8:383–407.

Li, Z., H. Yuan, K. Huang, C. Ni, Y. Ye, M. Chen *et al*. 2024. “Diffusion Model for Data-Driven Black-Box Optimization”. *arXiv Preprint arXiv:2403.13219*.

Lozano, J. A. 2006. *Towards a New Evolutionary Computation: Advances on Estimation of Distribution Algorithms*, Volume 192. Berlin, Germany: Springer Science & Business Media.

Mockus, J. 2005. “The Bayesian Approach to Global Optimization”. In *System Modeling and Optimization: Proceedings of the 10th IFIP Conference, New York City, USA, August 31–September 4, 1981*, 473–481. Berlin, Germany: Springer.

Reed, P., B. Minsker, and D. E. Goldberg. 2000. “Designing a Competent Simple Genetic Algorithm for Search and Optimization”. *Water Resources Research* 36(12):3757–3761.

Sohl-Dickstein, J., E. Weiss, N. Maheswaranathan, and S. Ganguli. 2015. “Deep Unsupervised Learning Using Nonequilibrium Thermodynamics”. In *International Conference on Machine Learning*, 2256–2265. PMLR.

Xu, J., B. L. Nelson, and J. L. Hong. 2010. “Industrial Strength COMPASS: A Comprehensive Algorithm and Software for Optimization via Simulation”. *ACM Transactions on Modeling and Computer Simulation (TOMACS)* 20(1):1–29.

Zhou, E., and X. Chen. 2013. “Sequential Monte Carlo Simulated Annealing”. *Journal of Global Optimization* 55:101–124.

Zhou, E., M. C. Fu, and S. I. Marcus. 2013. “Particle Filtering Framework for a Class of Randomized Optimization Algorithms”. *IEEE Transactions on Automatic Control* 59(4):1025–1030.

Zhou, E., and J. Hu. 2014. “Gradient-Based Adaptive Stochastic Search for Non-Differentiable Optimization”. *IEEE Transactions on Automatic Control* 59(7):1818–1832.

AUTHOR BIOGRAPHIES

YUHAO WANG is a Ph.D. candidate at the H. Milton Stewart School of Industrial and Systems Engineering at Georgia Institute of Technology. He received his B.S. degree from the Department of Mathematics at Nanjing University, China, in 2021. His research interests include simulation and stochastic optimization and reinforcement learning. His email address is yuhaowang@gatech.edu.

HAOWEI WANG is currently a senior research fellow at National University of Singapore (NUS). He received the B.Eng. degree in industrial engineering from Nanjing University in 2016 and the Ph.D. degree in industrial and systems engineering from NUS in 2021. His research interest includes Bayesian optimization, and simulation optimization under uncertainties. His email address is whw@nus.edu.sg.

ENLU ZHOU is a Fouts Family Professor in the H. Milton Stewart School of Industrial and Systems Engineering at the Georgia Institute of Technology. Her research interests include simulation optimization, stochastic optimization, and stochastic

control. She received the B.S. degree with highest honors in electrical engineering from Zhejiang University, China, in 2004, and received the Ph.D. degree in electrical engineering from the University of Maryland, College Park, in 2009. She is a recipient of the Best Theoretical Paper award at the Winter Simulation Conference, AFOSR Young Investigator award, NSF CAREER award, and INFORMS Outstanding Simulation Publication Award. She has served on the editorial boards of *Operations Research*, *IEEE Transactions on Automatic Control*, and *SIAM Journal on Optimization*, and is currently co-Editor-in-Chief for *Journal of Simulation*. She serves as the President of the INFORMS Simulation Society from 2024–2026. Her email address is enlu.zhou@isye.gatech.edu, and her website is https://www.enluzhou.gatech.edu/.

SZU HUI NG is an Associate Professor in the Department of Industrial Systems Engineering and Management at the National University of Singapore. She holds B.S., M.S. and Ph.D. degrees in Industrial and Operations Engineering from the University of Michigan. Her research interests include computer simulation analysis and optimization, applications of simulation to maritime transportation and quality engineering. She is a member of IEEE and INFORMS, and a senior member of IISE. Her e-mail address is isensh@nus.edu.sg, and her website is https://bonusisem.github.io.