arXiv:2505.12383v1 [math.OC] 18 May 2025

High-dimensional Optimization with Low Rank Tensor Sampling and Local Search

Konstantin Sozykin*
Skolkovo Institute
of Science and Technology
konstantin.sozykin@skoltech.ru

Andrei Chertkov*
Artificial Intelligence Research Institute (AIRI),
Skolkovo Institute of Science and Technology
a.chertkov@skoltech.ru

Anh-Huy Phan
Skolkovo Institute
of Science and Technology
a.phan@skoltech.ru

Ivan Oseledets
Artificial Intelligence Research Institute (AIRI),
Skolkovo Institute of Science and Technology
i.oseledets@skoltech.ru

Gleb Ryzhakov†
Central University, Moscow, Russia
g.ryzhakov@centraluniversity.ru

Abstract

We present a novel method called TESALOCS (TEnsor SAmpling and LOCal Search) for multidimensional optimization, combining the strengths of gradient-free discrete methods and gradient-based approaches. The discrete optimization in our method is based on low-rank tensor techniques, which, thanks to their low-parameter representation, enable efficient optimization of high-dimensional problems. For the second part, i.e., local search, any effective gradient-based method can be used, whether existing (such as quasi-Newton methods) or any other developed in the future. Our approach addresses the limitations of gradient-based methods, such as getting stuck in local optima; the limitations of discrete methods, which cannot be directly applied to continuous functions; and limitations of gradient-free methods that require large computational budgets. Note that we are not limited to a single type of low-rank tensor decomposition for discrete optimization, but for illustrative purposes, we consider a specific efficient low-rank tensor train decomposition. For 20 challenging 100-dimensional functions, we demonstrate that our method can significantly outperform results obtained with gradient-based methods like Conjugate Gradient, BFGS, SLSQP, and other methods, improving them by orders of magnitude with the same computing budget.

1 Introduction

A persistent challenge in numerical optimization arises from the susceptibility of gradient-based methods to converge to local optima, particularly in high-dimensional search spaces where solution quality exhibits strong dependence on initialization. While these methods demonstrate remarkable efficiency in local refinement, their global performance remains fundamentally constrained by the quality of starting points. To overcome this limitation, we propose a hybrid paradigm that employs gradient-free strategies for identifying promising initial regions in the parameter space followed by gradient-based local optimization.

*Equal contribution
†Corresponding author

Contemporary approaches in this vein often utilize sampling mechanisms based on multivariate normal distributions Hansen and Auger [2011], Krause et al. [2016] or their mixture variants He et al. [2021], which theoretically enable systematic exploration of complex landscapes. However, despite their mathematical elegance, such methods frequently underperform in practical high-dimensional scenarios due to the insufficient expressiveness of conventional sampling distributions to capture complex parameter correlations.

In the case of discrete parameter spaces, low rank tensor methods provide a powerful tool for building expressive yet low-parametric representations of high-dimensional models. State-of-the-art algorithms like TTopt Sozykin et al. [2022] and PROTES Batsheva et al. [2023] exemplify this paradigm, leveraging the well-known Tensor Train (TT) format Oseledets [2011], Cichocki et al. [2016, 2017] to solve complex discrete black-box optimization problems ranging from quantum computing Paradezhenko et al. [2024] to reinforcement learning Shetty et al. [2022]. The efficacy of these methods stems from the intrinsic properties of tensor networks like TT, Canonical Polyadic (CP) Harshman et al. [1970], and Hierarchical Tucker (HT) Hackbusch and Kühn [2009], which enable compact representations of high-dimensional data structures while maintaining computational tractability. This combination of expressiveness and efficiency makes tensor networks particularly suitable for modern optimization problems where dimensionality scaling and resource limitations are fundamental constraints.

In this work, we establish a bridge between discrete low-rank tensor-based optimization and gradient-driven approach. The motivation for our method is that conventional gradient methods are highly dependent on the starting point (approximation). We solve this problem by iteratively sampling starting points from a discrete low-rank representation (surrogate model) of the optimized function that probabilistically encodes promising regions of the optimization landscape. Moreover, the use of a low-rank approach allows us to compactly store the model, efficiently sample candidates for starting points, and dynamically update the model, since memory consumption and computational complexity grows linearly with dimension. While we focus on TT-decomposition due to its numerical stability, it is important to note that the choice of a specific type of low-rank tensor decomposition is not fundamental, and other approaches, such as CP or HT, can also be successfully applied.

We summarize our main contributions as follows:

- We develop a new method TESALOCS for optimization (finding the minimum or maximum[1] value) of multivariable functions based on a sampling approach from a dynamically updated probability distribution in the TT-format, combined with external optimization procedures such as the well-known second-order BFGS Nocedal and Wright [2006] or more heuristic zero-order methods like particle swarm optimization (PSA) Kennedy and Eberhart [1995].
- We demonstrate superiority of TESALOCS over 6 modern gradient-based methods and 3 popular gradient-free approaches through extensive benchmarks on 20 non-convex high-dimensional functions. Using fixed hyperparameters across all experiments, our method achieves consistent performance improvements under equivalent computational budgets.

2 Method

The TESALOCS algorithm is a novel method that decomposes high-dimensional optimization problem into two synergistic phases: global exploration through adaptive tensor sampling and local refinement via gradient-based or heuristic optimization. The whole procedure is schematically illustrated in Figure 1 and is formally described in Algorithm 2.1. At its core, our method leverages low-rank tensor decompositions to construct a probabilistic discrete surrogate model that identifies promising regions in the parameter space while maintaining computational tractability. The algorithm operates by iteratively sampling candidate solutions from the TT-decomposition, refining them using a local search procedure, and updating the low-rank model in the TT-format based on the best-performing candidates. This hybrid approach ensures both global exploration and local exploitation, making it highly effective for complex optimization problems.

Let us consider in more detail the mechanism of the TESALOCS method for optimization of d-dimensional function $f : \mathbb{R}^d \mapsto \mathbb{R}$ in accordance with the main points indicated in Figure 1.

[1]Further, for concreteness, we will consider the minimization problem in this paper, while the proposed method can be applied to the discrete maximization problem without any modifications.

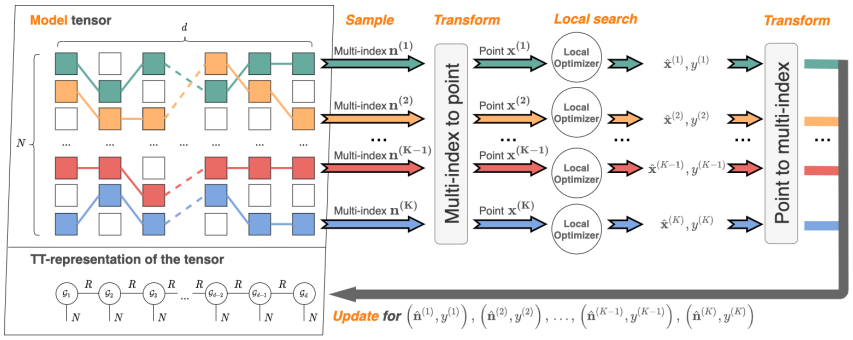


Figure 1: Schematic representation of the proposed optimization method TESALOCS.

Model. We use a discrete surrogate model $\mathcal{T}$ in the TT-format to store information about the function $f(\mathbf{x})$ being optimized. A d-dimensional tensor $\mathcal{T} \in \mathbb{R}^{N_1 \times N_2 \times \ldots \times N_d}$ is expressed in the TT-format Oseledets [2011] if its elements can be represented as:

$$\mathcal{T}[n_1, n_2, \ldots, n_d] = \sum_{r_1=1}^{R_1} \sum_{r_2=1}^{R_2} \cdots \sum_{r_{d-1}=1}^{R_{d-1}} \mathcal{G}_1[1, n_1, r_1]\mathcal{G}_2[r_1, n_2, r_2] \ldots \mathcal{G}_d[r_{d-1}, n_d, 1], \quad (1)$$

where $(n_1, n_2, \ldots, n_d)$ is a multi-index (with $n_i = 1, 2, \ldots, N_i$ for $i = 1, 2, \ldots, d$), and the integers $R_1, R_2, \ldots, R_{d-1}$ (with the convention $R_0 = R_d = 1$) controlling the expressiveness of the model and are called TT-ranks. The three-dimensional tensors $\mathcal{G}_i \in \mathbb{R}^{R_{i-1} \times N_i \times R_i}$ (for $i = 1, 2, \ldots, d$) are known as TT-cores. The TT-decomposition allows to represent a tensor in a compact and descriptive low-parameter form, which is linear in dimension d, i. e., it has less than $d \cdot \max_{i=1,\ldots,d}(N_i R_i^2)$ parameters, hence the TT-format is recognized for its ability to reduce memory consumption and enable efficient linear algebra operations Cichocki et al. [2016, 2017]. Note that we start with a random non-negative tensor $\mathcal{T}$ with fixed rank $R_1 = R_2 = \ldots = R_{d-1} = R$, which is generated by assigning random non-negative elements to all its TT cores. For simplicity, we use the equal number of discretization nodes for each dimension, i.e., $N_1 = N_2 = \ldots = N_d = N$.

Sample. Our algorithm proceeds iteratively within a given budget M, and in each iteration, a batch of k candidate solutions is generated by sampling from the current tensor $\mathcal{T}$, which models a discrete probability distribution in the TT-format (that is, it corresponds to the probability for a given discrete point to be a global optimum). The sampling is performed using an efficient procedure that has linear complexity in dimension d (see Dolgov et al. [2020] for details).

Transform. The generated discrete candidates $\mathbf{n}_{batch} = (\mathbf{n^{(1)}}, \mathbf{n^{(2)}}, \ldots, \mathbf{n^{(k)}})$ are projected onto a continuous grid using a projection operator Pr ("multi-index to point" in Fig. 1) and search space $\mathcal{X}$ defined as a tuple of $\mathbf{a}, \mathbf{b}, \mathbf{N}$ that represents lower and upper grid bound and a number of discretization nodes ("sample rate") for search space. In the case of uniform grid the projection operations Pr and Pr^{-1} ("point to multi-index" in Fig. 1) can be represented as follows:

$$Pr(\mathbf{n^{(l)}}, \mathbf{a}, \mathbf{b}, \mathbf{N}) = \frac{\mathbf{n^{(l)}}}{\mathbf{N} - 1} \cdot (\mathbf{b} - \mathbf{a}) + \mathbf{a}, \quad Pr^{-1}(\mathbf{x^{(l)}}, \mathbf{a}, \mathbf{b}, \mathbf{N}) = \frac{\mathbf{x^{(l)}} - \mathbf{a}}{\mathbf{b} - \mathbf{a}} \cdot (\mathbf{N} - 1).$$

However, for different purposes, we may use a non-uniform grid or even a mix of different grids on different components of the input vector. Note that we always have one-to-one mapping between multi-indices and corresponding float points.

Local search. A local search algorithm is applied to refine the projected candidates $\mathbf{x}_{batch} = (\mathbf{x^{(1)}}, \mathbf{x^{(2)}}, \ldots, \mathbf{x^{(k)}})$ to exploit high-potential regions identified by the global tensor sampler. We can employ any gradient-based method (e.g., BFGS), or even derivative-free heuristics (e.g., PSO) in the case of non-differentiable or noisy objectives. Each candidate $\mathbf{x^{(l)}}$ $(l = 1, 2, \ldots, k)$ serves as an initial point for an independent local optimization run. Based on the results of the local optimizer's work, we obtain a batch of refined continuous points $\widehat{\mathbf{x}}_{batch}$, the corresponding values of the objective function at these points y_{batch}, and the actual budget spent on this launch m_{loc}. Note that these calculations can be performed in parallel if necessary to maximize computational efficiency.

Algorithm 2.1: TEnsor SAmpling and LOCal Search

Input: Function $f(\mathbf{x})$, search space $\mathcal{X}$, budget M, discrete grid size N, TT-rank r, batch size k, elite size k_{top}, projectors Pr and Pr^{-1}, local search algorithm Loc, SGD algorithm.

Output: Optimal solution $\mathbf{x}^*$.

Initialize cores $\{\mathcal{G}_i\}_{i=1}^d$ of the TT-decomposition $\mathcal{T}$ with all $N_1, N_2, \ldots, N_d$ equal to N and all $R_1, R_2, \ldots, R_{d-1}$ equal to r ;

Set $m = 0$ and $\mathcal{P} = \emptyset$;

while $m < M$ **do**

  Generate discrete candidates $\mathbf{n}_{batch} = \{\mathbf{n}^{(l)}\}_{l=1}^k$ via sampling from $\mathcal{T}$ in the TT-format;

  Project discrete candidates on the continuous grid: $\mathbf{x}_{batch} = Pr(\mathbf{n}_{batch}, \mathcal{X})$;

  Make local search with the given algorithm, using $\mathbf{x}_{batch}$ as initials: $\{\widehat{\mathbf{x}}_{batch}, y_{batch}, m_{\text{loc}}\} = \texttt{Loc}(f, \mathbf{x}_{batch})$;

  Select the best k_{top} candidates: $\mathbf{x}_{\text{top}} = \text{Top}(\widehat{\mathbf{x}}_{batch}, y_{batch}, k_{top})$;

  Merge populations: $\mathcal{P} \leftarrow \mathcal{P} \cup \mathbf{x}_{\text{top}}$;

  Unproject back: $\mathbf{n}_{\text{top}} = Pr^{-1}(\mathbf{x}_{\text{top}}, \mathcal{X})$;

  Make SGD step to update the cores using Log Likelihood loss $\mathcal{L}$ on the obtained indices: $\{\mathcal{G}_i\}_{i=1}^d \leftarrow SGD(\{\mathcal{G}_i\}_{i=1}^d, \mathbf{n}_{\text{top}})$;

  Update current budget: $m \leftarrow m + m_{\text{loc}}$;

end

$\mathbf{x}^* = \arg\max_{\mathbf{x} \in \mathcal{P}} f(\mathbf{x})$;

Update. Following local refinement, the top-k solutions are selected from the batch $\mathbf{x}_{batch}$ based on their objective values, forming the elite subset $\mathbf{x}_{top}$. Then we project them back into discrete space and use selected discrete k_{top} candidates $\mathbf{n}_{top}$ in SGD with log-likelihood objective, which can be easily and efficiency computed using equation (1):

$$\mathcal{L} = -\sum_{\mathbf{n} \in \mathbf{n}_{top}} \log p_{\mathcal{T}}(\mathbf{n}),$$

where $p_{\mathcal{T}}(\mathbf{n})$ is the probability density encoded in the TT-cores (i.e. normalized value of the TT-tensor). By doing gradient ascend with such loss we will increase likelihood of points $\mathbf{n}_{top}$ refined with local optimization. Note that such update mechanism focuses the tensor sampler on promising subspaces while preserving rank constraints to maintain linear memory scaling.

3 Numerical experiments

To rigorously validate the efficacy of TESALOCS, we conducted systematic comparisons across 20 analytically defined benchmark functions with non-convex, multimodal landscapes Jamil and Yang [2013]. We compare the absolute error of the optimization result relative to the known exact global minimum for various existing optimization methods as part of our TESALOCS approach and without it (i.e., with random selection of the starting point). For each considered optimization method we report the average value E of the absolute error and its variance σ for 10 independent runs of each experiment.

All experiments were performed in a 100-dimensional parameter space ($d = 100$) under strict computational constraints, limiting the number of objective function evaluations to $M = 10^4$ and with fixed rank $r = 5$, sampling rate $N = 2^{20}$, batch size $k = 100$, elite size $k_{top} = 10$. The search space parameters $\mathbf{a}$ and $\mathbf{b}$ were selected according to benchmark function definitions.

In our initial experimental series, we evaluated widely-used gradient-based optimization methods Dembo and Steihaug [1983], Nocedal and Wright [2006] from the SciPy library[2]: BFGS (Broyden–Fletcher–Goldfarb–Shanno algorithm); CG (Conjugate Gradient algorithm); L-BFGS-B (limited-memory BFGS variant with box constraints); Newton-CG; SLSQP (Sequential Least Squares Programming); and TNC (Truncated Newton Algorithm). The computation results are reported in Table 1, and Table 2. For each benchmark-method pair, bold formatting denotes the lowest absolute error. Co-convergence is declared when both (random and our) approaches achieve errors $< 10^{-8}$,

[2]See SciPy optimization module: `https://docs.scipy.org/doc/scipy/reference/optimize.html`

Table 1: Optimization results for gradient-based methods BFGS, CG, and L-BFGS-B.

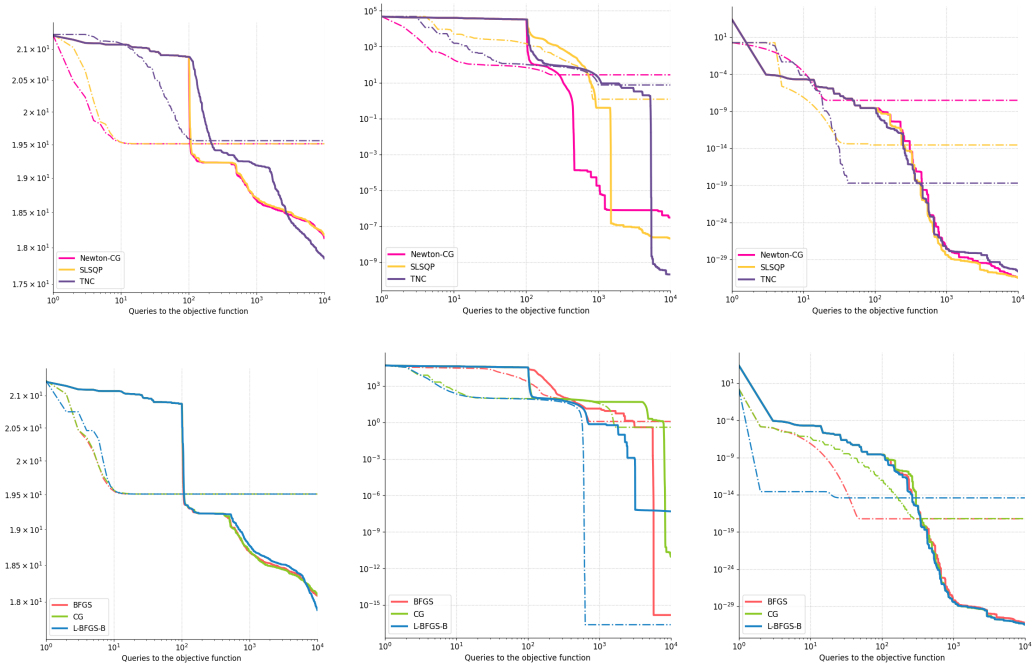
Function		BFGS		CG		L-BFGS-B	
		Random	TESALOCS	Random	TESALOCS	Random	TESALOCS
# of best results		9	**19**	7	**19**	10	**18**
Ackley	E	2.0e+01	**1.8e+01**	2.0e+01	**1.8e+01**	2.0e+01	**1.8e+01**
	σ	8.2e-02	5.1e-01	8.2e-02	4.2e-01	8.2e-02	5.3e-01
Alpine	E	3.8e-04	**1.8e-09**	2.2e+01	**7.5e+00**	6.2e+00	**8.0e-01**
	σ	7.8e-04	1.1e-09	9.2e+00	3.2e+00	2.9e+00	7.3e-01
Chung	E	**1.8e-22**	**1.2e-09**	6.9e+03	**1.7e-11**	**2.3e-17**	**1.6e-10**
	σ	2.9e+00	7.3e-01	8.2e+03	3.4e-11	8.2e+03	3.4e-11
Dixon	E	**6.7e-01**	**6.7e-01**	**6.7e-01**	**6.7e-01**	**6.7e-01**	**6.7e-01**
	σ	3.6e-15	0.0e+00	1.1e-15	0.0e+00	2.7e-16	0.0e+00
Exp	E	**0.0e+00**	**0.0e+00**	**0.0e+00**	**0.0e+00**	**0.0e+00**	**0.0e+00**
	σ	2.7e-16	0.0e+00	2.7e-16	0.0e+00	2.7e-16	0.0e+00
Griewank	E	**0.0e+00**	**0.0e+00**	7.4e-04	**2.7e-10**	**0.0e+00**	**0.0e+00**
	σ	2.7e-16	0.0e+00	2.2e-03	2.4e-10	2.2e-03	2.4e-10
Pathological	E	3.8e+01	**3.0e+01**	4.0e+01	**3.2e+01**	3.9e+01	**3.2e+01**
	σ	1.4e+00	1.5e+00	1.3e+00	2.0e+00	1.6e+00	1.5e+00
Pinter	E	1.5e+05	**6.0e+04**	1.7e+05	**6.6e+04**	1.7e+05	**6.4e+04**
	σ	2.1e+04	3.7e+03	2.0e+04	9.2e+03	2.0e+04	6.8e+03
Powell	E	**1.3e-18**	6.2e-08	**1.2e-17**	2.3e-08	**2.6e-11**	2.8e-08
	σ	1.2e-18	3.8e-08	1.0e-17	2.0e-08	2.0e-11	1.5e-08
Qing	E	**8.9e-23**	**3.8e-14**	**3.8e-26**	**2.7e-13**	**1.1e-16**	**1.6e-09**
	σ	2.0e-11	1.5e-08	2.0e-11	1.5e-08	2.0e-11	1.5e-08
Rastrigin	E	8.4e+02	**3.3e+02**	8.5e+02	**3.2e+02**	8.5e+02	**2.9e+02**
	σ	7.9e+01	4.5e+01	7.4e+01	5.0e+01	7.4e+01	4.6e+01
Rosenbrock	E	1.2e+00	**1.5e-16**	4.0e-01	**1.0e-11**	**4.8e-17**	2.3e-08
	σ	1.8e+00	4.5e-16	1.2e+00	1.8e-11	3.9e-17	4.7e-08
Salomon	E	5.7e+01	**1.5e+01**	5.7e+01	**1.0e+01**	4.6e+01	**9.0e-15**
	σ	2.5e+00	1.0e+01	2.5e+00	9.2e+00	2.2e+01	4.6e-15
Schaffer	E	4.5e+01	**4.0e+01**	4.5e+01	**4.0e+01**	4.5e+01	**4.0e+01**
	σ	7.3e-01	1.4e+00	7.2e-01	1.4e+00	7.2e-01	1.4e+00
Sphere	E	**7.6e-34**	**3.1e-34**	**6.4e-39**	**8.5e-19**	**8.9e-59**	**5.8e-37**
	σ	7.2e-01	1.4e+00	7.2e-01	1.4e+00	7.2e-01	1.4e+00
Squares	E	**1.8e-35**	**1.3e-20**	**1.4e-33**	**5.3e-12**	**2.7e-16**	**4.2e-10**
	σ	7.2e-01	1.4e+00	7.2e-01	1.4e+00	7.2e-01	1.4e+00
Trid	E	4.2e-08	**0.0e+00**	1.3e-05	**0.0e+00**	1.5e-06	**0.0e+00**
	σ	3.8e-08	0.0e+00	6.0e-06	0.0e+00	1.0e-06	0.0e+00
Trigonometric	E	3.9e+03	**3.0e+01**	1.3e+01	**7.4e-01**	2.2e+00	**2.4e-01**
	σ	8.1e+02	9.1e+01	1.3e+01	9.8e-01	5.9e+00	7.1e-01
Wavy	E	5.9e-01	**3.6e-01**	5.9e-01	**3.7e-01**	5.9e-01	**3.6e-01**
	σ	3.4e-02	1.8e-02	3.4e-02	2.4e-02	3.4e-02	2.8e-02
Yang	E	**5.9e-18**	**4.9e-32**	**6.2e-18**	**4.2e-32**	**3.8e-15**	**4.2e-32**
	σ	3.4e-02	2.8e-02	3.4e-02	2.8e-02	3.4e-02	2.8e-02

Table 2: Optimization results for gradient-based methods Newton-CG, SLSQP, and TNC.

Function		Newton-CG		SLSQP		TNC	
		Random	TESALOCS	Random	TESALOCS	Random	TESALOCS
# of best results		6	**20**	9	**15**	8	**20**
Ackley	E	2.0e+01	**1.8e+01**	2.0e+01	**1.8e+01**	2.0e+01	**1.8e+01**
	σ	8.3e-02	3.8e-01	8.2e-02	3.4e-01	9.2e-02	7.9e-01
Alpine	E	9.1e+01	**2.5e+01**	**6.9e-13**	3.1e-05	4.7e+01	**4.9e-04**
	σ	2.1e+01	5.8e+00	1.7e-13	5.1e-06	1.8e+01	7.8e-04
Chung	E	**6.6e-10**	**4.6e-11**	**6.9e-13**	3.7e-08	**4.1e-17**	**2.1e-11**
	σ	1.8e+01	7.8e-04	1.3e-12	5.3e-08	1.3e-12	5.3e-08
Dixon	E	7.6e+00	**6.7e-01**	**6.7e-01**	**6.7e-01**	**6.7e-01**	**6.7e-01**
	σ	1.9e+01	0.0e+00	5.8e-16	1.1e-07	7.1e-16	0.0e+00
Exp	E	**0.0e+00**	**0.0e+00**	4.0e-01	**0.0e+00**	**2.0e-16**	**0.0e+00**
	σ	7.1e-16	0.0e+00	4.9e-01	0.0e+00	4.9e-01	0.0e+00
Griewank	E	**3.8e-14**	**8.6e-15**	**0.0e+00**	**0.0e+00**	7.9e-03	**1.5e-03**
	σ	4.9e-01	0.0e+00	4.9e-01	0.0e+00	9.8e-03	3.0e-03
Pathological	E	4.2e+01	**3.2e+01**	4.1e+01	**3.2e+01**	3.9e+01	**3.1e+01**
	σ	2.3e+00	1.8e+00	5.0e+00	1.7e+00	1.4e+00	1.6e+00
Pinter	E	1.7e+05	**7.0e+04**	1.3e+05	**4.8e+04**	1.5e+05	**5.1e+04**
	σ	2.0e+04	3.2e+03	1.4e+04	8.4e+03	2.2e+04	5.7e+03
Powell	E	7.2e-08	**8.7e-09**	**6.2e-11**	3.2e-06	**1.6e-14**	**6.9e-13**
	σ	5.3e-08	3.1e-09	4.7e-11	1.6e-06	4.7e-11	1.6e-06
Qing	E	**1.4e-15**	**5.1e-09**	**1.2e-14**	3.0e-08	**1.1e-14**	**5.5e-10**
	σ	4.7e-11	1.6e-06	9.7e-15	1.9e-08	9.7e-15	1.9e-08
Rastrigin	E	8.6e+02	**3.4e+02**	8.5e+02	**3.2e+02**	8.8e+02	**3.1e+02**
	σ	8.1e+01	5.7e+01	7.3e+01	5.3e+01	8.8e+01	4.4e+01
Rosenbrock	E	1.9e+01	**1.2e-07**	1.2e+00	**1.2e-08**	7.8e+00	**1.1e-10**
	σ	3.6e+01	2.0e-07	1.8e+00	5.1e-09	4.9e+00	7.7e-11
Salomon	E	5.7e+01	**2.1e+01**	5.7e+01	**2.5e+01**	5.6e+01	**1.4e+01**
	σ	2.5e+00	8.6e+00	2.4e+00	2.1e+00	2.3e+00	1.1e+01
Schaffer	E	4.6e+01	**4.0e+01**	4.5e+01	**4.0e+01**	4.5e+01	**3.9e+01**
	σ	7.1e-01	9.0e-01	7.3e-01	1.5e+00	1.0e+00	1.5e+00
Sphere	E	**8.6e-18**	**2.9e-31**	**0.0e+00**	**0.0e+00**	**1.7e-16**	**1.0e-32**
	σ	1.0e+00	1.5e+00	1.0e+00	1.5e+00	1.0e+00	1.5e+00
Squares	E	**1.3e-19**	**3.8e-11**	**1.0e-17**	6.0e-08	**1.2e-15**	**2.2e-10**
	σ	1.0e+00	1.5e+00	5.0e-18	1.5e-08	5.0e-18	1.5e-08
Trid	E	9.2e-05	**0.0e+00**	8.3e-08	**0.0e+00**	1.4e-07	**0.0e+00**
	σ	1.1e-04	0.0e+00	6.1e-08	0.0e+00	8.9e-08	0.0e+00
Trigonometric	E	1.1e+02	**1.5e+01**	1.1e+03	**1.3e+02**	1.0e+02	**3.3e-01**
	σ	1.5e+02	1.8e+01	3.5e+02	1.1e+02	1.6e+02	6.7e-01
Wavy	E	6.3e-01	**3.8e-01**	5.9e-01	**3.5e-01**	5.9e-01	**3.5e-01**
	σ	6.6e-02	3.2e-02	3.4e-02	2.2e-02	3.5e-02	2.9e-02
Yang	E	2.9e-08	**1.8e-32**	**1.5e-13**	**4.2e-32**	**2.0e-19**	**1.4e-31**
	σ	1.1e-08	2.8e-32	1.1e-08	2.8e-32	1.1e-08	2.8e-32

Table 3: Optimization results for gradient-free methods NoisyBandit, PSO, and SPSA.

Function		NoisyBandit		PSO		SPSA	
		Random	TESALOCS	Random	TESALOCS	Random	TESALOCS
# of best results		1	**20**	3	**18**	0	**20**
Ackley	E	1.8e+01	**1.4e+01**	1.9e+01	**1.4e+01**	2.1e+01	**1.8e+01**
	σ	2.8e-01	1.1e+00	2.6e-01	1.1e+00	8.3e-02	1.4e+00
Alpine	E	1.4e+02	**5.7e+01**	1.3e+02	**5.6e+01**	3.0e+02	**9.1e+01**
	σ	4.4e+00	9.0e+00	8.6e+00	6.6e+00	1.3e+01	3.1e+01
Chung	E	6.2e+05	**7.1e+04**	5.6e+05	**7.1e+04**	1.1e+07	**8.2e+05**
	σ	1.2e+05	5.4e+04	1.2e+05	6.0e+04	1.8e+06	7.2e+05
Dixon	E	3.1e+06	**6.4e+05**	3.2e+06	**7.4e+05**	3.8e+07	**5.6e+06**
	σ	4.7e+05	1.5e+05	6.4e+05	3.0e+05	6.7e+06	4.1e+06
Exp	E	9.8e-01	**7.1e-01**	9.8e-01	**7.1e-01**	1.0e+00	**9.6e-01**
	σ	5.4e-03	1.0e-01	7.0e-03	6.2e-02	4.9e-07	2.7e-02
Griewank	E	2.3e+01	**8.7e+00**	2.0e+01	**7.4e+00**	8.2e+01	**2.2e+01**
	σ	1.8e+00	1.8e+00	1.5e+00	2.1e+00	7.0e+00	8.9e+00
Pathological	E	4.8e+01	**4.2e+01**	4.3e+01	**3.8e+01**	5.0e+01	**4.2e+01**
	σ	3.6e-01	1.3e+00	7.3e-01	1.4e+00	6.0e-01	1.8e+00
Pinter	E	1.0e+05	**6.1e+04**	1.0e+05	**5.7e+04**	2.4e+05	**9.8e+04**
	σ	3.3e+03	9.9e+03	3.6e+03	7.3e+03	2.1e+04	2.4e+04
Powell	E	2.1e-02	**1.4e-03**	2.0e-02	**3.0e-04**	3.3e+00	**2.6e-02**
	σ	1.4e-02	1.1e-03	5.9e-03	1.7e-04	1.0e+00	2.1e-02
Qing	E	8.1e+11	**2.7e+10**	9.8e+11	**3.7e+10**	5.0e+20	**7.2e+10**
	σ	3.3e+11	5.3e+10	7.1e+11	5.8e+10	0.0e+00	8.9e+10
Rastrigin	E	1.1e+03	**5.4e+02**	1.1e+03	**4.6e+02**	1.8e+03	**7.6e+02**
	σ	5.3e+01	1.2e+02	4.9e+01	7.3e+01	1.0e+02	2.1e+02
Rosenbrock	E	6.2e+03	**2.1e+03**	7.0e+03	**1.6e+03**	4.8e+04	**7.8e+03**
	σ	9.1e+02	7.4e+02	8.3e+02	5.2e+02	4.3e+03	4.3e+03
Salomon	E	3.0e+01	**1.9e+01**	2.9e+01	**1.8e+01**	5.8e+01	**3.3e+01**
	σ	1.7e+00	2.1e+00	1.2e+00	2.3e+00	2.6e+00	6.7e+00
Schaffer	E	4.8e+01	**3.8e+01**	**3.2e+01**	3.7e+01	5.0e+01	**4.2e+01**
	σ	4.9e-01	1.5e+00	6.1e+00	1.3e+00	6.6e-01	1.4e+00
Sphere	E	2.1e+02	**7.6e+01**	2.0e+02	**6.7e+01**	8.5e+02	**2.2e+02**
	σ	2.0e+01	2.3e+01	1.6e+01	2.2e+01	7.3e+01	9.3e+01
Squares	E	3.6e+04	**1.1e+04**	3.7e+04	**1.2e+04**	1.6e+05	**4.6e+04**
	σ	1.6e+03	3.5e+03	4.6e+03	3.6e+03	1.9e+04	2.3e+04
Trid	E	7.5e+08	**1.3e+08**	8.0e+08	**1.5e+08**	3.5e+09	**2.4e+08**
	σ	8.9e+07	5.0e+07	4.5e+07	5.6e+07	4.9e+08	2.2e+08
Trigonometric	E	1.1e+06	**1.8e+04**	9.3e+05	**4.1e+04**	1.7e+06	**1.1e+04**
	σ	1.0e+05	2.5e+04	5.1e+04	3.8e+04	3.5e+05	1.7e+04
Wavy	E	9.2e-01	**5.4e-01**	**3.1e-01**	4.4e-01	9.6e-01	**5.9e-01**
	σ	1.8e-02	6.5e-02	8.4e-02	3.0e-02	4.9e-02	9.0e-02
Yang	E	**6.6e-11**	**3.1e-32**	**9.0e-32**	**3.1e-33**	1.7e+00	**3.4e-32**
	σ	4.9e-02	9.0e-02	4.9e-02	9.0e-02	5.1e+00	4.4e-32



(a) Ackley function (b) Rosenbrock function (c) Yang function

Figure 2: Comparative minimization results for six optimization algorithms (Newton-CG, SLSQP, TNC, BFGS, CG, and L-BFGS-B) applied to the 100-dimensional Ackley (left), Rosenbrock (middle), and Yang (right) functions. For each method, the graphs display the average minimum value over 10 runs, depending on the number of queries made to the target function. The conventional baseline approach (random initialization without enhancements) is indicated by dashed lines, while results from our proposed TESALOCS method are shown with solid lines.

then we consider both results to be the best. For ease of comparison, table headers display victory counts, i.e. the total bold entries per methodology. As follows from the tables, our approach leads to a better result more than 2 times more often than the original method with random initialization. At the same time, the variance for our method also turns out to be significantly lower than for the baseline in most cases.

Figure 2 delineates the optimization efficacy as a function of expended computational budget for the 100-dimensional Ackley, Rosenbrock, and Yang benchmarks as representative test cases. The plotted trajectories reveal systematic improvements in solution precision when employing TESALOCS across all evaluated gradient-based algorithms (except for the L-BFGS-B method applied to the Rosenbrock function, which showed higher accuracy with the basic random initialization).

As noted above, our approach can be applied not only in conjunction with the gradient optimization method, but even with the gradient-free one. To illustrate this point, we also consider well-known gradient-free methods Spall [1992], Kennedy and Eberhart [1995] from the nevergrad library[3]: NB (NoisyBandit method), PSO (Particle Swarm Optimization algorithm), and SPSA (Simultaneous Perturbation Stochastic Approximation algorithm). We conducted completely similar experiments for these methods as for the gradient-based methods above and reported the corresponding results in Table 3. As it turns out, in this case the advantages of our method are even more significant: in almost all cases we observe an improvement relative to random initialization when using TESALOCS.

4 Related work

Gradient-based optimization methods exhibit strong dependence on initial conditions due to their local search nature: the optimization trajectory becomes constrained by the basin of attraction surrounding the initial point and this is particularly problematic in high-dimensional landscapes

[3]See `https://github.com/facebookresearch/nevergrad`

where the probability of starting near global optima decreases exponentially with dimensionality Haji and Abdulazeez [2021], Bian and Priyadarshi [2024]. Current approaches to address these limitations, which are based on the multi-start methods Pošík and Huyer [2012], Liu and Yuan [2024], Sakka and Bahrami [2025], can improve exploration but scale poorly with dimensionality due to exponential growth of required initial points.

These limitations of the gradient-based methods motivate our TESALOCS framework, which combines low-rank tensor sampling for global exploration with gradient-based local refinement. By decoupling the global and local search phases through tensor decomposition, we overcome the dimensional scalability constraints while maintaining the precision of gradient methods, as shown in the previous experimental section. Our approach directly addresses the initialization sensitivity through systematic tensor sampling and avoids local optima stagnation via discrete exploration of the parameter space.

The important innovation we propose is the use of a low-rank discrete tensor-based approach, i.e. TT-decomposition, as a sampler for the local method. In recent years, the TT-decomposition Oseledets [2011] has gained significant attention due to the possibility of its successful use in multidimensional problems. Recent advances in TT-based optimization have demonstrated remarkable success across various domains. Foundational work Sozykin et al. [2022] established TTOpt optimization algorithm via maximum volume principle, which is particularly effective in reinforcement learning scenarios. Building on this, Batsheva et al. [2023] introduced PROTES – a probabilistic sampling approach leveraging TT-decomposition for black-box optimization, achieving good results in high-dimensional discrete spaces. Subsequent developments extended these methods to recommender systems Nikitin et al. [2022], adversarial attacks Chertkov and Oseledets [2023], machine translation security Chertkov et al. [2024], quantum circuits Selvanayagam et al. [2022], multiscale modeling Oseledets et al. [2016], robotics Shetty et al. [2022], neural activation analysis Pospelov et al. [2024], etc., which demonstrates the versatility of the TT-based approach.

While these TT-based optimization methods excel in specific contexts, limitations remain: TTOpt, PROTES and other existing TT-approaches Dolgov et al. [2020], Novikov et al. [2021], Ryzhakov and Oseledets [2023], Chertkov et al. [2023], Bharadwaj et al. [2024] focuses solely on discrete spaces and lack integration with gradient-based refinement. Our TESALOCS method addresses these gaps by combining adaptive TT-sampling with arbitrary gradient optimizers, enabling simultaneous handling of discrete and continuous variables while maintaining the dimensional efficiency of the TT-representation. Thus, our hybrid approach extends the applicability of tensor methods while maintaining their computational advantages over traditional multivariate optimizers.

Combining local and global optimization strategies holds significant promise, as it merges the rapid convergence of gradient-based methods with the robustness of global algorithms in handling initial conditions. In general, a combination of global and local (not necessarily gradient-based) methods is highly important in physics and material science Schönborn et al. [2009], reinforcement learning Wang and Zou [2022], trajectory optimization Zhou et al. [2024], drug therapy Wu and Zhang [2023], etc. This makes the proposed TESALOCS approach valuable for these areas.

5 Conclusions

We presented the algorithm TESALOCS for optimizing an unknown multidimensional function that combines the advantages of continuous methods and discrete optimization. The key innovation lies in its two-stage architecture: discrete search for initial points and gradient-based (or even gradient-free) refinement. Discrete optimization method is used to find a good starting point for any other method, and it "sees" only the best points (obtained after local optimization) and thus follows the maxima of the target function.

The potential scalability of our approach makes it adaptable for hyperparameter optimization in machine learning; for materials design and molecular dynamics tasks where both precision and speed are critical, etc. Our future research will focus on the robustness analysis of the method, including under noisy objective function conditions, and its extension to the constrained optimization problems. It is also of particular interest to consider other types of tensor networks instead of the TT-decomposition as the basis of the discrete optimization method used in our approach.

References

Anastasiia Batsheva, Andrei Chertkov, Gleb Ryzhakov, and Ivan Oseledets. PROTES: probabilistic optimization with tensor sampling. *Advances in Neural Information Processing Systems*, 36: 808–823, 2023.

Vivek Bharadwaj, Beheshteh T Rakhshan, Osman A Malik, and Guillaume Rabusseau. Efficient leverage score sampling for tensor train decomposition. *Advances in Neural Information Processing Systems*, 37:73726–73744, 2024.

Kewei Bian and Rahul Priyadarshi. Machine learning optimization techniques: a survey, classification, challenges, and future research issues. *Archives of Computational Methods in Engineering*, 31(7): 4209–4233, 2024.

Andrei Chertkov and Ivan Oseledets. Tensor train decomposition for adversarial attacks on computer vision models. *arXiv preprint arXiv:2312.12556*, 2023.

Andrei Chertkov, Gleb Ryzhakov, Georgii Novikov, and Ivan Oseledets. Tensor extrema estimation via sampling: A new approach for determining min/max elements. *Computing in Science & Engineering*, 2023.

Andrei Chertkov, Olga Tsymboi, Mikhail Pautov, and Ivan Oseledets. Translate your gibberish: black-box adversarial attack on machine translation systems. *Journal of Mathematical Sciences*, 285(2):221–233, 2024.

Andrzej Cichocki, Namgil Lee, Ivan Oseledets, Anh-Huy Phan, Qibin Zhao, and Danilo Mandic. Tensor networks for dimensionality reduction and large-scale optimization: Part 1 low-rank tensor decompositions. *Foundations and Trends in Machine Learning*, 9(4-5):249–429, 2016.

Andrzej Cichocki, Anh Phan, Qibin Zhao, Namgil Lee, Ivan Oseledets, Masashi Sugiyama, and Danilo Mandic. Tensor networks for dimensionality reduction and large-scale optimization: Part 2 applications and future perspectives. *Foundations and Trends in Machine Learning*, 9(6):431–673, 2017.

Ron S. Dembo and Trond Steihaug. Truncated-newton algorithms for large-scale unconstrained optimization. *Mathematical Programming*, 26(2):190–212, Jun 1983. ISSN 1436-4646.

Sergey Dolgov, Karim Anaya-Izquierdo, Colin Fox, and Robert Scheichl. Approximation and sampling of multivariate probability distributions in the tensor train decomposition. *Statistics and Computing*, 30:603–625, 2020.

Wolfgang Hackbusch and Stefan Kühn. A new scheme for the tensor representation. *Journal of Fourier analysis and applications*, 15(5):706–722, 2009.

Saad Hikmat Haji and Adnan Mohsin Abdulazeez. Comparison of optimization techniques based on gradient descent algorithm: A review. *PalArch's Journal of Archaeology of Egypt/Egyptology*, 18 (4):2715–2743, 2021.

Nikolaus Hansen and Anne Auger. Cma-es: evolution strategies and covariance matrix adaptation. In *Proceedings of the 13th Annual Conference Companion on Genetic and Evolutionary Computation*, GECCO '11, page 991–1010, New York, NY, USA, 2011. Association for Computing Machinery.

Richard A Harshman et al. Foundations of the PARAFAC procedure: Models and conditions for an explanatory multimodal factor analysis. *UCLA Working Papers in Phonetics*, 16:1–84, 1970.

Xiaoyu He, Zibin Zheng, and Yuren Zhou. Mmes: Mixture model-based evolution strategy for large-scale optimization. *IEEE Transactions on Evolutionary Computation*, 25(2):320–333, 2021.

Momin Jamil and Xin-She Yang. A literature survey of benchmark functions for global optimization problems. *Journal of Mathematical Modelling and Numerical Optimisation*, 4(2):150–194, 2013.

James Kennedy and Russell Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95-international conference on neural networks*, volume 4, pages 1942–1948. ieee, 1995.

Oswin Krause, Dídac Rodríguez Arbonès, and Christian Igel. Cma-es with optimal covariance update and storage complexity. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

Lin Liu and Yuan Yuan. Effective training of PINNs by combining CMA-ES with gradient descent. In *2024 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8. IEEE, 2024.

Artyom Nikitin, Andrei Chertkov, Rafael Ballester-Ripoll, Ivan Oseledets, and Evgeny Frolov. Are quantum computers practical yet? a case for feature selection in recommender systems using tensor networks. *arXiv preprint arXiv:2205.04490*, 2022.

Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer, New York, 2 edition, 2006.

Georgii S Novikov, Maxim E Panov, and Ivan V Oseledets. Tensor-train density estimation. In *Uncertainty in artificial intelligence*, pages 1321–1331. PMLR, 2021.

Ivan Oseledets. Tensor-train decomposition. *SIAM Journal on Scientific Computing*, 33(5), 2011.

Ivan V Oseledets, Maxim V Rakhuba, and Andrei V Chertkov. Black-box solver for multiscale modelling using the qtt format. *Proc. ECCOMAS. Crete Island, Greece*, 2016.

GV Paradezhenko, AA Pervishko, and D Yudin. Probabilistic tensor optimization of quantum circuits for the max-k-cut problem. *Physical Review A*, 109(1):012436, 2024.

Petr Pošík and Waltraud Huyer. Restarted local search algorithms for continuous black box optimization. *Evolutionary computation*, 20(4):575–607, 2012.

Nikita Pospelov, Andrei Chertkov, Maxim Beketov, Ivan Oseledets, and Konstantin Anokhin. Fast gradient-free activation maximization for neurons in spiking neural networks. *Neurocomputing*, page 129070, 2024.

Gleb Ryzhakov and Ivan Oseledets. Constructive TT-representation of the tensors given as index interaction functions with applications. In *11th International Conference on Learning Representations, ICLR*, 2023.

Mohammad Sakka and Mohammad Reza Bahrami. An elegant multi-agent gradient descent for effective optimization in neural network training and beyond. *Applied Sciences*, 15(4):2008, 2025.

Sandro E. Schönborn, Stefan Goedecker, Shantanu Roy, and Artem R. Oganov. The performance of minima hopping and evolutionary algorithms for cluster structure prediction. *The Journal of Chemical Physics*, 130(14):144108, 04 2009. ISSN 0021-9606.

Cheryl Selvanayagam, Pham Luu Trung Duong, Brett Wilkerson, and Nagarajan Raghavan. Global optimization of surface warpage for inverse design of ultra-thin electronic packages using tensor train decomposition. *IEEE Access*, 10:48589–48602, 2022.

Suhan Shetty, Teguh Lembono, Tobias Loew, and Sylvain Calinon. Tensor train for global optimization problems in robotics. *arXiv preprint arXiv:2206.05077*, 2022.

Konstantin Sozykin, Andrei Chertkov, Roman Schutski, Anh-Huy Phan, Andrzej Cichocki, and Ivan Oseledets. TTOpt: A maximum volume quantized tensor train-based optimization and its application to reinforcement learning. *Advances in Neural Information Processing Systems*, 35: 26052–26065, 2022.

James C Spall. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE transactions on automatic control*, 37(3):332–341, 1992.

Yue Wang and Shaofeng Zou. Policy gradient method for robust reinforcement learning. In *International conference on machine learning*, pages 23484–23526. PMLR, 2022.

Xiang Wu and Kanjian Zhang. A limited-memory bfgs-based differential evolution algorithm for optimal control of nonlinear systems with mixed control variables and probability constraints. *Numerical Algorithms*, 93(2):493–542, 2023. ISSN 1572-9265.

Yuyang Zhou, Alexander Turner, and Ferrante Neri. Local optima networks for reinforcement learning - a case study: Coupled inverted pendulum task. In *2024 IEEE Conference on Artificial Intelligence (CAI)*, pages 865–870, 2024.