

RESEARCH ARTICLE | FEBRUARY 05 2024

Benchmarking inverse optimization algorithms for materials design

Hanfeng Zhai ; Hongxia Hao; Jingjie Yeo  



APL Mater. 12, 021107 (2024)

<https://doi.org/10.1063/5.0177266>



View
Online



Export
Citation

CrossMark



APL Materials

Special Topic: Ultrafast Materials Science:
Coherence and Dynamics

Submit Today!

 AIP
Publishing

Benchmarking inverse optimization algorithms for materials design

Cite as: APL Mater. 12, 021107 (2024); doi: [10.1063/5.0177266](https://doi.org/10.1063/5.0177266)

Submitted: 20 September 2023 • Accepted: 13 December 2023 •

Published Online: 5 February 2024



View Online



Export Citation



CrossMark

Hanfeng Zhai,^{1,a,b)} Hongxia Hao,^{2,c)} and Jingjie Yeo^{1,d)}

AFFILIATIONS

¹ Sibley School of Mechanical and Aerospace Engineering, Cornell University, Ithaca, New York 14850, USA

² Microsoft Research AI4Science, Beijing, China

Note: This paper is part of the Special Topic on Open Framework Materials.

^aElectronic mail: hz253@cornell.edu

^bPresent address: Department of Mechanical Engineering, Stanford University, Stanford, California 94305, USA.

^cElectronic mail: hongxiahao@microsoft.com

^dAuthor to whom correspondence should be addressed: jingjeyeo@cornell.edu

ABSTRACT

Machine learning-based inverse materials discovery has attracted enormous attention recently due to its flexibility in dealing with black box models. Yet, many metaheuristic algorithms are not as widely applied to materials discovery applications as machine learning methods. There are ongoing challenges in applying different optimization algorithms to discover materials with single- or multi-elemental compositions and how these algorithms differ in mining the ideal materials. We comprehensively compare 11 different optimization algorithms for the design of single- and multi-elemental crystals with targeted properties. By maximizing the bulk modulus and minimizing the Fermi energy through perturbing the parameterized elemental composition representations, we estimated the unique counts of elemental compositions, mean density scan of the objectives space, mean objectives, and frequency distributed over the materials' representations and objectives. We found that nature-inspired algorithms contain more uncertainties in the defined elemental composition design tasks, which correspond to their dependency on multiple hyperparameters. Runge–Kutta optimization (RUN) exhibits higher mean objectives, whereas Bayesian optimization (BO) displayed low mean objectives compared with other methods. Combined with materials count and density scan, we propose that BO strives to approximate a more accurate surrogate of the design space by sampling more elemental compositions and hence have lower mean objectives, yet RUN will repeatedly sample the targeted elemental compositions with higher objective values. Our work sheds light on the automated digital design of materials with single- and multi-elemental compositions and is expected to elicit future studies on materials optimization, such as composite and alloy design based on specific desired properties.

© 2024 Author(s). All article content, except where otherwise noted, is licensed under a Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>). <https://doi.org/10.1063/5.0177266>

07 February 2024 03:28:45

I. INTRODUCTION

Designing materials with desired tailored properties has always been a key goal in human society, from metallurgy, biomedicine, energy storage, to nanotechnology. Traditionally, most research studies conducted in the field of designing materials with multi-elemental compositions rely on costly experimental synthesis and time-consuming trial-and-error processes. Advances in computational engineering can greatly accelerate materials design efficiency. To achieve the design goal, many optimization methods were developed for the design process. These methods can be categorized as first and zeroth orders, where the first-order optimization methods are gradient-based methods, such as gradient descent and

Newton–Raphson, that rely on the problem being first-order differentiable. The first-order methods are widely applied to topology optimization coupled with finite element methods for mechanical design, including classical problems such as the design of beams¹ and trusses.² However, the primary drawback of first-order optimization is that the differential form of the problems may not exist, or be easily obtained, in many (or most) real-world application scenarios. In many cases, the governing equations for the problem do not exist as differentiable forms. Moreover, first-order optimization tends to get “trapped” in local optima close to the initial design, which heavily relies on prior physical understanding of the problem to ameliorate. Hence, metaheuristic optimization methods, such as genetic algorithms and particle swarm optimization,

are more flexible and adaptive to complex problem domains. These characteristics enable metaheuristics to be applicable to multi-scale materials design problems, such as molecular simulations, machine learning (ML) surrogates, and density functional theory calculations. For example, molecular dynamics (MD) simulations model the interactions of particles based on interatomic potentials,^{3,4} DFT simulations calculate the energy based on pseudo-potentials for the Hamiltonian,⁵ and ML surrogates represent the material's structure–property map with a neural network.⁶ In these situations, obtaining first-order derivatives is either an extremely taxing task or may even be intractable. To overcome these challenges, recent literature highlights the utility of metaheuristics for elemental composition design, in contrast to gradient-based optimizations. For example, Ramprasad and co-workers developed genetic algorithm methods coupled with deep learning surrogate models for designing polymer dielectrics⁷ and used similar strategies to design polymers with high glass transition temperatures and bandgaps.⁸ Winter *et al.* combined machine learning surrogates with particle swarm optimization (PSO) to design drugs.⁹ Weiel *et al.* used PSO to select parameters in molecular dynamics simulations.¹⁰

Many machine learning methods exist for design optimization, and two of the most widely applied methods are Bayesian optimization (BO) and deep reinforcement learning (DRL). BO leverages Gaussian process regression (GPR) to fit a surrogate model of the design space and leverages Bayesian statistics to iteratively improve the accuracy of such a surrogate by sampling more points via an acquisition function. BO is frequently applied in materials design. For example, Shin *et al.* combined the micromechanical experiments and BO to design spiderweb nanomechanical resonators.¹¹ Chen and co-workers developed a series of BO frameworks for mechanical design and elemental composition design using finite element simulations,¹² molecular simulations,^{13,14} and machine learning surrogate models.^{15,16} Zhai and Yeo combined BO and individual-based modeling of bacteria to design antimicrobial nanosurfaces.¹⁷ In another popular method, DRL utilizes a deep neural network (DNN) as an agent to interact with the environment to receive rewards from different output properties and learn the best strategy to achieve the design goal. When applying DRL for materials design, the evaluations of the material properties are treated as the environment for the agent to learn the correct policy to achieve the design goal. For instance, Gu and co-workers developed a DRL framework for designing tough graphene oxide¹⁸ and continuum composite materials.¹⁹ Barati Farimani and co-workers developed DRL frameworks for porous graphene for water desalination.²⁰

Can these optimization methods be uniformly applied to the same problem and how do they differ? To answer these questions, benchmark studies are urgently needed. One particular reason for the urgency is efficiency. Optimization methods are intended to improve the efficiency of the design problems. However, when companies or research groups want to employ design optimization for their problems of interest, it is of utmost importance to select optimization methods that are best suited to their specific problems. Selecting ideal optimization methods from the massive amount of the existing literature is non-trivial: one may fall into the “trial-and-error” trap, and the efficiency of solving the problem is significantly reduced. Moreover, ML-based studies for materials

design are often focused on either large-scale molecular structures (i.e., polymers) or continuum structures. This is in part because it is easier to extract structure–property relationships for larger structural bodies in the context of continuum modeling, where first-order derivatives are easy to obtain. Specifically, for materials composed of multiple elements, many studies leverage molecular graphs or use natural language processing to find this structure–property map. In contrast, while there are some studies on the design of small molecules using language representation with generative models,²¹ a huge gap exists in applying general optimization methods for designing such multi-elemental compositions. Therefore, following our previous discussion, the specific problem to be posed herein is how different optimization methods differ in the context of materials design.

To address these challenges, we propose and conduct a benchmark study by designing a framework to implement many state-of-the-art optimization methods. We benchmark many widely employed metaheuristic optimizations, including Bayesian optimization, genetic algorithms, particle swarm optimization, and simulated annealing. We further compared novel optimization methods, such as the arithmetic optimization algorithm and Runge–Kutta optimization, for a total of 11 different optimization algorithms. We took advantage of the large number of open-source software applications for computational elemental composition design, including pymatgen,²² Materials Project,²³ matminer,²⁴ and mealy.²⁵ The material evaluations are enabled by surrogate models: pretrained graph neural network models to predict the elemental compositions' properties based on input graph structure. The general schematic of our study is shown in Fig. 1. We begin with inspiration from the general premise of industrial or fundamental research problems: one needs to design elemental compositions based on specific applications. Given this application, the targeted physical properties must be identified based on the design requirements, such as high toughness or low thermal conductivity. Based on these targeted applications and properties, the design problem can then be formulated mathematically together with the corresponding optimization algorithms. The design space can be explored to extract the optimal elemental compositions from the materials database(s). Here, we will mainly focus on the optimization process [Figs. 1(c)–1(e)] to propose simplified target material properties and focus on the algorithmic implementations and benchmarking. The significance of this work lies in that there is a lack of studies that evaluate the performance of different optimization methods, in contrast with the abundance of published literature that applies specific optimization methods for tailored materials design tasks. Our study may benefit future research in the following ways: (1) Researchers can select different algorithms that correspond well with their potential applications based on our evaluation results, and (2) researchers can use or refer to our framework and benchmark other optimization algorithms of their interest. Our major contributions are to (1) propose a framework that can implement different algorithms, which generalizes the materials' inverse design tasks, (2) evaluate different optimization methods based on defined metrics, and (3) highlight ideal algorithms for different potential applications based on our evaluations. Some results were unexpected: the commonly employed BO did not stand out in terms of mean objective evaluation, while surprisingly RUN outperformed both the stability and objective evaluations.

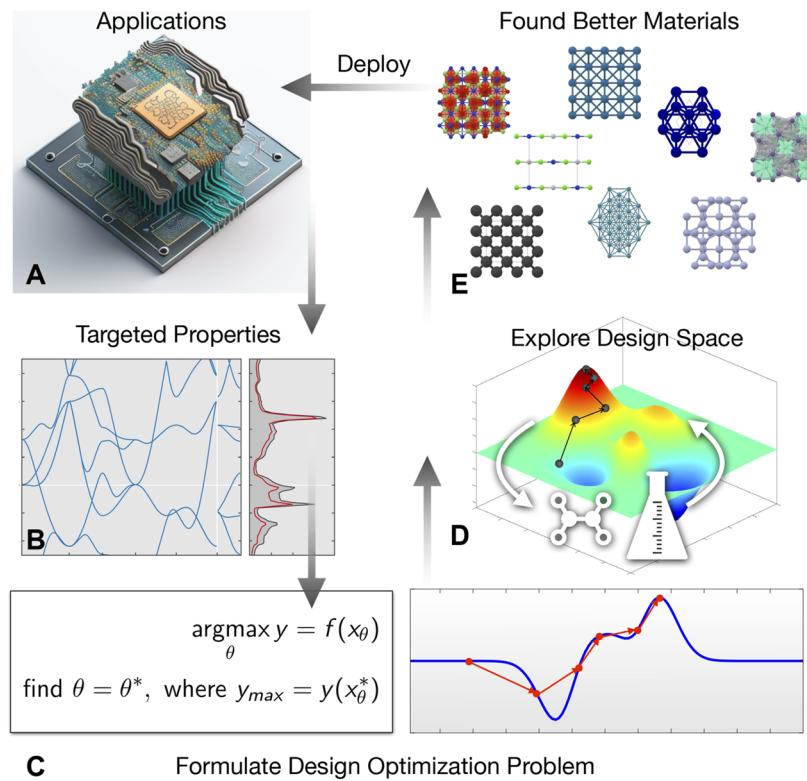


FIG. 1. General overview of the materials design process. (a) Schematic for defining the intended industrial application. (b) The process of identifying the desired material properties, such as high bulk modulus and low Fermi energy. (c) The schematic for formulating a digital design optimization problem using numerical models to discover or design elemental composition with the desired properties. (d) Different optimization methods use exploration/exploitation strategies to generate a design space for exploring the best materials. (e) The optimal elemental compositions are extracted from the design space exploration and are ready for real-world applications. In this study, we focus on formulating an optimization framework that enables the application of the most current popular optimization methods to search for desired elemental compositions and compare different optimizations based on specific metrics. Finally, the “mined” elemental compositions are analyzed.

07 February 2024 03:28:45

This paper is articulated as follows: We present an overview of the underlying methodology, describe the workflow of connecting machine-learned models of materials properties with different optimization methods, and benchmark the results obtained from 11 different optimization algorithms in Sec. III. We then discuss the algorithms that are developed to perform the automated optimizations in Sec. IV. A detailed explanation of the methodologies employed is presented in Sec. II, and further information is contained in the supplementary material.

II. METHODS

A. Inverse optimization framework

1. Problem formulation

Our goal is to benchmark design optimization algorithms for inverse elemental composition design, in which we formulate the problem as maximizing an objective function corresponding to materials properties. We benchmark using a model problem of maximizing the bulk modulus while simultaneously minimizing the Fermi energy. While not specific to any applications, our prototype model is useful for benchmarking and unveiling the characteristics

of different optimization algorithms for elemental composition design and discovery. To maximize an objective function, \mathcal{J} , as a single-objective optimization problem, which is defined as the difference between the bulk modulus and the Fermi energy, $\mathcal{J} = K - E_{\text{Fermi}}$,

$$\underset{n_{\text{atom}}, \xi_n, \eta}{\operatorname{argmax}} \mathcal{J} = K - E_{\text{Fermi}},$$

where

$$K, E_{\text{Fermi}} = \text{MEGNet}(\mathcal{G}_\Theta), \quad \rightarrow \mathcal{G} = \Omega(n_{\text{atom}}, \xi_n, \eta); \quad (1)$$

$$\Theta = [n_{\text{atom}}, \xi_n, \eta]$$

subject to

$$n_{\text{atom}} \in [1, 4] \text{ or } \equiv 1, \quad \xi_n \in [0, 100], \quad \eta \in [0, 100].$$

Here, the problem is defined as a discrete constrained optimization problem. The input graph structure \mathcal{G} is related to the properties K and E_{Fermi} , which can be considered as a forward mapping. The forward mapping of fast property predictions is enabled based on the graph neural network structure MatErials Graph Network (MEGNet).²⁶ Further details are provided in Sec. II and the supplementary material.

The optimization problem can be solved using various optimization algorithms. Here, we explored 11 different state-of-the-art optimizations to conduct benchmarking for materials discovery. The methods include Bayesian optimization,²⁷ deep reinforcement learning,²⁸ genetic algorithm,²⁹ particle swarm optimization,³⁰ hybrid gray wolf optimization,³¹ (hybrid-improved) whale optimization algorithm,³² ant colony optimization,³³ differential evolution,³⁴ simulated annealing,³⁵ bees optimization,³⁶ arithmetic optimization,³⁷ and Runge–Kutta (RUN) optimization.³⁸ These algorithms fall into two main categories: machine learning-based optimization and metaheuristic optimization [Figs. 2(b) and 2(c)]. We use different strategies to perturb Θ to maximize \mathcal{J} , which will be elaborated in Sec. II and the supplementary material. Here, we consider single- and multi-element composition optimization scenarios. For single-element composition optimization, the design variable n_{atom} is restricted to 1, and there are only two tunable design variables ξ_n and η . For multi-element composition optimization, the screened elemental compositions contain $n_{atom} \in [1, 4]$. The materials libraries as indicated in Fig. 2 consist of a series of MP IDs, and the randomly generated η is used to select the corresponding MP ID.

Following the discussion from Eq. (1), the optimization is benchmarked and connected through the material evaluations. We utilize MEGNet for materials properties evaluation mainly for two considerations: (1) Fast inference: by using the pretrained MEGNet model (mp-2019.4.1), the properties can be obtained “on the fly” given the input graph structure, getting rid of the lengthy computational process via traditional numerical methods such as molecular dynamics (MD) or density functional theory (DFT). (2) Better screenability of the design space. Given the pretrained MEGNet model, one can infer the corresponding properties given the generated input graph \mathcal{G} , hence allowing the evaluation of a large amount of “untouched” materials. Meanwhile, due to the lack of existing data for the properties of many potential materials, one may not directly

obtain the properties given the input elements just from a query. In addition, due to the lack of pseudo- and/or interatomic potentials, MD or DFT calculations may not be conducted. Here, we employ MEGNet due to its reported accuracy and stability. As reported by Chen *et al.*,²⁶ the MAE for the prediction of Fermi energy and bulk modulus is around 0.028 and 0.05, respectively. MEGNet is not the only option for materials properties inference; other existing frameworks with a competitive performance include CGCNN,³⁹ MODNet,³⁹ H-CLMP(T),⁴⁰ and crystal graph attention networks.⁴¹ Since our focus is on optimization methods, we do not emphasize the surrogate model evaluations. Notably, many recent studies focus on materials property prediction benchmarks.^{42,43}

For this problem, the inverse optimization problem is to directly perturb the graph structure such that the materials’ properties can be tailored. However, it is difficult to represent different materials’ graph structures containing different elemental numbers in the same dimensions for optimization. Hence, we parameterize the graph structure $\mathcal{G} = \mathcal{G}_\Theta$ so that the algorithms tune the design variables $\Theta = [n_{atom}, \xi_n, \eta]$ to change the elemental composition \mathcal{G}_Θ . Here, as visualized in Fig. 2(a), the optimization begins with a variable n_{atom} , as the number represents different types of elements in the elemental compositions. We may denote the different atoms in the elemental compositions as the *atomic dimensions*. Associated with this n_{atom} , we define a random number distributed in [0,100] as ξ_n to assign atomic types to the atomic dimensions, which we may define as the *material basis*. For example, if the atomic dimension $n_{atom} = 3$, we may obtain an element basis from ξ_n like {C–H–O}. Given this material basis, one can generate a list of potential elemental compositions from the *Materials Project*, for which we denote the *materials library*. One can then use η to select the candidate elemental compositions from the materials library. If following our previous example, the materials library may look like {CH₃O, C₂H₆O, C₃H₆O, ...} and our selected material may be C₂H₆O. For how ξ_n and η determine the

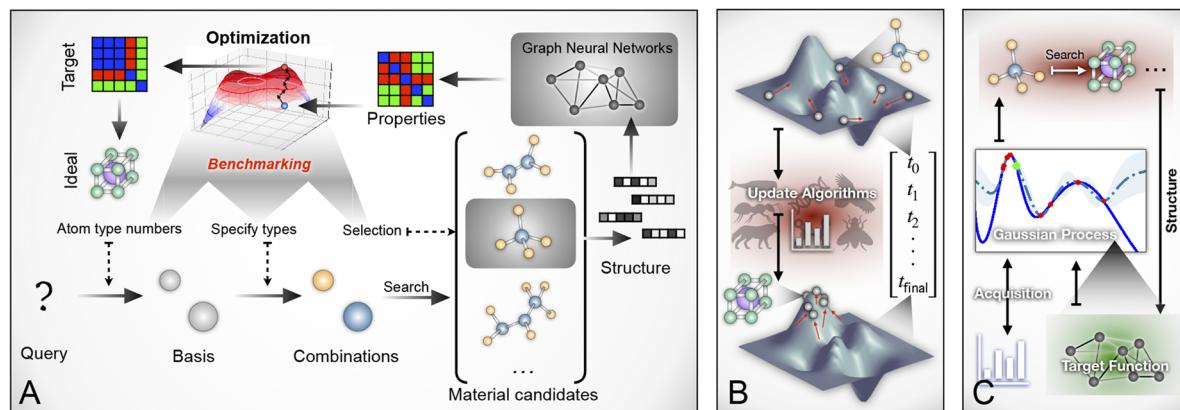


FIG. 2. Schematic for optimization benchmarking and different optimization algorithms employed in this study. (a) Workflow for discovering elemental compositions by combining graph neural network (GNN) surrogate models with defined design variable randomization to select optimal elemental compositions with targeted materials properties. See Sec. II and the supplementary material for details. (b) Schematic for optimization through metaheuristic algorithms, starting with a set of random particles (or population) sampled in the design space, to iteratively update and eventually cluster around globally optimal values, indicating the search for elemental compositions with desired properties. (c) Bayesian optimization methods for elemental composition discovery, seeking to build a surrogate model of the design space by employing Gaussian process regression and searching for the next material evaluations based on an acquisition function. Optimal elemental compositions can be extracted from the design space surrogate.

atomic graph structure \mathcal{G} in the algorithm, please refer to *Extension on Problem Formulation* in the supplementary material.

2. Workflow automation

One of the main contributions of our work is the development of an optimization framework that is flexible for many implementations of state-of-the-art optimization methods. The key point in such a framework is the automation of the optimization, where the update of the next set and the current set of materials evaluation is connected. The automation is enabled through the graph neural network surrogate model as formulated in Eq. (1), the MEGNet. The whole workflow can be simplified to the following expression:

$$\begin{aligned} & \underbrace{\left(\text{MEGNet}(\Theta_{n+1}) \Theta_{n+1} \right)}_{\text{exploration}} \\ & \xrightarrow{\text{search}} \text{IOA}(\Theta_n, \underbrace{\text{MEGNet}(\Theta_n); \mathbf{P}_{IOA}}_{\text{exploitation}}), \quad (2) \\ & \xleftarrow{\text{feedback}} \end{aligned}$$

where IOA stands for “inverse optimization algorithm.” Assuming that the algorithm evaluates the n th set of elemental compositions with the design variable Θ_n , obtaining the corresponding properties $\text{MEGNet}(\Theta_n)$, and by employing the hyperparameters \mathbf{P}_{IOA} specified by different optimization methods, one then obtain the design variables for the next set of evaluation Θ_{n+1} , for which the evaluated properties $\text{MEGNet}(\Theta_{n+1})$ then feed back to the IOA and iteratively search for the next set. This loop allows one to get rid of the traditional “trial-and-error” approach and only need to set the hyperparameters to obtain the optimal elemental compositions given the targeted properties.

The algorithmic flow chart is visualized in Fig. 3 [Eqs. (1) and (2)]. The formulated basic optimization method is shown in the right block, while the materials generation and evaluation processes are laid out in the left blocks. In our approach, the algorithm is initialized and repeated based on different random seeds for the five attempts in Fig. 5. The hyperparameters for each algorithm are provided in the supplementary material and were selected from commonly used default values for each algorithm.

B. Optimization algorithms

1. Bayesian optimization

The core of machine learning (ML)-based optimization algorithms is to employ the explored materials data to fit a surrogate assisting the optimization process. Here, there are two main ways of adopting ML methods for design optimization: (1) building up a surrogate model of the design space for final exploration, namely the Bayesian optimization method; (2) utilizing a deep neural network (DNN) to learn the interactions of input elemental compositions and their corresponding material properties, which can be considered as the interaction of actions and their environment in which this DNN is considered as an “agent.” This method is known as deep reinforcement learning. The detailed mathematical formulation and derivations are shown in the *Machine Learning Based Algorithms* and *Metaheuristic Optimization* in the supplementary material.

For approach (1), following the formulated problem in Eq. (1), we now assume that the optimization algorithm is initiated by a set of m materials evaluations from MEGNet: $K_1, K_2, K_3, \dots, K_m, (E_{\text{Fermi}})_1, (E_{\text{Fermi}})_2, (E_{\text{Fermi}})_3, \dots, (E_{\text{Fermi}})_m = \text{MEGNet}(\Omega(\Theta_1), \Omega(\Theta_2), \dots, \Omega(\Theta_m))$; one then obtains data that lie in the mapping of $M : \Theta \rightarrow \mathcal{J}$, in which we assume that M exists and is continuous. Here, this M represents the design space, i.e., the projection from the input design variables to the objectives. M can be approximated using the Gaussian process regression (GPR) from the generated data in the materials evaluation, which can be simplified to the following form:

$$(\mathcal{J}_1, \mathcal{J}_2, \mathcal{J}_3, \dots, \mathcal{J}_m) = GP(\Theta_1, \Theta_2, \dots, \Theta_m; \mathbf{p}_{GP}), \quad (3)$$

where \mathbf{p}_{GP} are all the hyperparameters involved in setting up the regression model, elaborated in detail in the supplementary material. Given a surrogate model of the design space built from GPR, one can improve the accuracy of such a surrogate by iteratively sampling more points. The core of BO here would be the strategies to sample the next set of material evaluations from Θ .

The next set of design variables Θ_{next} is selected via maximizing the acquisition function \mathcal{A} ,

$$\Theta_{next} = \operatorname{argmax} \mathcal{A}(\Theta_1, \dots, \Theta_m, \mathcal{J}_1, \dots, \mathcal{J}_m; \mathbf{p}_{Aq}), \quad (4)$$

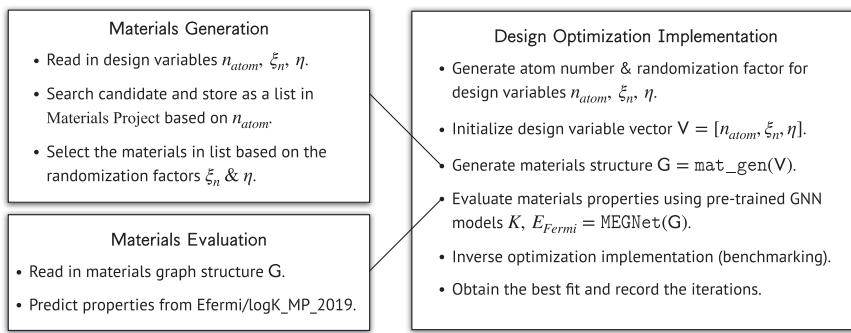


FIG. 3. Algorithmic flow chart for the optimization framework, corresponding to Eqs. (1) and (2) and Fig. 2.

where we use the upper confidence bound as the acquisition function (see *Bayesian Optimization* in the supplementary material). BO utilizes GPR and acquisition function to efficiently sample the design space for a more accurate approximation toward the formulated optimization goal ($\max \mathcal{J}$).

2. Metaheuristic optimization

Metaheuristic optimization is a family of optimization techniques that are used to find the optimal solution by exploring a large search space. They are often inspired by natural processes or social behavior, such as animal behavior (e.g., ant, whale, and wolf), physical process (i.e., annealing of metals), biological processes (i.e., gene mutation), or fundamental mathematics.

Metaheuristic algorithms are typically applied to complex optimization problems where traditional methods are either not feasible

or inefficient. They are highly flexible and can be adapted to a wide range of problem domains, in which elemental composition design is our target scenario. Metaheuristics are often considered as “black box” optimization methods, as they do not require knowledge of the problem structure. More specifically, they do not require the existence of the first-order differentiation form of the governing equations of the problem.

If we assume the simplest form, one can simplify and interpret the metaheuristic optimization process by sampling a set of random particles in the design space and the optimization process refers to the update of these sampled particles (for the generalization of this form, see *Extension on Problem Formulation* in the supplementary material). The update rules may be inspired by different natural processes, corresponding to the optimization methods employed. If one assumes that there are a total of d dimensions of the problem domain

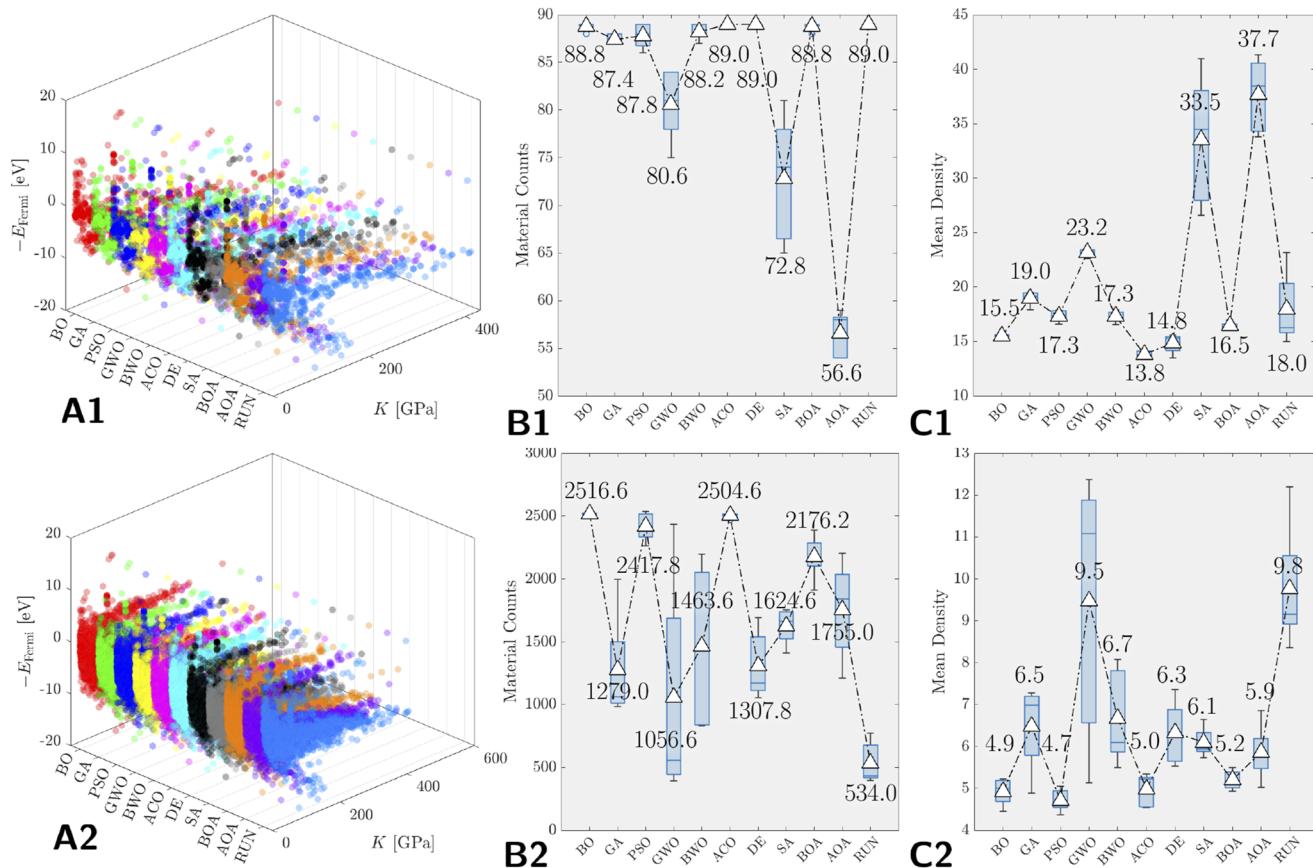


FIG. 4. Objective properties exploration, objective value evaluation, and targeted design space count for benchmarking different optimization methods for single- and multi-element composition design cases. (a1) Overall design space benchmarking for the 11 different optimization schemes used, marked in different colors for the single-element composition design (for the projections in 2D, please refer to Fig. S6 in the supplementary material). (b1) The distribution (for the five repeated experiments) and mean values of the overall material counts during the evaluation processes. (c1) The distribution (for the five repeated experiments) and mean values of the scanned material evaluation densities (see Sec. II and *Density Scan* in the supplementary material for the calculation of “mean density”) during the evaluation processes. (Related details can be found in the analysis of Fig. S7 in the supplementary material.) (a2) Overall design space benchmarking for the multi-element composition design (refer to Fig. S6 in the supplementary material). (b2) The distribution and mean values of the overall material counts. (c2) The distribution and mean values of the scanned material evaluation densities. The white triangular dots are the mean values for the 11 evaluations in both (b) and (c). The corresponding numbers are marked on top of the bar plot for both (b) and (c). BO, GA, PSO, GWO, HIWOA, ACO, DE, SA, BOA, AOA, and RUN denote the Bayesian optimization, genetic algorithm, particle swarm optimization, hybrid gray wolf optimization, hybrid-improved whale optimization algorithm, ant colony optimization, differential evolution, simulated annealing, bees optimization algorithm, arithmetic optimization algorithm, and Runge-Kutta optimization, respectively.

and denotes the rule as \mathcal{R} , the update rules for the i th particle write

$$X_i^d \leftarrow X_i^d + \mathcal{R}(X_i^d, X_j^d, \mathbf{p}_{MH}), \quad (5)$$

where X is the location of the particle and the i th particle's location may be related to its adjacent particle(s) j . \mathbf{p}_{MH} denotes the hyperparameters employed in the metaheuristic optimization. Here, we only present the generalized form to identify the key essence of

metaheuristics (for the details of each algorithm, see *Metaheuristic Optimization* in the supplementary material).

III. RESULTS

A. Elemental composition design benchmarks

Figure 4 shows the benchmarks of the 11 different inverse optimization algorithms (IOAs) and their objective properties explorations [column (a)], the distribution of the material evaluation

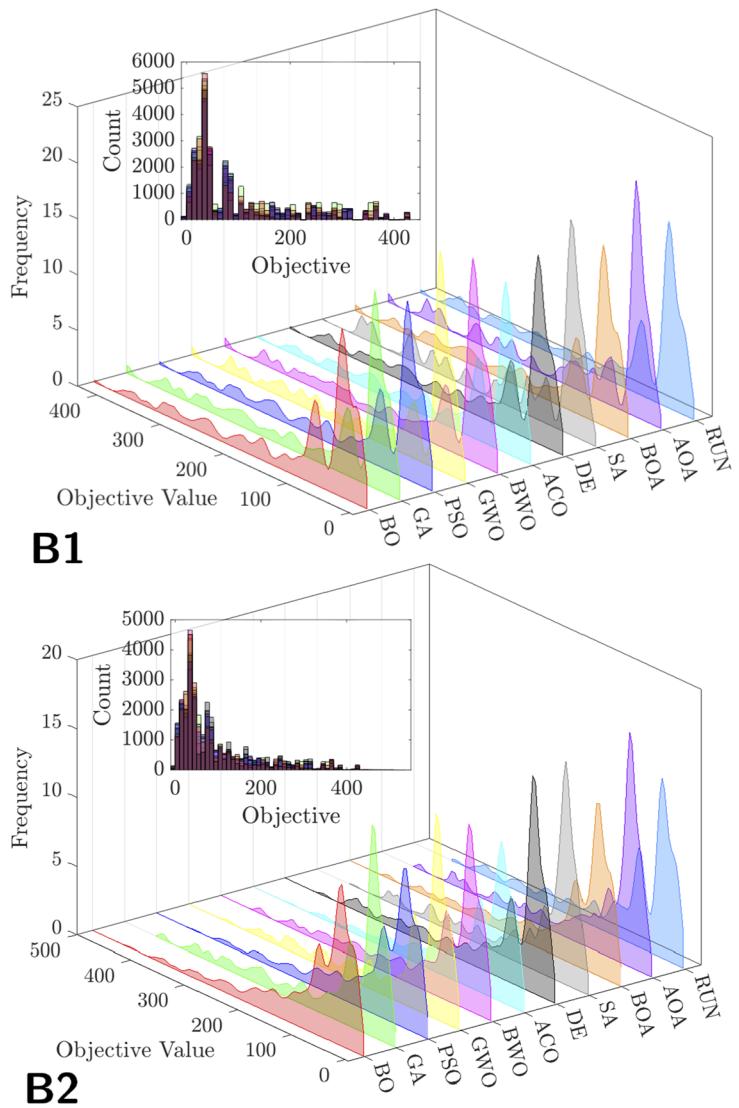
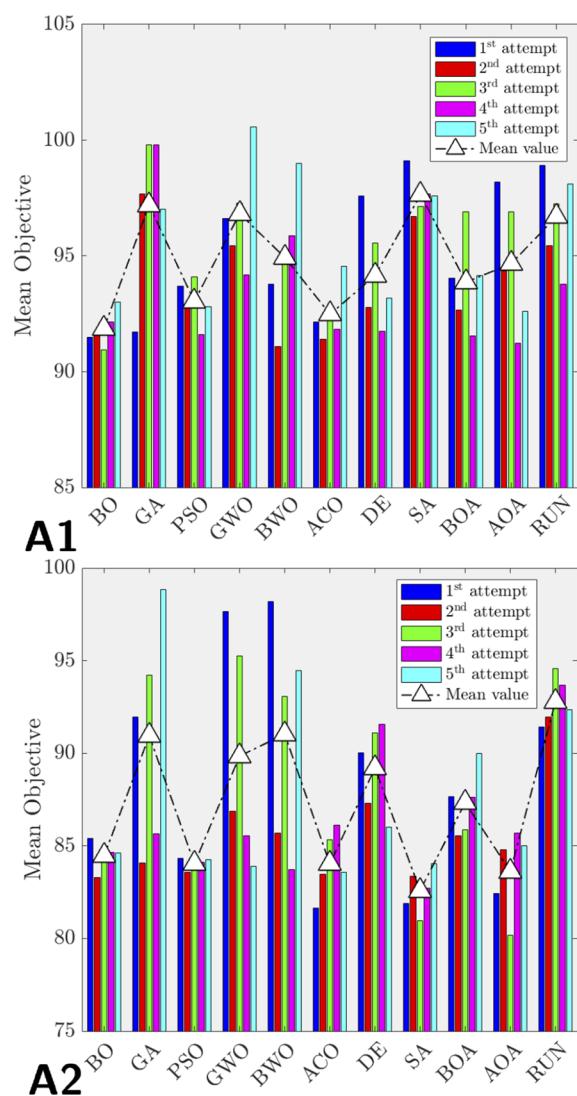


FIG. 5. Statistical evaluations of the optimization processes. (a1) The distribution of the mean objective values (from five attempts) for the 11 evaluated optimization algorithms for single-element composition cases. Different colors mark different attempts, as indicated in the legend in the top-right corner of this figure. The white triangular dots stand for the mean values of the five attempts. The “mean objectives” are calculated as $\frac{\sum_{\text{attempts}} (\text{objectives})}{\text{attempts}}$ per optimization attempt (under the same random seed). (b1) The frequency distributions (i.e., repeated evaluations) using kernel fits of different objective values for the 11 different optimization algorithms (for details, see Sec. II and Additional Results in the supplementary material). Note that the original data distribution is shown in the top left corner of the inset. (a2) The distribution of the mean objective values for the 11 evaluated optimization algorithms for multi-element composition cases, where the white triangular dots are the mean values. (b2) The frequency distributions of different objective values for the 11 optimization algorithms.

numbers with the corresponding mean values [column (b)], and the distributions of scanned density with the overall mean values [column (c)], for both the single- (row 1) and multi-element (row 2) composition design cases. In the objective distributions, different IOAs are marked in different colors, and the corresponding optimization methods are abbreviated. Here, the materials count represents the total number of unique elemental compositions scanned by the different IOAs. The mean densities of objective spaces are computed to estimate the repetitive evaluations of the same elemental compositions by computing the mean value of the density scan space (see Sec. II and the supplementary material). Both metrics help us understand the “strategy” of different algorithms when they scan the chemical design space to search for the ideal elemental compositions.

There are five major findings from Fig. 4. (1) Comparing (a1) and (a2), the scanned objective spaces are smaller for GWO compared with other optimization methods for both the single- and multi-element composition design cases. For the single-element composition design case: (2) the evaluated elemental composition numbers are similar for most methods, whereas SA and AOA are observably smaller. Both GWO and SA display relatively high variances compared with other methods. (3) The mean scanned densities are relatively higher for the SA and AOA methods, with both displaying higher variances. Both SA and AOA possess higher variances. For the multi-element composition design cases: (4) RUN has the lowest evaluated elemental composition numbers. GWO and BWO have the highest variances. (5) For the mean scanned densities for multi-element compositions, GWO and RUN have the highest evaluated elemental composition numbers and variances.

From Fig. 5, we can draw five key observations. (1) When comparing (a1) and (a2), the mean objective distributions differ for the single-element and multi-element composition design cases. (2) For single-element design, GA, SA, and RUN display high mean objective values. SA and RUN also exhibit relatively low variances.

GWO shows relatively high mean objectives but with higher variances, while BO demonstrates a stable performance with a low mean objective value and low variance. (3) DE and AOA exhibit repetitive scans in the same composition region, indicating a tendency to get “trapped” in a local space. This aligns with their relatively lower mean objectives in (a1). Moving on to multi-element design, RUN achieves the highest mean objective values with low variances. GA, GWO, and BWO show higher variances. (4) AOA and SA again exhibit repeated scanning of the same elemental compositions, suggesting a tendency for optimization methods to get trapped in local composition spaces. (5) When considering both single- and multi-element cases, different optimization algorithms demonstrate similar elemental composition search strategies, as evidenced by similar frequency distributions.

To further quantitatively support our observations on the uncertainties of different optimizations for repeated elemental composition design experiments in Figs. 4 and 5, Tables I and II show the variances among different experiments per different optimizations. Both Table I and Fig. 4 show that the materials count variances are very different for single- and multi-element compositions, due to the very limited single-element compositions existing in the chemistry table. For the multi-element composition design, BO has the lowest variance and GWO has the highest variance of materials count, with GWO’s variance \sim 7398 times higher than BO. Other than that, GWO’s variance is \sim 27 times higher than that of RUN of materials count. GA and AOA have approximately the same level of uncertainty for materials count. The relative magnitude of the variances of the materials count and the mean density scan does not follow the exact same trend. Yet for multi-element composition design, GWO still exhibits the highest variance, with 366.93% higher than that of BO and 9992.08% higher than that of BO. Generally, it can still be deduced from Table I that, for both single- and multi-element composition design, nature-inspired algorithms (i.e., GA, GWO, and BWO) display relatively higher variances, indicating higher uncertainties, which will be further discussed in Sec. IV. Table II further

TABLE I. Variances for the five repeated experiments of different optimization algorithms for single- and multi-element composition design cases, expressed with single-digit precision. Note that MC denotes “materials count” and MDS stands for “mean density scan.” The four columns correspond to Figs. 4(b) and 4(c). For better presentation, we rescale the variances of MDS 100 times larger (indicated in the form $\times 10^{-2}$).

Optimization	MC (single-element)	MDS (single-element)	MC (multi-element)	MDS (multi-element)
BO	0.2	6.5×10^{-2}	104.3	10.1×10^{-2}
GA	0.3	42.9×10^{-2}	179 704.5	101.7×10^{-2}
PSO	1.7	28.9×10^{-2}	12 575.7	7.1×10^{-2}
GWO	14.3	13.3×10^{-2}	771 728.3	1019.3×10^{-2}
BWO	0.7	26.0×10^{-2}	402 679.3	131.2×10^{-2}
ACO	0	9.7×10^{-2}	427.3	15.8×10^{-2}
DE	0	79.6×10^{-2}	73 394.7	56.8×10^{-2}
SA	45.2	3576.7×10^{-2}	19 843.8	12.5×10^{-2}
BOA	0.2	8.4×10^{-2}	29 907.2	5.2×10^{-2}
AOA	5.8	1156.6×10^{-2}	150 198.5	44.1×10^{-2}
RUN	0	1098.7×10^{-2}	27 895.5	218.3×10^{-2}

TABLE II. Mean objective values and variances for the five repeated experiments of different optimization algorithms for single- and multi-element composition design cases, expressed with single-digit precision. Note that MO denotes “mean objective” and Vr stands for “variances.” The four columns correspond to Fig. 5(a). For better presentation, we rescale the variances of MDS 100 times larger (indicated in the form $\times 10^{-2}$).

Optimization	MO (single-element)	Vr (single-element)	MO (multi-element)	Vr (multi-element)
BO	91.9	58.7×10^{-2}	84.5	57.5×10^{-2}
GA	97.2	1097.1×10^{-2}	91.0	3726.5×10^{-2}
PSO	93.0	90.7×10^{-2}	84.0	8.8×10^{-2}
GWO	96.8	580.6×10^{-2}	89.8	3816.3×10^{-2}
BWO	94.9	838.0×10^{-2}	91.0	3719.7×10^{-2}
ACO	92.5	148.3×10^{-2}	84.0	307.5×10^{-2}
DE	94.2	557.5×10^{-2}	89.2	593.5×10^{-2}
SA	97.6	81.5×10^{-2}	82.6	145.8×10^{-2}
BOA	93.9	404.5×10^{-2}	87.3	314.1×10^{-2}
AOA	94.7	840.2×10^{-2}	83.6	523.2×10^{-2}
RUN	96.7	429.3×10^{-2}	92.8	166.6×10^{-2}

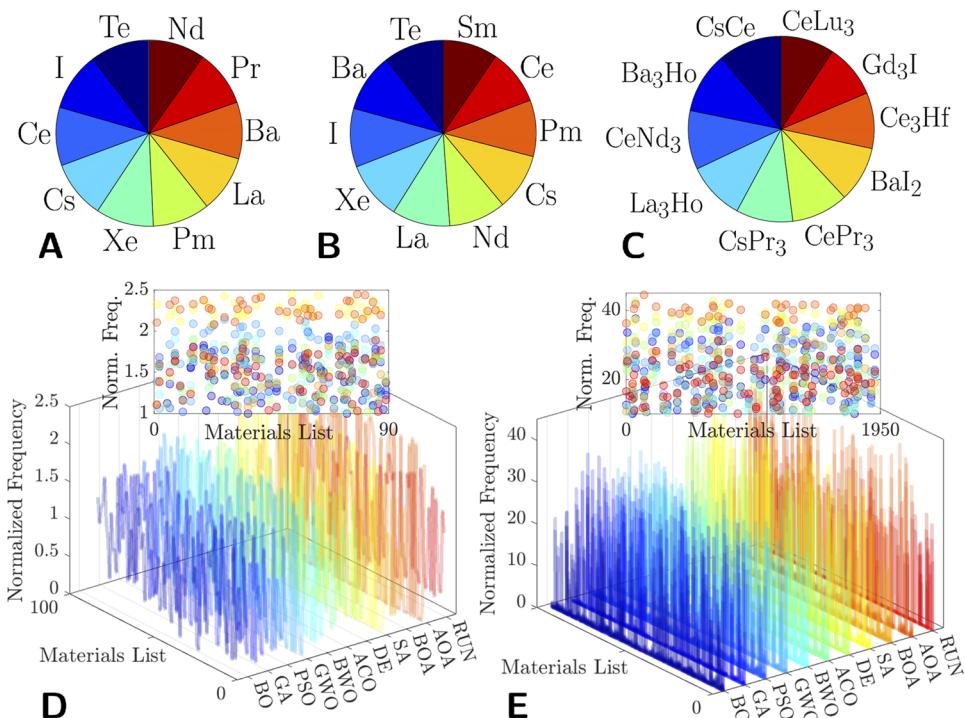


FIG. 6. Extracted elemental composition evaluations for the optimization benchmarking. (a) The top ten elemental compositions extracted from the single-element design case. (b) The top ten elemental compositions extracted from the multi-element composition design case. (c) The top ten chemical compounds extracted from the multi-element composition design case. (d) The normalized frequency (see Additional Results in the supplementary material for the extended analysis for frequency evaluations of optimizations) for different extracted elemental compositions under the single-element composition design case. The top inset dots stand for the projection of the normalized frequency in the range [1, 2.5]. (e) The normalized frequency under the multi-element composition design case. The top inset dots stand for the projection of the normalized frequency in the range [10, 45].

TABLE III. Assessment and conclusion from the comparison results for optimization methodologies benchmarked. Note that the (dis)advantages concluded from the previous results may be subjective. Note that the “low selectivity” for RUN indicates that the variance is relatively low compared with DE and AOA.

Optimization	Advantages	Disadvantages	Potential use
BO	Low variances	Low objective	Large-scale materials exploration
GA	High objectives	High variances	Targeted materials selection
PSO	Low variances	Low objective	Large-scale materials exploration
GWO	High objective	High variances	Not recommended
BWO	High objective	High variances	Not recommended
ACO	Low variances	Low objective	Large-scale materials exploration
DE	Decent objective	Relatively high variances	Not recommended
SA	High selectivity	Problem-specific	Not recommended
BOA	Decent objective	Low selectivity	Not recommended
AOA	High selectivity	High variance	Targeted materials selection
RUN	High objective and stable	Low selectivity	Materials exploration and design

confirms the observation that nature-inspired algorithms present higher uncertainties. For both single- and multi-element composition design cases, GA, GWO, and BWO show evidently higher variances of the mean objectives than those of other optimizations. For the single-element composition design, GA presents 155.56% and 1093.8% relatively higher variances, GWO presents 35.24% and 531.77% relatively higher variances and BWO presents 95.20% and 811.86% relatively higher variances than those of RUN and BO, respectively. For the multi-element composition design, GA presents 2136.79% and 6380.87% relatively higher variances, GWO presents 2190.70% and 6537.04% relatively higher variances, and BWO presents 2132.71% and 6369.04% relatively higher variances than those of RUN and BO, respectively.

From Fig. 6, we can derive five main observations. (1) Comparing (a) and (b), we observe that the top evaluated elemental compositions are similar. (2) The majority of the top chemical compounds are binary alloys, consisting of elements present in the top single-element compositions, such as Cs, Ce, Ba, and others. (3) RUN demonstrates effective scanning strategies and consistently converges toward local optimal elemental compositions. (4) SA demonstrates similar search strategies as RUN, evidenced by the high normalized frequency along the materials list. (5) BO appears to struggle in identifying the “ideal” elemental compositions and tends to scan the same region repeatedly, indicating a potential trap in the local optimal area. From Figs. 4–6, we have also added qualitative comparison and suggested potential applications in Table III. Since the overall goal is to design materials digitally with low uncertainty and targeted desired properties quantified through objectives, therefore, low variance, and high mean objectives can be viewed as “advantages” of the algorithm. “Targeted materials selection” is recommended due to repeated materials sampling, and “large-scale exploration” is recommended based on more materials candidates sampled with decent objective values.

Among all the metrics we evaluated, it can be deduced that RUN achieved superior optimization results compared with the

others (Figs. 4 and 5; Tables I and II). This out-performance may be due to several reasons: (1) RUN is free of hyperparameters. The update process of RUN does not depend on fine-tuned hyperparameters, making it more robust especially for highly nondeterministic situations like elemental composition design. (2) The updating strategies of RUN are very different from those of other methods. The RUN optimization searches for the next set of evaluations by mimicking the update strategies of Runge–Kutta’s discretization of differential equations (see the supplementary material), which are fundamentally different from other metaheuristic methods. (3) From (2), the enhanced solution quality (ESQ) mechanism is well suited for our formulated elemental composition discovery cases. In the authors’ original proposal,³⁸ the ESQ mechanism is employed to avoid the local optimal solutions and increase the convergence speed. In our case, as the material evaluation numbers are fixed at 5000, the faster convergence induced by ESQ enables the robust performance of the RUN algorithm in our benchmark cases. Further comprehensive characteristics are discussed in Sec. IV.

IV. DISCUSSION

Based on the results presented earlier, our findings raise several intriguing questions, for which we proffer the following discussions. First, we observe variations in the performance of different optimization algorithms across the two elemental composition design scenarios, such as the contrasting results of material counts and the mean density of RUN and AOA [Figs. 4(b) and 4(c)]. One explanation is that the algorithm effectiveness is problem-specific. Moreover, the optimization methods may also be sensitive to the dimensionality of the problem such that changing from single-element to multi-element composition design induces the change of metrics evaluation shown in Figs. 4 and 5. In addition, in the case of single-element composition design, the limited number of available elemental compositions can impact the performance of certain algorithms, potentially influencing their search capabilities. An

interesting observation pertains to the consistent success of the RUN algorithm in achieving high mean objective values in both problem scenarios, especially in the second case [Fig. 5(a)]. This achievement can be attributed to the fact that the sampling strategy employed by RUN is hyperparameter-free and adept at storing information, which proves advantageous in handling high-dimensional spaces.

Another intriguing finding is that BO exhibits lower mean objectives accompanied by low variances in both cases. This trend becomes more apparent when considering Figs. 5(b) and 6(e). A plausible explanation would be that BO is sampling a wider range of diverse elemental compositions during the optimization process. This behavior can be attributed to the uncertainty-based sampling strategy employed by BO, which facilitates the exploration of a diverse design space and aids in constructing a more accurate surrogate model. AOA tends to repeatedly sample similar elemental compositions over an extensive number of evaluations [Fig. 5(b)]. This behavior suggests that AOA may become trapped within a specific region of the design space, necessitating further investigation to understand the underlying causes and identify potential remedies. Finally, we explore the major disparity in the performance of SA between the two elemental composition discovery cases. A possible explanation for this discrepancy is that SA is sensitive to the dimension of the optimization problem, leading to its different performances tested from the designed metrics in Figs. 4 and 5.

Each optimization algorithm demonstrates unique characteristics and strategies that contribute to its specific outcomes. The low variances of the objectives in Fig. 5 could be explained by BO's low dependence on hyperparameters. BO aims to approximate an accurate surrogate model to optimize the objective, resulting in small mean objective values. This strategy utilizes uncertainty-based sampling, which decreases the overall objective value. The low uncertainty and variance in the final results are attributed to the small number of hyperparameters and the large sampling base. In addition, BO exhibits high material counts, as observed in Fig. 4(b). GA heavily relies on hyperparameters, such as mutation, offspring generation, and randomization (for details, see the supplementary material). A strong dependence on hyperparameters leads to high uncertainty in the mean objective values for both cases [Figs. 5(a1) and 5(a2)]. Moreover, GA repeated the samples of the same elemental compositions once ideal properties are detected, resulting in high sampling density compared with BO, PSO, and other related methods [Figs. 5(b1) and (b2)]. Meanwhile, PSO exhibits a lower uncertainty and variance as it has fewer hyperparameters, contributing to its more stable performance, as depicted in Figs. 5(a1) and (a2).

GWO is similar to GA in terms of hyperparameter dependency, resulting in a high uncertainty and sampling density focused on specific objective regions (Fig. 5). Notably, GWO displays a higher uncertainty even in material counts and mean density, which differentiates it from GA (Fig. 4). Similarly, BWO exhibits a high uncertainty in mean objective values [Fig. 5(a)], which may also be attributed to its dependence on hyperparameters and very similar searching strategies of mimicking animal dynamical behavior. However, BWO displays more material counts and less sampling density scan [Figs. 4(b) and 4(c)], potentially attributed to the setting of the encircling prey parameter, which enhances the sampling stability.

ACO employs continuous probability-based sampling, resulting in relatively less uncertainty in the mean objective values [Fig. 5(a)]. ACO has similar mean objective values as BO and PSO while exhibiting lower sampling density scans [Fig. 4(c1)], indicating that ACO avoids oversampling the same regime for the single-element composition design case. DE algorithmically operates as a variant of GA, yielding comparable mean objective results [Fig. 5(a)]. With fewer hyperparameters compared to GA, DE exhibits a lower uncertainty [Fig. 5(a)]. The evolutionary strategies of DE demonstrate similar material counts to GA, as shown in Fig. 4, and repeated sampling for favorable materials, as depicted in Fig. 6. The performance of SA is observed to be more problem-specific, as both the material counts and density scan vary per different design cases [Figs. 4(b), 4(c), and a5(a)]. SA exhibits a low uncertainty for the mean objective values [Fig. 5(a)] due to fewer hyperparameters and the implementation of a Markov chain for the acceptance-rejection sampling criteria (for details, see the supplementary material). BOA shares similarities with GWO and BWO in terms of high uncertainty in the mean objectives [Fig. 5(a)] and low material counts for the multi-element composition design [Fig. 4(b2)], as they use particle-based, animal behavior-inspired search criteria (for details, see the supplementary material). However, BOA exhibits a relatively lower variance [Fig. 5(a)], indicating a lower dependence on, and having fewer, hyperparameters.

AOA uses a distinct searching strategy based on mathematical symbolized operations. AOA exhibits a high uncertainty with relatively low mean objectives [Fig. 5(a)]. AOA tends to oversample the local regime, leading to low material counts [Fig. 5(b)]. This behavior can be attributed to a large number of hyperparameters and the overdependence on certain factors, such as MOA and MOP (for details, see *Arithmetic Optimization Algorithm* in the supplementary material), resulting in a scaling factor that underestimates previous locations. Finally, RUN achieves high objectives with a low uncertainty and avoids oversampling in specific objective regions (Fig. 5). The material count is sensitive to dimensionality, yet all dimensions yield successful search results, as judged by the high overall mean objectives [Fig. 5(a)]. RUN is characterized by being hyperparameter-free, incorporating numerous intermediate hyperparameters to increase the dimensionality of the iteration scheme, and relying on previous solutions for updates based on search history. Following our previous discussion on RUN's ESQ search mechanism and being hyperparameter-free, we would like to examine why RUN had outperformed other algorithms by having high objective values and low uncertainties. RUN employs an update scheme that has four update parameters,⁴⁴ each containing randomization coefficients⁴⁵ and are connected. This randomization and "parameter-connection" could help explain the low uncertainty observed in our numerical experiments. For the high objective, RUN's ability to avoid local optima as reported by Ahmadianfar *et al.*³⁸ could be a possible explanation, in which they contend that the adaptive mechanism employed to update the parameter and the ESQ are the main mechanisms that assure a good transition from exploration to exploitation.

V. CONCLUSION

In this paper, we propose a framework that can incorporate most state-of-the-art optimization algorithms to perform

multi-elemental composition discovery and design for crystals. Our framework parameterizes the input elemental graph structure to tailor the corresponding bulk modulus K and Fermi energy E_{Fermi} . The key goal is to conduct a comprehensive benchmark for many of the state-of-the-art optimization methods and provide further insights for future implementation of these algorithms for elemental composition design. We optimized both single- and multi-element composition discovery cases for maximized K and minimized E_{Fermi} by formulating a single objective optimization. We benchmarked 11 different inverse optimization algorithms by fixing the material evaluations. We found that GA, GWO, and BWO exhibit higher variances and BO and RUN display generally lower variances for materials count, density scan (Table I), and mean objectives (Table II). We further conclude that nature-inspired algorithms contain more uncertainties in elemental design cases, which can be attributed to the dependency on hyperparameters. RUN exhibits higher mean objectives, whereas BO displayed low mean objectives compared with other optimization methods. Combined with materials count and density scan, it is proposed that BO samples more elemental compositions aiming to approximate a more accurate surrogate of the design space and, hence, has lower mean objectives, yet RUN will repeatedly sample the discovered optimal elemental compositions for the successful search strategy [Figs. 5(b), 6(d), and 6(e)]. We also proffer detailed discussions on the exhibited results that correspond to each optimization's characteristics.

SUPPLEMENTARY MATERIAL

See the supplementary material for the extension for our problem formulation for detailed explanations. We provide further details on hyperparameters and basic derivation for the 12 optimization algorithms employed (the 11 employed optimizations for repeated experiments with deep reinforcement learning). We proffer some additional results in supplementary to our proposition in the main article.

ACKNOWLEDGMENTS

J.Y. acknowledges the support from the U.S. National Science Foundation under Award Nos. CMMI-2038057, ITE-2236190, and EFMA-2223785 and the Cornell faculty startup grant. The authors also acknowledge the computational resources provided by the NSF Advanced Cyberinfrastructure Coordination Ecosystem: Services and Support (ACCESS) program under Grant No. BIO210063 and the computational resources provided by the G2 cluster from Cornell University.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

H.Z. and H.H. conceived the research. H.Z. designed and conducted the experiment(s). H.Z., H.H., and J.Y. analyzed the results.

J.Y. acquired the funding and resources. All authors reviewed the manuscript.

Hanfeng Zhai: Conceptualization (lead); Data curation (lead); Formal analysis (lead); Investigation (lead); Methodology (lead); Software (lead); Validation (lead); Visualization (lead); Writing – original draft (lead); Writing – review & editing (equal). **Hongxia Hao:** Conceptualization (equal); Investigation (supporting); Methodology (supporting); Supervision (equal); Writing – review & editing (equal). **Jingjie Yeo:** Conceptualization (supporting); Funding acquisition (lead); Methodology (supporting); Project administration (equal); Resources (lead); Supervision (equal); Writing – review & editing (equal).

DATA AVAILABILITY

The data that support the findings of this study are available at https://github.com/hanfengzha/materials-discovery-benchmark/tree/master/inverse_opt.

REFERENCES

- ¹E. Andreassen, A. Clausen, M. Schevenels, B. S. Lazarov, and O. Sigmund, "Efficient topology optimization in MATLAB using 88 lines of code," *Struct. Multidiscip. Optim.* **43**, 1–16 (2010).
- ²M. P. Bendsøe and N. Kikuchi, "Generating optimal topologies in structural design using a homogenization method," *Comput. Methods Appl. Mech. Eng.* **71**, 197–224 (1988).
- ³H. Zhai and J. Yeo, "Multiscale mechanics of thermal gradient coupled graphene fracture: A molecular dynamics study," *Int. J. Appl. Mech.* **15**, 2350044 (2023).
- ⁴T. Ng, J. Yeo, and Z. Liu, "A molecular dynamics study of the thermal conductivity of graphene nanoribbons containing dispersed Stone–Thrower–Wales defects," *Carbon* **50**, 4887–4893 (2012).
- ⁵E. T. Ritz and N. A. Benedek, "Interplay between phonons and anisotropic elasticity drives negative thermal expansion in PbTiO₃," *Phys. Rev. Lett.* **121**, 255901 (2018).
- ⁶M. J. Buehler, "Modeling atomistic dynamic fracture mechanisms using a progressive transformer diffusion model," *J. Appl. Mech.* **89**, 121009 (2022).
- ⁷A. Mannodi-Kanakkithodi, G. Pilania, T. D. Huan, T. Lookman, and R. Ramprasad, "Machine learning strategy for accelerated design of polymer dielectrics," *Sci. Rep.* **6**, 20952 (2016).
- ⁸C. Kim, R. Batra, L. Chen, H. Tran, and R. Ramprasad, "Polymer design using genetic algorithm and machine learning," *Comput. Mater. Sci.* **186**, 110067 (2021).
- ⁹R. Winter, F. Montanari, A. Steffen, H. Briem, F. Noé, and D.-A. Clevert, "Efficient multi-objective molecular optimization in a continuous latent space," *Chem. Sci.* **10**, 8016–8024 (2019).
- ¹⁰M. Weiel, M. Götz, A. Klein, D. Coquelin, R. Floca, and A. Schug, "Dynamic particle swarm optimization of biomolecular simulation parameters with flexible objective functions," *Nat. Mach. Intell.* **3**, 727–734 (2021).
- ¹¹D. Shin, A. Cupertino, M. H. J. de Jong, P. G. Steeneken, M. A. Bessa, and R. A. Norte, "Spiderweb nanomechanical resonators via Bayesian optimization: Inspired by nature and guided by machine learning," *Adv. Mater.* **34**, 2106248 (2021).
- ¹²Y. Zhang, S. Tao, W. Chen, and D. W. Apley, "A latent variable approach to Gaussian process modeling with qualitative and quantitative factors," *Technometrics* **62**, 291–302 (2019).
- ¹³U. F. Ghuman, A. van Beek, J. Munshi, T. Chien, G. Balasubramanian, and W. Chen, "Designing active layer of organic solar cells using multi-fidelity molecular simulations and spectral density function," *Comput. Mater. Sci.* **211**, 111491 (2022).

- ¹⁴Y. Comlek, T. D. Pham, R. Snurr, and W. Chen, “Rapid design of top-performing metal-organic frameworks with qualitative representations of building blocks,” *npj Comput. Mater.* **9**, 170 (2023).
- ¹⁵H. Zhang, W. Chen, A. Iyer, D. W. Apley, and W. Chen, “Uncertainty-aware mixed-variable machine learning for materials design,” *Sci. Rep.* **12**, 19760 (2022).
- ¹⁶A. Iyer, Y. Zhang, A. Prasad, P. Gupta, S. Tao, Y. Wang, P. Prabhune, L. S. Schadler, L. C. Brinson, and W. Chen, “Data centric nanocomposites design via mixed-variable Bayesian optimization,” *Mol. Syst. Des. Eng.* **5**, 1376–1390 (2020).
- ¹⁷H. Zhai and J. Yeo, “Computational design of antimicrobial active surfaces via automated Bayesian optimization,” *ACS Biomater. Sci. Eng.* **9**, 269–279 (2022).
- ¹⁸B. Zheng, Z. Zheng, and G. X. Gu, “Designing mechanically tough graphene oxide materials using deep reinforcement learning,” *npj Comput. Mater.* **8**, 225 (2022).
- ¹⁹F. Sui, R. Guo, Z. Zhang, G. X. Gu, and L. Lin, “Deep reinforcement learning for digital materials design,” *ACS Mater. Lett.* **3**, 1433–1439 (2021).
- ²⁰Y. Wang, Z. Cao, and A. Barati Farimani, “Efficient water desalination with graphene nanopores obtained using artificial intelligence,” *npj 2D Mater. Appl.* **5**, 66 (2021).
- ²¹N. De Cao and T. Kipf, “MolGAN: An implicit generative model for small molecular graphs,” *arXiv:1805.11973* (2018).
- ²²S. P. Ong, W. D. Richards, A. Jain, G. Hautier, M. Kocher, S. Cholia, D. Gunter, V. L. Chevrier, K. A. Persson, and G. Ceder, “Python materials genomics (pymatgen): A robust, open-source Python library for materials analysis,” *Comput. Mater. Sci.* **68**, 314–319 (2013).
- ²³A. Jain, S. P. Ong, G. Hautier, W. Chen, W. D. Richards, S. Dacek, S. Cholia, D. Gunter, D. Skinner, G. Ceder, and K. A. Persson, “Commentary: The materials project: A materials genome approach to accelerating materials innovation,” *APL Mater.* **1**, 011002 (2013).
- ²⁴L. Ward, A. Dunn, A. Faghaninia, N. E. Zimmermann, S. Bajaj, Q. Wang, J. Montoya, J. Chen, K. Bystrom, M. Dylla, K. Chard, M. Asta, K. A. Persson, G. J. Snyder, J. Foster, and A. Jain, “Matminer: An open source toolkit for materials data mining,” *Comput. Mater. Sci.* **152**, 60–69 (2018).
- ²⁵N. V. Thieu and S. Mirjalili, “MEALPY: A framework of the state-of-the-art meta-heuristic algorithms in Python,” *J. Syst. Archit.* **139**, 102871 (2022).
- ²⁶C. Chen, W. Ye, Y. Zuo, C. Zheng, and S. P. Ong, “Graph networks as a universal machine learning framework for molecules and crystals,” *Chem. Mater.* **31**, 3564–3572 (2019).
- ²⁷P. I. Frazier, “A tutorial on Bayesian optimization,” *Recent Advances in Optimization and Modeling of Contemporary Problems* **2018**(October), 255–278 (2018).
- ²⁸R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction* (MIT Press, 2018).
- ²⁹J. Carr, *An Introduction to Genetic Algorithms* (Whitman College Department of Mathematics, 2014).
- ³⁰J. Liang, A. Qin, P. Suganthan, and S. Baskar, “Comprehensive learning particle swarm optimizer for global optimization of multimodal functions,” *IEEE Trans. Evol. Comput.* **10**, 281–295 (2006).
- ³¹O. O. Obadina, M. A. Thaha, K. Althoefer, and M. H. Shaheed, “Dynamic characterization of a master-slave robotic manipulator using a hybrid grey wolf-whale optimization algorithm,” *J. Vib. Control* **28**, 1992–2003 (2021).
- ³²C. Tang, W. Sun, W. Wu, and M. Xue, “A hybrid improved whale optimization algorithm,” in *2019 IEEE 15th International Conference on Control and Automation (ICCA)* (IEEE, 2019).
- ³³K. Socha and M. Dorigo, “Ant colony optimization for continuous domains,” *Eur. J. Oper. Res.* **185**, 1155–1173 (2008).
- ³⁴A. W. Mohamed, A. A. Hadi, and K. M. Jambi, “Novel mutation strategy for enhancing SHADE and LSHADE algorithms for global numerical optimization,” *Swarm Evol. Comput.* **50**, 100455 (2019).
- ³⁵D. Bertsimas and J. N. Tsitsiklis, “Simulated annealing,” *Stat. Sci.* **8**, 10–15 (1993).
- ³⁶D. Karaboga and B. Basturk, “A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm,” *J. Global Optim.* **39**, 459–471 (2007).
- ³⁷L. Abualigah, A. Diabat, S. Mirjalili, M. Abd Elaziz, and A. H. Gandomi, “The arithmetic optimization algorithm,” *Comput. Methods Appl. Mech. Eng.* **376**, 113609 (2021).
- ³⁸I. Ahmadianfar, A. A. Heidari, A. H. Gandomi, X. Chu, and H. Chen, “RUN beyond the metaphor: An efficient optimization algorithm based on Runge Kutta method,” *Expert Syst. Appl.* **181**, 115079 (2021).
- ³⁹P.-P. De Breuck, G. Hautier, and G.-M. Rignanese, “Materials property prediction for limited datasets enabled by feature selection and joint learning with MODNet,” *npj Comput. Mater.* **7**, 83 (2021).
- ⁴⁰S. Kong, D. Guevarra, C. P. Gomes, and J. M. Gregoire, “Materials representation and transfer learning for multi-property prediction,” *Appl. Phys. Rev.* **8**, 021409 (2021).
- ⁴¹J. Schmidt, L. Pettersson, C. Verdozzi, S. Botti, and M. A. L. Marques, “Crystal graph attention networks for the prediction of stable materials,” *Sci. Adv.* **7**, eabi7948 (2021).
- ⁴²V. Fung, J. Zhang, E. Juarez, and B. G. Sumpter, “Benchmarking graph neural networks for materials chemistry,” *npj Comput. Mater.* **7**, 84 (2021).
- ⁴³A. Dunn, Q. Wang, A. Ganose, D. Dopp, and A. Jain, “Benchmarking materials property prediction methods: The Matbench test set and Automatminer reference algorithm,” *npj Comput. Mater.* **6**, 138 (2020).
- ⁴⁴ k_1, k_2, k_3, k_4 in Ref. 38.
- ⁴⁵ $rand_1$ and $rand_2$ in Ref. 38.