Review

From Grid Search to Modern Deterministic Ideas: A Brief Review of Hyper-Parameter Optimisation in Deep and Foundation Model Era

Mehdi Neshat *

Posted Date: 30 March 2026

doi: 10.20944/preprints202603.2300.v1

Keywords: deterministic hyper-parameters optimisation; grid search; hyper-gradient; surrogate; multi-fidelity methods

 doi:10.20944/preprints202603.2300.v1

Review

From Grid Search to Modern Deterministic Ideas: A Brief Review of Hyper-Parameter Optimisation in Deep and Foundation Model Era

Mehdi Neshat

Faculty of Engineering and Information Technology, University of Technology Sydney, Sydney, Australia; mehdi.neshat@uts.edu.au

Abstract

As deep and machine learning systems grow in scale and are deployed in sensitive, high-stakes environments, the need for reliable and fully reproducible model tuning has never been greater. This paper presents a clear and structured overview of Deterministic Hyper-parameter Optimisation (DHPO), covering four key families of methods: Direct Search, Surrogate-based, Hyper-gradient, and Multi-Fidelity approaches. We frame DHPO within a reproducible bilevel optimisation setting and discuss how each method performs in terms of efficiency, scalability, and practical applicability. Our analysis shows that Hyper-gradient and Multi-Fidelity techniques generally provide the best balance of speed and scalability for deep learning, while Surrogate-based methods are strong options when compute resources are limited. Direct Search remains appealing for its simplicity and guaranteed repeatability, but faces challenges in high-dimensional or expensive training scenarios. Using visual comparisons—including heatmaps, radar profiles, and cost–performance plots—we show that no single DHPO approach is universally superior, and that method selection should depend on task constraints and reproducibility needs. Reported results in the literature indicate that hyper-gradient and multi-fidelity DHPO methods can reduce training costs by 40–70% in deep-learning settings while achieving near-baseline performance with an accuracy deviation of 1–3%. Finally, we conclude by outlining the key gaps and future research opportunities, including scalable DHPO for foundation models, hybrid deterministic–stochastic designs, differentiable architecture and data optimisation, and the path toward fully deterministic AutoML.

Keywords: deterministic hyper-parameters optimisation; grid search; hyper-gradient; surrogate; multi-fidelity methods

1. Introduction

Hyper-parameter optimisation (HPO) is crucial for creating effective and reliable machine learning (ML) and deep learning (DL) systems. Even with rapid progress in model designs, from convolutional networks to transformers and large-scale foundation models, choosing the best hyper-parameters, such as learning rates, regularisation strengths, network depth, and training schedules, is still costly and complex [1]. HPO is typically seen as a black-box optimisation problem. The objective function can be expensive to assess, non-convex, and often does not have simple gradients [2]. In addition, the evaluation is often affected by the randomness of the initial conditions, data sampling, and hardware variability. This uncertainty makes systematic search strategies important to ensure consistent and strong performance [3]. In safety-critical tasks like medical diagnosis, stochastic HPO can produce non-repeatable outcomes; for example, two random-search runs on identical cardiovascular data showed 4–6% sensitivity variation due solely to random initialisation—unacceptable in regulated clinical settings [4].

Deterministic hyper-parameter optimisation (DHPO) methods are a unique group of strategies that avoid random sampling and create consistent search paths. Determinism is crucial in safety-critical

 doi:10.20944/preprints202603.2300.v1

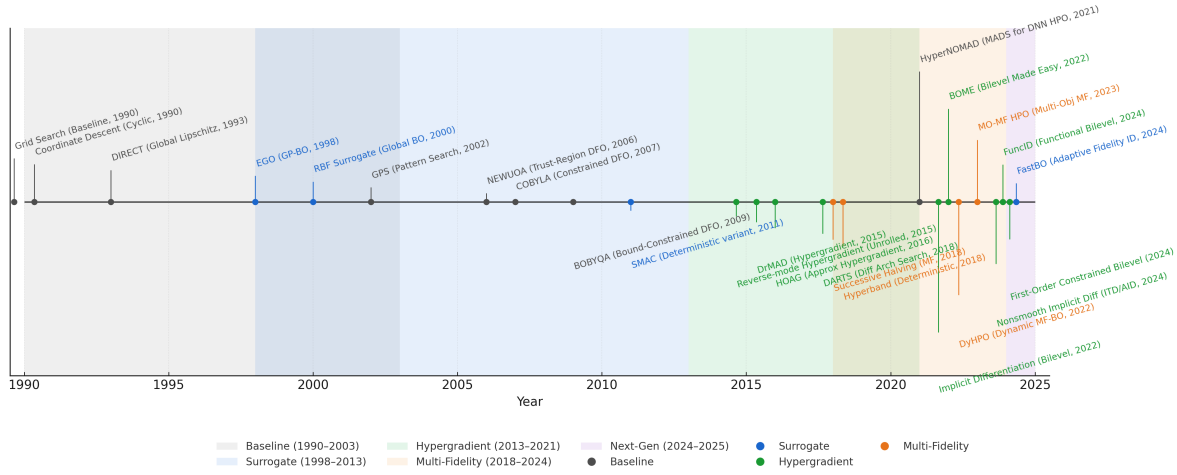
fields that are tightly regulated or scientific, like healthcare analytics, autonomous systems, finance, and engineering design. In these areas, the clarity, auditability, and repeatability of the optimisation process are essential [5,6]. Among DHPO methods, grid search (GS) is still popular because it is simple, thorough, and easy to run in parallel. By breaking the space into a Cartesian product of predefined hyper-parameter sets, GS allows for a systematic evaluation and full reproducibility of results. However, this thorough approach results in exponential scaling with increased dimensions and high computational costs, making GS less efficient for high-dimensional or resource-heavy deep learning tasks [5,7].

To address these limitations, several straightforward improvements to GS have been suggested. Multi-resolution and coarse-to-fine schedules cut down unnecessary evaluations at less promising granularities. Space-filling sampling designs, like Latin Hypercube Sampling (LHS), enhance coverage and reduce discretisation bias [8]. Budget-aware early-stopping strategies, such as Successive Halving [5] and Hyper-band [6], cut down computation by directing more resources to promising configurations. Additionally, hybrid deterministic-surrogate methods, such as Efficient Global Optimisation (EGO) [1] and deterministic Radial Basis Function (RBF) surrogates [9], maintain reproducibility by using fixed initial designs and clear acquisition and selection rules during the surrogate-guided search. These advancements create more scalable options for DHPO without losing determinism. Beyond GS, the broader DHPO range includes direct-search techniques, such as Hooke-Jeeves pattern search [10] and MADS [11], along with derivative-free local optimisation, like Nelder-Mead [12] and BOBYQA [13]. Each offers different trade-offs in how quickly they converge, how they handle constraints, and how they suit non-smooth objectives. At the same time, hyper-gradient-based methods have connected HPO with differentiable optimisation through implicit differentiation and bi-level formulations. This allows efficient optimisation of large hyper-parameter spaces with limited evaluation budgets [14–16]. Multi-fidelity deterministic HPO [17] evaluates configurations at different training budgets or data subsets, which proves to be effective for expensive DL pipelines. Recent studies, like DyHPO, show significant efficiency gains in practice [18]. On a larger scale, bandit-style schedulers can work with deterministic surrogates or pruning heuristics to provide a strong, cost-aware search under strict compute limits [6,7,11,19]. Figure 1 shows the development of HPOs from 1990 to 2025 across four main approaches: Direct Search, Surrogate-based, Hyper-gradient, and Multi-Fidelity. Essential methods are placed along their research lines, with shaded areas indicating the main development phases. Both visuals emphasise the recent “Next-Gen Hub” (2024-2025), where hybrid and convergent methods are appearing. This signals a move toward integrated, scalable deterministic HPO frameworks.

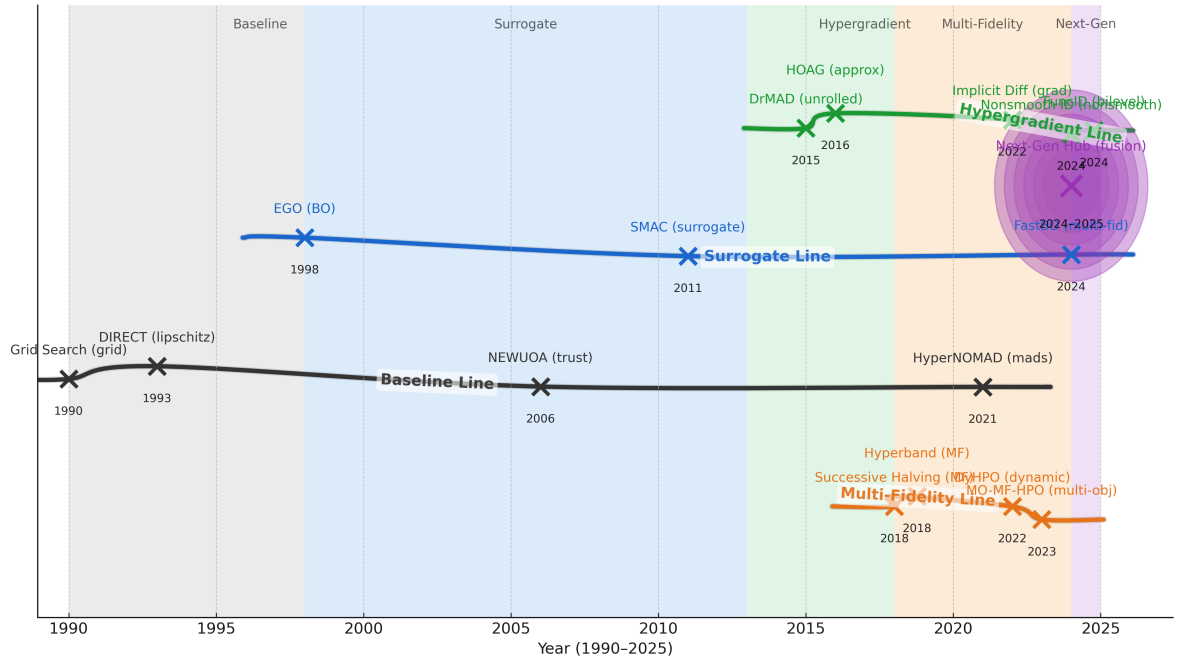
Given the growing use of ML models in critical settings, there is a clear need for a structured and technically sound review of DHPO techniques. This paper meets this need by examining GS as a baseline. It looks at its theoretical properties, limitations, and practical deployment considerations. It also places GS within the changing landscape of deterministic HPO. The goal is to provide researchers and practitioners with a better understanding of when and how DHPO can be effectively used in DL. Additionally, this paper highlights promising directions for cost-effective, scalable, and reproducible hyper-parameter search.

In this work, deterministic refers to algorithmic determinism, fixed optimisation trajectories given identical initial conditions, whereas reproducibility refers to end-to-end result consistency across runs and hardware environments. We note that GPU nondeterminism and floating-point variability can affect reproducibility even when the optimiser is deterministic.

 doi:10.20944/preprints202603.2300.v1



(**a**)



(**b**)

Figure 1. Evolution timeline of deterministic hyper-parameter optimisation (HPO) methods from 1990 to 2025.

2. Problem Formulation

Let $\boldsymbol{\theta} \in \Theta$ denote a vector of d hyperparameters, where the search space is defined as

$$\Theta = \mathbb{R}^{d_c} \times \mathcal{C}^{d_m} \times \mathbb{Z}^{d_i}, \tag{1}$$

comprising continuous, categorical, and integer-valued components.

Given a training set $\mathcal{D}_{\text{train}}$, a learning algorithm $\mathcal{A}_{\boldsymbol{\theta}}$ trains a model with parameters $\mathbf{w}$ by solving the inner optimization

$$\mathbf{w}^*(\boldsymbol{\theta}) = \arg\min_{\mathbf{w}} \mathcal{J}(\mathbf{w}, \boldsymbol{\theta}; \mathcal{D}_{\text{train}}), \tag{2}$$

where $\mathcal{J}(\cdot)$ denotes the training loss.

The goal of HPO is to identify an optimal hyperparameter configuration $\boldsymbol{\theta}^*$ that minimises the validation loss on $\mathcal{D}_{\text{val}}$:

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta} \in \Theta} f(\boldsymbol{\theta}), \quad \text{where} \quad f(\boldsymbol{\theta}) = \mathcal{L}(f_{\boldsymbol{\theta}}, \mathcal{D}_{\text{val}}). \tag{3}$$

 doi:10.20944/preprints202603.2300.v1

A deterministic HPO algorithm $\mathcal{H}$ generates a sequence of evaluated configurations such that

$$\boldsymbol{\theta}^{(t+1)} = \mathcal{H}\Big(\boldsymbol{\theta}^{(1:t)},\ f\big(\boldsymbol{\theta}^{(1:t)}\big)\Big), \tag{4}$$

ensuring reproducible optimization trajectories for identical initial conditions.

If a computational budget is imposed, the search is constrained as

$$\mathrm{Cost}(\boldsymbol{\theta}) \leq B, \tag{5}$$

where B represents a fixed resource limit (e.g., runtime, FLOPs, or GPU hours).

3. Review of Deterministic HPO Methods

Deterministic hyperparameter optimisation (DHPO) methods are becoming more important because they are reproducible, easy to understand, and suitable for high-stakes machine learning applications where random search behaviour is not acceptable or does not meet audit requirements. Unlike random or probabilistic optimisation strategies, DHPO provides consistent search paths. This consistency allows for clear model tuning and trustworthy performance attribution, which is crucial in areas like healthcare, finance, autonomous systems, and scientific computing [1–3]. Additionally, deterministic search avoids the variability and instability [4] that often come with random hyperparameter sampling. This makes it a better choice when computing resources are limited or when regulations require results to be repeated. Foundational work in deterministically guided model selection are as follows.

3.1. Direct Search HPO Methods

Direct Search (DS) methods are among the first deterministic ways to optimise hyper-parameters. They work without using gradient information and depend entirely on function evaluations. Based on the HPO objective in (Eq. 3), these methods update the hyperparameter vector $\boldsymbol{\theta}^{(t)}$ by exploring a neighbourhood or search pattern around the current point as follows:

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} + \Delta^{(t)}, \tag{6}$$

where the search direction $\Delta^{(t)}$ is generated based on previous evaluations $f(\boldsymbol{\theta}^{(1:t)})$. Classical DS algorithms, such as Hooke-Jeeves pattern search, alternate between exploratory and pattern moves to improve candidate solutions without random sampling [20]. However, more advanced versions, such as Mesh Adaptive DS (MADS), introduced a mesh and polling set that ensure convergence to a Clarke stationary point under mild regularity conditions [21]. Similarly, the Nelder-Mead (NM) simplex method maintains a geometric simplex with $d+1$ vertices and uses deterministic reflection, expansion, and contraction operations to navigate the search space [22]. DS methods are attractive for DHPO because they can be reproduced easily, do not require derivatives, and work well with mixed or discrete hyper-parameter domains. However, they can be costly in terms of computational budgets and may struggle with scalability in high-dimensional spaces.

3.2. Surrogate-Based Deterministic HPO Methods

Surrogate-based deterministic methods aim to reduce the evaluation cost of hyperparameter optimisation by approximating the expensive objective function $f(\boldsymbol{\theta})$ with a computationally cheaper surrogate model $s(\boldsymbol{\theta})$. At each iteration, the surrogate is constructed using previously evaluated configurations, such that

$$s(\boldsymbol{\theta}) \approx f(\boldsymbol{\theta}) \quad \text{for all } \boldsymbol{\theta} \in \Theta, \tag{7}$$

and is then used to guide the following deterministic query for evaluation. A widely used framework is Efficient Global Optimisation (EGO). It employs a Gaussian Process (GP) surrogate and uses a deterministic acquisition function optimiser to select the next candidate point [23]. Radial Basis Function (RBF) surrogates provide an alternative. They often serve as a more scalable model for higher-

 doi:10.20944/preprints202603.2300.v1

dimensional search spaces because of their interpolation-based structure and lower training costs [24]. Unlike stochastic Bayesian optimisation approaches, surrogate-based DHPO fixes the initial design, sampling policy, and acquisition optimiser. This pattern ensures that the optimisation trajectory can be reproduced. Surrogate models can allow a significant reduction in the necessary function evaluations. They are particularly helpful when $f(\boldsymbol{\theta})$ involves computationally expensive model training, though their performance may decline as dimensionality increases.

3.3. Hypergradient-Based Deterministic HPO Methods

Hypergradient-based methods address hyper-parameter optimisation by differentiating through the training process, thereby exploiting gradient information of the outer objective with respect to hyperparameters. Considering the bi-level formulation in (2)–(3), the hyper-gradient of the validation loss with respect to $\boldsymbol{\theta}$ is obtained via

$$\nabla_{\theta}\mathcal{L}(f_{\theta}, \mathcal{D}_{\text{val}}) = \frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}} + \frac{\partial \mathcal{L}}{\partial \mathbf{w}^*}\left(\frac{\partial \mathbf{w}^*}{\partial \boldsymbol{\theta}}\right), \tag{8}$$

where $\partial \mathbf{w}^*/\partial \boldsymbol{\theta}$ is computed through either unrolling training steps or using implicit differentiation. Early work, such as reversible learning [25], showed that it is possible to differentiate over long training periods. HOAG [26] later proposed an approximate implicit differentiation method that significantly reduced memory requirements. Recent developments enable scalable bilevel optimisation via efficient implicit differentiation and matrix-free solvers [27]. Hypergradient-based HPO methods achieve reliable convergence and higher sample efficiency than black-box search. However, they depend on the differentiability of both the learning algorithm and the training dynamics, which limits their use to models and losses that support stable differentiation.

3.4. Multi-Fidelity Deterministic HPO Methods

Multi-fidelity hyper-parameter optimisation methods aim to lower the computational cost of evaluating $f(\boldsymbol{\theta})$. They do this by using less expensive, lower-fidelity approximations of the objective before fully committing to training. Let $\mathcal{F} = \{f^{(1)}, f^{(2)}, \ldots, f^{(K)}\}$ denote a set of fidelity levels, where $f^{(1)}$ is the lowest-cost and $f^{(K)} = f$ is the full-fidelity evaluation. The optimisation process selectively evaluates

$$f^{(k)}(\boldsymbol{\theta}) \approx f(\boldsymbol{\theta}), \qquad \text{with cost } C^{(k)} < C^{(K)}, \tag{9}$$

Poorly performing configurations are discarded early at low fidelity. Deterministic Successive Halving [28] allocates resources across candidate configurations. It repeatedly halves the pool by keeping only the top performers at each fidelity. Hyperband builds on this idea by scheduling multiple resource allocation brackets. This pattern helps balance exploration and exploitation [29]. More recent developments like DyHPO [30] use deterministic performance modelling to adjust evaluation fidelity as the search moves forward. Multi-fidelity DHPO methods lower computational costs while ensuring reproducibility. This makes them ideal for expensive deep learning tasks where fully training every candidate is not practical.

4. Comparative Evaluation and Discussion

As can be seen in Table 1, we compare 30 deterministic HPO methods based on key features and practical use. DS methods are simple and easy to repeat, but they tend to be local and often require a lot of evaluations. Surrogate-based methods allow for more efficient global searches, although they struggle with scalability in high dimensions. Hyper-gradient methods work very well for deep learning; however, they require differentiability and are more complex to implement. Multi-Fidelity strategies save on computation by adjusting resource allocation, making them suitable for situations with limited budgets. In summary, each method works best in specific situations, and no single approach is the best across all factors. It is noted that GP-BO/EGO, FastBO, DyHPO are deterministic *iff* we fix: initial design, acquisition optimiser, and seeds.

 doi:10.20944/preprints202603.2300.v1

Table 1. Overview of deterministic hyper-parameter optimisation methods, characteristics, scope and their computational budgets

Category	Method	Supports Constraints	Search Scope	Parallel-Friendly	Differentiability Required	Typical Eval Count	DL Suitability (1-5)	Scope Color	Citation
Direct Search	Coordinate Descent (Cyclic)	✗	Local	✗	✗	Low&Medium	3	🟢	[31]
	Grid Search	✓	Global	✓	✗	High	3	🔵	[32]
	Hooke-Jeeves (Pattern Search)	✗	Local	✓	✗	Medium	3	🟢	[10]
	Greedy Depth-First Search (DFS)	✓	Local	✗	✗	Medium	4	🟢	[33]
	Nelder-Mead Simplex	✗	Local	✓	✗	Medium	3	🟢	[12]
	Piyavskii-Shubert (Lipschitz, 1-D)	✗	Global	✗	✗	Medium&High	2	🔵	[34]
	Golden-Section Line Search (1-D)	✗	Local	✗	✗	Low&Medium	2	🟢	[35]
	NEWUOA	✗	Local	✓	✗	Medium	4	🟢	[36]
	MADS (Mesh Adaptive Direct Search)	✓	Local	✓	✗	Medium&High	4	🟢	[11]
	BOBYQA	✓	Local	✓	✗	Medium	4	🟢	[13]
	COBYLA	✓	Local	✓	✗	Medium	4	🟢	[37]
	DIRECT (DIviding RECTangles)	✗	Global	✓	✗	Medium&High	4	🔵	[38]
	GPS (Generalized Pattern Search)	✓	Local	✓	✗	Medium	3	🟢	[39]
	HyperNOMAD (MADS for DNN HPO)	✓	Local	✓	✗	Medium&High	5	🟢	[40]
Surrogate	EGO / GP-based BO (deterministic)	✓	Global	✓	✗	Low&Medium	5	🔵	[1]
	RBF Surrogate (Global)	✓	Global	✓	✗	Low&Medium	5	🔵	[19]
	FastBO (Adaptive Fidelity Identification)	✓	Global	✓	✗	Low&Medium	5	🔵	[41]
Hyper-gradient	Reverse-mode Hyper-gradient (Unrolled)	✓	Local	✓	✓	Very Low	5	🟢	[16]
	HOAG (Approx. Hyper-gradient)	✓	Local	✓	✓	Low	5	🟢	[14]
	Implicit Differentiation	✓	Local	✓	✓	Low	5	🟢	[42]
	Nonsmooth Implicit Differentiation	✓	Local	✓	✓	Low	5	🟢	[43]
	BOME (Bilevel Optimization Made Easy)	✓	Local	✓	✓	Low	5	🟢	[44]
	Nonsmooth Implicit Diff (Deterministic ITD/AID)	✓	Local	✓	✓	Low	5	🟢	[45]
	FuncID (Functional Bilevel Optimization)	✓	Local	✓	✓	Low	5	🟢	[46]
	First-Order Linearly Constrained Bilevel	✓	Local	✓	✓	Low	5	🟢	[47]
Multi-Fidelity*	Successive Halving (deterministic)	✗	Global	✓	✗	Low	5	🔵	[48]
	Hyper-band (deterministic brackets)	✗	Global	✓	✗	Low	5	🔵	[49]
	MO-MF HPO (Multi-Objective Multi-Fidelity)	✓	Global	✓	✗	Low&Medium	4	🔵	[49]
	DyHPO (Dynamic MF Bayesian Optimization)	✓	Global	✓	✗	Low	5	🔵	[18]
	Frozen Layers MF-HPO	✓	Global	✓	✗	Low	5	🔵	[50]

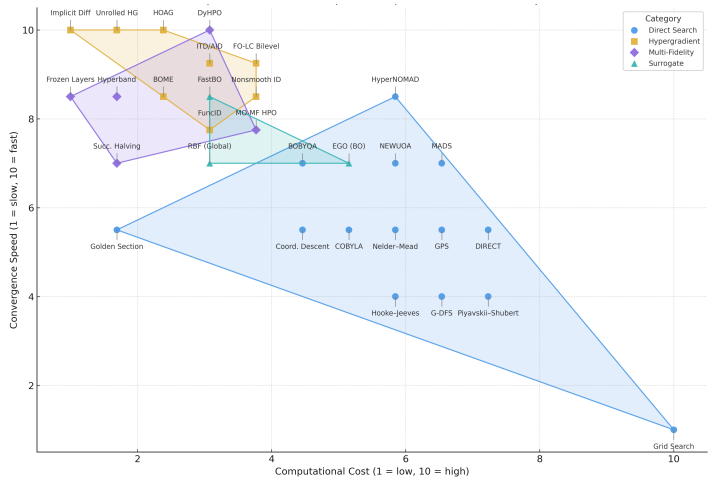
In order to provide a comprehensive comparison among the methods, we focus on specific characteristics such as the performance trade-offs among deterministic HPO families. In Figure 2(a), Direct Search methods usually have low to medium convergence speed and moderate to high computational costs. In contrast, Surrogate-based approaches achieve faster convergence at a lower cost, providing a better balance. Furthermore, hyper-gradient methods are the most efficient, and delivering rapid convergence with a relatively low cost. Multi-Fidelity techniques offer strong speed and cost efficiency through adaptive resource allocation. Figure 2(b) emphasises scalability differences; for example, hyper-gradient and Multi-Fidelity methods perform better in high-dimensional spaces.

Figure 3 reveals clear patterns of drawbacks across deterministic HPO categories. DS methods consistently score high in severity across various areas, especially in high-dimensional scalability (D3), expensive search (D2), and lack of surrogate learning (D8). This shows their limitations for large or complex optimisation tasks. On the other hand, both surrogate-based and hyper-gradient methods represent significantly lower severity in terms of scalability and convergence-related issues. However, hyper-gradient approaches score higher in sensitivity to differentiability and complexity of implementation. Finally, Multi-Fidelity methods show the lowest severity in most cases, highlighting their efficiency and flexibility. Still, they exhibit moderate limitations in managing categorical search spaces. Overall, Figure 3 indicate that each method family has specific structural weaknesses. This emphasises the need to choose deterministic HPO solutions based on specific task requirements instead of using a general approach.

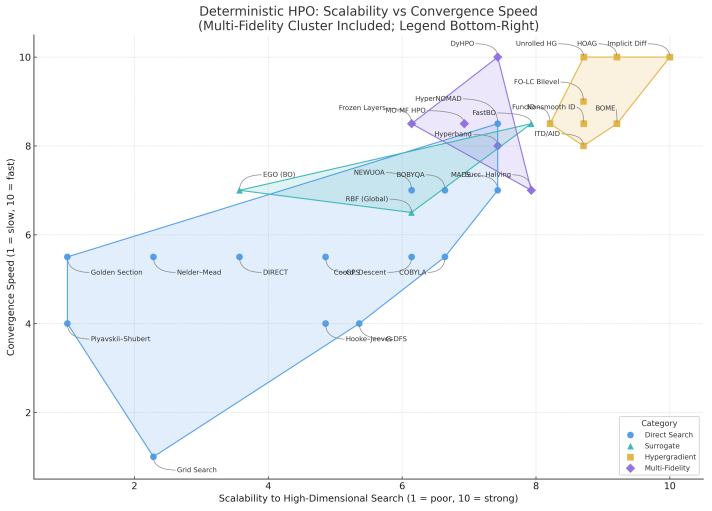
Figure 4 shows clear drawback patterns across deterministic HPO families, which reveal their structural limitations. For instance, DS methods consistently score high in severity across most dimensions. This is particularly true for scalability (D3), cost (D2), and the absence of surrogate or multi-fidelity integration (D7, D8). These findings highlight their weaknesses in modern large-scale optimisation. On the other hand, Surrogate and Hyper-gradient approaches have more balanced profiles. They show lower severity in convergence and scalability but greater sensitivity to differentiability and implementation complexity. Multi-Fidelity methods are notable for having the lowest overall drawback intensity. This makes them the most efficient and scalable option when computational resources are limited. The compact drawbacks checklist that can be seen in Table 2 shows the common

 doi:10.20944/preprints202603.2300.v1

limitations of deterministic DS HPO methods. Most techniques struggle to escape local minima, scale to high-dimensional spaces, and lack support for surrogates or multi-fidelity approaches. This makes them less suitable for large-scale tuning tasks today. While some of the DS methods, like Grid Search and DIRECT, avoid certain issues, most still face problems with expensive searches, slow convergence compared to hyper-gradient-based methods, and sensitivity to initialisation. Overall, Table 2 highlights that DS methods, despite being simple and reproducible, have structural drawbacks that reduce their effectiveness in complex optimization situations. Furthermore, Table 3 shows that although each method has its own limitations, surrogate and multi-fidelity HPO methods have fewer structural issues than DS approaches. Surrogate-based methods perform well in most areas, but their main weakness is lower scalability in high-dimensional settings and their reliance on the quality of the surrogate model. Hyper-gradient methods have the best overall profile, addressing almost all the listed issues. However, they depend on differentiability and add more complexity to implementation. Multi-fidelity methods have the least severe drawbacks, with few limitations across all areas. This confirms their efficiency and flexibility, especially when the computational budget is tight or early stopping is important.



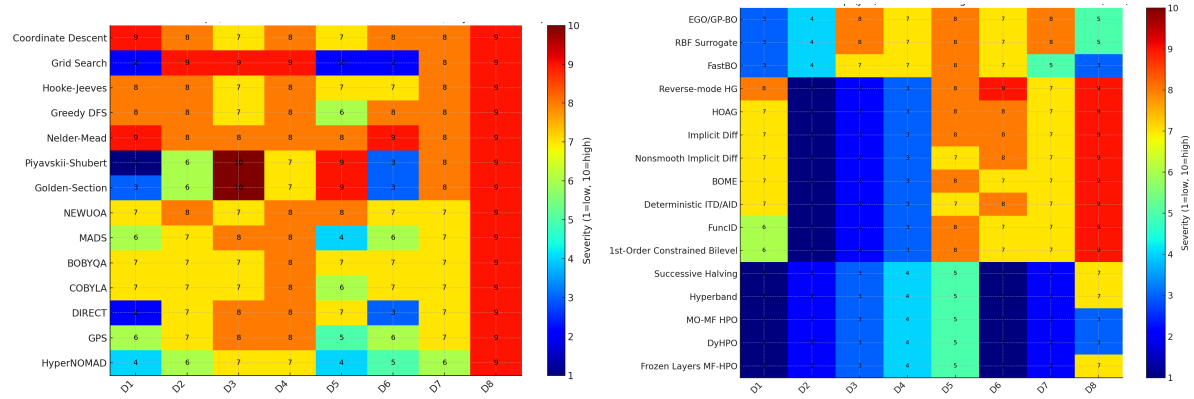
(**a**)



(**b**)

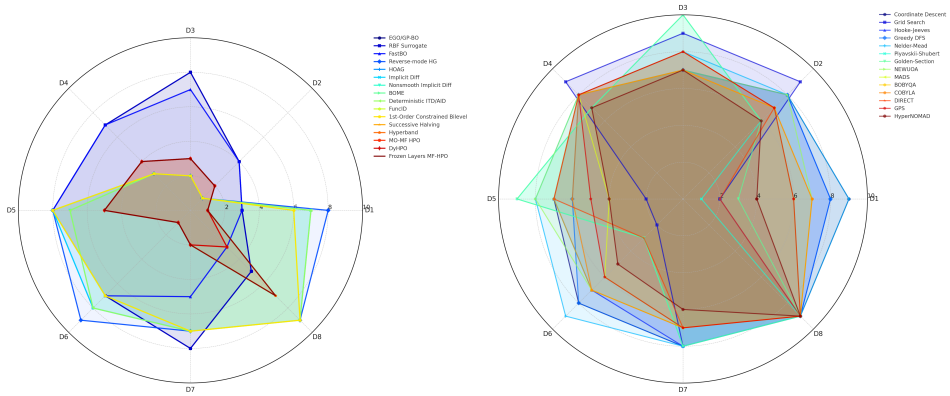
Figure 2. (a) Cost–Speed trade-off landscape of deterministic hyperparameter optimisation (HPO) methods. Methods are grouped by category, with convex-hull regions illustrating cluster boundaries. (b) Comparison in terms of scalability to high-dimensional search spaces. Higher scores indicate stronger scalability or faster convergence.

 doi:10.20944/preprints202603.2300.v1



(**a**) (**b**)

Figure 3. Annotated heatmap (1–10 severity scale) comparing the major drawbacks associated with deterministic (a) DS and (b) Surrogate, Hyper-gradient and Multi-Fidelity hyper-parameter optimisation methods. The analysis benchmarks 14 methods across eight limitation dimensions: (D1) local-minima vulnerability, (D2) costly search, (D3) poor scaling in high-dimensional spaces, (D4) slowness relative to hyper-gradient-based approaches, (D5) weak support for categorical/mixed search spaces, (D6) sensitivity to initialization, (D7) limited use of multi-fidelity information, and (D8) lack of surrogate-based learning. Higher values indicate more severe drawbacks.



(**a**) (**b**)

Figure 4. Comparison of drawback severity profiles for all deterministic HPO categories, highlighting distinct weakness patterns and trade-offs across method families.

The Figure 5 describes the development in the approach adopted in the scientific literature on deterministic hyper-parameter optimisation. This flow chart reveals the development over time from direct search, then surrogate and hyper-gradients, and, latterly, the adoption of multi-fidelity optimisation. The flow charts show that there has been an early preponderance in the use of the grid search algorithm, in which optimisation has high costs and average robustness against local minima, corresponding to difficulty levels 1 through 3. Conversely, there has lately been an increase in the use of hyper-gradients and, later, multi-fidelity optimisation, in which the optimisation process requires differentiability, has lower evaluation costs, and takes advantage of deep architecture, corresponding to scores ranging from 5, and an increased vulnerability against local minima, corresponding to difficulty levels 4-5. The intersecting edges highlight the alignment between conceptual bins and algorithmic characteristics, resulting in a rich representation in the scientific optimisation landscape.

 doi:10.20944/preprints202603.2300.v1

Table 2. Direct Search (deterministic) HPO: compact drawbacks checklist.

Method	Local-minima	Costly Search	High-D issue	Slow vs hyper-grad	Cat/mixed weak	Init-sensitive	Weak multi-fidelity	No surrogate learning
Coordinate Descent (Cyclic)	✓	✓	✓	✓	✓	✓	✓	✓
Grid Search	X	✓	✓	✓	X	X	✓	✓
Hooke–Jeeves (Pattern)	✓	✓	✓	✓	✓	✓	✓	✓
Greedy DFS	✓	✓	✓	✓	✓	✓	✓	✓
Nelder–Mead Simplex	✓	✓	✓	✓	✓	✓	✓	✓
Piyavskii–Shubert (1-D)	X	✓	✓	✓	✓	X	✓	✓
Golden–Section (1-D)	X	✓	✓	✓	✓	X	✓	✓
NEWUOA	✓	✓	✓	✓	✓	✓	✓	✓
MADS	✓	✓	✓	✓	X	✓	✓	✓
BOBYQA	✓	✓	✓	✓	✓	✓	✓	✓
COBYLA	✓	✓	✓	✓	✓	✓	✓	✓
DIRECT	X	✓	✓	X	✓	X	✓	✓
GPS	✓	✓	✓	✓	X	✓	✓	✓
HyperNOMAD	✓	✓	✓	✓	X	✓	✓	✓

Table 3. The listed drawbacks of Surrogate, Hyper-gradient, and Multi-Fidelity HPO methods.

Method	Local-minima	Costly Search	High-D issue	Slow vs hyper-grad	Cat/mixed weak	Init-sensitive	Weak multi-fidelity	No surrogate learning
Surrogate HPO methods								
EGO / GP-based BO (deterministic)	X	X	✓	✓	✓	✓	✓	X
RBF Surrogate (Global)	X	X	✓	✓	✓	✓	✓	X
FastBO (Adaptive Fidelity Identification)	X	X	✓	✓	✓	✓	X	X
Hyper-gradient HPO methods								
Reverse-mode Hyper-gradient (Unrolled)	✓	X	X	X	✓	✓	✓	✓
HOAG (Approx. Hyper-gradient)	✓	X	X	X	✓	✓	✓	✓
Implicit Differentiation	✓	X	X	X	✓	✓	✓	✓
Nonsmooth Implicit Differentiation	✓	X	X	X	✓	✓	✓	✓
BOME (Bilevel Optimization Made Easy)	✓	X	X	X	✓	✓	✓	✓
Nonsmooth Implicit Diff (Deterministic ITD/AID)	✓	X	X	X	✓	✓	✓	✓
FuncID (Functional Bilevel Optimization)	✓	X	X	X	✓	✓	✓	✓
First-Order Linearly Constrained Bilevel	✓	X	X	X	✓	✓	✓	✓
Multi-Fidelity HPO methods								
Successive Halving (deterministic)	X	X	X	X	X	X	X	✓
Hyper-band (deterministic brackets)	X	X	X	X	X	X	X	✓
MO-MF HPO (Multi-Objective Multi-Fidelity)	X	X	X	X	X	X	X	X
DyHPO (Dynamic MF Bayesian Optimization)	X	X	X	X	X	X	X	X
Frozen Layers MF-HPO	X	X	X	X	X	X	X	✓

 doi:10.20944/preprints202603.2300.v1

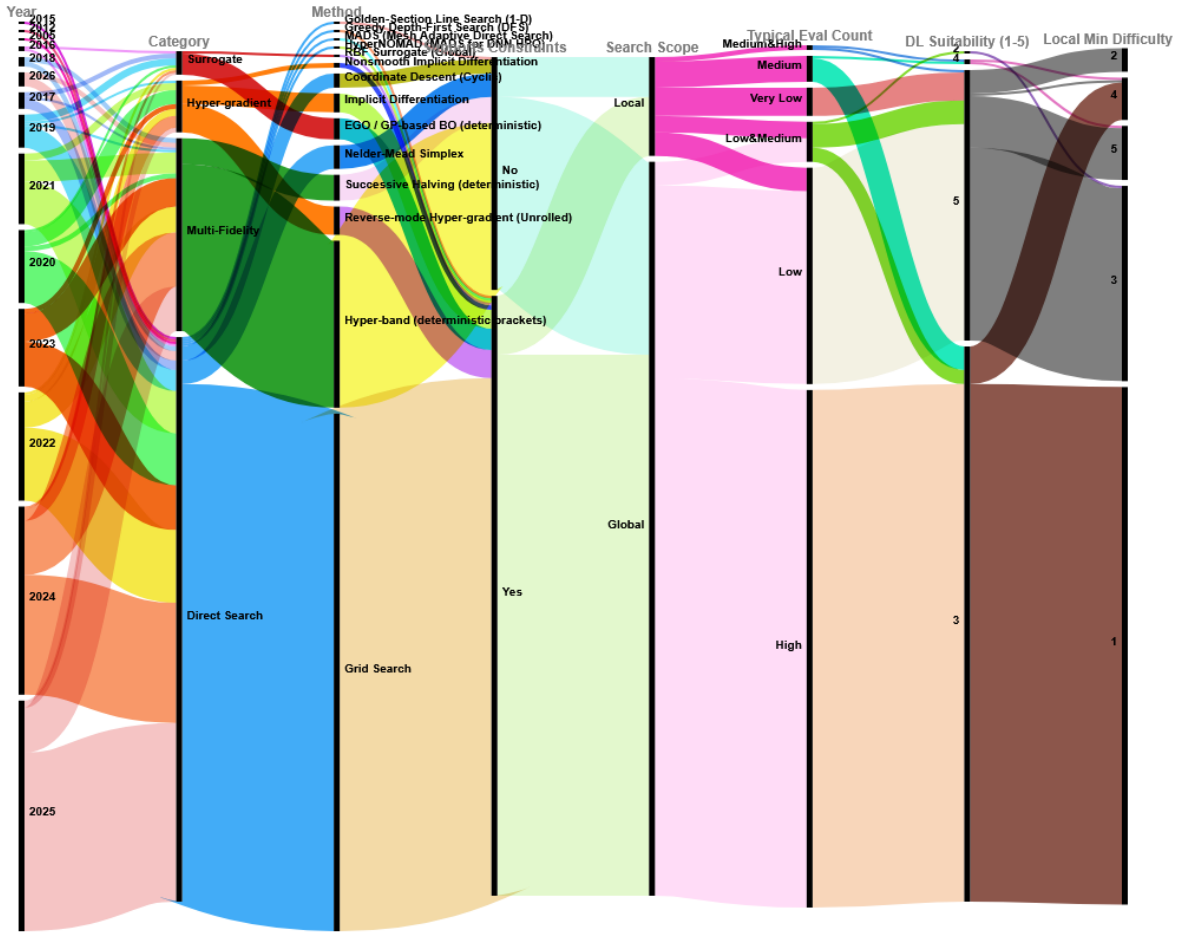


Figure 5. Alluvial diagram illustrating the relationships between publication year, DHPO categories, specific methods, search scope, evaluation cost, deep-learning suitability, and susceptibility to local minima based on Scopus-indexed literature

5. Benchmark Datasets and Frameworks

Evaluations of differentiable hyperparameter optimisation tools must be conducted under controlled, reproducible test settings focusing on test complexity, runtime performance, and faithfulness. Containerised hyperparameter optimisation benchmarking with identical interfaces across application areas has been provided in HPOBench [51] and with cost models supported in YAHPO Gym [52]. In NAS-HPO-Bench-II, pre-evaluated learning curves enable optimisation across architecture and hyperparameters in unison [53]. Generalisation over different data sets in DHPO testing has become possible using OpenML-CC18 [54]. In the MLCommons benchmark environment, platform-independent testing and benchmarking are supported [55], and in the AutoML-Benchmark, everything-to-end comparability under identical conditions has now become available [56]. Realistic large-scale testing has become possible using DeepHyper [57] testing tools, including deterministic settings. Testing with rich domain knowledge has become available using Optuna-Bench [58] and Meta-Dataset [59].

6. Gaps, Challenges, and Future Directions

6.1. Current Gaps in Deterministic HPO

Although deterministic HPO (DHPO) has advanced across different method families, several apparent gaps remain. First, most existing approaches are not designed for today's large-scale foundation models (FMs), where training costs and memory demands are significantly higher. Second, DHPO methods typically optimise hyper-parameters in isolation, without considering architecture or data choices, even though these three are strongly connected in practice. Third, achieving true reproducibility is still difficult at the system level. Even if the optimiser is deterministic, the pipeline may still rely

 doi:10.20944/preprints202603.2300.v1

on random data splits, augmentations, or model selection steps. Finally, DHPO research lacks widely accepted benchmark standards, especially ones that evaluate both accuracy and computational cost across different model sizes and training budgets.

6.2. Key Challenges Hindering Progress

Scaling DHPO to large-scale deep learning and foundation-model settings brings several practical and technical challenges. Because deterministic search typically explores the space more narrowly, it can struggle to navigate high-dimensional and highly non-convex landscapes effectively. Surrogate-based DHPO methods [23,24] tend to lose accuracy as the number of hyper-parameters grows, while hyper-gradient methods [25–27] demand differentiability, stable long-horizon training, and careful memory management to remain usable in large models. Multi-Fidelity strategies [28,29] also rely heavily on how fidelity levels are defined, and these settings do not always transfer cleanly across architectures or training regimes. Achieving full determinism at the system level adds another layer of difficulty—hardware differences, data loading behaviour, random seeds, mixed precision, parallelism, and dynamic computation graphs can all introduce subtle non-determinism. Finally, combining hyper-parameter, architecture, and data optimization into one deterministic framework is inherently complex, especially when discrete design choices must be represented in a reproducible and differentiable way.

6.3. Future Research Directions

Several promising directions can significantly advance DHPO in the coming years. (i) Foundation-Model DHPO: Future work should focus on making DHPO practical for foundation models by introducing deterministic low-fidelity training shortcuts, and reproducible parameter-efficient tuning strategies. These should be supported by budget-aware multi-fidelity schedulers [28,29]. (ii) Hybrid Deterministic–Stochastic DHPO: A balanced approach that combines a deterministic search backbone with controlled, reproducible randomness may improve search coverage while still maintaining full auditability and repeatability. (iii) Differentiable Architecture and Data HPO: Extending hyper-gradient-based DHPO to architecture search and data-centric optimisation will require robust deterministic bilevel solvers, stable implicit differentiation [25–27], reproducible training over long horizons, and deterministic relaxations to handle discrete design choices. (iv) Deterministic AutoML: A fully deterministic AutoML pipeline should guarantee reproducibility across every stage—from data handling to model training, HPO, and evaluation—leveraging deterministic multi-fidelity and surrogate-based optimizers [23,24].

7. Conclusion

In this paper, we reviewed and compared deterministic hyper-parameter optimization (DHPO) across four primary method families: Direct Search, Surrogate-based, Hyper-gradient, and Multi-Fidelity approaches. We defined DHPO within a reproducible bilevel optimisation framework and explained why deterministic methods are increasingly important, especially in situations where reliability, auditability, and repeatability matter. Our analysis showed apparent performance differences. Hyper-gradient and Multi-Fidelity methods usually provide the best balance of speed, efficiency, and scalability for modern deep learning tasks. In contrast, Surrogate-based approaches are still effective for reducing search costs when compute budgets are limited. The findings confirmed that no single DHPO method is the best; the right choice largely depends on the problem's dimensionality, resource limits, and the required level of determinism. Overall, our findings suggest a shift in DHPO research and practice. As models increase in size and reproducibility becomes a requirement, there is a clear need for DHPO methods that operate at foundation-model scale, combine hyper-parameter, architecture, and data optimization, and offer system-level determinism. Promising directions include hybrid deterministic-stochastic search, more efficient deterministic Multi-Fidelity and surrogate-driven pipelines, differentiable architecture and data-centric HPO, and ultimately, fully deterministic AutoML frameworks that can ensure transparent and trustworthy model development for high-stakes applications.

 doi:10.20944/preprints202603.2300.v1

References

1. Donald R Jones, Matthias Schonlau, and William J Welch. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13(4):455–492, 1998.
2. James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *The journal of machine learning research*, 13(1):281–305, 2012.
3. Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. *Advances in neural information processing systems*, 25, 2012.
4. Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.
5. Kevin Jamieson and Ameet Talwalkar. Non-stochastic best arm identification and hyperparameter optimization. In *Artificial intelligence and statistics*, pages 240–248. PMLR, 2016.
6. Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. Hyperband: A novel bandit-based approach to hyperparameter optimization. *Journal of Machine Learning Research*, 18(185):1–52, 2018.
7. Stefan Falkner, Aaron Klein, and Frank Hutter. Bohb: Robust and efficient hyperparameter optimization at scale. In *International conference on machine learning*, pages 1437–1446. PMLR, 2018.
8. Michael D McKay, Richard J Beckman, and William J Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 42(1):55–61, 2000.
9. Rommel G Regis and Christine A Shoemaker. A stochastic radial basis function method for the global optimization of expensive functions. *INFORMS Journal on Computing*, 19(4):497–509, 2007.
10. Robert Hooke and Terry A Jeeves. "direct search"solution of numerical and statistical problems. *Journal of the ACM (JACM)*, 8(2):212–229, 1961.
11. Charles Audet and John E Dennis Jr. Mesh adaptive direct search algorithms for constrained optimization. *SIAM Journal on optimization*, 17(1):188–217, 2006.
12. John A Nelder and Roger Mead. A simplex method for function minimization. *The computer journal*, 7(4):308–313, 1965.
13. Michael JD Powell et al. The bobyqa algorithm for bound constrained optimization without derivatives. *Cambridge NA Report NA2009/06, University of Cambridge, Cambridge*, 26:26–46, 2009.
14. Fabian Pedregosa. Hyperparameter optimization with approximate gradient. In *International conference on machine learning*, pages 737–746. PMLR, 2016.
15. Jonathan Lorraine, Paul Vicol, and David Duvenaud. Optimizing millions of hyperparameters by implicit differentiation. In *International conference on artificial intelligence and statistics*, pages 1540–1552. PMLR, 2020.
16. Paul Micaelli and Amos J Storkey. Gradient-based hyperparameter optimization over long horizons. *Advances in Neural Information Processing Systems*, 34:10798–10809, 2021.
17. Kirthevasan Kandasamy, Gautam Dasarathy, Jeff Schneider, and Barnabás Póczos. Multi-fidelity bayesian optimisation with continuous approximations. In *International conference on machine learning*, pages 1799–1808. PMLR, 2017.
18. Martin Wistuba, Arlind Kadra, and Josif Grabocka. Supervising the multi-fidelity race of hyperparameter configurations. *Advances in Neural Information Processing Systems*, 35:13470–13484, 2022.
19. Felipe AC Viana, Raphael T Haftka, and Layne T Watson. Efficient global optimization algorithm assisted by multiple surrogate techniques. *Journal of Global Optimization*, 56(2):669–689, 2013.
20. Robert Hooke and Terry A. Jeeves. "direct search" solution of numerical and statistical problems. *Journal of the ACM*, 8(2):212–229, 1961.
21. Charles Audet and John E. Dennis Jr. Mesh adaptive direct search algorithms for constrained optimization. *SIAM Journal on Optimization*, 17(1):188–217, 2006.
22. John A Nelder and Roger Mead. A simplex method for function minimization. *The Computer Journal*, 7(4):308–313, 1965.
23. Donald R Jones, Matthias Schonlau, and William J Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492, 1998.
24. Rommel G Regis and Christine A Shoemaker. A stochastic radial basis function method for the global optimization of expensive functions. *Journal of Global Optimization*, 37(1):197–219, 2007.
25. Dougal Maclaurin, David Duvenaud, and Ryan Adams. Gradient-based hyperparameter optimization through reversible learning. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, pages 2113–2122, 2015.

 doi:10.20944/preprints202603.2300.v1

26. Fabian Pedregosa. Hyperparameter optimization with approximate gradient. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, pages 737–746, 2016.
27. Jonathan Lorraine, Paul Vicol, and David Duvenaud. Optimizing millions of hyperparameters by implicit differentiation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.
28. Kevin Jamieson and Ameet Talwalkar. Non-stochastic best arm identification and hyperparameter optimization via successive halving. In *AISTATS*, pages 240–248, 2016.
29. Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. Hyperband: A novel bandit-based approach to hyperparameter optimization. *Journal of Machine Learning Research*, 18(185):1–52, 2018.
30. Martin Wistuba, Arlind Kadra, and Josif Grabocka. Supervising the multi-fidelity race of hyperparameter configurations. In *NeurIPS*, 2022.
31. Stephen J Wright. Coordinate descent algorithms. *Mathematical programming*, 151(1):3–34, 2015.
32. Jinhyo Kim. *Iterated grid search algorithm on unimodal criteria*. Virginia Polytechnic Institute and State University, 1997.
33. Dexter C Kozen. Depth-first and breadth-first search. In *The design and analysis of algorithms*, pages 19–24. Springer, 1992.
34. Kaan Gokcesu and Hakan Gökcesu. Cumulative regret analysis of the piyavskii–shubert algorithm and its variants for global optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 20700–20708, 2024.
35. Richard P Brent. *Algorithms for minimization without derivatives*. Courier Corporation, 2013.
36. Michael JD Powell. The newuoa software for unconstrained optimization without derivatives. In *Large-scale nonlinear optimization*, pages 255–297. Springer, 2006.
37. Patrick Koch, Samineh Bagheri, Wolfgang Konen, Christophe Foussette, Peter Krause, and Thomas Bäck. A new repair method for constrained optimization. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, pages 273–280, 2015.
38. Donald R Jones, Cary D Perttunen, and Bruce E Stuckman. Lipschitzian optimization without the lipschitz constant. *Journal of optimization Theory and Applications*, 79(1):157–181, 1993.
39. Charles Audet and John E Dennis Jr. Analysis of generalized pattern searches. *SIAM Journal on optimization*, 13(3):889–903, 2002.
40. Dounia Lakhmiri, Sébastien Le Digabel, and Christophe Tribes. Hypernomad: Hyperparameter optimization of deep neural networks using mesh adaptive direct search. *ACM Transactions on Mathematical Software (TOMS)*, 47(3):1–27, 2021.
41. Jiantong Jiang, Zeyi Wen, Atif Mansoor, and Ajmal Mian. Efficient hyperparameter optimization with adaptive fidelity identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26181–26190, 2024.
42. Paul Vicol, Jonathan P Lorraine, Fabian Pedregosa, David Duvenaud, and Roger B Grosse. On implicit bias in overparameterized bilevel optimization. In *International Conference on Machine Learning*, pages 22234–22259. PMLR, 2022.
43. Tianlong Chen, Xiaohan Chen, Wuyang Chen, Howard Heaton, Jialin Liu, Zhangyang Wang, and Wotao Yin. Learning to optimize: A primer and a benchmark. *Journal of Machine Learning Research*, 23(189):1–59, 2022.
44. Bo Liu, Mao Ye, Stephen Wright, Peter Stone, and Qiang Liu. Bome! bilevel optimization made easy: A simple first-order approach. *Advances in neural information processing systems*, 35:17248–17262, 2022.
45. Riccardo Grazzi, Massimiliano Pontil, and Saverio Salzo. Nonsmooth implicit differentiation: Deterministic and stochastic convergence rates. *arXiv preprint arXiv:2403.11687*, 2024.
46. Ieva Petrulionytė, Julien Mairal, and Michael Arbel. Functional bilevel optimization for machine learning. *Advances in Neural Information Processing Systems*, 37:14016–14065, 2024.
47. Guy Kornowski, Swati Padmanabhan, Kai Wang, Zhe Zhang, and Suvrit Sra. First-order methods for linearly constrained bilevel optimization. *Advances in Neural Information Processing Systems*, 37:141417–141460, 2024.
48. Sotetsu Koyamada, Soichiro Nishimori, and Shin Ishii. A batch sequential halving algorithm without performance degradation. *arXiv preprint arXiv:2406.00424*, 2024.
49. Florian Karl, Tobias Pielok, Julia Moosbauer, Florian Pfisterer, Stefan Coors, Martin Binder, Lennart Schneider, Janek Thomas, Jakob Richter, Michel Lang, et al. Multi-objective hyperparameter optimization in machine learning—an overview. *ACM Transactions on Evolutionary Learning and Optimization*, 3(4):1–50, 2023.
50. Timur Carstensen, Neeratyoy Mallik, Frank Hutter, and Martin Rapp. Frozen layers: Memory-efficient many-fidelity hyperparameter optimization. *arXiv preprint arXiv:2504.10735*, 2025.

 doi:10.20944/preprints202603.2300.v1

51. Katharina Eggensperger, Philipp Müller, Neeratyoy Mallik, Matthias Feurer, René Sass, Aaron Klein, Noor Awad, Marius Lindauer, and Frank Hutter. Hpobench: A collection of reproducible multi-fidelity benchmark problems for hpo. *arXiv preprint arXiv:2109.06716*, 2021.
52. Florian Pfisterer, Lennart Schneider, Julia Moosbauer, Martin Binder, and Bernd Bischl. Yahpo gym-an efficient multi-objective multi-fidelity benchmark for hyperparameter optimization. In *International Conference on Automated Machine Learning*, pages 3–1. PMLR, 2022.
53. Yoichi Hirose, Nozomu Yoshinari, and Shinichi Shirakawa. Nas-hpo-bench-ii: A benchmark dataset on joint optimization of convolutional neural network architecture and training hyperparameters. In *Asian Conference on Machine Learning*, pages 1349–1364. PMLR, 2021.
54. Bernd Bischl, Giuseppe Casalicchio, Matthias Feurer, Pieter Gijsbers, Frank Hutter, Michel Lang, Rafael G Mantovani, Jan N van Rijn, and Joaquin Vanschoren. Openml benchmarking suites. *arXiv preprint arXiv:1708.03731*, 2017.
55. Yen-Hsiang Chang, Jianhao Pu, Wen-mei Hwu, and Jinjun Xiong. Mlharness: A scalable benchmarking system for mlcommons. *BenchCouncil Transactions on Benchmarks, Standards and Evaluations*, 1(1):100002, 2021.
56. Pieter Gijsbers, Marcos LP Bueno, Stefan Coors, Erin LeDell, Sébastien Poirier, Janek Thomas, Bernd Bischl, and Joaquin Vanschoren. Amlb: an automl benchmark. *Journal of Machine Learning Research*, 25(101):1–65, 2024.
57. Prasanna Balaprakash, Michael Salim, Thomas D Uram, Venkat Vishwanath, and Stefan M Wild. Deephyper: Asynchronous hyperparameter search for deep neural networks. In *2018 IEEE 25th international conference on high performance computing (HiPC)*, pages 42–51. IEEE, 2018.
58. Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2623–2631, 2019.
59. Eleni Triantafillou, Tyler Zhu, Vincent Dumoulin, Pascal Lamblin, Utku Evci, Kelvin Xu, Ross Goroshin, Carles Gelada, Kevin Swersky, Pierre-Antoine Manzagol, et al. Meta-dataset: A dataset of datasets for learning to learn from few examples. *arXiv preprint arXiv:1903.03096*, 2019.