nature machine intelligence

Article

https://doi.org/10.1038/s42256-023-00642-4

Machine learning-enabled globally guaranteed evolutionary computation

Received: 11 October 2022

Accepted: 8 March 2023

Published online: 10 April 2023

Check for updates

Bin Li[1,2,6], Ziping Wei[1,6], Jingjing Wu[1], Shuai Yu[3], Tian Zhang[3], Chunli Zhu[2], Dezhi Zheng[2], Weisi Guo[4,5], Chenglin Zhao[1] & Jun Zhang[2]

Evolutionary computation, for example, particle swarm optimization, has impressive achievements in solving complex problems in science and industry; however, an important open problem in evolutionary computation is that there is no theoretical guarantee of reaching the global optimum and general reliability; this is due to the lack of a unified representation of diverse problem structures and a generic mechanism by which to avoid local optima. This unresolved challenge impairs trust in the applicability of evolutionary computation to a variety of problems. Here we report an evolutionary computation framework aided by machine learning, named EVOLER, which enables the theoretically guaranteed global optimization of a range of complex non-convex problems. This is achieved by: (1) learning a low-rank representation of a problem with limited samples, which helps to identify an attention subspace; and (2) exploring this small attention subspace via the evolutionary computation method, which helps to reliably avoid local optima. As validated on 20 challenging benchmarks, this method finds the global optimum with a probability approaching 1. We use EVOLER to tackle two important problems: power grid dispatch and the inverse design of nanophotonics devices. The method consistently reached optimal results that were challenging to achieve with previous state-of-the-art methods. EVOLER takes a leap forwards in globally guaranteed evolutionary computation, overcoming the uncertainty of data-driven black-box methods, and offering broad prospects for tackling complex real-world problems.

Evolutionary computation is essential to complex real-world problems that cannot be solved by classical gradient-based methods, for example, molecular docking or dynamics simulation[1–3], nanophotonics[4–7], ultra-fast lasers[8,9], medical diagnosis[10], nuclear physics[11], power grid dispatch[12–14], aerodynamics[15,16] and so on. Particle swarm optimization (PSO), a well-known evolutionary computation method, is a powerful tool for such challenging problems[17–20]. By reconciling two conflicting aspects—exploitation and exploration[19]—PSO learns from the best experience of the particle itself and the best experience of the whole population[17,21]; it has therefore attracted great attention due to its high efficiency and flexibility[22,23]. Various new mechanisms have recently been designed to avoid vanishing diversity or achieve fast convergence[21], including: multiple populations[24–26], learning strategies[27,28], velocity update[29], parameter settings[30], population topology[31,32] and so on. Furthermore, existing theoretical works have analysed the trajectory of the simplified particle[33,34], the second-order and Lyapunov stabilities[35–37], and the effect of parameters on stability and convergence[35,38–40]. But there are still two long-standing pitfalls in

[1]School of Information and Communication Engineering, Beijing University of Posts and Telecommunications (BUPT), Beijing, China. [2]School of Information and Electronics, Beijing Institute of Technology, Beijing, China. [3]State Key Laboratory of Information Photonics and Optical Communications, Beijing University of Posts and Telecommunications, Beijing, China. [4]Centre for Autonomous and Cyberphysical Systems, Cranfield University, Bedford, UK. [5]The Alan Turing Institute, London, UK. [6]These authors contributed equally: Bin Li, Ziping Wei. e-mail: Binli@bupt.edu.cn; buaazhangjun@vip.sina.com

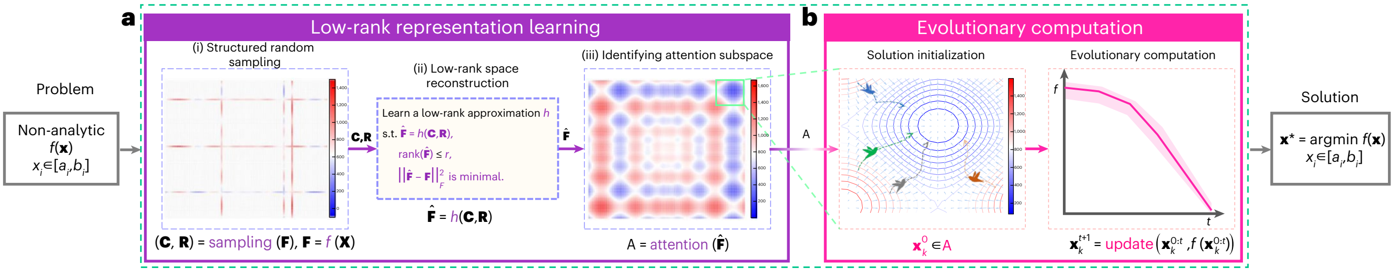
a
Low-rank representation learning
Problem
Non-analytic
$f(\mathbf{x})$
$x_i \in [a_i, b_i]$
(i) Structured random sampling
$(\mathbf{C}, \mathbf{R})$ = sampling $(\mathbf{F})$, $\mathbf{F} = f(\mathbf{X})$
C,R
(ii) Low-rank space reconstruction
Learn a low-rank approximation h
s.t. $\hat{\mathbf{F}} = h(\mathbf{C},\mathbf{R})$,
$\mathrm{rank}(\hat{\mathbf{F}}) \le r$,
$\|\hat{\mathbf{F}} - \mathbf{F}\|_F^2$ is minimal.
$\hat{\mathbf{F}} = h(\mathbf{C},\mathbf{R})$
$\hat{\mathbf{F}}$
(iii) Identifying attention subspace
A = attention $(\hat{\mathbf{F}})$
A
b
Evolutionary computation
Solution initialization
$\mathbf{x}_k^0 \in A$
Evolutionary computation
f
t
$\mathbf{x}_k^{t+1}$ = update$\left(\mathbf{x}_k^{0:t}, f(\mathbf{x}_k^{0:t})\right)$
Solution
$\mathbf{x}^* = \mathrm{argmin}\, f(\mathbf{x})$
$x_i \in [a_i, b_i]$

Fig. 1 | Schematic flow of EVOLER. a, Learning a low-rank representation of unknown problem space: (i) randomly sampling s columns/rows of unknown space $\mathbf{F} \in \mathbb{R}^{M\times N}$ ($d = 2$, $s = 3$; taking a 2D Schwefel function for example); (ii) reconstructing the low-rank representation space $\hat{\mathbf{F}} \in \mathbb{R}^{M\times N}$ (s.t., subject to); (iii) identifying an attention subspace. **b**, Exploring the small attention subspace via evolutionary computation and finding the global optimum.

classical PSO methods that seriously undermine trust in their application to real-world problems emerging in science and industry.

There being no theoretical guarantee of attaining the global optimum of evolutionary computation methods[21] has been an important open problem for more than 50 years[41]. Due to the lack of a unified description of diverse problem structures and a generic mechanism to avoid local optima, globally guaranteed evolutionary computation remains a challenging task, especially for complex non-convex problems. This leads to the so-called uncertainty problem; that is, after a limited computation time, one can hardly know whether or not the returned solution is globally optimal. For another, according to the celebrated no free lunch (NFL) theorem[42], almost all such heuristic methods are problem-dependent. That is to say, every refined mechanism is only suitable to specific geometric structures (for example, convex envelope, smoothness and so on), yet ineffective or even problematic to the other problems of different structures. Consequently, there is no universal best method for general problems[21,43,44]. This presents another critical difficulty, especially when handling complex real-world problems with unexplored properties.

In this article we report a new evolutionary computation framework aided by machine learning, named EVOLER, which enables the theoretically guaranteed global optimization of complex non-convex problems. By leveraging the common low-rank structure of real-world problems—which is barely considered in evolutionary computation—our new method involves two stages (Fig. 1). First, with a small portion of structured samples (Fig. 1ai), a low-rank representation of the problem is learned (Fig. 1aii), which helps to identify one small attention subspace (Fig. 1aiii). Second, this identified subspace is explored via evolutionary computation methods (Fig. 1b), which offers a generic mechanism to reliably avoid local optima. As shown by empirical and theoretical results, EVOLER finds the global optimum with a probability approaching 1 for rank-restricted problems (Figs. 2c and 3u,v). Moreover, it substantially extends the applicability of evolutionary computation in diverse problems with ubiquitous low-rank structures, and exhibits the best performance in 20 challenging benchmarks[30,45] (see Fig. 3). Furthermore, the convergence can be accelerated by orders of magnitude (Figs. 2f and 3).

We demonstrate our new optimizer in two important real-world problems: power grid dispatch and the inverse design of nanophotonics devices. For the first problem, EVOLER almost surely finds the optimal economic cost that was not reported, making global optimization possible subject to equality/inequality constraints. For the second problem, it consistently gains the optimal nanophotonic structures and attains the most favourable transmittance responses. Past evolutionary computation methods, however, find only local optima or acquire bad performance. Meanwhile, the number of function evaluations (or samples) is reduced by ~5–10-fold with EVOLER, thus enabling many rapid-prototyping applications. By a marriage of low-rank representation and evolutionary computation, our method makes a substantial stride towards theoretically guaranteed global optimization, overcoming the inherent uncertainty of black-box methods and offering broad prospects for solving challenging real-world problems.

Evolutionary computation with machine learning

We are interested in applying evolutionary computation to solve a prevalent problem, that is, finding the global optimum of a complex non-analytic function $f : \mathbb{R}^d \mapsto \mathbb{R}^1$. Our new method consists of two stages: (1) learning the low-rank representation, and (2) performing the evolutionary computation (Fig. 1 and Supplementary Fig. 2). First we reconstruct a low-rank representation of unknown problem space from limited samples (Fig. 1a). This learned global representation helps to identify one small attention subspace, which potentially contains the global optimum. Second, we use the optimization methods to explore this small subspace (Fig. 1b) by reliably rejecting the local optima. As the learned low-rank representation has the analytical expression and the bounded error, the theoretical probability of finding the global optimum can be derived, thus overcoming the uncertainty of black-box evolutionary computation.

Learning low-rank representation

Classical machine learning methods can be applied in the first stage. For example, one may directly use J samples $\{\mathbf{x}_j, f(\mathbf{x}_j)\}_{j=1}^{J}$ to train one neural network, thus modelling the complex unknown function $\hat{f} : \mathbb{R}^d \mapsto \mathbb{R}^1$ by $\hat{f}(\mathbf{x}) = g(\mathbf{W}\mathbf{x} + b)$ (a single layer neural network is considered here for simplicity), where $\mathbf{W} \in \mathbb{R}^{1\times d}$ is the learned network weight, $g(\cdot)$ is the nonlinear activation function and $b \in \mathbb{R}$ is the bias. However, there are two limitations to this classical method in low-rank representation learning. First, it requires huge samples to explore the whole problem space, each of which $\{\mathbf{x}_j, f(\mathbf{x}_j)\}$ corresponds to one time function evaluation. Second, it acquires unexplainable weights (that is, $\mathbf{W}$), which hinders further theoretical analysis. To this end, we design a low-rank representation learning method to reconstruct a problem space from limited samples (Fig. 1aii). Our method is inspired by the ubiquitous low-rank structures of real-world problems (see Supplementary Fig. 1 for diverse benchmark problems, Fig. 5b for the power-dispatch problem and Fig. 6b for the nanophotonics inverse-design problem).

For simplicity, we start with the two-dimensional problems. We first explore the discrete problem space $\mathbf{F} = f(\mathbf{X}) \in \mathbb{R}^{M\times N}$ by randomly sampling s rows/columns (M and N denote the grid sizes of a solution space $\mathbf{X} \in \mathbb{R}^{M\times N}$). The sampled function values $\{f(\mathbf{x}_j)\}_{j=1}^{s(M+N)-s^2}$ are structured into two sub-matrices $\mathbf{C} \in \mathbb{R}^{M\times s}$ and $\mathbf{R} \in \mathbb{R}^{s\times N}$ (where $s \ll \min(M, N)$); see Fig. 1ai for the sampling result of a non-convex Schwefel function (Supplementary Table 1); $s = 3$, $M = N = 100$. Targeting a sample-efficient reconstruction of $\mathbf{F}$, we turn to learning another mapping function $h : \mathbb{R}^{(M\times s)\times(s\times N)} \mapsto \mathbb{R}^{M\times N}$, so that $\hat{\mathbf{F}} = h(\mathbf{C}, \mathbf{R}) = \hat{f}(\mathbf{X})$ (Fig. 1aii). This formulated model may be intractable as it is usually underdetermined. Fortunately, for the low-rank problems

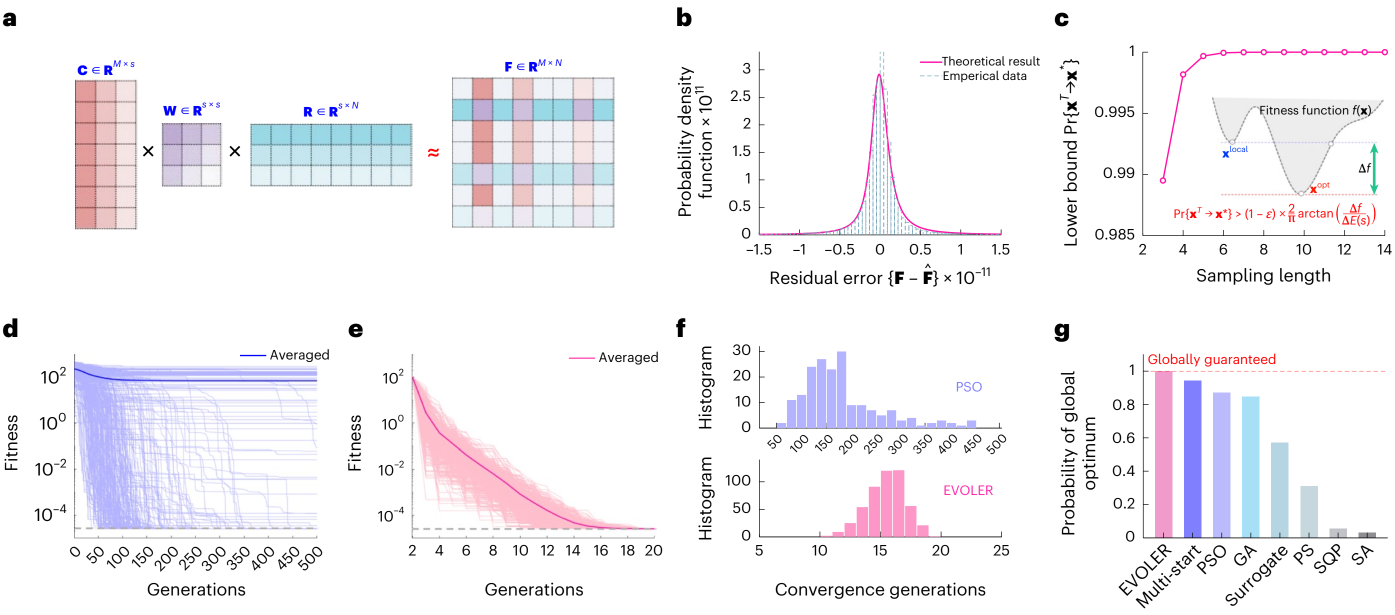

Fig. 2 | EVOLER enables globally guaranteed evolutionary computation. **a**, Reconstructing a low-rank representation $\hat{\mathbf{F}} \in \mathbb{R}^{M\times N}$ from two sampling matrices $\mathbf{C} \in \mathbb{R}^{M\times s}$ and $\mathbf{R} \in \mathbb{R}^{s\times N}$, as well as one weighting matrix W∈$\mathbb{R}^{s\times s}$. **b**, Probability density of the residual error of a representation space. We still consider a 2D Schwefel function in this example. **c**, Lower bound probability of finding the global solution (see Supplementary Section 2); Δf is distance between the global optimum and the next local optimum, whereas $\Delta E(s)$ is the upper-bound residual error related to a sampling length s. **d**, Convergence of classical evolutionary computation for a standard PSO with 50 particles. **e**, Convergence of EVOLER. **f**, EVOLER reduces the sample complexity by more than tenfold. **g**, EVOLER gains the global solution with probability 1 (500 random trials). By contrast, classical deterministic/heuristic global optimization methods all fail to do so (see ref. 50 for details on such methods). GA, genetic algorithm; PS, pattern search; SA, simulated annealing; SQP, sequential quadratic programming; multi-start, SQP with multiple starts.

(that is, rank ($\mathbf{F}$) $\simeq r \ll \min(M, N)$), we are able to apply randomized matrix approximation techniques[46–49] and hence approximate h via one special form: $\hat{\mathbf{F}} = h(\mathbf{C}, \mathbf{R}) \triangleq \mathbf{CWR}$ (Fig. 2a). Here, $\mathbf{W} \in \mathbb{R}^{s\times s}$ is one weighting matrix that can be directly obtained (see Methods for more details and the extension to d-dimensional problems).

Taking a 2D Schwefel function, for example, the learned low-rank representation is shown in Fig. 1aiii; the residual error between real function values $f(\mathbf{x})$ and reconstructed ones $\hat{f}(\mathbf{x})$ is given in Fig. 2b. Our new method would have two key advantages compared with classical neural networks. First, it demonstrates a promising extrapolation capacity and thus attains very high sample efficiency. To be specific, we can reconstruct the unknown problem space of MN samples by only using $[s(M+N) - s^2]$ samples, where $s \approx \mathcal{O}(r\log r) \ll \min(M, N)$. Second, it permits an analytical expression for the reconstructed space $\hat{\mathbf{F}}$, as well as the bounded residual error[46,47]. As such, it further enables a theoretical analysis on the probability of finding the global optimum (see Supplementary Section 2 for details).

Evolutionary computation

Relying on the low-rank representation learned from the first stage, we identify one attention subspace $\mathcal{A}$. As its name suggests, this small subspace $\mathcal{A}$ probably contains the global solution and hence claims the attention of the next evolutionary computation. The initial population $\mathbf{x}_k^0 (k = 1, 2, \cdots, K)$ is thus placed around $\mathcal{A}$, and then canonic or local-version PSO[18,19,21] can be used to exploit this small subspace (see Methods for more details). Note that other evolutionary computation methods[50] such as genetic algorithm and differential evolution can be combined with our method. Similarly, classical optimization methods such as the downhill simplex algorithm[51] will be also applicable.

After the total T generations, the final solution $\mathbf{x}^T$ is obtained. As noted, the identification of attention subspace offers a generic mechanism to reliably avoid the local optima (Supplementary Theorem 3). Incorporating the evolutionary computation or other methods, EVOLER converges to the global solution $\mathbf{x}^*$ with a probability approaching 1, as demonstrated in widespread rank-restricted problems (see approximately low-rank problems in Supplementary Fig. 1b–f and Fig. 6b, and strictly low-rank problems in Supplementary Figs. 1a,g,j and Fig. 5b). More importantly, EVOLER establishes the first theoretical result on the global optimum of the non-convex problems with common low-rank structures; that is, regardless of diverse geometry properties of problems, the lower bound probability of finding the global optimum is given by

$$\Pr\{\mathbf{x}^T \to \mathbf{x}^*\} > (1-\epsilon) \times \frac{2}{\pi}\arctan\left(\frac{\Delta f}{\Delta E(s)}\right).$$

Here ϵ is a small constant (see Supplementary Section 2). This lower bound probability relates to two important properties of the objective function. First, $\Delta f \triangleq |f(\mathbf{x}^*) - f(\mathbf{x}^{\mathrm{local}})|$ reflects the local difference (Fig. 2c), that is, the distance between the global optimum $f(\mathbf{x}^*)$ and the next local optimum $f(\mathbf{x}^{\mathrm{local}})$. Second, $\Delta E(s) \propto \left[\frac{1}{MN}\sum_{j=s+1}^{\min(M,N)} \sigma_j^2\right]^{1/2}$ is the error bound of $\| \mathbf{F} - \hat{\mathbf{F}}\|_F^2$ and reflects the global coherence, that is, the extent of the accuracy of representing the other values with few samples. Here, σ_j denotes the jth singular value of $\mathbf{F}$. We consider again the non-convex Schwefel function to illustrate our theoretical result. After obtaining the low-rank representation $\hat{\mathbf{F}}$ (Fig. 1aiii), we estimate $\Delta\hat{f} > 100$and $\Delta\hat{E}(s) \simeq 10^{-12} (s = 3, M = N = 100)$. In line with the estimated lower bound in Fig. 2c, EVOLER finds the global optimum almost surely, with probability approaching 1 (Fig. 2e). By contrast, classical optimization methods almost inevitably fall into the local optimums in this case (Fig. 2d,g).

Results

Representative benchmark problems

We examine EVOLER in 20 benchmark problems that are commonly used in evolutionary computation[30,45], including: unimodal, banana-valley, non-continuous, non-differentiable and multimodal functions, the

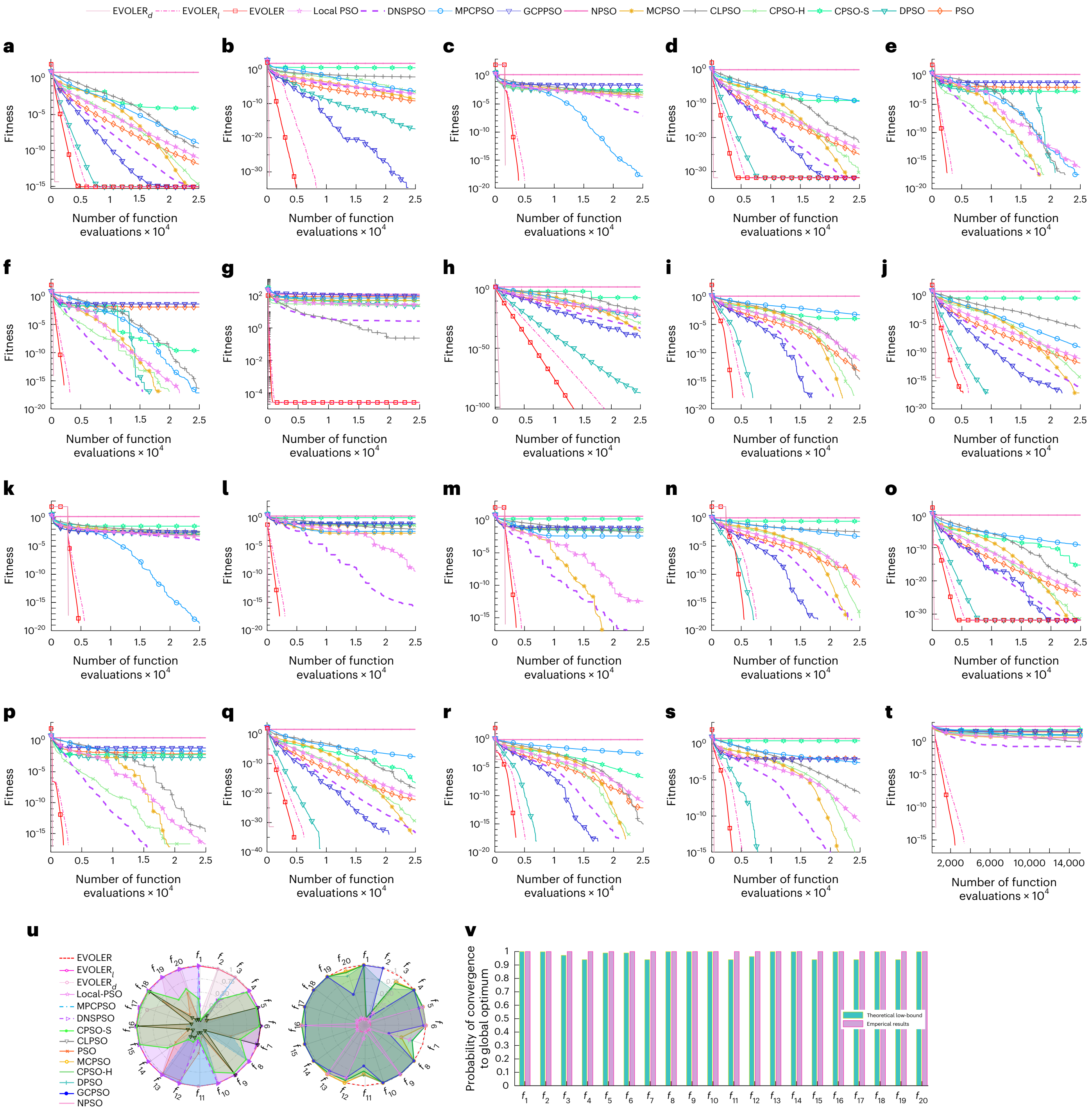

Fig. 3 | Finding the global optimum almost surely. **a**–**v**, Averaged convergence curves are derived via 500 independent trials with random initializations. In EVOLER, canonic PSO is used in the second stage, whereas the local PSO and downhill methods are used in EVOLER$_l$ and EVOLER$_d$, respectively. **a**, Ackley function (f_1). **b**, Rosenbrock function (f_2). **c**, Griewank function (f_3). **d**, Levy function (f_4). **e**, Non-continuous Rastrigin function (f_5). **f**, Rastrigin function (f_6). **g**, Schwefel function (f_7). **h**, Sphere function (f_8). **i**, Weierstrass function (f_9). **j**, Rotated Ackley function (f_{10}). **k**, Rotated Griewank function (f_{11}). **l**, Rotated non-continuous Rastrigin function (f_{12}). **m**, Rotated Rastrigin function (f_{13}). **n**, Rotated Weierstrass function (f_{14}). **o**, Shifted Levy function (f_{15}). **p**, Shifted Rastrigin function (f_{16}). **q**, Shifted Sphere function (f_{17}). **r**, Shifted Weierstrass function (f_{18}). **s**, Hybrid Composite function 2 (f_{19}). **t**, Hybrid composite function 3 (f_{20}). **u**, Empirical probabilities of different methods in finding the global optimum (maximum generation = 500). **v**, Theoretical low-bound probability of convergence to the global optimum versus the empirical results.

rotated versions, the shifted versions and the hybrid composite functions (see Supplementary Table 1 for details on these benchmark functions). Classical methods of various new mechanisms were evaluated, including: multiple populations, for example, MPCPSO[26], MCPSO[25] and CPSO-H/S[24]; improved learning strategies, for example, CLPSO[28] and NPSO[27]; refined position update, for example, DPSO[29]; parameter settings, for example, DNSPSO[30]; and population topology, for example, GCPSO[31] (see Methods for the parameter settings of EVOLER and Supplementary Table 2 for the parameter settings of classical methods). In the experiments, each method was evaluated for $N_t = 500$

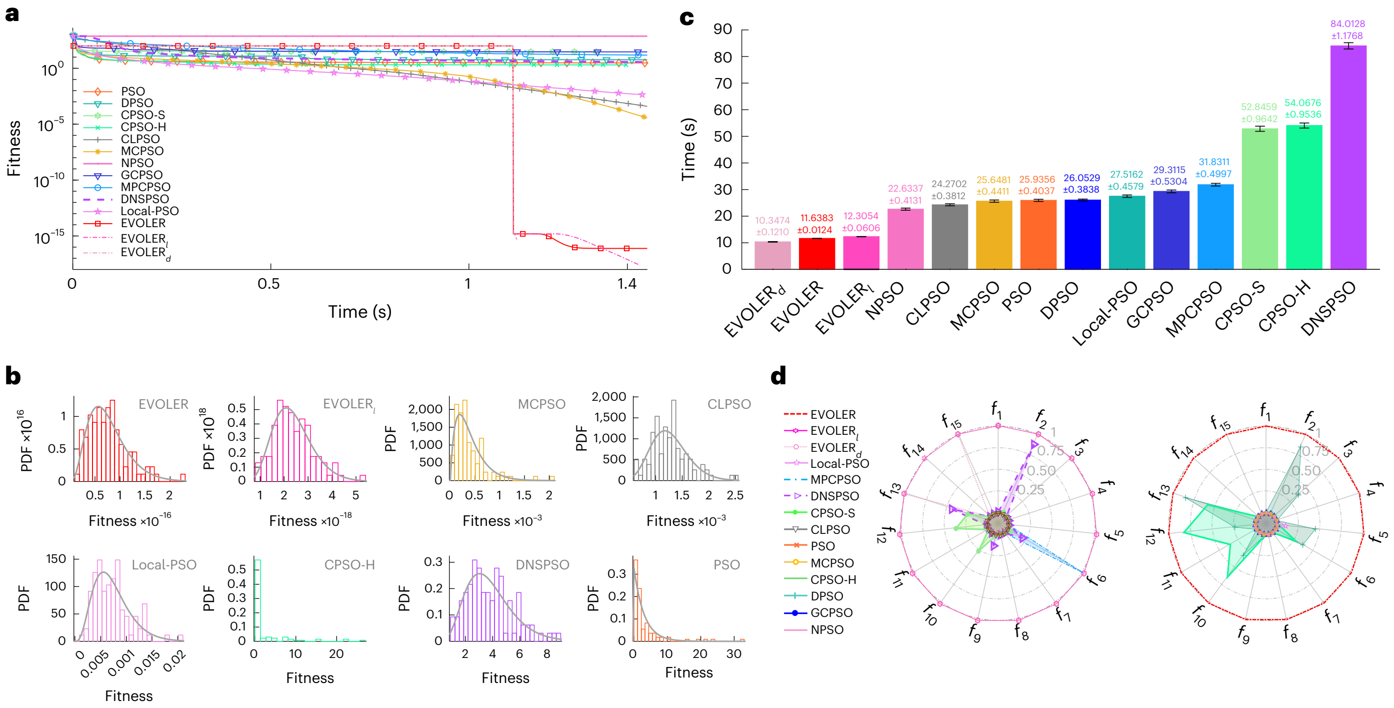

Fig. 4 | Performance in 30-dimensional benchmark problems. **a**, Averaged convergence performance of various evolutionary computation method (30-dimensional shifted Levy function, 100 independent trials). **b**, Probability density function (PDF) of the achieved fitness with equal time budgets (CPU time 1.45 s). **c**, CPU running time of various methods. The data are presented as mean values ± s.e.m. (sample size 20) (30-dimensional hybrid composition function 1, maximum generation = 900). **d**, Probability of convergence to the global optimum of various methods (500 independent trials with random initializations, maximum generation = 900; see the averaged convergence curves in Supplementary Fig. 5). f_1, hybrid composite function 1; f_2, hybrid composite function 2; f_3, Levy function; f_4, Rastrigin function; f_5, Rastrigin function with noise; f_6, rotated Griewank function; f_7, rotated Rastrigin function; f_8, Schwefel function; f_9, shifted composite function; f_{10}, shifted Levy function; f_{11}, shifted Rastrigin function; f_{12}, shifted sphere function; f_{13}, shifted Weierstrass function; f_{14}, sphere function; f_{15}, Weierstrass function.

independent trials with random restarts, so that $1 - 1/N_t \geq \Pr\{\mathbf{x}^T \to \mathbf{x}^*\}$. The averaged convergence curves are thus derived (see Supplementary Table 3 for the statistical results); the empirical probability of finding the global optimum was also obtained. As the number of function evaluations in each generation may be different for various methods, we focus on the convergence performance with regards to the number of function evaluations. For $d = 2$, we assume $K = 50$ and the experimental results are shown in Fig. 3; for $d = 30$, $K = 100$ and the results are shown in Fig. 4 and Supplementary Fig. 5.

As mentioned, one critical challenge of classical evolutionary computation methods is the uncertainty problem; that is, we can never know whether or not the returned solution is globally optimal. Apart from this, they are especially vulnerable to varying problem structures. Although some methods may find the global optimum in special cases (for example, MPCPSO in the Griewank function with the convex envelope), none of them can gain the good performance in all such benchmarks (Fig. 3a,t). By contrast, our method potentially overcomes the two pitfalls inherent to evolutionary computation.

From Fig. 3, EVOLER finds the global optimum with probability 1, regardless of the different benchmark problems of diverse geometry properties. This further validates our theoretical result. As the benchmark problems generally have low-rank structures (see Supplementary Fig. 1), when $s \approx \mathcal{O}\{r\log(r/\epsilon)\}$, we have $\Pr\{\mathbf{x}^T \to \mathbf{x}^*\} \to 1$. See Fig. 2c for the theoretical lower bound probability of the Schwefel function, and Fig. 3v for the theoretical lower bound probability of the convergence to the global optimum of the other benchmark functions, as well as the empirical results. As such, even for the challenging non-convex problems, EVOLER would gain the global optimum almost surely, whereas classical deterministic or heuristic methods fail to do so (Figs. 2g and 3u).

We also observe that EVOLER attains the best performance in all cases. In contrast to classical PSO methods explicitly or implicitly assuming specific geometry properties, in EVOLER a discrete problem space needs only to be rank-restricted. Note that this low-rank structure is easy to evaluate in practice (see the online rank estimation in Supplementary Section 1a and Supplementary Fig. 3) and, moreover, is ubiquitous in application. As a result, EVOLER substantially extends the applicability of evolutionary computation in diverse problems with common low-rank structures, providing great promise to real-world problems with unexplored properties. Moreover, with the sample-efficient reconstruction of the attention subspace, the sample/time complexity is substantially reduced. For example, the required samples of EVOLER are reduced by more than 30-fold compared with CLPSO in the case of Schwefel function (Fig. 3g). To even things up, we also take samples of the first stage into account.

Our method could be scalable to the more challenging 30-dimensional problems. Taking the shifted Levy function, for example, EVOLER almost surely converges to the near-global optimum within 1.45 s; by contrast, classical PSO methods fail to do so with equal time budgets (see the averaged convergence curves in Fig. 4a, and the statistical distribution of the achieved fitness of various methods in Fig. 4b). Similar results are obtained in the other 30-dimensional benchmark problems (see Supplementary Fig. 5 for the averaged convergence curves and Fig. 4d for the probability of convergence to the global optimum; 500 independent trials with random restarts). Provided the maximum generation of 900, the averaged running time of various methods is also obtained (Fig. 4c); 30-dimensional composite function, 20 trails, CPU = 4 GHz, RAM = 32 GB.

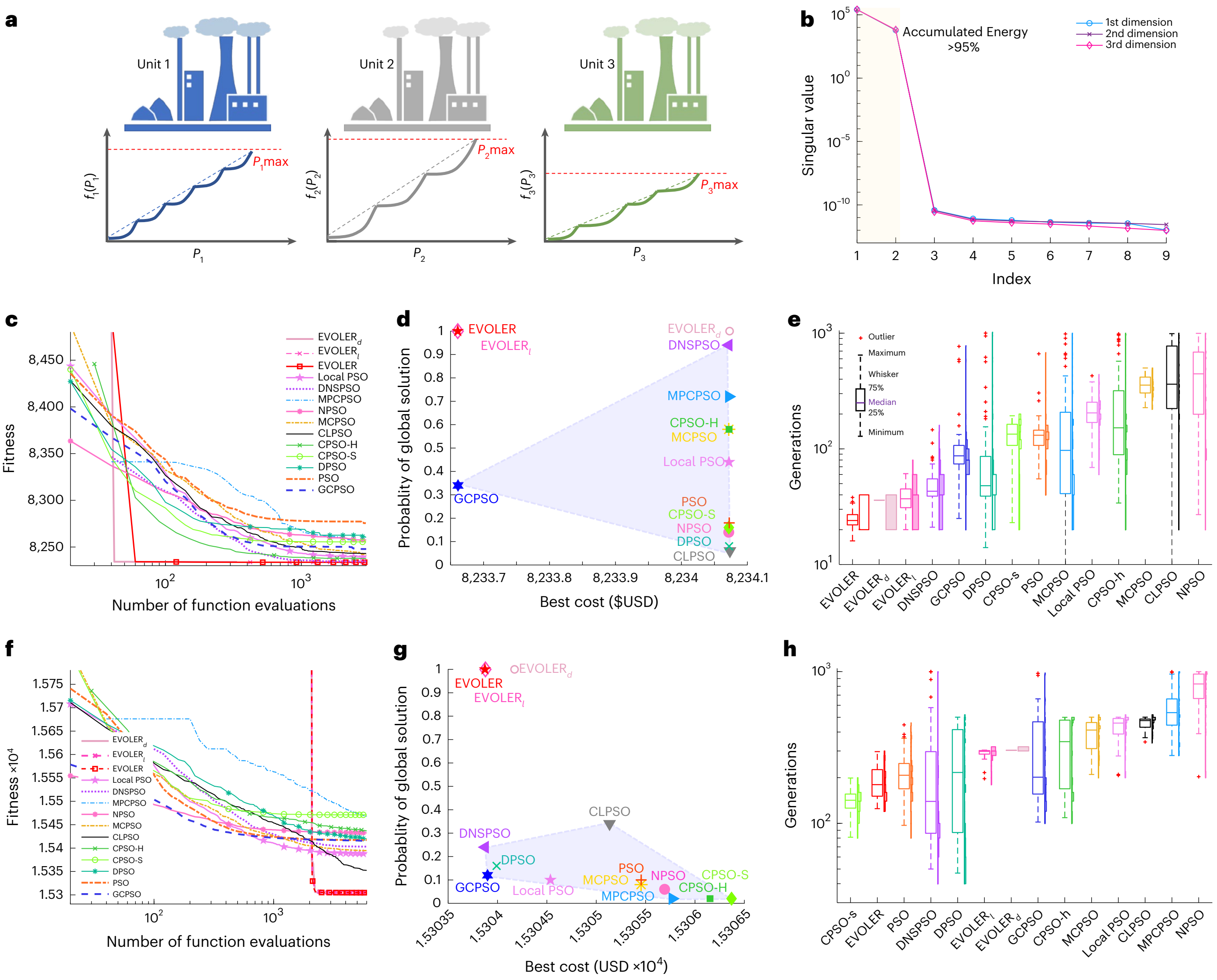

Fig. 5 | Optimal dispatch of power grid. a, Illustration of power dispatch with $d = 3$. **b**, Singular values of d-dimensional fitness ($d = 3$). **c**, Averaged results of various evolutionary computation methods ($d = 3$, $K = 20$, 50 independent trials). **d**, Achievable minimal costs of different methods and the probabilities of finding them. **e**, Required generations for convergence (sample size 50). **f**, Averaged results of various evolutionary computation methods ($d = 6$, $K = 20$, 50 independent trials). **g**, Achievable minimal costs of different methods and the probabilities of finding them. **h**, Required generations for convergence (sample size 50).

Dispatch of power grid

Economic dispatch of a power grid is a classical yet still challenging real-world problem, characterized by the intrinsic difficulties in global optimization, that is, non-smooth fitness with many local optima[12–14,52]. Despite the applications of evolutionary computation[14,52–54], this important problem still remains unsolved. On the one hand, it is not known whether the global solution could be found; on the other hand, the probability of finding it was rarely considered. Assume the power grid consists of d units; each of them has the maximum power $P_j^{\max}$ and the minimal power $P_j^{\min}$ so that the output power satisfies $P_j^{\min} \le P_j \le P_j^{\max}$ (see Fig. 5a). Meanwhile, the accumulated power of d units must be a constant, that is, $\sum_{j=1}^{d} P_j = P^{\text{total}}$. Each unit has a non-smooth cost function $f_j(P_j) = a_j + b_jP_j + c_jP_j^2 + |d_j \sin[e_j(P_j^{\min} - P_j)]|$;[$a_j, b_j, c_j, d_j, e_j, P_j^{\min}, P_j^{\max}$]is the parameters set of unit j, which are available in ref. 55. The goal is to optimize the output powers of d units, that is, $\{P_j\}_{j=1}^{d}$, so as to minimize the accumulated economic cost $f = \sum_{j=1}^{d} f_j(P_j)$, subject to the above inequality and equality constraints.

We first study the power dispatch with $d = 3$ by applying evolutionary computation methods. As we are interested in the trusted global optimization, each method was evaluated independently for 50 trials and the averaged fitness was obtained (Fig. 5c). From the singular values of 3D fitness, this real-world problem is strictly low ranked (Fig. 5b). We thus expect EVOLER would reliably find the global optimum. As seen, EVOLER indeed converges to a cheaper optimal cost than that of classical methods, after a few number of function evaluations. The minimal expected cost of classical methods is $USD 8,234.51 per 860 MWh (that is, DSNPSO), whereas that of EVOLER is $USD 8,233.66.

We then evaluate the best solutions of different methods and the probabilities of finding them. For classical methods, we estimate an upper-bound probability of successful results, that is, the sub-optimal results falling to $[f_j^{\text{best}}, (1+\kappa)f_j^{\text{best}}]$are treated as successful ones, where f_j^{best} is the best result achieved by method j and $\kappa = 10^{-3}$. From Fig. 5d, EVOLER finds the optimum solution with probability 1. The best cost of EVOLER is $USD 8,233.66 per 860 MWh. Among classical methods, the minimal cost was achieved by GCPSO, that is, $USD 8,233.66, yet

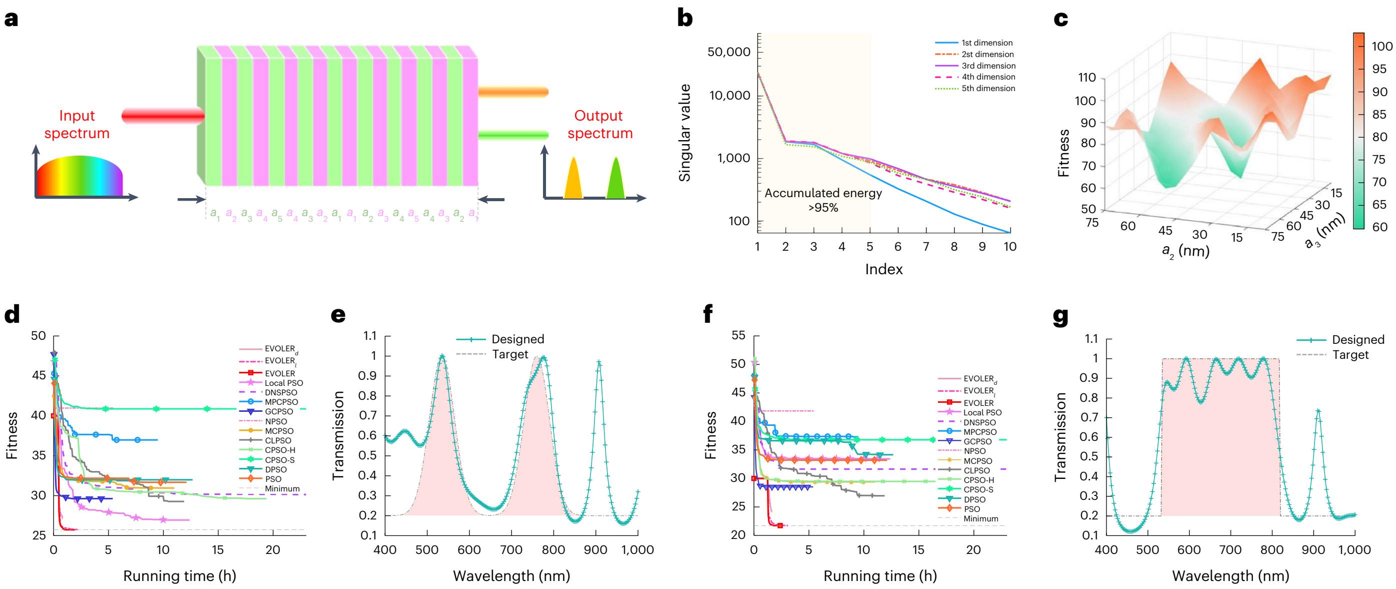

Fig. 6 | Inverse design of nanostructures. a, Illustration of multilayer thin-film structures. In the experiment, the thickness of $L = 18$ dielectric layers needs to be designed, subject to the targeted transmittance response. SiO_2, green; Si_3N_4, purple. **b**, Singular values of the problem space ($d = 5$). **c**, One 2D slice of d-dimensional fitness. **d**, Averaged optimization curves of various evolutionary computation methods; double bell-shaped transmittance bands; five independent trials. **e**, Designed optimal transmittance response versus the targeted response. **f**, Averaged optimization curves of various evolutionary computation methods; single transmittance band; five independent trials. **g**, Designed optimal transmittance response vs the targeted response.

with a small probability of 0.34. Combining the theoretical result on the low-rank problems, it is convinced that EVOLER would gain the global optimum almost surely. Furthermore, EVOLER accelerates the evolutionary computation by more than 15-fold compared with CLPSO (Fig. 5e).

We further demonstrate the advantage of EVOLER for $d = 6$ (see Supplementary Fig. 6 for the case of $d = 13$). From Fig. 5f, it converges stably to the minimal cost, that is, \$USD 15,303.872 per 1,260 MWh, whereas classical methods can only achieve \$USD 15,352.396 in expectation. By rough calculation, the economic cost will be reduced by 1.1×10^9 dollars per year, provided that the power consumption of the whole world in 2021 was about 28.32×10^{12} KWh. From Fig. 5g, EVOLER again finds the global solution with probability 1. By contrast, classical methods more likely end up with local solutions. Although classical methods have an intrinsic defect of problem dependency (for example, GCPSO for $d = 3$, DNSPSO for $d = 6$, CLPSO for $d = 13$), our method consistently gains the best performance (see Fig. 5d,g and Supplmentary Fig. 6).

Inverse design of nanophotonics devices

Inverse design of nanostructures is essential to nanophotonics[5,6], which has begun to reshape the landscape of structures available to nanophotonics[4], and has been applied in many fields[7,56,57] such as optical physics, biosensors, optical communications and so on. This computational task is very challenging, with elusive nanophotonics mechanisms and non-unique local solutions. At the present, there are two common strategies to handle it[4,8]: machine learning and evolutionary computation. The former adopts neural networks to model the complex relationship between nanophotonics structures and spectrum responses[58–60], yet lacking the critical interpretability. The latter directly optimizes nanophotonic devices[7,56,61], but inevitably acquires non-unique local solutions. Most importantly, both of them fail to gain the optimal structure, severely limiting the functionality of various wavelength-sensitive nanophotonics devices such as biosensors[5,56,62], optical switches[63] and so on. Despite the great importance, the optimal inverse design of nanophotonics devices remains an open problem.

Here we consider one representative inverse-design problem, multilayer thin-film structures[4–6,56,59], which aims to optimize the thickness of L dielectric layers (SiO_2 and Si_3N_4), subject to the target transmittance response. Note that our method can be generalized to similar problems, such as jointly optimizing the materials type or the number of periods[6]. In the experiment, $d = 5$ thickness parameters of $L = 18$ dielectric layers need to be adjusted for two desirable responses (Fig. 6a). In the first case, the desirable response involves two bell-shaped transmittance bands (Fig. 6e; 3 dB bandwidth = 85 nm, the two centre wavelengths are located at 535 nm and 760 nm). In the second case, the response contains one sharp transmittance band (Fig. 6g; bandwidth = 200 nm, the centre wavelength is located at 750 nm). As noted, this real-world problem is inherently rank-restricted; see the singular values (Fig. 6b) and 2D slice of d-dimensional fitness (Fig. 6c; a_2 versus a_3; $a_1 = 120$ nm, $a_4 = 80$ nm, $a_5 = 80$ nm; two bell-shaped bands).

Given the thickness parameters $\mathbf{a} \in \mathbb{R}^d$, the designed response was obtained by finite difference time domain (FDTD) methods[64]. The fitness is defined as mean square error between the targeted response $r_t(\lambda; \mathbf{a})$ and a designed response $r_d(\lambda; \mathbf{a})$; that is, $f(\mathbf{a}) = \sum_{i=0}^{I-1} |r_t(\lambda_i; \mathbf{a}) - r_d(\lambda_i; \mathbf{a})|^2$; $I = 201$, λ_i is the discrete wavelength of interest. Note that each run is time consuming, for example, ~10–20 h for one trial (maximum generation = 200, $K = 50$; CPU = 4 GHz, RAM = 32 GB).

To demonstrate the global optimality of EVOLER, we first consider one special case with an achievable optimal response, whereby it reliably gains the global solution (Supplementary Fig. 7). Various PSO methods were then evaluated in two designed cases, and the convergence curves were derived via five independent trials (Fig. 6d,f). In the first case, EVOLER consistently converges to the best result, acquiring a minimal mean square error of 24.7 with a probability of 1. By contrast, the best expected fitness of classical methods is only 26.95, that is, local PSO (see Supplementary Fig. 8 for their designed responses). Similar results are obtained in the second case (Fig. 6f and Supplementary Fig. 9). When it comes to realistic applications, for example, the wavelength-sensitive optical communications[6,7], the transmission efficiency and reliability are substantially enhanced, as the in-band desirable signal is maximized while the out-band interference signal is minimized.

Another important advantage of EVOLER is that it permits the highly efficient computation, which is very promising to the time-consuming inverse-design tasks. Recall that each sample was independently evaluated in the first stage, which can be implemented in parallel, thus remarkably reducing the time complexity. From Fig. 6d,f, the computational time is reduced by around tenfold compared with classical PSO; two computers are used in the first stage. That is to say, with a dramatically reduced complexity, EVOLER attains the global optimum with a theoretical guarantee. As such, it provides great potentials to the inverse design of nanophotonics structures, as well as the other complex real-world problems.

Discussion

Many important problems emerging in science and industry call for judicious data-driven and derivative-free optimization. Despite broad application prospects, current evolutionary computation methods depend on specific problem structures and fail to acquire globally guaranteed solutions, severely undermining the trust in their applications. Here we report a new framework of evolutionary computation aided by machine learning, EVOLER, which makes it possible to attain the theoretically guaranteed global optimization of complex non-convex problems. As demonstrated on 20 challenging benchmark problems, and two real-world problems with diverse properties yet common low-rank structures, it finds the global optimum almost surely, thus substantially extending applicability. Meanwhile, the number of samples can be remarkably reduced. As such, our method takes a leap forwards in theoretically guaranteed evolutionary computation, overcoming the uncertainty in data-driven black-box methods, and paving the way for solving complex real-world problems. Note that EVOLER also embodies two important biological characteristics that rarely emerge in classical evolutionary computation[65]: (1) neutrality or weak selection at an early stage to preserve the diversity; and (2) major transitions to suddenly discover the promising region of solutions.

According to the theoretical result, EVOLER would be generally applicable to d-dimensional problems with low-rank structures. The complexity of our method is $\mathcal{O}(dr^{d-1}N + TK)$ (see Methods and Supplementary Section 3). Our method would be less attractive for problems with high d (for example, $d > 100$) or without a low-rank structure. There are two remedies for the potential limitations when tackling high-dimensional problems. First, various dimension reduction techniques are readily applicable. Taking the challenging aerodynamic design problem (for example, ref. 16), a set of orthogonal basis functions was first identified; the high-dimensional problem of $d > 20$ was then transformed to optimize the coefficients of the $d \le 10$ principle orthogonal basis. Second, as a future direction, the incorporation of the hierarchy concept (for example, the hierarchy reconstruction of d-dimensional space; see Supplementary Fig. 4) could substantially reduce the complexity. For example, assume the number of blocks to be $L \le \lfloor\sqrt{d}\rfloor$, the complexity of our method will be $\mathcal{O}\{dr^{(d/L-1)}N + TK\}$. Thus, the complexity in solving high-dimensional problems could be effectively reduced (see Supplementary Fig. 5 for the challenging benchmark problems of $d = 30$, and Supplementary Fig. 6 for the power-dispatch problem of $d = 13$). Although the wide applicability of EVOLER has been demonstrated, another open question is whether it is possible to further relax the requirements for unknown problems, which may deserve further study.

Methods

Evolutionary computation with PSO

Particle swarm optimization is inspired by collective intelligence in nature, for example, birds flocking or fish schooling[17–19]. In canonical PSO[14,18,21], K particles (or agents) cooperatively search the d-dimensional problem space and update their positions (or solutions) $\mathbf{x}_k^{t+1} \in \mathbb{R}^d$, that is,

$$\begin{aligned}\mathbf{v}_k^{t+1} &= w\mathbf{v}_k^t + c_1\mathbf{u}_1 \circ \left(\mathbf{g}_{\mathrm{pbest},k}^t - \mathbf{x}_k^t\right) + c_2\mathbf{u}_2 \circ \left(\mathbf{g}_{\mathrm{gbest}}^t - \mathbf{x}_k^t\right),\\ \mathbf{x}_k^{t+1} &= \mathbf{v}_k^t + \mathbf{x}_k^t.\end{aligned} \tag{1}$$

As seen, updating the particle velocity $\mathbf{v}_k^t$ involves three parts. The first part is the inertia term—the dependence of the particle's velocity on time t, that is, $\mathbf{v}_k^t(k = 0, 1, 2, \cdots, K-1)$; the inertia weight is $w \in \mathbb{R}^+$. The second part is the cognition item—the influence from the particle's own experience so far; $\mathbf{g}_{\mathrm{pbest},k}^t = \arg\min_t f(\mathbf{x}_k^t) \in \mathbb{R}^d$ is the optimal position of the kth particle and $f(\mathbf{x}) : \mathbb{R}^d \mapsto \mathbb{R}$ is the fitness function, $c_1 \in \mathbb{R}^+$ is a cognition learning factor, $\mathbf{u}_1 \in \mathbb{R}^d$ is a Gaussian variable and $\circ$ is the Hadamard (element-wise) product. The third term is the social part—the influence from the experience of the whole population; $\mathbf{g}_{\mathrm{gbest}}^t = \arg\min_{k,t} f(\mathbf{x}_k^t) \in \mathbb{R}^d$ is the optimal position of K particles; $c_2 \in \mathbb{R}^+$ is the social learning factor and $\mathbf{u}_2 \in \mathbb{R}^d$ is a Gaussian variable.

The above canonic PSO is referred to as the global version PSO, as the particles will track the global best $\mathbf{g}_{\mathrm{gbest}}^t$ of the whole population. There also exists another local-version PSO, whereby the particles only track the best position of its local neighbourhood, which is denoted as $\mathbf{g}_{\mathrm{lbest}}^t$. The particle positions (or solutions) will then be updated via:

$$\begin{aligned}\mathbf{v}_k^{t+1} &= w\mathbf{v}_k^t + c_1\mathbf{u}_1 \circ \left(\mathbf{g}_{\mathrm{pbest},k}^t - \mathbf{x}_k^t\right) + c_2\mathbf{u}_2 \circ \left(\mathbf{g}_{\mathrm{lbest}}^t - \mathbf{x}_k^t\right),\\ \mathbf{x}_k^{t+1} &= \mathbf{v}_k^t + \mathbf{x}_k^t.\end{aligned}$$

Exploring a global problem space

Taking $d = 2$ for example, the original problem space—which is composed of the fitness values on a discrete grid $\mathbf{X} \in \mathbb{R}^{M\times N}$ of continuous variables $\mathbf{x} \in \mathbb{R}^2$—is structured to one matrix $\mathbf{F} = \{f(\mathbf{X}_{m,n})\}$, where $m = 1, 2, \ldots, M$, and $n = 1, 2, \ldots, N$; M and N denote the grid sizes. To explore this unknown problem space $\mathbf{F}$, we randomly sample s columns and rows, thus obtaining two sub-matrices:

$$\begin{aligned}\mathbf{C} &= \mathbf{F}(:, \mathcal{C}) \in \mathbb{R}^{M\times s}, \quad \mathcal{C} \subset \{1, 2, \cdots, N\};\\ \mathbf{R} &= \mathbf{F}(\mathcal{R}, :) \in \mathbb{R}^{s\times N}, \quad \mathcal{R} \subset \{1, 2, \cdots, M\}.\end{aligned}$$

Here, $\mathcal{C}$ and $\mathcal{R}$ are two indexing sets, with the cardinality of $|\mathcal{C}| = |\mathcal{R}| = s \ll \min\{M, N\}$ (see Supplementary Section 1 for the probability densities of random sampling).

Reconstructing a low-rank representation

Based on the sub-matrices **C** and **R**, a low-rank representation space is reconstructed (see Fig. 2a).

$$\hat{\mathbf{F}} = \mathbf{CWR} \in \mathbb{R}^{M\times N}, \tag{2}$$

$$\mathbf{W} = \arg\min_{\mathbf{W}\in\mathbb{R}^{s\times s}} ||\mathbf{F} - \mathbf{CWR}||_F^2 \simeq [\mathbf{F}(\mathcal{R}, \mathcal{C})]^{\dagger}. \tag{3}$$

Here, a small weighting matrix $\mathbf{W} \in \mathbb{R}^{s\times s}$ is directly computed by minimizing the residual error[46–48,66]; $\mathbf{Y}^{\dagger}$ is the Moore–Penrose pseudo-inverse of $\mathbf{Y}$. When the original space $\mathbf{F} \in \mathbb{R}^{M\times N}$ is rank-restricted, that is, $\mathrm{rank}(\mathbf{F}) = r \ll \min(M, N)$ and the sampling length satisfies $s \approx \mathcal{O}\{r\log(r/\epsilon)\}$, then the residual error is upper bounded with probability $1 - \epsilon$ (refs. 46,47), that is,

$$\| \mathbf{F} - \hat{\mathbf{F}}\|_F^2 \le (1 + \eta) \sum_{j=s+1}^{\min(M,N)} \sigma_j^2,$$

where $\| \mathbf{X}\|_F^2$ is the Frobenius norm of $\mathbf{X}$, σ_j is the jth singular value of $\mathbf{F}$ and η is a constant related to s. For convenience, we define the upper bound of the averaged residual error of each element in $\hat{\mathbf{F}}$ as $\Delta E(s) \triangleq \left[\frac{1}{MN} \times (1+\eta) \sum_{j=s+1}^{\min(M,N)} \sigma_j^2\right]^{1/2}$ so that we have $\frac{1}{MN} \parallel \mathbf{F} - \hat{\mathbf{F}} \|_F^2 \leq \Delta E^2(s)$.

Identifying an attention subspace

Provided a low-rank representation space $\hat{\mathbf{F}} \in \mathbb{R}^{M\times N}$, an initial solution $\hat{\mathbf{x}}^{\mathrm{opt}} \in \mathbf{X}$ is then identified:

$$\hat{\mathbf{x}}^{\mathrm{opt}} = \arg\min_{\mathbf{x}\in\mathbf{X}} \hat{f}(\mathbf{x}). \qquad (4)$$

We define an attention subspace as $\mathcal{A} \triangleq \{\mathbf{x} \in \mathbf{X}; |\mathbf{x} - \hat{\mathbf{x}}^{\mathrm{opt}}| < \delta|\}$, where δ is the subspace radius. As discussed, the learned low-rank representation $\hat{\mathbf{F}}$ is highly accurate and thus the attention subspace $\mathcal{A}$ probably contains the global optimum $\mathbf{x}^{\mathrm{opt}} = \arg\min_{\mathbf{x}\in\mathbf{X}} f(\mathbf{x})$ of a discrete space $\mathbf{F}$ (Supplementary Section 2). When N and M are sufficiently large, $\mathbf{X}$ has a very high grid resolution and hence subspace $\mathcal{A}$ would contain the global solution $\mathbf{x}^* = \arg\min_{\mathbf{x}\in\mathbb{R}^2} f(\mathbf{x})$ (Fig. 1aiii). We thus initialize K particles via one Gaussian distribution:

$$\mathbf{x}_k^0 \sim \mathcal{N}\left(\hat{\mathbf{x}}^{\mathrm{opt}}, \delta\right) \in \mathcal{A}, \delta = \mathcal{O}\{\max(\delta_1, \delta_2)\}, \qquad (5)$$

where $\delta_1 = (x_1^{\max} - x_1^{\min})/M$ and $\delta_2 = (x_2^{\max} - x_2^{\min})/N$ denote the grid resolution.

Reconstructing low-rank representation for $d > 2$

Similar to the case in which $d = 2$, the high-dimensional ($d > 2$) space can be reconstructed via few structured samples. For clarity, a high-dimensional problem space is denoted by the tensor $\mathcal{F} \in \mathbb{R}^{N\times N\times\cdots\times N}$, and we assume each dimensionality of the tensor $\mathcal{F}$ is equal to N.

Taking the three-dimensional representation space reconstruction as an example, we first apply a structured random sampling and abstract three small tensors[67–69] $\mathcal{C}_1 \in \mathbb{R}^{N\times s_2\times s_3}$, $\mathcal{C}_2 \in \mathbb{R}^{s_1\times N\times s_3}$, $\mathcal{C}_3 \in \mathbb{R}^{s_1\times s_2\times N}$ and one intersection tensor $\mathcal{R} \in \mathbb{R}^{s_1\times s_2\times s_3}$ ($s_1, s_2, s_3 \ll N$) (Supplementary Fig. 14). A representation tensor is then reconstructed via:

$$\hat{\mathcal{F}} \simeq \mathcal{R}_{\times 1}(\mathbf{C}_1\mathbf{U}_1^{\dagger})_{\times 2}(\mathbf{C}_2\mathbf{U}_2^{\dagger})_{\times 3}(\mathbf{C}_3\mathbf{U}_3^{\dagger}), \qquad (6)$$

where $\mathbf{C}_1 \in \mathbb{R}^{N\times(s_2 s_3)}$ was obtained by expanding the tensor $\mathcal{C}_1$ to one 2D matrix along the first dimension; $\mathbf{U}_1 \in \mathbb{R}^{s_1\times(s_2 s_3)}$ was obtained by expanding a tensor $\mathcal{R}$ to 2D matrix along the first dimension, $\mathbf{U}_1^{\dagger}$ denotes the pseudo-inverse of $\mathbf{U}_1$ and $\mathcal{R}_{\times 1}(\mathbf{C}_1\mathbf{U}_1^{\dagger}) \in \mathbb{R}^{N\times s_2\times s_3}$ denotes the one-mode product[70].

When it comes to the arbitrary d-dimensional tensor, the above sampling and reconstruction procedure can be directly applicable. In this case, the required samples for a structured exploration is $\mathcal{O}\left\{ds^{d-1}N\right\}, s = \lfloor\hat{r} \times \log(\hat{r})\rfloor; r$ is the rank of the tensor. To reduce the samples, we may apply a multiscale exploration, that is, we first reconstruct one down-sampling $\left(\frac{N}{2} \times \frac{N}{2} \times \ldots \times \frac{N}{2}\right)$ tensor and then interpolate it to form the other $(N \times N \times \ldots \times N)$ tensor (also see Supplementary Figs. 5, 6 and 15 for practical methods to further reduce the samples).

Parameter settings

EVOLER. The hyperparameters in EVOLER can be classified into two subsets. In the first stage, there are two parameters: the discrete length N (for simplicity, we assume $N_1 = N_2 = \ldots = N_d = N$) and the sampling length s. In practice, the discrete length N is directly set to ~20–120. The rank of a discrete problem space is then estimated (see Supplementary Section 1a for more details), which is denoted as $\hat{r}$. On this basis, the sampling length is configured to be $s = \lfloor\hat{r} \times \log(\hat{r})\rfloor$ (in practice, we find $s = \hat{r}$ is also applicable). In the second stage, there are three parameters in the PSO method: the inertia weight w, the cognition learning factor c_1 and the social learning factor c_2. In the tth generation ($0 \leq t \leq T$), we have $w = 0.3$, $c_1 = 2 - 1.5 \times t/T$ and $c_2 = 1.5 + 0.5 \times t/T$; whereby the maximum generation is T. In each independent trial, K particles are randomly initialized, and the sampling indexes are also randomly determined. For the high-dimensional problems (for example, $d = 30$), the hierarchy exploration is applied (see Supplementary Fig. 15). In this case, the discrete length N is ~3–6, and the iterative hierarchy exploration was terminated when the change on solutions was smaller than 10^{-3}.

Classical methods. In the experiments, the parameter settings of classical PSO methods remain exactly the same as the related references. For these comparative methods, their parameter settings are summarized in Supplementary Table 2.

Data availability

The representative benchmarks for testing evolutionary computation methods are partly available at http://www.sfu.ca/~ssurjano/optimization.html, and partly presented in refs. 30,45. The power-dispatch dataset, that is, the Economic Load Dispatch Test Systems Repository[55], is available at https://al-roomi.org/economic-dispatch. The dataset of inverse design of multilayer thin-film structures is available at Gitbub (https://github.com/BinLi-BUPT/EVOLER.git)[71], which was generated by the FDTD method. Source Data are provided with this paper.

Code availability

Matlab code of EVOLER is available at https://github.com/BinLi-BUPT/EVOLER.git (ref. 71).

References

1. Weiel, M. et al. Dynamic particle swarm optimization of biomolecular simulation parameters with flexible objective functions. *Nat. Mach. Intell.* **3**, 727–734 (2021).
2. Morris, G. M. et al. Automated docking using a Lamarckian genetic algorithm and an empirical binding free energy function. *J. Comput. Chem.* **19**, 1639–1662 (1998).
3. Quignot, C. et al. Interevdock2: an expanded server for protein docking using evolutionary and biological information from homology models and multimeric inputs. *Nucleic Acids Res.* **46**, W408–W416 (2018).
4. Molesky, S. et al. Inverse design in nanophotonics. *Nat. Photon.* **12**, 659–670 (2018).
5. Sreekanth, K. V. et al. Biosensing with the singular phase of an ultrathin metal-dielectric nanophotonic cavity. *Nat. Commun.* **9**, 1–8 (2018).
6. Qiu, C. et al. Simultaneous inverse design continuous and discrete parameters of nanophotonic structures via back-propagation inverse neural network. *Optics Commun.* **483**, 126641 (2021).
7. Fischer, B. et al. Autonomous on-chip interferometry for reconfigurable optical waveform generation. *Optica* **8**, 1268–1276 (2021).
8. Genty, G. et al. Machine learning and applications in ultrafast photonics. *Nat. Photon.* **15**, 91–101 (2021).
9. Andral, U. et al. Fiber laser mode locked through an evolutionary algorithm. *Optica* **2**, 275–278 (2015).
10. Inbarani, H. H., Azar, A. T. & Jothi, G. Supervised hybrid feature selection based on PSO and rough sets for medical diagnosis. *Comput. Methods Prog. Biomed.* **113**, 175–185 (2014).
11. Wild, S. M., Sarich, J. & and Schunck, N. Derivative-free optimization for parameter estimation in computational nuclear physics. *J. Phys. G* **42**, 034031 (2015).
12. Park, J.-B., Lee, K.-S., Shin, J.-R. & Lee, K. Y. A particle swarm optimization for economic dispatch with nonsmooth cost functions. *IEEE Trans. Power Syst.* **20**, 34–42 (2005).
13. Park, J. B., Jeong, Y. W., Shin, J. R. & Lee, K. Y. An improved particle swarm optimization for nonconvex economic dispatch problems. *IEEE Trans. Power Syst.* **25**, 156–166 (2010).

14. Del Valle, Y., Venayagamoorthy, G. K., Mohagheghi, S., Hernandez, J.-C. & Harley, R. G. Particle swarm optimization: basic concepts, variants and applications in power systems. *IEEE Trans. Evolut. Comput.* **12**, 171–195 (2008).
15. Skinner, S. N. & Zare-Behtash, H. State-of-the-art in aerodynamic shape optimisation methods. *Appl. Soft Comput.* **62**, 933–962 (2018).
16. Yasong, Q., Junqiang, B., Nan, L. & Chen, W. Global aerodynamic design optimization based on data dimensionality reduction. *Chinese J. Aeronaut.* **31**, 643–659 (2018).
17. Kennedy, J. & Eberhart, R. Particle swarm optimization. In *Proc. International Conference on Neural Networks (ICNN'95)* Vol. 4, 1942–1948 (IEEE, 1995).
18. Shi, Y. & Eberhart, R. C. A modified particle swarm optimizer. In *Proc. IEEE ICEC Conference* (IEEE, 1998).
19. Shi, Y. & Eberhart, R. C. Empirical study of particle swarm optimization. In *Proc. 1999 Congress on Evolutionary Computation (CEC'99)* Vol. 3, 1945–1950 (IEEE, 1999).
20. Mendes, R., Kennedy, J. & Neves, J. The fully informed particle swarm: simpler, maybe better. *IEEE Trans. Evolut. Comput.* **8**, 204–210 (2004).
21. Wang, D., Tan, D. & Liu, L. Particle swarm optimization algorithm: an overview. *Soft Comput.* **22**, 387–408 (2018).
22. Shi, Y. et al. Particle swarm optimization: developments, applications and resources. In *Proc. 2001 Congress on Evolutionary Computation (CEC)* Vol. 1, 81–86 (IEEE, 2001).
23. Poli, R., Kennedy, J. & Blackwell, T. Particle swarm optimization. *Swarm Intell.* **1**, 33–57 (2007).
24. Van den Bergh, F. & Engelbrecht, A. P. A cooperative approach to particle swarm optimization. *IEEE Trans. Evolut. Comput.* **8**, 225–239 (2004).
25. Niu, B., Zhu, Y., He, X. & Wu, H. MCPSO: A multi-swarm cooperative particle swarm optimizer. *Appl. Mathematics Comput.* **185**, 1050–1062 (2007).
26. Li, W., Meng, X., Huang, Y. & Fu, Z.-H. Multipopulation cooperative particle swarm optimization with a mixed mutation strategy. *Inf. Sci.* **529**, 179–196 (2020).
27. Yang, C. & Simon, D. A new particle swarm optimization technique. In *18th International Conference on Systems Engineering (ICSEng'05)* 164–169 (IEEE, 2005).
28. Liang, J. J., Qin, A. K., Suganthan, P. N. & Baskar, S. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE Trans. Evolut. Comput.* **10**, 281–295 (2006).
29. Xie, X.-F., Zhang, W.-J. & Yang, Z.-L. Dissipative particle swarm optimization. In *Proc. 2002 Congress on Evolutionary Computation (CEC'02)* Vol. 2, 1456–1461 (IEEE, 2002).
30. Zeng, N. et al. A dynamic neighborhood-based switching particle swarm optimization algorithm. *IEEE Trans. Cybernetics* **52**, 9290–9301 (2022).
31. Peer, E. S., van den Bergh, F. & Engelbrecht, A. P. Using neighbourhoods with the guaranteed convergence PSO. In *Proc. 2003 IEEE Swarm Intelligence Symposium (SIS'03)* 235–242 (IEEE, 2003).
32. Blackwell, T. & Kennedy, J. Impact of communication topology in particle swarm optimization. *IEEE Trans. Evolut. Comput.* **23**, 689–702 (2018).
33. Kennedy, J. The behavior of particles. In *International Conference on Evolutionary Programming* 579–589 (Springer, 1998).
34. Ozcan, E. & Mohan, C. K. Analysis of a simple particle swarm optimization system. *Intell. Eng. Syst. Artificial Neural Netw.* **8**, 253–258 (1998).
35. Clerc, M. & Kennedy, J. The particle swarm-explosion, stability, and convergence in a multidimensional complex space. *IEEE Trans. Evolut. Comput.* **6**, 58–73 (2002).
36. Van Den Bergh, F. et al. An analysis of particle swarm optimizers. PhD thesis, Univ. Pretoria (2007).
37. Kadirkamanathan, V., Selvarajah, K. & Fleming, P. J. Stability analysis of the particle dynamics in particle swarm optimizer. *IEEE Trans. Evolut. Comput.* **10**, 245–255 (2006).
38. Van den Bergh, F. & Engelbrecht, A. P. A study of particle swarm optimization particle trajectories. *Inf. Sci.* **176**, 937–971 (2006).
39. Fernandez-Martinez, J. L. & Garcia-Gonzalo, E. Stochastic stability analysis of the linear continuous and discrete PSO models. *IEEE Trans. Evolut. Comput.* **15**, 405–423 (2010).
40. Van den Bergh, F. & Engelbrecht, A. P. A convergence proof for the particle swarm optimiser. *Fundam. Inform.* **105**, 341–374 (2010).
41. Holland, J. H. Genetic algorithms and the optimal allocation of trials. *SIAM J. Comput.* **2**, 88–105 (1973).
42. Wolpert, D. H. & Macready, W. G. No free lunch theorems for optimization. *IEEE Trans. Evolut. Compu.* **1**, 67–82 (1997).
43. Garcia-Martinez, C., Rodriguez, F. J. & Lozano, M. Arbitrary function optimisation with metaheuristics. *Soft Comput.* **16**, 2115–2133 (2012).
44. Adam, S. P., Alexandropoulos, S.-A. N., Pardalos, P. M. & Vrahatis, M. N. in *Approximation and Optimization* 57–82 (Springer 2019).
45. Suganthan, P. N. et al. Problem definitions and evaluation criteria for the CEC2005 special session on real-parameter optimization. In *Proc. IEEE Congr. Evol. Comput. (CEC)* 1–5 (2005).
46. Drineas, P., Mahoney, M. W. & Muthukrishnan, S. Relative-error *CUR* matrix decompositions. *SIAM J. Matrix Anal. Appl.* **30**, 07070471X (2008).
47. Wang, S. & Zhang, Z. Improving *CUR* matrix decomposition and the Nyström approximation via adaptive sampling. *J. Mach. Learning Res.* **14**, 2729–2769 (2013).
48. Li, B. et al. Random sketch learning for deep neural networks in edge computing. *Nat. Comput. Sci.* **1**, 221–228 (2021).
49. Liu, H., Wei, Z., Zhang, H., Li, B. & Zhao, C. Tiny machine learning (TINY-ML) for efficient channel estimation and signal detection. *IEEE Trans. Vehicular Technol.* **71**, 6795–6800 (2022).
50. Younis, A. & Dong, Z. Trends, features, and tests of common and recently introduced global optimization methods. *Eng. Optimization* **42**, 691–718 (2010).
51. Nelder, J. A. & Mead, R. A simplex method for function minimization. *Comput. J.* **7**, 308–313 (1965).
52. Yang, H.-T., Yang, P.-C. & Huang, C.-L. Evolutionary programming based economic dispatch for units with non-smooth fuel cost functions. *IEEE Trans. Power Syst.* **11**, 112–118 (1996).
53. Singh, R. P., Mukherjee, V. & Ghoshal, S. Optimal reactive power dispatch by particle swarm optimization with an aging leader and challengers. *Appl. Soft Comput.* **29**, 298–309 (2015).
54. Xu, S., Xiong, G., Mohamed, A. W. & Bouchekara, H. R. Forgetting velocity based improved comprehensive learning particle swarm optimization for non-convex economic dispatch problems with valve-point effects and multi-fuel options. *Energy* **256**, 124511 (2022).
55. Al-Roomi, A. R. Economic Load Dispatch Test Systems Repository. *Electric Power Systems Analysis & Nature-Inspired Optimization Algorithms* https://www.al-roomi.org/economic-dispatch (2016).
56. Rodrigo, D. et al. Mid-infrared plasmonic biosensing with graphene. *Science* **349**, 165–168 (2015).
57. Lin, Z., Liang, X., Loncar, M., Johnson, S. G. & Rodriguez, A. W. Cavity-enhanced second-harmonic generation via nonlinear-overlap optimization. *Optica* **3**, 233 (2016).
58. Liu, Z., Zhu, D., Raju, L. & Cai, W. Tackling photonic inverse design with machine learning. *Adv. Sci.* **8**, 2002923 (2021).
59. Sheverdin, A., Monticone, F. & Valagiannopoulos, C. Photonic inverse design with neural networks: the case of invisibility in the visible. *Phys. Rev. Appl.* **14**, 024054 (2020).

60. Zhang, T. et al. Machine learning and evolutionary algorithm studies of graphene metamaterials for optimized plasmon-induced transparency. *Opt. Express* **28**, 18899–18916 (2020).
61. Yu, Z., Cui, H. & Sun, X. Genetically optimized on-chip wideband ultracompact reflectors and Fabry–Perot cavities. *Photon. Res.* **5**, B15–B19 (2017).
62. Zhang, T. et al. Plasmon induced absorption in a graphene-based nanoribbon waveguide system and its applications in logic gate and sensor. *J. Phys. D* **51**, 055103 (2018).
63. Miller, K. J., Hallman, K. A., Haglund, R. F. & Weiss, S. M. Silicon waveguide optical switch with embedded phase change material. *Opt. Express* **25**, 26527–26536 (2017).
64. Lumerical, F. *FDTD Solutions 6.5* (Ansys, 2023); http://www.lumerical.com/tcad-products/fdtd
65. Miikkulainen, R. & Forrest, S. A biological perspective on evolutionary computation. *Nat. Mach. Intell.* **3**, 9–15 (2021).
66. Li, B., Wang, S., Zhang, J., Cao, X. & Zhao, C. Ultra-fast accurate AoA estimation via automotive massive-MIMO radar. *IEEE Trans. Vehicular Technol.* **71**, 1172–1186 (2021).
67. Mahoney, M. W., Maggioni, M. & Drineas, P. Tensor-*CUR* decompositions for tensor-based data. *SIAM J. Matrix Anal. Appl.* **30**, 957–987 (2008).
68. Cai, H., Hamm, K., Huang, L. & Needell, D. Mode-wise tensor decompositions: multi-dimensional generalizations of CUR decompositions. *JMLR* **22**, 8321–8356 (2021).
69. Song, Z., Woodruff, D. P. & Zhong, P. Relative error tensor low rank approximation. In *Proc. 30th Annual ACM-SIAM Symposium on Discrete Algorithms* 2772–2789 (SIAM, 2019).
70. Kolda, T. G. & Bader, B. W. Tensor decompositions and applications. *SIAM Rev.* **51**, 455–500 (2009).
71. Li, B., Wei, Z. & Wu, J. *Machine Learning-enabled Globally Guaranteed Evolutionary Computation [Source Code]* (Zenodo, 2023); https://doi.org/10.5281/zenodo.7688653

Acknowledgements

This work was supported by National Natural Science Foundation of China under grant nos. U1805262, 62088101, 61971050 and 62171055.

Author contributions

J.Z. supervised the study. B.L. conceived the idea. B.L. and Z.P.W. designed and implemented source code. B.L., T.Z., S.Y. and Z.P.W. performed the experiments on inverse design of nanophotonics structures. All of the authors interpreted the findings and wrote the paper.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s42256-023-00642-4.

Correspondence and requests for materials should be addressed to Bin Li or Jun Zhang.

Peer review information *Nature Machine Intelligence* thanks Jing Liang and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.