Numerical Algorithms (2020) 85:145–169
https://doi.org/10.1007/s11075-019-00807-6

ORIGINAL PAPER

Double descent and intermittent color diffusion for landscape exploration

Luca Dieci[1] · **Manuela Manetta**[2] · **Haomin Zhou**[1]

Received: 7 March 2019 / Accepted: 5 September 2019 / Published online: 6 November 2019

Abstract
In this work, we present a method to explore the landscape of a smooth potential in the search of global minimizers, combining a double-descent technique and a basin-escaping technique based on intermittent colored diffusion. Numerical results illustrate the performance of the method.

Keywords Double descent · Color noise · Intermittent diffusion · Optimization

Mathematics Subject Classification (2010) 65H99 · 65K99

1 Introduction

Let $g : \mathbb{R}^n \to \mathbb{R}$, $n \geq 1$ be a sufficiently smooth function (say, $\mathcal{C}^\infty$); we call g the "potential," or "objective function." Let ∇g be the gradient of g, and H be the Hessian. Finally, we also let $G : \mathbb{R}^n \to \mathbb{R}^+$ be defined as $G = \frac{1}{2}(\nabla g)^T(\nabla g)$; we call G the "auxiliary potential." Our goal is to minimize g.

Finding global minimizers for a general objective function g is one of the oldest and most challenging problems in applied mathematics. Whereas it is at times possible to exploit a-priori knowledge for specific potentials, it remains an outstanding task to devise effective general optimization strategies which can be applied to a general problem. In the literature, one finds extensive collections of real-world

✉ Manuela Manetta
manuela.manetta@emory.edu

Luca Dieci
dieci@math.gatech.edu

Haomin Zhou
hmzhou@math.gatech.edu

[1] School of Mathematics, Georgia Institute of Technology, Atlanta, USA

[2] Department of Mathematics, Emory University, Atlanta, USA

potentials, arising from chemistry, physics, mathematics, etc., as well as artificial problems. Many challenging problems in the first group are obtained as models of interatomic forces, and are distinguished by having a reduced region of interest, expensive computation of the potential and its gradient, and full (not sparse) Hessians. Problems in the second group are useful to validate an optimization technique, to illustrate it, and to have objective functions with selectively distinguished features: a single global minimum, multiple global minima in the presence of many local minima (in which case any deterministic technique will be trapped in the basin of attraction of a minimizer without being able to escape it), long narrow valleys (which will slow down the search process), and flat surfaces. Unsurprisingly, some methods perform well on some problems, and poorly on others, and—aside from knowing ahead of time what method to use on a specific potential—one is left wondering on what to use for a given problem.

We are often confronted with this frustrating state of affairs when teaching a course on numerical methods for optimization. Even absolutely marvelous textbooks (e.g., [7]) are ultimately having to accept some uncertainties, and to deal with fine-tuning of parameters, and empirical choices. To be fair, these difficulties are intrinsic to the task at hand, and surely not the result of negligence. So, when we teach such a course, we end up teaching local techniques, maybe continuation and embedding techniques, emphasize gradient descent and Newton's method and their variants, stress convex or maybe polynomial functions, but in the end we fail at providing rigorous answers to the recurring questions of alert students relative to a general smooth function g: "how do we know that we found the global minimum? how do we know that we have visited the interesting regions of configuration space?" We do not know, and most likely we will not know for the foreseeable future. Indeed, barring a painstaking and extensive search of the configuration space, we have few hints to offer to our students for answering their questions above. Motivated by our classroom experience, one of our purposes in this work is to present methods and ideas that can be taught in a numerical optimization course. That said, quite honestly, we have no pretense that our work is an answer to the above questions, but we are hopeful to be taking a (small) step in the right direction.

Let us immediately stress that we are putting forth some ideas for a general purpose method, one which does not rely on the specific properties of the potential. With this purpose in mind, we may recall that the main components of a global minimization algorithm are to explore the landscape and to locate minima. The shape of the level sets of g dictates the nature of the landscape: flat, rough, predictable, and crowded with minima. At the same time, the shape of the level sets of G is more directly responsible for properly identifying the basins of attraction of the minima of g for important techniques, such as Newton's method. Naturally, the level sets of g and of G are often of a very different nature; for example, in Fig. 1, we show them relative to the function of Example 6.1.1.

Unfortunately, graphical insight provided by the level sets is all but unavailable for problems in several variables, and exploration of the landscape remains a mix of randomization and subdivision ideas. Most global optimization techniques switch repeatedly between local and global phases. In the former, a restricted area is explored, whereas in the latter points, they are generated in order to explore the

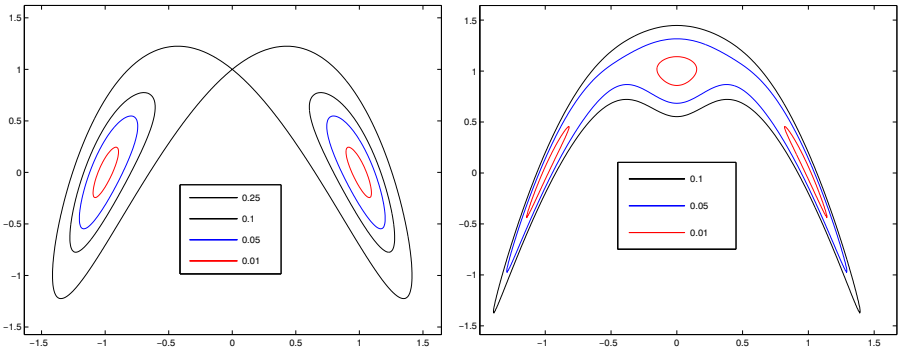

Fig. 1 Level sets for g (on the left) and G (on the right); Example 6.1.1

search domain. The global phase usually consists in a random generation of points that are generally far from the regions that have already been visited. As for the local phase, various techniques have been presented, such as the so-called variable neighborhood search [18], basin hopping [15], and iterated local search [17]. If the reader is missing a general glance at optimization methods, some elementary methods such as pure random search, best start, and multistart or clustering methods, together with the abovementioned methods and the well-known simulated annealing, genetic and particle swarm algorithms are excellently summarized in [16]. Another characteristic of global optimization methods concerns the sequential or parallel nature of the algorithm. An example of sequential method is simulated annealing, which makes use of a probability function to decide how to move in the search space; briefly, the method can be described as follows: call x the current state, x_{new} a randomly selected neighbor of x, and let the probability function be given by

$$P(x, x_{\text{new}}, T) = \begin{cases} 1, & \text{if } g(x_{\text{new}}) < g(x), \\ \exp\left(\frac{-[g(x_{\text{new}}) - g(x)]}{T}\right), & \text{if } g(x_{\text{new}}) > g(x), \end{cases}$$

where T is called the temperature and is a function of the ratio between the current iteration number and the total number of iterations allowed. Given this setting, a random number r is generated: if the probability to move from x to x_{new} is greater than r, then the new state is accepted; otherwise, it is rejected; note that a downhill direction will always be accepted, though one may also take uphill steps. This method is inherently better suited for discrete problems, and it is sequential in nature (see [4] for more details on simulated annealing).

It is useful to mention some of the techniques from the 1970s and 1980s, when several works were concerned with global optimization, e.g., the collection of works in the two volumes [9] and [10], and especially the global optimization method proposed in [1] and [2]. The techniques in these latter works can be viewed as a steepest descent differential equation perturbed by adding a stochastic white noise term and a further penalization function to ensure the paths remain in a given region is used in

the implementation. A key aspect of our approach in this work represents an extension of [1] and [2], in that a minimization method is modified by perturbing with a colored noise stochastic term.

In this work, we introduce a method which uses a combination of new techniques, namely a double-descent method to search for minima and an intermittent colored diffusion technique to escape critical points. The global phase avoids the usual randomness of the exploration of the search domain, by using the information gathered at the critical points to move in an educated way in the landscape exploration.

All along, we will tacitly assume that critical points (i.e., values x where $G(x) = 0$) are simple, in the sense that the Hessian is invertible there. In particular, at minima, the associated Hessian will be positive definite. Nevertheless, the proposed method will also be able to solve problems where at the critical point the Hessian has eigenvalues equal to 0.

We conclude this introduction with some practical considerations.

(i) Although we are considering an unconstrained minimization problem, the case of constrained optimization is of course also important, and we expect to consider it in the future.

(ii) Of the many minimization methods proposed during the years, some use only gradient information, some also Hessian, and some use only functional evaluations (so-called direct search algorithms). Of course, the specific problem at hand may inhibit using the gradient and/or the Hessian, we will assume that these are available to us. In fact, in our technique, we make use of repeated eigen-decomposition of the Hessian. Of course, this is an expense which may be prohibitive for truly large problems, though by today's standards, it is easily doable for dimensions of up to a few hundreds. It is not by coincindence that a lot of people have been concerned with efficient updating of Hessian factorizations (e.g., the BFGS (Broyden-Fletcher-Goldfarb-Shanno) or the DFP (Davidon-Fletcher-Powell) updates); see [11].

(iii) The prevailing wisdom (e.g., see the well-known Levenberg-Marquardt algorithm, trust-region methods, and the discussion in [7] and [14]) is to use Newton's method near a minimizer. Our technique is designed to automatically do Newton's method as well, as we reach the neighborhood of a minimizer, or of another critical point.

(iv) Many recent advances in global optimization (e.g., genetic algorithms, direct search techniques, multiple random initializations) have found their place in public domain software; e.g., see [12] and the `Matlab` Global Optimization Toolbox. In particular, the latter contains three routines which we have used for cross-comparison of our results: `GlobalSearch`, `simulannealbnd`, and `MultiStart`. The function `simulannealbnd` is the `Matlab` implementation of simulated annealing. Instead, `GlobalSearch` finds minimizers at different stages: first a local search (carried out by the function `fmincon`) starts from an initial point x_0 provided by the user, and then a list of trial point is generated as potential starting points, taking into account penalty functions, spherical basins of attractions, and run-time, to eventually perform the local search from a large number of initial points. (For fair comparison with the

results of our method, we used this function by providing gradient and Hessian information). The `MultiStart` routine runs a local solver (`fmincon`) from a different given number of starting points.

(v) Finally, we must stress that it is very delicate to implement any method, and that methods that look good on paper may not deliver according to expectations. For this reason, we will detail our implementation choices so that our results may be replicated.

A plan of this paper is as follows. In Section 2, we give some background material. In Section 3, we introduce the double-descent technique, and in Section 4, we give the combined "double-descent color-intermittent diffusion" method (DD-CID, for short). An overview of our numerical method is in Section 5, and several numerical experiments are reported in Section 6.

2 Preliminaries

2.1 Intermittent diffusion

In the recent work [5], the authors proposed a general methodology, called intermittent diffusion (ID, for short), motivated by the fact that the most widely used stochastic technique available for global optimization, the simulated annealing mentioned before, needs a deterministic part to speed up the convergence towards the minimizers. In order to do so, ID alternates between gradient descent and diffusion processes, by turning on and off a white noise term. In mathematical terms, the ID methodology can be summarized by the following stochastic differential equation:

$$dx(t, w) = -\nabla g(x(t, w))dt + \sigma(t)dW(t),\ t \in [0, +\infty] \tag{1}$$

where $W(t)$ is Brownian motion in $\mathbb{R}^n$, w is a random path in the Wiener space, and $\sigma(t)$ is a piecewise constant function of time alternating between positive and zero values. In particular, when the noise is off ($\sigma(t) = 0$), the method reduces to the gradient-descent technique, leading the trajectory towards a minimizer of the potential; when the noise is on ($\sigma(t) > 0$), the trajectory should leave a neighborhood of the minimizer and, eventually, reach the basins of attraction of different minima.

The discontinous diffusion term is given in [5] as

$$\sigma(t) = \sum_{j=1}^{N} \sigma_j I_{[S_j, T_j]}(t) \tag{2}$$

where $0 = S_1 < T_1 < S_2 < T_2 \cdots < S_N < T_N < S_{N+1} = T$ and $I_{[S_j,T_j]}$ is the characteristic function of the interval $[S_j, T_j]$. In [5], the length of the intervals $T_j - S_j$, and the constants σ_j are supposedly chosen randomly for all $j = 1, \ldots, N$; therefore, when the characteristic function is 1, the minimizer is perturbed by a positive random number for a certain amount of time, and when the noise is off, namely $I_{[S_j,T_j]} = 0$, the method reduces to gradient descent and slowly converges to a local minimum.

As originally proposed, the ID approach is a general methodology, but to make it become a practical method requires making a lot of choices; for example, to decide how long the diffusion process needs to be carried out. In our experimentation of this technique, at first, we used the discrete analog of (1):

$$x_{k+1} = x_k - h\nabla g(x_k) + \sqrt{h}\sigma(t_k)W \ , \tag{3}$$

where $W \in N(0,1)^n$. However, when using this technique, we faced the need to adjust too many parameters based on the potential we were trying to minimize, and realized that there were some key aspects to be addressed:

- The local convergence towards minima, using the gradient descent, was too slow, and a faster method (eventually, Newton-like) was desirable.
- White noise–based diffusion did not account for the local landscape of the potential, and we eventually wanted to modify this with color noise diffusion.
- Criteria were needed to replace choosing the interval length randomly, finding instead a deterministic criterion to switch the noise on and off.

We addressed all of the above concerns in the present paper.

In order to achieve our goal to build up a method which automatically adjusts to the optimization problem, we resorted to exploiting the Hessian's spectral information.

2.2 $\mathbb{R}^n$ via the Hessian

Below, we clarify the structure of regions of $\mathbb{R}^n$ where the eigenvalues of H have a specified signature (inertia).

Definition 1 Given a symmetric matrix H, the inertia of H is the triplet

$$\nu(H) = \{n_+(H), n_0(H), n_-(H)\},$$

where n_+, n_0, and n_- are the number of positive, zero, or negative eigenvalues of H, counted with their multiplicities. H is called hyperbolic if $n_0(H) = 0$.

Observe that $H : \mathbb{R}^n \to \mathbb{R}^{n\times n}$ is a smooth symmetric function of n parameters; hence, the reasonings below are valid.

We will always order the eigenvalues of H as $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$, and $v_1, \ldots, v_n$ will be the corresponding orthogonal eigenvectors. According to $\nu(H)$, we will also use the notation $V = [v_1, \ \ldots \ , \ v_n] = [V_+, \ V_0, \ V_-]$, and will call the columns of V_+ the basis for the positively dominant subspace, or simply (with improper language) the dominant subspace, etc..

Lemma 2 *Consider the set* $\mathcal{P} := \{x \in \mathbb{R}^n \ : \ y^T H(x) y > 0 \ , \ \forall y \in \mathbb{R}^n\}$. *We have the following properties:*

(1) $\mathcal{P}$ *is open.*
(2) $\mathcal{P} = \cup_i \mathcal{P}_i$, *where each* $\mathcal{P}_i$ *is open and connected and* $\mathcal{P}_i \cap \mathcal{P}_j = \emptyset$ *for* $i \neq j$.
(3) *Each* $\mathcal{P}_i$ *is path connected.*

Proof (1) follows from these considerations.

(i) The eigenvalues of the function H are continuous functions of x. (This is a standard result).

(ii) If A is a symmetric positive definite function, and $B = B^T$ is such that $\|A - B\|_2 < \lambda_n(A)$, then B is positive definite. (This is also well known).

Thus, if we take a value x_0 where $H(x_0)$ is pos-def, consider the smallest eigenvalue of H as a function of x, and use the continuity of $\lambda_n(\cdot)$, then $H(\cdot)$ must remain positive definite for $x \in B_r(x_0)$ (an open ball centered at x_0, of radius r): $B_r(x_0) = \{y \in \mathbb{R}^n \ : \ \|x - y\| < r\}$, or also $B_r(x_0) = \{x \in \mathbb{R}^n \ : \ x = x_0 + \rho w, \ \|w\| = 1, \ \rho < r\}$.

As far as (2), the observation is that the function H ceases to be positive definite when $\lambda_n(x) = 0$. So, we define a set $\mathcal{P}_i$ as that component of $\mathcal{P}$ such that for any two $x, y \in \mathcal{P}_i$ there is a curve joining x and y such that along this curve $\lambda_n > 0$. As above, $\mathcal{P}_i$ is open, and thus the set $\mathcal{P}$ is separated into open connected components $\mathcal{P}_i$'s, and $\mathcal{P}_i \cap \mathcal{P}_j = \emptyset$, for $i \neq j$.

(3) follows from a classical result in topology, telling that "open connected sets in $\mathbb{R}^n$ are path connected." [It is also possible to argue directly, since, given that the $\mathcal{P}_i$'s are open, an open ball centered at any point $x \in \mathcal{P}$ must intersect all coordinate directions]. □

Remark 3 Properties similar to (1)-(2)-(3) above are still true in case the Hessian is hyperbolic. Indeed, considering the sets

$$\mathcal{P} := \{x \in \mathbb{R}^n \ : \ n_0(H(x)) = 0 \ , \ n_+(H(x)) \text{ and } n_-(H(x)) \text{ constant} \neq 0\} \ ,$$

any of these sets for a fixed constant of n_+ and n_- is open and the union of (path) connected components. The reason is that perturbation of a hyperbolic Hessian renders a hyperbolic one, with same inertia, as a consequence of the fact that invertible matrices form an open set.

2.3 Courant's theorem

As we will see in the following, a main idea of our method is to escape the basin of attraction of a minimizer by searching for a saddle point. Now, it is well known that if the potential is a function of one real variable, $g : \mathbb{R} \to \mathbb{R}$, and x_1 and x_2 are two strict minima, that is, $g''(x_{1,2}) > 0$, then g must have another critical point x_3 between x_1 and x_2. However, as soon as we consider a real-valued function of two variables, a similar result does not hold, in general.[1] Nevertheless, under certain conditions, the existence of other non-extremal critical points has been proved, and this result, due to Courant, dates back to 1950 (see [13, p.49], where g is only assumed to be $\mathcal{C}^1$).

[1] As an example, consider the function $g_1(x, y) = (x^2 y - x - 1)^2 + (x^2 - 1)^2$: it has two local minima at $(1, 2)$ and $(-1, 0)$, and no other critical point.

Theorem 4 *Suppose that g is coercive*[2] *and possesses two distinct strict relative minima x_1 and x_2. Then g possesses a third critical point distinct from x_1 and x_2, characterized by*

$$g(x_3) = \inf_{\Sigma\in\Gamma} \max_{x\in\Sigma} g(x) \ ,$$

where $\Gamma = \{\Sigma \subset \mathbb{R}^N;\ \Sigma$ is compact and connected and $x_1, x_2 \in \Sigma\}$.

Moreover, x_3 is not a relative minimizer; that is, in every neighborhood of x_3, there exists a point x such that $g(x) < g(x_3)$.

Theorem 4 is part of "mountain pass theory." An accessible introduction to this theory and its applications is in [3], a comprehensive treatment is [13], and the report [19] and the work [20] propose numerical methods to approximate mountain pass points (here, the authors use the characterization of mountain pass points as critical points where the (nonsingular) Hessian has exactly one negative eigenvalue).

3 Descent directions and the double descent

Here we introduce the double-descent direction. First, we recall the definition of (gradient) descent and Newton's directions.

(a) (*Descent direction*). Assuming that $\nabla g(x_0) \neq 0$, any direction v such that $g(x_0 + \alpha v) < g(x_0)$, for all sufficiently small $\alpha > 0$, is called a *direction of descent* for the potential g. A trivial computation shows that a direction of descent v must satisfy $(\nabla g(x_0))^T v < 0$. The classic choice is $v = -\nabla g(x_0)$ (the so-called *gradient descent* choice).

(b) (*Newton's direction*). This is the direction resulting from using Newton's method to solve the problem $\nabla g(x) = 0$. In other words, it is the direction (assuming that $\nabla g(x_0) \neq 0$ and that the Hessian is invertible) given by $v(x_0) = -H(x_0)^{-1}\nabla g(x_0)$. We note that this is a descent direction for the functional G at x_0, since $\nabla G = H\nabla g$.

Remark 5 Of course, we can always normalize a descent (and/or Newton's) direction to be a vector of norm 1.

3.1 Descent direction within a positive definite region

The following result, which is both fundamental and well known (see [7, p.114]), serves as motivation for some of our later algorithmic choices.

Lemma 6 *Let x_0 be a point where $\nabla g \neq 0$ and let $H(x_0)$ be positive definite. Then, the Newton's direction is a direction of descent for g.*

[2]Recall $g : \mathbb{R}^n \to \mathbb{R}$ is coercive if $\lim_{\|x\|\to\infty} g(x) = +\infty$, that is, for any constant M, there is a constant R_M such that $\|g(x)\| > M$ whenever $\|x\| > R_M$.

Proof We need to show that—at x_0—we have $(\nabla g)^T v < 0$ when $v = -H^{-1}\nabla g$. But this is obvious since H is positive definite. □

With the help of Lemma 6, the following is immediate.

Proposition 7 *Suppose that $x_0 \in \Omega_0$, where Ω_0 is a path-connected component where H is positive definite. Then, either $\nabla g(x_0) = 0$ or there exists a direction $v \in \mathbb{R}^n$, and a scalar $\alpha > 0$, such that both $G(x_0+\alpha v) < G(x_0)$ and $g(x_0+\alpha v) < g(x_0)$. Further, when $\nabla g(x_0) \neq 0$, one can also choose τ so that both potentials decrease and $H(x_0 + \tau v)$ is positive definite.*

Proof We want v such that both of these relations hold at x_0:

$$v^T \nabla G < 0 \quad \text{and} \quad v^T \nabla g < 0\,.$$

Since $\nabla G = H\nabla g$, these relations are

$$v^T H \nabla g < 0 \quad \text{and} \quad v^T \nabla g < 0\,.$$

Each of the above inequalities defines an open half space, and the Newton's direction is in both of these. Therefore, the existence of a unit vector v giving us the sought decrease is established, and there exists a scalar α, positive, such that both $G(x_0 + \alpha v) < G(x_0)$ and $g(x_0 + \alpha v) < g(x_0)$.

Further, since $H(x_0)$ is positive definite, then there is an open ball centered at x_0 and of radius $r > 0$, $B_r(x_0)$, such that $H(x)$ remains positive definite for any $x \in B_r(x_0)$. Therefore, there exist τ so that $H(x)$ is positive definite for $x = x_0+\tau v$. □

Remark 8 Because of Lemma 6, in Proposition 7, we can choose v to be the Newton's direction.

We note right away that it is often not advisable to select the step length τ so that the Hessian remains positive definite. Indeed, in our numerical experiments, doing so often resulted in a severe restriction of the step length and inefficient computations, and it was quite preferable to allow a decrease in the potentials without forcing a fixed inertia for the Hessian. For this reason, we now define the double-descent direction allowing for the Hessian to be indefinite.

3.2 Descent direction within an indefinite region

Here, we generalize the above result to the case of regions with different Hessian's inertia.

Proposition 9 *Let Ω be a path-connected region where $\nu(H) = \{n_+, n_0, n_-\}$ for all $x \in \Omega$, with $n_+ \geq 1$. Let $x_0 \in \Omega$, and let $V_+ = span\{v_1, \ldots, v_{n_+}\}$, where $v_1, \ldots, v_{n_+}$ are eigenvectors corresponding to positive eigenvalues of $H(x_0)$.*

Then, if $V_+^T \nabla g(x_0) \neq 0$, there exists a direction $v \in \mathbb{R}^n$, and a scalar $\alpha > 0$, such that both $G(x_0 + \alpha v) < G(x_0)$ and $g(x_0 + \alpha v) < g(x_0)$. Further, if $n_0 = 0$, i.e., $H(x_0)$ is hyperbolic, then there exists $\tau > 0$ so that $\nu(H(x_0 + \tau v)) = \nu(H(x_0))$.

Finally, in all cases above, the direction v can be taken as $v = -(H_+(x_0))^\dagger \nabla g(x_0)$, where H_+ is the closest positive semidefinite matrix to H; that is, if $H = V\Lambda V^T$, and $\Lambda = diag(\Lambda_+, \Lambda_0, \Lambda_-)$, then $H_+ = V\Lambda^+ V^T$, with $\Lambda^+ = diag(\Lambda_+, 0_{n_0}, 0_{n_-})$, and thus $(H_+)^\dagger = V_+(\Lambda_+)^{-1}V_+^T$.

Proof We want v such that both of these relations hold at x_0:

$$v^T H\nabla g < 0 \quad \text{and} \quad v^T \nabla g < 0\,.$$

Considering the direction v given in the statement, we have

$$v^T \nabla g = -\nabla g(x_0)^T (H_+(x_0))^\dagger \nabla g(x_0) < 0$$

since $V_+^T \nabla g(x_0) \neq 0$. For the same reason, we also have

$$\begin{aligned} v^T H\nabla g &= -\nabla g(x_0)^T (H_+(x_0))^\dagger H \nabla g(x_0) \\ &= -(V_+^T \nabla g)^T (\Lambda_+)^{-1} (V_+^T \nabla g) < 0\,. \end{aligned}$$

Therefore, the existence of a unit vector v giving us the sought decrease is established, and there exists a scalar α, positive, such that both $G(x_0 + \alpha v) < G(x_0)$ and $g(x_0 + \alpha v) < g(x_0)$. Further, if $H(x_0)$ is hyperbolic, then $\nu(H(x_0)) = \{n_+, n_-\}$. Therefore, there is an open ball centered at x_0 and of radius $r > 0$, $B_r(x_0)$, such that $\nu(H(x)) = \nu(H(x_0))$ for any $x \in B_r(x_0)$. Thus, we can choose $\tau > 0$ such that $\nu(H(x_0 + \tau v)) = \nu(H(x_0))$. □

Remark 10 The direction $v = -(H_+(x_0))^\dagger \nabla g(x_0)$ of Theorem 9 is effectively the Newton's direction restricted to the subspace associated with the positive eigenvalues.

Summary 11 *To sum up, as long as the point x_0 is in a region where the Hessian has at least one positive eigenvalue, and ∇g has a nontrivial component in the subspace spanned by the eigenvectors corresponding to the positive eigenvalues, we can always find a direction v which is of descent for both G and g. If $H(x_0)$ has no 0 eigenvalue, we can also maintain the inertia of H by taking a step in the direction v; however, this may be counterproductive (as our computations showed), since it may unduly restrict the step τ, and it is much more desirable to let the iterate enter and exit regions of different Hessian's inertia while decreasing the potentials g and G.*

Remark 12 One more comment is needed about the condition $V_+^T(x)\nabla g(x) \neq 0$. The dimension of the subspace spanned by the columns of V_+ is n_+, while ∇g is a vector in $\mathbb{R}^n$. Therefore, in general, the requirement $V_+^T \nabla g = 0$ would define a set of n_+ equations in the n variables $x \in \mathbb{R}^n$. Generally, these define an $n - n_+$ dimensional manifold immersed in $\mathbb{R}^n$. Therefore, we should expect that, at any x, the vector ∇g will have a nontrivial component in V_+. This is the truer the larger is n_+. In the limiting case of $n_+ = n$, Lemma 6 already told us that. At the same time, if $n_+ = 0$, then obviously there is no direction v_+ to begin with. In this case, there is no double-descent direction to begin with, and our method (see below) will revert to using the gradient direction.

4 Double-descent colored-diffusion method

The main idea of our technique is to take advantage of information about the inertia of the Hessian, in order to explore the landscape going from a saddle point to a minimum and vice versa, avoiding being trapped in the basins of attraction of the critical points while following an educated path.

There are two types of processes we use: "local zoom-in" and "basin-escaping" methods.

4.1 Reaching a critical point: local search

This part is based on the developments of Section 3. We distinguish between the cases of searching for a minimum or a saddle. In the former case, we have a host of possibilities: the double-descent algorithm, gradient descent, and Newton's method (damped); in the latter, we use a (damped) Newton's method. Still, we must make some careful implementation choices.

For example, when searching for a minimum, beside the usual concerns on how to choose the step length (see [7]), we also accounted for the following aspects when implementing the double-descent method.

(i) When using the double-descent direction, we demand that both g and G have appreciably decreased.

(ii) To declare that ∇g has no meaningful component in the subspace spanned by V_+, we used the criterion $\|V_+^T(x_k)\frac{\nabla g(x_k)}{\|\nabla g(x_k)\|}\| \leq \sqrt{n}/10$, where n is the ambient dimension. When this happens, we reverted to the direction v given by the simpler gradient-descent direction, and kept this descent strategy for 5 steps before retrying the double-descent direction. Likewise, we reverted to the gradient-descent direction when too many damping steps are taken with the double-descent direction.

(iii) Another important consideration is about the stopping criterion (both when searching for a minimum or a saddle). In our implementation, we chose the following stopping criterion (always within the maximum allowed number of iterations). We iterate as long as:

$$\|\nabla g(x_k)\| \geq \texttt{atol} + \|\nabla g(x_0)\|\texttt{rtol} \text{ and } \|x_k - x_{k-1}\| \geq \texttt{atol} + \|x_0\|\texttt{rtol}\,,$$

where `atol` and `rtol` stand for absolute and relative tolerance, and they are chosen by the user.

4.2 Basin escaping by color diffusion

Here we adopt (some of) the ideas in Section 2.1 in order to leave the basin of attraction (for Newton's method) of a critical point. In particular, compare (4), (5), and (6) with (3) from Section 2.1; just as (3) can be viewed as a discretization of the SDE (1), our equations (4), (5), and (6) can be thought of as discretizations of an underlying SDE in regions where the inertia of the Hessian and the sign of the dominant (respectively, smallest) eigenvalue are not changing.

As seen in Section 3, the double-descent method is designed to lead to a minimum when the Hessian at the starting point is either positive definite or indefinite, and—when properly implemented—it will really be Newton's method close to a minimum. In contrast, when the initial value lies in a region in which both $n_+(H)$ and $n_-(H)$ are nonzero, we will presume that (damped) Newton method will converge to a saddle point (or perhaps to a maximum); this expectation has effectively been borne out in practice for the vast majority of our experiments.

Regardless, if we are at either a minimum or at a saddle, we need to leave the respective basins of attraction for (damped) Newton method. To do this, we implemented a colored intermittent diffusion method as follows, by reflecting the choices made above to look for either a saddle or a minimum, and the discussion in Section 2.1.

(a) From a min x_0, trying to go to a saddle. Let us first assume that, $\lambda_1 > \lambda_2 \geq \cdots \geq \lambda_n > 0$ along our iterates. There are three basic steps.

(i) Select $\alpha > 0$, and generate

$$x_1 = x_0 + \alpha v_1(x_0) v_1^T(x_0) W \ , \quad W \in N(0,1)^n \ .$$

(ii) Find h such that $|G(\hat{x}_{k+1})| < |G(x_k)|$, with $\hat{x}_{k+1} = x_k - h\left(H^\dagger(x_k)\nabla g(x_k)\right)$, and then

$$\begin{aligned} x_{k+1} &= x_k - h\left(H^\dagger(x_k)\nabla g(x_k)\right) + \alpha\sqrt{h}\sigma(x_k)W \text{ , where} \\ \sigma(x_k) &= -v_1(x_k)v_1(x_k)^T \ , \quad \text{and} \quad W \in N(0,1)^n \ . \end{aligned} \tag{4}$$

(iii) Continue with the diffused damped Newton's above until the Hessian has some negative eigenvalues or the maximum number of diffusive iteration has been exceeded. At that point, use (damped) Newton's method. Hence, select the Newton direction v and the step length h to decrease the auxiliary potential G; say, $G(x_k + hv) < G(x_k)$. If g_m denotes the value of the potential g at the minimum from which we started, we observed that consistently $g(x_k + hv) > g_m$ which betrays that we are not going back to the starting minimum.

The rationale for the colored noise diffusive step is to move away as quickly as possible from the basin of attraction of the minimum. If x_0 is a minimum, the standard quadratic approximation in a ϵ-ball around x_0 will give:

$$g(x_0 + \epsilon y) = g(x_0) + \epsilon \nabla g(x_0)^T y + \frac{1}{2}\epsilon^2 y^T H(x_0) y + \ldots$$

and therefore, with $\|y\| = 1$, the fastest increase is for $y = v_1$. In the (very unlikely) case that the dominant eigenvalue has multiplicity greater than 1, we select a random vector in the span of the dominant eigenvectors.

(b) *From a saddle x_0, trying to go to a min*. Let us first assume that $\lambda_1 \geq \cdots \geq \lambda_{n-1} > \lambda_n$, with $\lambda_1 > 0$ and $\lambda_n < 0$. Even here, there are two basic steps, getting out of the saddle and going to a minimum. The second step, see below, can be carried out with the double-descent method, or with gradient descent, or

possibly with a (damped) Newton approach. In all cases, first we use colored diffusion steps to move out of the saddle.

(i) Select α and generate

$$x_1 = x_0 + \alpha v_n(x_0) v_n^T(x_0) W \ , \quad W \in N(0,1)^n \ .$$

As before, the choice of the colored noise diffusive step is to move away as quickly as possible from the basin of attraction of the saddle, while decreasing the potential g. If x_0 is a saddle, in a ϵ-ball around x_0, we have:

$$g(x_0 + \epsilon y) = g(x_0) + \epsilon \nabla g(x_0)^T y + \frac{1}{2}\epsilon^2 y^T H(x_0) y + \dots$$

and therefore, with $\|y\| = 1$, the fastest decrease is for $y = v_n$.

(ii) **Double descent**. If $\nabla g(x_k)$ has a meaningful component in the direction of $V_+(x_k)$, do (ii-a), otherwise do (ii-b).

(ii-a) Find h such that both $|G(\hat{x}_{k+1})| < |G(x_k)|$ and $|g(\hat{x}_{k+1})| < |g(x_k)|$ with $\hat{x}_{k+1} = x_k - h\left(H_+^\dagger(x_k)\nabla g(x_k)\right)$, and then

$$\begin{aligned} x_{k+1} &= x_k - h\left(H_+^\dagger(x_k)\nabla g(x_k)\right) + \alpha\sqrt{h}\sigma(x_k)W \ , \text{ where} \\ \sigma(x_k) &= -v_n(x_k)v_n(x_k)^T \ , \quad \text{and} \quad W \in N(0,1)^n \ . \end{aligned} \tag{5}$$

(ii-b) Find h such that $|g(\hat{x}_{k+1})| < |g(x_k)|$, with $\hat{x}_{k+1} = x_k - h\,\nabla g(x_k)$, and then

$$\begin{aligned} x_{k+1} &= x_k - h\,\nabla g(x_k) + \alpha\sqrt{h}\sigma(x_k)W \ , \text{ where} \\ \sigma(x_k) &= -v_n(x_k)v_n(x_k)^T \ , \quad \text{and} \quad W \in N(0,1)^n \ . \end{aligned} \tag{6}$$

(iii) **Diffused double descent**. Continue with the diffused double descent above until the Hessian has all positive eigenvalues or the maximum number of diffusive iteration has been exceeded. At that point, use double-descent method. Hence, select the direction v and the step length h to decrease the potential g and the auxiliary potential G; say, $g(x_k + hv) < g(x_k)$ and $G(x_k + hv) < G(x_k)$.

Again, in the (very unlikely) event that the smallest eigenvalue has multiplicity greater than 1, we select a random unit vector in the corresponding subspace.

5 The method at a glance

In the previous sections, we presented only the two key components of the method, namely, the local search and the basin escaping. Here, we give a broader idea of the method.

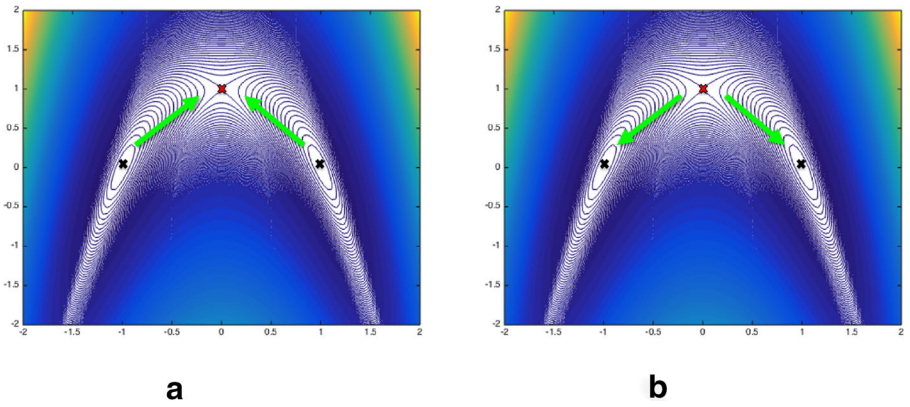

Fig. 2 Basin-escaping main idea. Consider the function of Example 6.1.1, which has two minima and a saddle point. To escape the basin of attraction of a minimum, the idea is to move towards a saddle point, as in **a**. Starting from a saddle point, instead, the goal is to find a direction to move as quick as possible to a minimum, as illustrated in **b**

(0) The very first initial datum x_0 is randomly chosen (within a region of interest). A local search for a minimum starts with the double-descent method, and the point found is stored in a table of critical points.

(1) A point from the table is randomly selected. Colored diffusion to escape the basin of this critical point is performed (see Fig. 2), followed by a local search for the next critical point. The new point is stored in the table,[3] and step (1) is thus repeated until a predefined number of critical points is found.

5.1 Sketch of the algorithm

(1) Choose a random point x_0 in the search region.
(2) Look for a minimum $x^{\min}$ by using the double-descent method.
(3) Store $x^{\min}$ in the list of critical points.
(4) LOOP BEGINS - to be repeated for a preassigned number of iterations.

(a) Randomly choose a point x from the list.
(b) If x is a minimum, start diffusion according to (4) until $n_-(H) \neq 0$ (see Fig. 3a) or maximim number of diffusive steps is exceeded.
Apply (damped) Newton method to find a saddle point.
Store the saddle in the list of critical points.
(c) If x is a saddle point, take diffusive steps according to (5) (or (6)), until $n_-(H) = 0$ (see Fig. 3b) or the maximum number of diffusive steps is exceeded. Apply double descent to find a minimizer.
Store the minimum in the list of critical points.

LOOP ENDS

[3] The same point can thus appear multiple times in the table.

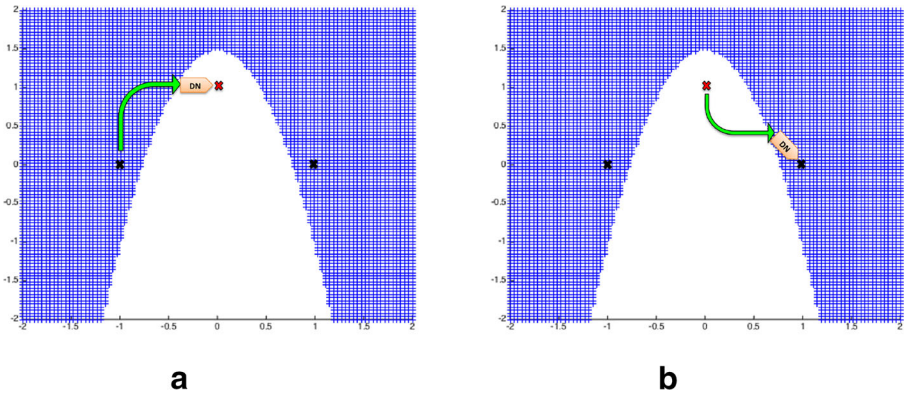

Fig. 3 Switch timing for colored diffusion. The blue region represents the set of points at which the Hessian is positive definite, while the white region has an indefinite Hessian. Suppose we need to escape the basin of attraction of a minimum (**a**), the diffusion process is triggered until reaching the white region, where a local search starts. Conversely, starting from a saddle point, as in **b**, the diffusion process stops when the blue region is reached, and the local search begins

5.2 Computational considerations

A main drawback of Newton's type technique, hence also of the double-descent method, is the need to form, evaluate, and decompose, the Hessian. Except for problems where the Hessian is simple to evaluate, and very structured (e.g., tridiagonal), this can be very expensive and it restricts applicability of Hessian based techniques to small dimension (say, up to a few hundred variables on a typical laptop). We also note that for some problems, evaluating the Hessian is itself an expensive task; e.g., this is the case for interatomic potentials, such as the Morse and Lennard-Jones potentials (see Section 6). Although our purpose in this work has not been to deal specifically with efficient implementations, but rather to give ways to explore the phase space (the landscape), for large (possibly sparse) problems, we have experimented with Lanczos techniques, and a subspace version of Newton's method, whereby we project the Hessian in the direction of the most dominant eigenvalues (positive and negative). We will report on these aspects in other works.

An important consideration pertains to the colored noise diffusion. To perform this diffusion, and to monitor when to stop it, it is straightforward to bypass the Hessian factorization. In fact, to form the color noise and to decide when to stop diffusion, we only need the two eigen-directions v_1 and v_n. These are inexpensive to obtain with a well-designed Lanczos technique (e.g., `eigs` in `Matlab`), by asking for (respectively) the largest and smallest eigenvalues. This feature is particularly useful when using just a damped Newton's method with color noise (as in our basic intermittent diffusion method from a min to a saddle), since the linear systems arising during the iteration are then solved without resorting to a full eigen-decomposition. To elucidate and to account for the possibility of singular Hessian, we first form the

QR factorization with column pivoting of the Hessian: $HP = QR$ with diagonal of R ordered in decreasing magnitude. Then solve the resulting triangular system, possibly for the minimum norm solution (if the Hessian was singular, which is betrayed by $R_{nn} = 0$).

Finally, as seen in Section 4.2, we use a variable stepsize controlled through the requirement of moving in the descent direction(s). The initial stepsize is set to 1, and the stepsize is always required to remain in $[2^{-26}, 2^5]$, where $2^{-26} = \sqrt{\texttt{eps}}$, the square root of the machine precision. When one step is taken in the desired direction, and the computation is immediately accepted, then the stepsize is doubled; if the computation is rejected, the stesize is halved. If we reach the minimum allowed stepsize, the algorithm halts and restarts from a different critical point in the table (or a different random point, the very first time).

6 Applications and examples

In this section, we show performance of our method on several problems, both standard model problems, with an illustrative purpose and to validate the method on different landscape features, and those arising from chemical potentials.

6.1 Test problems

6.1.1 An illustrative example

Consider the following elementary potential:

$$g(x, y) = (x^2 - 1)^2 + (x^2 + y - 1)^2. \tag{7}$$

It has 2 minimizers, located at $(-1, 0)$ and $(1, 0)$, and a saddle point at $(0, 1)$. Starting from a random point x_0, the double-descent technique quickly leads to a minimizer, and the diffusion combined with Newton method allows to find the saddle point, from which the algorithm looks again for a minimizer. By using our technique, and asking the algorithm for at most 4 critical points, we were able to find, in a single run, the two minimizers and the saddle point. Indeed, the method is behaving exactly like we were hoping: first, it converges to $(-1, 0)$, then it goes through the saddle point and from there localizes the other minimum at $(1, 0)$, and then it goes back to the saddle point. On average, we counted 3 diffusive steps and 8.25 local search iterations.

6.1.2 Shubert function

The Shubert function is a highly multimodal potential: it has several local minima and many global ones. Naturally, the function presents many saddle points and maxima. Moreover, the global minima and the global maxima are extremely close, and this is one of the reason why it may be difficult to find the global minimizers.

Although our algorithm is designed to find minima and saddles, it could end up finding maxima as well, due to the Newton's basins of attractions, which are nontrivial. Schubert's potential,

$$g(x, y) = \left(\sum_{i=1}^{5} i \cos\left[(i+1)\, x + i\right]\right)\left(\sum_{i=1}^{5} i \cos\left[(i+1)\, y + i\right]\right), \tag{8}$$

is represented in Fig. 4a. Figure 4 b is a zoom of the contour plot around the global minimizer, and the points are the minimizers found by just applying the double descent technique, starting from random initial values. While finding the global minimizer by applying a deterministic technique requires a starting point in its neighborhood, the ability to explore the landscape eliminates this necessity. One single run of our technique, asking for 100 possible critical points, gave us 45 minima, 3 of which were global, at different locations. On average, for attempt, we counted one diffusive step and 5.9 local search iterations.

6.1.3 Biggs function

Let us consider the following function:

$$g(X) = \sum_{i=1}^{10} \left(e^{-t_i\, x_1} - 5e^{-t_i\, x_2} - y_i\right)^2 \tag{9}$$

where $t_i = 0.1\, i$ and $y_i = e^{-t_i} - 5e^{10\, t_i}$.

There are two critical points: a minimum at $(1, 10)$ and a saddle at $(16.7047, 16.7047)$, as shown in Fig. 5.

The challenges in this problem are the flat landscape of the potential and the presence of narrow regions in which the Hessian is positive (negative) definite, but that do not contain a minimum (maximum), as shown in Fig. 6. Asking for 2 critical points,

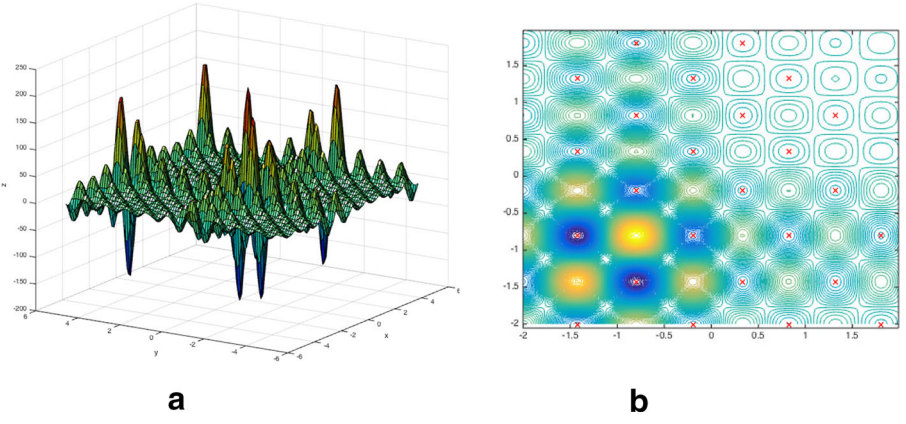

Fig. 4 Shubert function. **a** Potential landscape. **b** Contour plot around a global minimizer

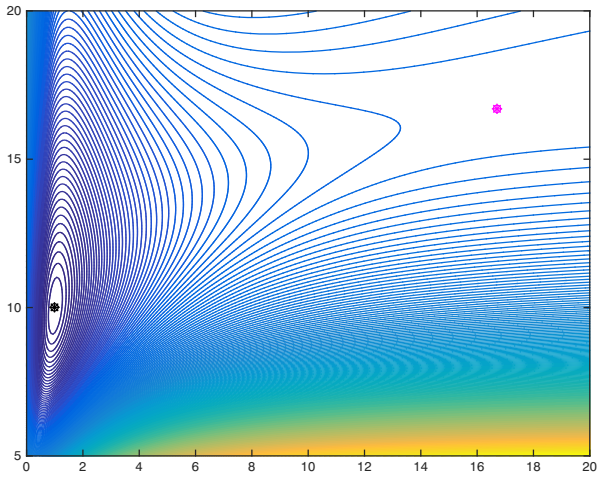

Fig. 5 Biggs function's critical points. (Global) Minimum in black and saddle point in magenta

we were able to find, in a single run, both the minimum and the saddle point. On average, we performed 2.5 diffusive steps and 18 local search iterations.

6.1.4 Camel function

Let us consider the following function:

$$g(x, y) = \left(4 - 2.1\, x^2 + \frac{x^4}{3}\right) x^2 + xy + 4(y^2 - 1)y^2 \tag{10}$$

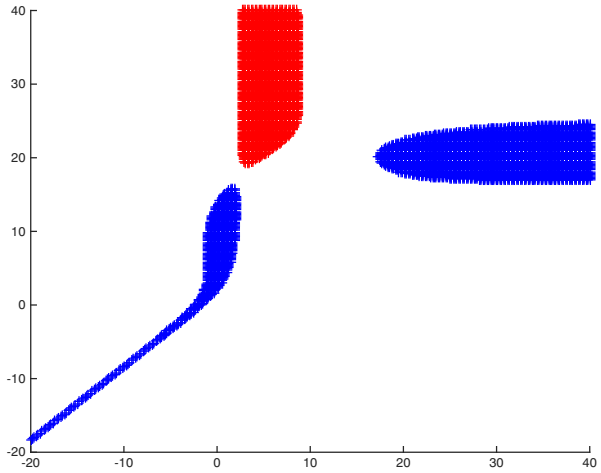

Fig. 6 Biggs function. In blue, the regions of the plane in which the Hessian is positive definite. In redm the one in which H is negative-definite; elsewhere, H is indefinite

This function is a standard test function for global optimization, but it is also useful as a test for the mountain passes' search (see [19]). In fact, our algorithm can be also used to compute mountain passes. As in [19], these are characterized as critical points whose Hessian has exactly one negative eigenvalue, that is, $n_-(H) = 1$.

Consider the region $[-2, 2]\times[-1, 1]$. Here, there are 14 critical points: 6 mountain passes, 6 minima, and 2 maxima. Our method has no difficulty in computing all of these points in one single execution. Results are summarized in Fig. 7 and in Table 1.

6.1.5 Rosenbrock function

This function is given by

$$g(x) = \sum_{i=1}^{N-1} \left[100\,(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \right], \tag{11}$$

and it has a global minimum value of 0 at at $(1, 1, ..., 1)$, for any value of N. The Hessian is very inexpensive to compute and factor, being tridiagonal. The global minimum lies inside a long, narrow, parabolic shaped flat valley; while finding the valley is trivial, detecting the global minimizer is not.

We take $N = 50$. A Monte Carlo gradient-descent technique using 20 random initial guesses did not find the global minimizer. A single implementation of our technique with the possibility to find at most 20 critical points found 16 minimizers, 9 of which were global. On average, per attempt to find a critical point, we counted 1.8 diffusive steps and 136 local search steps. For comparison sake, the `Matlab` routine `GlobalSearch` found the global minimizer, whereas the `Matlab` simulated annealing routine `simulannealbnd` gave a best value for the minimum of

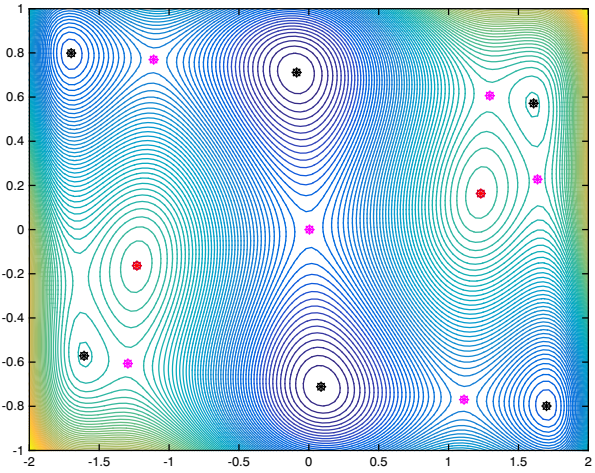

Fig. 7 Camel Function's critical points. In the region of interest, represented in this figure, there are 6 minima (in black), 2 maxima (in red), and 6 mountain passes (in magenta)

Table 1 Camel critical points

x	y	$f(x, y)$	Spectrum of H	
0.0898	− 0.7127	− 1.0316	7.6822	16.4932
− 0.0898	0.7127	− 1.0316	7.6823	16.4932
1.6071	0.5687	2.1043	7.1215	10.0216
− 1.6071	− 0.5687	2.1043	7.1215	10.0216
1.7036	− 0.7961	− 0.2155	18.8171	22.6975
− 1.7036	0.7961	− 0.2155	18.8171	22.6975
1.2302	0.1623	2.4963	− 8.0149	− 5.9537
− 1.2302	− 0.1623	2.4963	− 8.0149	− 5.9537
0	0	0	− 8.0623	8.0623
1.1092	− 0.7683	0.5437	− 7.9026	20.3667
− 1.1092	0.7683	0.5437	− 7.9026	20.3667
1.2961	0.6051	2.2295	− 6.1772	9.6376
− 1.2961	− 0.6051	2.2295	− 6.1772	9.6376
1.6381	0.2287	2.2294	− 5.5458	2.4367

40.7188, and with `Matlab` routine `MultiStart` none of the 20 local solver runs converged, providing no result.

6.2 Chemical potentials

An interesting application of global optimization is protein folding. Mathematically, this consists in finding the equilibrium configuration of d atoms, assuming that the forces between the atoms are known. In the end, one has to find the minimizer of a potential energy function depending on $3d$ variables.

The Lennard-Jones and Morse clusters are two well-known systems of this kind and they have been extensively studied, and the minima tabulated. For example, the (currently best) global minima for Lennard-Jones and Morse potential can be found at the database [6]. These results were obtained with the methods presented in [8] and [15]. Both are "basin-hopping" techniques; they exploit the funneling structure of the potentials (that is, the global minimizer lies at the bottom of a monotonically descending sequence of minimizers), and make a number of choices explicitly based on the specific problem at hand. For example, the authors perform a continuation based upon an optimal configuration reached with d atoms to initiate the search for $(d + 1)$ atoms.

With no pretense of comparing with these other results, below we present some of the results we obtained applying our general technique on both Morse and Lennard-Jones potentials. These potentials depend on the mutual distances (in $\mathbb{R}^3$) between the atoms, namely $r_{ij} = \|P_j - P_i\|$, with $1 \leq i < j \leq d$ and $P_k = (x_k, y_k, z_k)$, for all $k = 1, ..., d$.

Given obvious symmetries in the problem, we imposed the following location constraints: we fix one atom at the origin ($P_1 = (0, 0, 0)$), another one on the x axis

($P_2 = (x_2, 0, 0)$), and a third one in the xy-plane ($P_3 = (x_3, y_3, 0)$). All other atoms are unconstrained. With this setup, each configuration will be identified by the vector of coordinates

$$X = (x_2, x_3, y_3, x_4, y_4, z_4, \ldots, x_d, y_d, z_d),$$

and the dimension of the problem becomes $N = 3d - 6$.

In our experiments, we compute the gradient analytically, and the Hessian numerically, by forward finite differences.

6.2.1 Lennard-Jones potential

This is defined as follows:

$$V(r) = 4\,\varepsilon \sum_{i<j}^{d} \left[\left(\frac{\sigma}{r_{ij}} \right)^{12} - \left(\frac{\sigma}{r_{ij}} \right)^{6} \right], \tag{12}$$

where ε and $2^{1/6}\sigma$ are the pair equilibrium well depth and separation respectively. We take $\varepsilon = \sigma = 1$.

This is a problem where a simple gradient-descent technique, coupled with a Monte Carlo randomization, performs reasonably well; as a matter of fact, our own double-descent method quite often automatically reverts to gradient descent. On this problem, the basin-hopping techniques of [15] is an effective way to find the global minima, since the knowledge of the potential landscape is exploited in the algorithm itself; our method is really a landscape exploration approach. Nevertheless, the method worked well for small values of d, as reported in Table 2.

For the sake of comparison, we ran `MultiStart` with 100 starting points. This technique provided global solutions for $d \leq 4$, but gave local results for $d > 4$.

Table 2 Lennard-Jones global minima

d	N	Potential at global minima
2	1	− 1
3	3	− 3
4	6	− 6
5	9	− 9.104
6	12	− 12.712
7	15	− 16.505
8	18	− 19.821
9	21	− 24.113
10	24	− 28.422
11	27	− 32.766
12	30	− 37.968
13	33	− 44.327
14	36	− 47.845

Table 3 Morse global minima ($d = 11$, $N = 27$)

ρ	Global minima
3	− 37.930817
6	− 31.521880
10	− 30.265230
14	− 29.596054

6.2.2 Morse potential

The Morse potential is defined as follows:

$$V_\rho(r) = \sum_{i<j}^{d} \left[e^{\rho(1-r_{ij})} \left(e^{\rho(1-r_{ij})} - 2 \right) \right] \tag{13}$$

where ρ is a parameter which determines the width of the well. We treat this problem as truly unconstrained, and this may create difficulties to descent techniques, since descent directions may well identify points "at infinity" (i.e., some coordinates grow unbounded); e.g., this happens to the `Matlab` code `GlobalSearch`. A further difficulty is that global minima become harder to locate when ρ increases. In Table 3, we report the results of our method for 11 atoms and $\rho = 3, 6, 10, 14$; our minima match those of [6].

For the sake of comparison, we remark that `Matlab` functions `GlobalSearch`, `simulannealbnd`, and `MultiStart` did not give acceptable results. Namely, we obtained the results in Table 4.

6.3 Nonlinear systems

Our technique can also be used to solve nonlinear systems. Indeed, a nonlinear system

$$S(x) = 0, \text{ with } x \in \mathbb{R}^n, \tag{14}$$

can be reformulated as an optimization problem, simply by considering the objective function given by

$$g(x) = \frac{1}{2} S(x)^T S(x) \tag{15}$$

Table 4 Results obtained using `Matlab` global optimization toolbox (`GlobalSearch`, `simulannealbnd`, and `MultiStart`) on Morse Potential ($d = 11$, $N = 27$)

ρ	`GlobalSearch`	`simulannealbnd`	`MultiStart`
3	− 31.2539	− 11.1367	− 6
6	− 19.3274	− 3.1483	− 6
10	− 0.0036	− 1	− 1
14	− 8.006	− 1	− 3.2060e-32

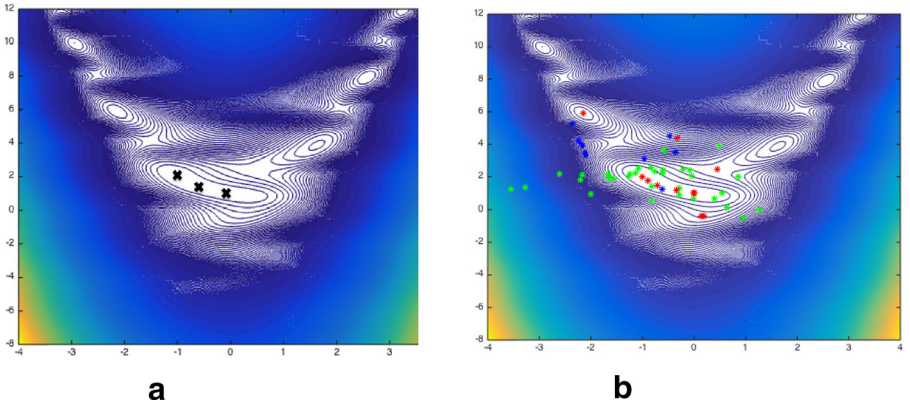

Fig. 8 Boggs system. **a** Contour plot of the potential (15), and solutions of the nonlinear system. **b** Critical points of (15) found by our method; the green dots correspond to diffusion steps

In this case, $\nabla g(x) = J_S(x)^T S(x)$, where J_S indicates the Jacobian of S. Therefore, the critical points of the objective function g correspond to both the zeros of $S(x)$, and the points for which $S(x)$ is in the left null space of the Jacobian.

6.3.1 Boggs system

Given the nonlinear system

$$\begin{bmatrix} x^2 - y + 1 \\ x - \cos\left(\frac{\pi}{2} y\right) \end{bmatrix} = 0, \tag{16}$$

we construct the objective function according to (15).

The solutions of the problem are $(-1, 2)$, $(0, 1)$, and $(-\sqrt{2}/2, 3/2)$, illustrated in Fig. 8a.

Numerical results from a single run of the algorithm, asking for at most 20 critical points, are shown in Table 5 and illustrated in Fig. 8.

Table 5 Critical points—Boggs system

x	y	$g(x, y)$	id
0	1	0	Global minima
− 1	2	0	
− 0.7071	1.5	0	
− 2.1530	5.9055	0.7139	Local minima
0.1301	− 0.3768	1.2161	
0.1890	− 0.3663	1.1941	
− 0.8898	1.7671	0.0013	Saddle points
− 0.3319	1.1830	0.0038	
0.4555	2.4926	1.5111	
− 0.3277	4.3927	6.0502	

7 Conclusions

In this work, we presented a method apt at exploring the landscape of a smooth (at least $\mathcal{C}^2$) potential, in order to locate global minima. The new components of our method are a double-descent technique (to locate minima) and a colored intermittent diffusion (to escape basin of attraction of minima and other critical points). The idea of the technique is to use Hessian information in order to bias the exploration of the landscape. We illustrated performance of our technique on several problems from the literature, observing that our method is able, in most cases, to adapt to different features of the potential. We believe that our method can be easily taught in an optimization course, along with other well-established techniques.

Funding information Manuela Manetta and Haomin Zhou's work was supported under NSF Awards DMS-1419027, DMS-1620345 and ONR Award N000141310408.

References

1. Aluffi-Pentini, F., Parisi, V., Zirilli, F.: Global optimization and stochastic differential equations. J. Optim. Theo. App. **47** (1986)
2. Aluffi-Pentini, F., Parisi, V., Zirilli, F.: A global optimization algorithm using stochastic differential equations. ACM Trans. Math. Softw. **14**, 345–365 (1988)
3. Bisgard, J.: Mountain passes and saddle points. SIAM Review **57**(2), 275–292 (2015)
4. Bertsimas, D., Tsitsiklis, J.: Simulated Annealing. Stat. Sci. **8**(1), 10–15 (1993)
5. Chow, S.N., Yang, T.S., Zhou, H.M.: Global optimizations by intermittent diffusion, Chaos, CNN, Memristors and Beyond: a Festschrift for Leon Chua. In: Adamatzky et al. (eds.) World Scientific Publishing Co. Ltd. (2013)
6. Wales, D.J., Doye, J.P.K., Dullweber, A., Hodges, M.P., Naumkin, F.Y., Calvo, F., Hernandez-Rojas, J., Middleton, T.F.: The Cambridge cluster database. http://www-wales.ch.cam.ac.uk/CCD.html
7. Dennis, J.E., Schnabel, R.B.: Numerical methods for unconstrained optimization and nonlinear equations. SIAM Classics in Applied Mathematics **16** (1996)
8. Doye, J.P.K., Leary, R.H., Locatelli, M., Schoen, F.: Global optimization of Morse clusters by potential energy transformations. INFORMS J. Comput. **16**(4), 371–379 (2004)
9. Dixon, L., Szegö, G.P.: Towards Global Optimization. American Elsevier Pub. Co., North Holland (1975)
10. Dixon, L., Szegö, G.P.: Towards Global Optimization. Elsevier Science Publishing Co Inc., North Holland (1978)
11. Gill, P.E., Murray, W., Wright, M.H.: Practical Optimization. Academic Press, London (1982)
12. Holmström, K.: New optimization algorithms and software. Theory of Stochastic Processes **5**(21), 55–63 (1999)
13. Jabri, Y.: The Mountain Pass Theorem: Variants, Generalizations and Some Applications. Cambridge University Press, Cambridge (2003)
14. Kelley, T.: Iterative Methods for Linear and Nonlinear Equations. SIAM Publications, Philadelphia (1995)
15. Wales, D.J., Doye, J.P.K.: Global optimization by Basin-Hopping and the lowest energy structures of Lennard-Jones clusters containing up to 110 atoms. J. Phys. Chem. A **101**, 5111 (1997)
16. Locatelli, M., Shoen, F.: Global Optimization. Theory, Algorithms, and Applications. SIAM-MOS, Philadelphia (2013)
17. Lourenco, H.R., Matin, O.C., Stütze, T.: Iterated local search. In: Glover, F.W., Kochenberger, G.A. (eds.) Handbook of Metaheuristics, pp. 321–353. Kluwer Academic Publishers, Boston
18. Mladenovic, N., Hansen, P.: Variable neighborhood search. Comput. Oper. Res. **24**(11), 1097–110 (1997)

19. Moré, J.J., Munson, T.S.: Computing Mountain Passes, Preprint ANL/MCS-P957-0502, Argonne National Laboratory (2002)
20. Zhang, J., Du, Q.: Shrinking dimer dynamics and its applications to saddle point search. SIAM J. Numer. Anal. **50-4**, 1899–1921 (2012)