Hopping between distant basins

Maldon Goodridge\textsuperscript{1} \, \cdot \, John Moriarty\textsuperscript{1} \, \cdot \, Jure Vogrinc\textsuperscript{2} \, \cdot \, Alessandro Zocca\textsuperscript{3}

Received: 13 August 2021 / Accepted: 16 March 2022
© The Author(s) 2022

Abstract
We present and numerically analyse the Basin Hopping with Skipping (BH-S) algorithm for stochastic optimisation. This algorithm replaces the perturbation step of basin hopping (BH) with a so-called skipping mechanism from rare-event sampling. Empirical results on benchmark optimisation surfaces demonstrate that BH-S can improve performance relative to BH by encouraging non-local exploration, that is, by hopping between distant basins.

Keywords Basin hopping \cdot Stochastic optimisation \cdot Skipping sampler \cdot Rare events \cdot Markov chains

Mathematics Subject Classification 65K10 \cdot 90C26

1 Introduction and background

Global optimisation of a multi-modal function is one of the oldest and most essential problems of computational mathematics, regularly manifesting itself throughout the applied sciences [20, 38].

A source of inspiration for tackling global optimisation has been methods from the theory of rare-event sampling. Examples include the methods of cross-entropy for combinatorial and continuous optimisation [30] and, more recently, splitting for optimisation [5]. In stochas-

tic optimisation algorithms such as random search [32], basin hopping [17, 37], simulated annealing [15] and the multistart method [10, 21], one or more initial points $X_0$ are per-
turbed in order to discover new neighbourhoods (or ‘basins’) of lower energy, which may then be explored by a local procedure such as gradient descent. As such algorithms discover progressively smaller energy values, the remaining lower-energy basins form a decreasing sequence of sets. Viewing the optimisation domain heuristically as a probability space and these basins as events, the discovery of smaller energy values can then also be likened to rare-event sampling.

In this analogy, the local perturbation step plays a similar role to the proposal step in a Markov Chain Monte Carlo (MCMC) sampler (see [4], [28]). Thus in order to enhance performance, one may explore the use of alternative MCMC proposal distributions developed in the context of rare event sampling as alternative perturbation steps within stochastic optimisation routines. This is the approach we take in the present paper.

To illustrate the potential benefit of this approach, consider an energy landscape having multiple, well-separated basins whose minimum energies are approximately equal to the global minimum. Then, if $X_0$ lies in one such basin, separation means that local perturbations are not well suited to the direct discovery of another basin. Instead, algorithms using local perturbations to minimise over such a landscape should be non-monotonic, accepting transitions from $X_0$ to states of higher energy in the hope of later reaching lower-energy basins. In contrast, since non-local perturbation steps offer the possibility of direct moves between distant low-energy basins, they may possibly be effective on such surfaces within a monotonic optimisation algorithm. In this paper we explore the use of a particular non-local perturbation, the ‘skipping perturbation’ of [24].

Although other non-local perturbations have been proposed in the literature (see for example [1, 16, 27, 33, 34, 36] in the context of MCMC), skipping has the advantage of being just as straightforward to implement as a local random walk perturbation. That is, it requires no additional information about the energy landscape beyond the ability to evaluate it pointwise.

In principle, the skipping perturbation, regarded as a sampling step within the framework of [31], can be paired with any optimisation step and any stopping rule. For demonstrations in this paper we restrict ourselves to a simple stopping rule and state-of-the-art quasi-Newton methods. Overall the method can be understood as a modification of the basin hopping (BH) algorithm [17, 37], which combines local optimisation with perturbation steps and requires only pointwise evaluations of the energy function $f$ and its gradient. The resulting ‘basin hopping with skipping’ (BH-S) algorithm is thus as generally applicable as the BH algorithm.

The BH algorithm works as follows: the current state $X_n$ is perturbed via a random walk step to give $Y_n$ which is, in turn, mapped via deterministic local minimisation to a local minimum $X_{n+1}$. This local minimum point is then either accepted or rejected as the new state with a probability given by the Metropolis acceptance ratio, and the procedure is repeated until a pre-determined stopping criterion is met. Due to its effectiveness and ease of implementation, the BH algorithm has been used to solve a wide variety of optimisation problems (see [9, 25, 26] for more details).

In contrast with the non-monotonic BH algorithm, BH-S is monotonic and replaces the random walk step with a skipping perturbation over the sublevel set of the current state $X_n$. Like a flat stone skimming across water, this involves repeated perturbations in a straight line until either a point of lower energy is found, or the skipping process is halted. The BH-S algorithm, which was first outlined in [24], thus provides a direct mechanism to escape local minima which contrasts with the indirect approach taken by BH. Another perspective is that BH-S alters the balance between the computational effort expended on local optimisation versus the effort spent on perturbation, typically increasing the latter while decreasing the former (cf. Table 1 below).
This paper investigates the interaction between the choice of algorithm and type of landscape by using a set of benchmark problems to present a systematic overview on the types of optimisation problem on which BH-S tends to outperform BH (and vice versa). The rest of the paper is structured as follows: Sect. 2 introduces the algorithms, empirical results are presented and discussed in Sects. 3, and 4 concludes.

2 The BH-S algorithm

Consider the box-constrained global optimisation problem on a rectangular subdomain $D \subseteq \mathbb{R}^d$, of the form

$$
\min f(x) \quad \text{s.t. } x \in D := \prod_{i=1}^{d} [l_i, u_i],
$$

for some scalars $l_i \leq u_i$, $i = 1, \ldots, d$. In the rest of the paper, we will often refer to $f$ as the energy function and to its graph as the energy landscape. This terminology, which is similar to that of simulated annealing, is appropriate, since the BH algorithm was originally conceived as a method to find the lowest energy configuration of a molecular system [37]. In this section, we review the BH algorithm and then introduce basin hopping with skipping (BH-S).

2.1 Basin hopping algorithm

The core idea of the basin hopping algorithm [37], which is presented in Algorithm 1, is to supplement local deterministic optimisation by alternating it with a random perturbation step capable of escaping local minima. More specifically, inside the RandomPerturbation procedure at step 5 of Algorithm 1, a random perturbation $W \in \mathbb{R}^d$ is drawn and added to the current state $X_n$ giving a state $Y_n = X_n + W$. Most commonly, the increment $W$ is either spherically symmetric or has independent coordinates. The state $Y_n$ becomes the starting point of a deterministic local minimisation routine. In our implementation of Algorithm 1, the LocalMinimisation procedure at step 6 is performed using the limited-memory BFGS algorithm [18], a quasi-Newton method capable of incorporating boundary constraints, although we note that other choices are possible. The resulting local minimum $U_n$ is then either accepted or rejected as the new state with probability equal to

$$
\min \left(1, \exp \left(-\frac{f(U_n) - f(X_n)}{T}\right)\right),
$$

where $T \geq 0$ is a fixed temperature parameter. This means, in particular, that downwards steps for which $f(U_n) < f(X_n)$ are always accepted. The BH algorithm prescribes to repeat this basic step until a pre-defined stopping criterion is satisfied. Commonly used stopping criteria for the BH algorithm include, among others, a limit on the number of evaluations of the function $f$ or the absence of improvement over several consecutive iterations [25, 29]. The monotonic basin hopping method introduced in [17] is the BH variant corresponding to the limiting case $T = 0$, in which all steps that increase the energy are rejected.

In this paper, similarly to [19], we use the term basin to refer to any basin of attraction with respect to the chosen local search procedure (in our case BFGS). Basin hopping can thus be viewed as a random walk on the set of local minima of the energy landscape which, with its transition probabilities favouring moves to lower minima, aims to find the global minimum.
and, hence, to solve global optimisation problems. Its transition probabilities depend in a complex way on the current position, the landscape, and the perturbation step. The BH-S algorithm introduced in the next section modifies these transition probabilities, aiming to accelerate optimisation.

2.2 Skipping perturbations and the BH-S algorithm

In this subsection we introduce the BH-S algorithm, which differs from BH only in the perturbation step of line 5 in Algorithm 1. Instead of the random walk perturbation described above, the RandomPerturbation procedure described in Algorithm 2 below is applied in order to obtain \( Y_n \). The LocalMinimisation and acceptance steps remain identical to those in Algorithm 1.

Given the current state \( X_n \) and a fixed probability density \( q \) on \( \mathbb{R}^d \), the random walk perturbation of the BH algorithm can be understood as drawing a state \( Y_n \) from the density \( y \mapsto q(y - X_n) \). In contrast, the skipping perturbation of BH-S depends on both the current state \( X_n \) and a target set \( C \subseteq \mathbb{R}^d \) of states. The target set \( C_n \) for the \( n \)-th skipping perturbation is the sublevel set of the energy function \( f \) at the current point \( X_n \), i.e.,

\[
C_n := \{ x \in D : f(x) \leq f(X_n) \} \subset \mathbb{R}^d.
\]  

A state \( Z_1 \) is drawn according to the density \( q \) just as in the random walk perturbation and, if \( Z_1 \) does not lie in the target set \( C_n \), further states \( Z_2, Z_3, \ldots \) are drawn such that \( X_n, Z_1, Z_2, \ldots \) lie in order on a straight line, with each distance increment \( |Z_{j+1} - Z_j| \) having the same distribution as that of \( |Z_1 - X_n| \) conditioned on the line’s direction \( \frac{Z_1 - X_n}{|Z_1 - X_n|} \). The first state of this sequence to land in the target set \( C_n \) becomes the state \( Y_n \). If \( C_n \) is not entered before the skipping process is halted, then \( Y_n \) is set equal to \( X_n \). The skipping perturbation is illustrated in Fig. 1.

More precisely, let \( x = (r, \varphi) \) be polar coordinates on \( \mathbb{R}^d \) with the angular part \( \varphi \) lying on the \( d - 1 \) dimensional unit sphere \( S^{d-1} \). Write \( \varphi \mapsto q_\varphi(\varphi) \) for the marginal density of \( q \) with respect to the angular part \( \varphi \), which we may call the directional density (and which we

---

**Algorithm 1: Basin hopping**

1. Generate a random initial state \( Y_0 \in D \);
2. \( X_0 = \text{LocalMinimisation}(Y_0) \);
3. \( n = 0 \);
4. while Stopping criterion for \( \{ X_j \}_{j \leq n} \) is not satisfied: do
5. \( Y_n = \text{RandomPerturbation}(X_n) \);
6. \( U_n = \text{LocalMinimisation}(Y_n) \);
7. Generate \( V \sim \text{Uniform}([0, 1]) \);
8. if \( V < \min \left( 1, \exp \left( - \frac{f(U_n) - f(X_n)}{T} \right) \right) \) then
9. \( X_{n+1} = U_n \);
10. else
11. \( X_{n+1} = X_n \);
12. end
13. Increase \( n \) by 1;
14. end
The skipping perturbation is therefore $Y_n = Z_4$. The LocalMinimisation step maps $Y_n$ to the local minimum $U_n$.
we set \( Y_n = X_n \). Note that although in [24] the halting index \( K \) can be randomised, in the present setting with a known bounded domain \( D \) it is sufficient to consider only fixed halting indices.

For clarity, in the remainder of the paper we will understand the BH algorithm to mean setting \( K = 1 \) in Algorithm 2. In all simulations we set the perturbation \( q \) to be a spherically symmetric and Gaussian with standard deviation \( \sigma \), although other choices are possible (see the discussion in Sect. 4.1 of [24]). In the next section we explore for which types of energy function \( f \) BH-S offers an advantage over BH, and also discuss the choice of the halting index.

Algorithm 2: RandomPerturbation subroutine for BH-S

\[
\begin{align*}
\text{Input} & : \text{State } X_n \in \mathbb{R}^d \\
\text{Output: } & \text{Randomly perturbed state } Y_n \in \mathbb{R}^d \\
1 & \text{Set } Z_0 = X_n; \\
2 & \text{Generate an initial perturbation } W \text{ distributed according to the density } w \mapsto q(w - X_n); \\
3 & \text{Calculate the direction } \Phi = \frac{(W - X_n)}{\|W - X_n\|}; \\
4 & \text{Set } k = 1 \text{ and } Z_1 := W; \\
5 \text{while } & f(Z_k) > f(X_n) \text{ and } k < K \text{ do} \\
6 & \text{Generate an independent distance increment } R \text{ distributed as } \|W - X_n\| \text{ given } \Phi; \\
7 & \text{Set } Z_{k+1} = Z_k + \Phi R; \\
8 & \text{Increase } k \text{ by one; } \\
9 & \text{Set } Y_n := Z_k; \\
\end{align*}
\]

3 Empirical results

In this section, we aim to explore on which types of optimisation problem BH-S tends to outperform BH and vice versa using a set of benchmark energy landscapes with known global minima from [7, 11, 35]. To facilitate discussion of landscape geometry, we initially restrict attention to two-dimensional energy functions, before considering higher dimensions in Sect. 3.6.

In Sect. 3.3 we show that, if an energy landscape has distant basins (recall that with the word ‘basin’ we refer to the set of points from which the local optimisation method will lead to a given local minimum) then BH-S tends to offer an advantage. Otherwise, as described in Sect. 3.4, BH is to be preferred since any benefit from BH-S is then typically outweighed by its additional computational overhead. We also explore the effect of the state space dimension \( d \) on the performance of both algorithms and offer guidance on tuning the BH-S method, including strategies to improve exploration of challenging energy landscapes.

3.1 Methodology

For each benchmark energy landscape, we compare the performance of BH-S to that of BH with temperature \( T = 1 \). (Note that the monotonic basin-hopping algorithm was also compared to BH-S. Since this gave the same qualitative conclusions, however, in the interests
of clarity we present only the former comparison.) In both cases, we take the density $q$ of the initial perturbation as the centred Gaussian

$$q \sim \mathcal{N}(0, \sigma^2 \cdot I_d),$$

where $I_d$ is the $d \times d$ identity matrix and the parameter $\sigma$ allows for tuning, as follows. Both the BH and BH-S algorithms are run on a set of uniformly distributed initial states $I := \{X_0^{(n)} \in D, \ n = 1, \ldots, |I|\}$. These initial states are used sequentially until the computational budget of 300 seconds of execution time has elapsed, and the corresponding set of final states is recorded. To account for numerical tolerance, we consider a run to have succeeded if its final state lies in the set $G$ of all points within distance $10^{-5}$ of any global minimiser of $D$. This tolerance was chosen to exclude all non-global minima for all benchmark landscapes.

The performance of each algorithm is assessed with respect to two metrics:

- **Effectiveness**, (also known as **reliability**), defined as the proportion of runs terminating in $G$.
- **Efficiency**, defined as the number of runs terminating in $G$.

We write $\rho_c$ and $\rho_s$ for the effectiveness of the BH and BH-S algorithms respectively, while $\epsilon_c$ and $\epsilon_s$ denote their respective efficiencies. The BH and BH-S algorithms are individually tuned for each function by selecting $\sigma$ and $K$ to maximise their efficiency.

In order to understand the role played by the skipping perturbation, we also record diagnostics on the average size of perturbations. For each new state $X_{n+1} \neq X_n$ accepted in Algorithm 1, define the perturbation distance $J$ as $\|Y_n - X_n\|$, the Euclidean distance between the state $X_n$ at step $n$ and its perturbation $Y_n$. For each run of an algorithm, the mean $J$ of these perturbation distances is recorded. Then for each 300 second budget, the expected mean jump distance $\nu$ is the average $\nu := N^{-1} \sum_{n=1}^{N} J^{(n)}$, where $N$ is the number of runs realised within the execution time budget. For the BH-S algorithm, $\nu$ is calculated separately for the accepted random walk perturbations (that is, those for which $Y_n = Z_1$ in Algorithm 2) and the accepted skipping perturbations (those for which $Y_n = Z_k$ with $k \geq 2$ in Algorithm 2), denoting these by $\nu_1$ and $\nu_s$ respectively.

The simulations were conducted on a single core using Python 3.7, using the basinhopping and limited-memory BFGS routines in SciPy version 1.6.2 for the BH algorithm. Results for all considered landscapes are presented in the Appendix.

### 3.2 Exploratory analysis

As an exploratory comparison between BH and BH-S, their relative effectiveness $\rho_s/\rho_c$ and relative efficiency $\epsilon_s/\epsilon_c$ are plotted for each benchmark energy landscape in Fig. 2.

Landscapes in the first quadrant of Fig. 2 represent cases where the BH-S algorithm exhibits both greater effectiveness and greater efficiency than BH. The common feature among these landscapes, which are plotted in the Appendix, might be called **distant basins**: that is, basins separated by sufficient Euclidean distance that the random walk performed by the BH algorithm is unlikely to transition directly between them. While indirect transitions between such basins may be possible, they require a suitable combination of steps to be made. Such indirect transitions may carry significant computational expense, for example if suitable combinations of steps are long or relatively unlikely. In the BH-S algorithm, by contrast, the linear sequence of steps taken by the skipping perturbation enables direct transitions even between distant basins.
Conversely, landscapes lying in the lower-left quadrant of Fig. 2 represent cases where the BH-S algorithm is both less reliable and less efficient than BH. As explored more extensively later in Sect. 3.4, for each of these landscapes, if the energy of the state $X_n$ is close to the global minimum value then the corresponding sublevel set $C_n$ has almost zero volume. This means that even if the skipping perturbation traverses the distance between basins, the states $Z_1, \ldots, Z_k$ are unlikely to fall in $C_n$ due to its small volume. Since the BH algorithm is non-monotonic, it does not suffer from the same issue and outperforms BH-S for these landscapes.

Figure 2 displays a positive correlation between relative efficiency $\mathcal{E} = \ln(\epsilon_s/\epsilon_c)$ versus relative effectiveness $\mathcal{P} = \ln(\rho_s/\rho_c)$ for the BH and BH-S algorithms on benchmark energy landscapes.

Further exploratory analysis is provided in Table 1, which indicates average execution time spent on the perturbation versus the local minimisation steps for each algorithm. To facilitate comparison between the two algorithms, in each case the total execution time spent is normalised by the algorithm’s efficiency, as defined in Sect. 3.1. This demonstrates that the BH algorithm spends most execution time on the local minimisation step, with relatively little devoted to the perturbation step. While the ratio between execution time spent on local minimisation and perturbation is more problem-dependent for BH-S, the balance appears to be shifted in favour of perturbation.

The BH-S perturbation step is more expensive by construction, since it requires between 1 and $K$ evaluations of the energy function $f$ (depending on the sublevel set of the current state), whereas each BH perturbation requires just one evaluation of $f$. However in Table 1, for the Damavandi, Schwefel, Modified Rosenbrock and Egg-holder functions for which BH-S works well (cf. Fig. 2), after normalisation the BH-S algorithm spent approximately the
Table 1  Execution time spent on the perturbation and local minimisation steps by the BH and BH-S algorithms for the test functions discussed in Sects. 3.3–3.5, normalised by efficiency $\epsilon$.

| Location in Fig. 2 | Function      | BH          | Local Minimisation | BH-S            | Skipping Perturbation | Local Minimisation |
|--------------------|---------------|-------------|--------------------|------------------|-----------------------|--------------------|
| First quadrant     | Egg-holder    | 0.72        | 6.76               | 0.73             | 0.19                  |
|                    | Modified Rosenbrock | 0.46        | 13.53              | 0.22             | 0.10                  |
| Third quadrant     | Mishra-03     | 0.02        | 1.75               | 1.74             | 3.21                  |
|                    | Whitley       | 0.01        | 0.68               | 4.92             | 1.58                  |
| Special cases      | Rosenbrock    | 0.01        | 0.13               | 0.05             | 0.01                  |
|                    | Styblinks     | 0.02        | 0.06               | 0.06             | 0.01                  |
same or less execution time than BH on perturbation, in addition to spending less execution time on local minimisation. Thus, for these landscapes which favour BH-S, perturbation steps were not only less frequent for BH-S (again, after normalisation by efficiency) than BH, but the monotonic BH-S perturbations also reduced the total computational burden arising from the local minimisation step.

Conversely it was noted above for landscapes in the third quadrant of Fig. 2, if the energy of the state $X_n$ is close to the global minimum value, then the corresponding sublevel set $C_n$ has almost zero volume. This represents the worst case for the BH-S perturbation: if the states $Z_1, \ldots, Z_k$ all lie outside the sublevel set, then the perturbation requires the maximum number $k$ of evaluations of the energy function. However, the perturbed state $Y_n$ is rejected and $X_{n+1} = X_n$, so the optimisation procedure does not advance. Indeed for the Mishra-03 and Whitley functions in Table 1, the efficiency-normalised execution time invested in perturbations is two orders of magnitude greater for BH-S than for BH. For these landscapes, the efficiency normalised computational burden from local minimisation is also observed to be greater for BH-S than for BH, although the reasons for this are less clear.

Guided by the exploratory analysis of Fig. 3, in Sects. 3.3–3.5 we study the performance of both algorithms on specific energy landscapes in greater detail.

### 3.3 Landscapes favouring the BH-S algorithm

Figure 3 plots two landscapes from the first quadrant of Fig. 2—that is, landscapes which favour the BH-S algorithm over BH. For each landscape, the sublevel set of a level above the global minimum value is also plotted. The Modified Rosenbrock energy function is given by
\[ f(x) = 74 + 100(x_2-x_1^2)^2 + (1-x_1)^2 - 400 \exp \left( \frac{- (x_1+1)^2 + (x_2+1)^2}{0.1} \right), \]
and we take the domain \( D = [-2, 2]^2 \), with global minimum \( x^* = (-0.95, -0.95) \) [7].

The Egg-Holder energy function is
\[ f(x) = -(x_2 + 47) \sin \left( \sqrt{|x_2 + \frac{x_1}{2} + 47|} \right) - x_1 \sin \left( \sqrt{|x_1 - (x_2 + 47)|} \right), \]
and we take the domain \( D = [-512, 512]^2 \), with global minimum at \( x^* = (512, 404.2319) \) [11].

Observing Figure 3a, the modified Rosenbrock function has two basins: a larger basin with a U-shaped valley and a smaller, well-shaped basin. To transition from the minimum of the valley to the minimum of the well, the BH algorithm would require a relatively large perturbation step landing directly in the well, otherwise the local optimisation procedure would take it back to the minimum of the valley. Even for an optimal choice of \( \sigma \), which would require \textit{a priori} knowledge about the landscape, such perturbations would be unlikely.

In contrast, if the initial point \( X_0 \) lies at the minimum of the valley, the BH-S algorithm aims to skip across the domain and enter its sublevel set \( C_0 \) as defined in (2). From Fig. 3c, this will correspond to entering an approximately circular basin near the point \((-1, -1)\) in the domain. By Algorithm 2, the skipping perturbation has the potential to enter the basin that the straight line issuing from \( X_0 \) in the initial direction \( \Phi \) in Algorithm 2 intersects it. In particular, this ability is robust to the choice of standard deviation \( \sigma \) provided that the halting index \( K \) is chosen appropriately (see the discussion on tuning in Sect. 3.7 below).

Figure 3b illustrates that the Egg-Holder function has multiple basins, many of which have near-global minima. Figure 3d shows that the deepest basins lie in four groups, one group per corner of the domain. Within each group, the basins are close in the Euclidean distance and so perturbations are likely to enter different basins within that group. Also, the basins in each group have similar depths (that is, similar local minimum energies), making the acceptance ratio in Algorithm 1 high for such within-group perturbations. As a result the BH algorithm is likely to walk regularly between within-group local minima. Also from Fig. 3b, the Egg-Holder function has shallower basins distributed throughout its domain. As discussed in Sect. 3.2 these provide an indirect, although potentially computationally expensive, route for BH to cross between the four groups of Fig. 3d.

However between groups the Euclidean distance is large, creating the same challenge for BH as with the modified Rosenbrock function: even for optimally chosen \( \sigma \), which would require \textit{a priori} knowledge of the landscape, transitions between groups are relatively rare.

In contrast, the BH-S algorithm is capable of moving between the four groups in Fig. 3d provided the initial direction \( \Phi \) of its skipping perturbation intersects a different group. The likelihood of such an intersection is increased by both the length of the skipping chain and the use of periodic boundary conditions in the BH-S algorithm, and is again robust with respect to the choice of standard deviation \( \sigma \).

Regarding the application of periodic boundary conditions to the domain \( D \), we have argued that they are natural for BH-S, since otherwise long skipping chains would tend to exit the domain \( D \). In contrast, they are not implemented for the BH algorithm in the results of Fig. 2 and Table 3. One may therefore ask whether it is their use, rather than the skipping perturbation of BH-S, which yields any observed improvement. To explore this, Table 2 illustrates the effect of imposing periodic boundary conditions on the performance of both the BH and BH-S algorithms. Interestingly the performance of BH-S on the Egg-Holder
Table 2  effectiveness $\rho$ and efficiency $\epsilon$ of the BH and BH-S algorithms, with and without the application of periodic boundary conditions, for the test functions discussed in Sects. 3.3–3.5.

| Location in Fig. 2 | Function      | Boundary Condition | $\rho_c$ | $\epsilon_c$ | $\rho_s$ | $\epsilon_s$ |
|-------------------|---------------|-------------------|---------|-------------|---------|-------------|
| First quadrant (Section 3.3) | Egg-Holder     | Periodic          | 1.8%       | 77          | 29.4%   | 280         |
|                    |               | Non-periodic      | 1.4%       | 58          | 81.7%   | 815         |
|                    | Modified Rosenbrock | Periodic      | 18.8%      | 783         | 100%    | 1249        |
|                    |               | Non-periodic      | 17.2%      | 597         | 96.7%   | 321         |
| Third quadrant (Section 3.4) | Mishra-03     | Periodic          | 77.9%      | 811         | 6.2%    | 78          |
|                    |               | Non-periodic      | 49.6%      | 474         | 68.4%   | 576         |
|                    | Whitley       | Periodic          | 90.1%      | 137         | 13.9%   | 33          |
|                    |               | Non-periodic      | 86.8%      | 138         | 13.4%   | 34          |
| Special cases (Section 3.5) | Rosenbrock    | Periodic          | 100%       | 942         | 100%    | 1376        |
|                    |               | Non-periodic      | 100%       | 1006        | 100%    | 1524        |
|                    | Styblinski    | Periodic          | 99.7%      | 366         | 99.9%   | 932         |
|                    |               | Non-periodic      | 99.4%      | 526         | 92.3%   | 784         |
landscape is improved without their use. This tendency appears to be driven by the proximity of its global minimiser $x^*$ to the boundary. In general, it is clear from Table 2 that for both algorithms their benefit or disbenefit is problem-dependent and the skipping perturbation explains a distinct and material part of the observed improvements relative to BH.

It can be observed from Table 3 in the Appendix that the expected mean jump distances $\nu_s$ and $\nu_c$ (defined in Sect. 3.1) typically satisfy $\nu_s \gg \nu_c$ for landscapes in the first quadrant of Fig. 2. This confirms quantitatively the success of BH-S in hopping between distant basins. The cost of this feature is that the BH-S skipping perturbation is more computationally intensive than the random walk perturbation of BH.

Without skipping (that is, using the halting index $K = 1$ in Algorithm 2), BH-S would reduce to the monotonic basin hopping method of [17] and the initial perturbation $W$ of Algorithm 2 would simply be either accepted or rejected. One may therefore also ask whether this increase in the expected mean jump distance is induced by the skipping mechanism of BH-S, or simply by its monotonicity. To address this, recall that Algorithm 2 first perturbs the current state $X_n$ to give an initial perturbation $Z_1 := W$. Then if $f(W) > f(X_n)$, the initial perturbation is modified to $Z_2$, and so on, until either a state $Z_k$ is generated with $f(Z_k) \leq f(X_n)$ or skipping is halted. If such a $Z_k$ is found, then it may be accepted by setting $Y_{n+1} = Z_k$ and performing a local optimisation or rejected. The Appendix records the proportion of accepted BH-S perturbations for which $k > 1$. Indeed, for many landscapes in the first quadrant of Fig. 2 this proportion is 100%. That is, for such landscapes, each accepted perturbation $X_{n+1}$ required the skipping mechanism since none of the initial perturbations had lower energy than the current state $X_n$.

### 3.4 Landscapes favouring the BH algorithm

Figure 4 plots two landscapes from the third quadrant of Fig. 2, on which the BH algorithm outperforms BH-S, each with two sublevel sets above the global minimum value. The Mishra-03 function $f : \mathbb{R}^2 \to \mathbb{R}$ is given by

$$f(x) := \sqrt{|\cos(\sqrt{|x_1^2 + x_2^2|})|} + 0.01(x_1 + x_2),$$

and, on the domain $D = [-10, 10]^2$, has $x^* = (-8.466, -10)$ as global minimum [11]. The Whitley function $f : \mathbb{R}^2 \to \mathbb{R}$, given by

$$f(x) := \sum_{i=1}^{2} \sum_{j=1}^{2} \left( \frac{(100(x_i^2 - x_j)^2 + (1 - x_j)^2)^2}{4000} - \cos(100(x_i^2 - x_j)^2 + (1 - x_j)^2 + 1) \right),$$

has global minimum $x^* = (1, 1)$ on the domain $D = [0, 1.5]^2$ [11].

From Fig. 4a, the Mishra-03 function is highly irregular and has many basins which appear almost point-like. Figure 4e confirms that the situation outlined in Sect. 3.2 applies to this landscape. That is, for states $X_n$ with energy close to the global minimum value $f(x^*)$, the corresponding sublevel set $C_n$ has almost zero volume and the states $Z_1, Z_2, \ldots$, of Algorithm 2 are unlikely to fall in $C_n$.

The deepest basins of Mishra-03 form groups arranged in concentric circular arcs. Since the Euclidean distances both within and between these groups are relatively small, the BH algorithm is able to move frequently both within and between groups without requiring precise tuning of the standard deviation parameter $\sigma$. In particular, it outperforms BH-S on this landscape.
Similarly, by observing Fig. 4d, the deepest basins of the Whitley function can be seen either as forming one group, or as a small number of groups close to each other. Thus, as for Mishra-03, the BH algorithm is able to move frequently between them while nevertheless being robust to the choice of the standard deviation parameter $\sigma$. As with Mishra-03, however, from Fig. 4f the sublevel sets $C_n$ corresponding to near-global minimum states $X_n$ have low volume. Thus it is more challenging for BH-S to transition between the deepest basins, and BH outperforms BH-S on this landscape.

These limitations of the BH-S routine can be mitigated by alternating between a monotonic and non-monotonic perturbation step. In Sect. 3.8 we provide a discussion on how this alternating perturbation can be implemented.
Fig. 5 Comparison of BH and BH-S performance when applied to the Schwefel-07 function while varying the dimension $d$ of the domain $D$. We set $\sigma = 20$ for both algorithms and the BH-S has halting index $K = 50$. These parameters were close to optimal for both algorithms. Each simulation used an execution time budget of 300s.

3.5 Special cases

It was noted in Sect. 3.2 that for several landscapes lying near the vertical axis, both BH and BH-S algorithms have effectiveness close to 100%. For these surfaces BH-S typically has greater efficiency simply because of its monotonicity, since no further computational effort is expended on local optimisation once the global optimum is reached. The Holder Table and Carrom Table landscapes have multiple distant ‘legs’, each leg being the basin of a global minimum point. In this case, the ability of BH-S to skip between distant basins is not reflected in either its efficiency or its effectiveness, although it would clearly be beneficial if the goal was to identify the number of global minima in the landscape.

3.6 Scaling with dimension

In this section we aim to illustrate the performance of the BH-S algorithm as the dimension of the optimisation problem increases. For this we focus on Schwefel-07, a landscape with ‘distant basins’ which is also defined for higher dimensions. It is given by the function $f_d: \mathbb{R}^d \rightarrow \mathbb{R}$, where

$$f_d(x) = 418.9829 \times d - \sum_{i=1}^{d} x_i \sin(\sqrt{|x_i|}),$$

and has global minimum $x^*(421.0)^d$ on the domain $D_d = [-500, 500]^d$ [11].

Figure 5a illustrates that the effectiveness and also the efficiency of both algorithms decrease approximately linearly with increased dimension. Recall that relative to BH, the strength of BH-S lies in its ability to transition directly between distant basins. With reference to Algorithm 2, in order to transition directly to the global minimum basin, it is necessary for the line from the current state in the random direction $\Phi$ to intersect that basin. As $\Phi$ is drawn from a space of dimension $d - 1$, heuristically this becomes less likely as $d$ increases.

In contrast, the BH algorithm should rely to a greater extent on indirect transitions from its current state to the global minimum. By statistical independence, the probability of a particular indirect transition is the product of the probabilities of its constituent steps. Since
the probability of each step decays with dimension as discussed above for BH-S, this suggests that the performance of BH will degrade more rapidly with dimension than BH-S.

This is illustrated in Fig. 5a, where BH fails to locate $x^*$ within the 300 second budget for any dimension $d \geq 4$, while BH-S continues to locate $x^*$ (albeit with decreasing effectiveness and efficiency) until dimension $d = 11$. Indeed, the effectiveness of BH-S for this landscape is above 50% for dimensions $d \leq 7$.

### 3.7 Adapting parameter values to the landscape

Both BH and BH-S have the parameter $\sigma$, the standard deviation of the centred Gaussian density $q$ used to generate the initial perturbation. As noted above, the initial perturbation is analogous to a Metropolis-Hastings (MH) proposal in MCMC. The MH literature highlights the importance of tuning such proposals, guided either by theory or by careful experimentation [6, 22]. Following this analogy, in this section we explore the choice of $\sigma$ and also of the BH-S halting index $K$. To facilitate this discussion we restrict attention to the two-dimensional Egg-Holder function.

Figure 6 plots the effectiveness and efficiency of both BH and BH-S as $\sigma$ varies between 0 and 300 (recall that the domain $D = [-512, 512]^2$; also, we set $K = 25$ for BH-S). Clearly, for both algorithms $\sigma$ should not be very small ($\leq 10$). In that case the random walk step $W$ is likely to land in the same basin as the current point $X_n$, so that the local optimisation step maps the perturbation back to $X_n$ and the algorithms do not advance.

We note first from Fig. 6 that both the effectiveness and efficiency of the BH algorithm increase approximately linearly within this range as $\sigma$ increases. As discussed in Sect. 3.3, this reflects the fact that as $\sigma$ increases, direct transitions between the four groups of deepest basins become more likely. In contrast, and again confirming the discussion in Sect. 3.3, both the efficiency and effectiveness of BH-S appear to be rather robust to the choice of $\sigma$.

Figure 7 illustrates the impact on effectiveness and efficiency of the choice of halting index $K$. From Algorithm 2, the maximum linear distance covered by the skipping procedure is $\sum_{k=1}^{K} R_k$, where each $R_k$ is distributed as the radial part of a centred Gaussian with standard deviation $\sigma$. This suggests that $K$ should not be too small, and the plot of efficiency in Fig. 7b indicates that $K$ should be at least 5 in our example (by default we take $K = 25$).

It is seen that provided ($K \leq 5$), increasing $K$ tends to increase effectiveness while decreasing efficiency. This reflects the fact that larger $K$ allows the skipping procedure to travel further, thus increasing the likelihood of a direct transition to the global minimum basin, after which the BH-S algorithm would stop due to its monotonicity. In this way, greater $K$ increases effectiveness. On the other hand, greater $K$ increases the length of unsuccessful skipping trajectories. That is, each time the perturbed state $Y_n$ of Algorithm 2 is not accepted (after the local minimisation step of Algorithm 1), the landscape is evaluated up to $K$ times without advancing the optimisation. This implies that increased $K$ also typically leads to decreased efficiency.

The considerations with respect to parameters discussed above for the BH-S algorithm can be summed up as follows. It should first be checked that $\sigma$ is large enough that the initial perturbation regularly falls outside the basin of the current state $X_n$. Having selected $\sigma$, $K$ should then be taken large enough that the skipping procedure regularly enters the sublevel set $C_n$. A practical suggestion here is to choose $K$ so that $K\sigma$ exceeds the diameter of the domain $D$.

Figure 8 confirms these guidelines in higher dimensions, by plotting the BH-S effectiveness and efficiency in dimensions up to 10 as $\sigma$ varies with the fixed choice $K = 50$. It
confirms that these performance metrics are relatively robust to the value of $\sigma$, provided that $\sigma$ is sufficiently large.

### 3.8 Alternating BH-S and BH

In this section we explore a hybrid approach which is intended to overcome the challenges identified in Sect. 3.4 for the monotonic BH-S algorithm by regularly including non-monotonic BH steps. Figure 9 plots the effectiveness and efficiency metrics for this hybrid algorithm on various landscapes, as the ratio between BH-S and BH steps varies.
Fig. 8 Performance of the BH-S algorithm on the Schwefel-07 function for different combinations of domain dimension $d$ and perturbation variance $\sigma$. Note: the halting index was set to $K = 50$ with an execution time budget of 300s for all simulations.

Fig. 9 Performance of the hybrid algorithm with varying proportions of BH to BH-S steps.

It can be seen that for the Mishra-03 and Whitley functions of Sect. 3.4, this hybrid improves both effectiveness and efficiency compared to BH-S. Indeed, the performance of a 1:1 ratio of BH and BH-S steps is comparable to that of BH for these landscapes. Further, on the landscapes of Sect. 3.3, this 1:1 ratio achieves performance superior to that of BH and somewhat comparable to that of BH-S. Thus if little is known about the problem’s energy landscape a priori, these results indicate that the 1:1 hybrid is to be preferred.

4 Discussion and future work

Basin hopping with skipping (BH-S) is a global optimisation algorithm inspired by both the basin hopping algorithm and the skipping sampler, an MCMC algorithm. As such, the MCMC literature also suggests potential extensions of this work. In adaptive MCMC, parameter tuning is an online procedure driven by the progress of the chain [2]. A similar idea has been proposed for BH in [8] and is part of the SciPy implementation of the BH method. We believe it could be interesting as future work to devise an adaptive scheme for the halting index $K$ and the standard deviation $\sigma$, possibly reducing in this way the amount of parameter adaptation required to implement BH-S. Alternatively, a priori global knowledge of the energy landscape could be used where available (see [3]), for example, by favourably biasing the choice of skipping direction.

We also explored the idea of sampling several directions and skipping in all of them simultaneously. As a negative finding, we report that preliminary results indicated that the
computational effort is best spent searching over a single, rather than multiple, directions. Our heuristic explanation is that the line is the shortest route between two sets, and so is the most efficient way to cover distance. An alternative, more sophisticated approach would be to introduce multiple BH-S particles which explore the energy landscape in a coordinated way. This could for instance be inspired by selection-resampling procedures as in sequential Monte Carlo sampling [23], clustering as in [12, 13], or by an optimisation procedure such as particle swarm optimisation [14].

When tackling a challenging global optimisation problem one could also take a divide-and-conquer approach, sequentially (or hierarchically) splitting the problem into a number of simpler ones. The BH-S algorithm could, for instance, be applied to a (deterministic or stochastic) sequence of lower-dimensional linear sub-spaces. Related to this idea, it would be of interest to examine how incorporating the multi-level framework introduced in [19] influences the stochastic sequence of local minima returned by the BH-S algorithm.

**Funding**  MG was supported by a Queen Mary University of London Principal’s Studentship Award. JM was partially supported by EPSRC grant number EP/P002625/1 and by the Lloyd’s Register Foundation-Alan Turing Institute programme on Data-Centric Engineering under the LRF grant G0095. JV was supported by EPSRC grant number EP/R022100/1.

**Availability of data and material**  Not applicable.

**Code Availability**  Jupyter notebook files used to conduct simulations are available at https://github.com/ahw493/Basin-Hopping-with-Skipping.git.

**Compliance with ethical standards**  The authors declare there are no conflicts of interest when presenting this research.

**Open Access**  This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

**Appendix**  Table 3 records the results for all landscapes in Fig. 2. For each landscape, the parameters of BH and BH-S were chosen following the approach of Sect. 3.7. The following notation is used:

- \( \rho_c \) and \( \rho_s \) are the effectiveness of BH and BH-S respectively;
- \( \epsilon_c \) and \( \epsilon_s \) are the efficiency of BH and BH-S respectively;
- \( \nu_1 \) is the expected mean jump distance among random walk steps;
- \( \nu_s \) is the expected mean jump distance among skipping transitions, i.e., when \( k > 1 \);
- \( P_s \) is the probability that, conditional on the BH-S perturbation being accepted, skipping had occurred (\( k > 1 \));
- \( \nu_1 \) and \( \nu_s \) are the expected mean jump distances among random walk steps (\( k = 1 \)) and skipping steps (\( k > 1 \)), respectively.
| Function               | BH | BH-S |
|-----------------------|----|------|
|                       | $\sigma$ | $\rho_c$ | $\epsilon_c$ | $\nu_1$ | $\sigma$ | $K$ | $\rho_s$ | $\epsilon_s$ | $\nu_1$ | $\nu_s$ | $P_s$ |
| Carrom Table          | 2  | 99.8  | 556 | 1.5 | $\sqrt{2}$ | 10 | 100 | 925 | 2.9 | 9.4 | 86.8 |
| Cross in Tray         | 2  | 96.5  | 446 | 1.7 | $\sqrt{2}$ | 10 | 100 | 676 | 2.7 | 9.3 | 85.1 |
| Cross Leg Table       | 0.4 | 15.5  | 48  | 0.8 | $\sqrt{2}$ | 10 | 12.7 | 45  | 1.8 | 6.6 | 45.7 |
| Damavandi             | 0.1 | 0.2   | 3   | 0.5 | 0.3 | 150 | 32.9 | 28  | N/A | 34.9 | 100 |
| Eggcrate              | 1  | 99.7  | 377 | 1   | 1  | 10  | 99.7 | 647 | 1.9 | 7.6 | 94.8 |
| Egg-holder            | 100 | 2.2   | 13  | 12.5 | 10 | 25  | 38.7 | 116 | 7.1 | 178.1 | 98.6 |
| Function                        | BH | BH-S |
|--------------------------------|----|------|
|                                | $\sigma$ | $\rho_c$ | $\epsilon_c$ | $\nu_1$ | $\sigma$ | $K$ | $\rho_s$ | $\epsilon_s$ | $\nu_1$ | $\nu_s$ | $P_s$ |
| El Attar Vidyasagar Dutta EAVD | 8  | 99.6 | 231 | 3.2 | 5  | 10 | 63.4 | 393 | 4.8 | 5.8 | 39.3 |
| Freudenstein-Roth              | 2  | 82.6 | 176 | 1.6 | $\sqrt{2}$ | 10 | 97.2 | 551 | 4.9 | 11.3 | 99.5 |
| Holder Table                   | 2  | 99.8 | 453 | 1.4 | $\sqrt{2}$ | 10 | 100 | 695 | 2.4 | 9.1 | 82.4 |
| Keane                          | 2  | 47   | 272 | 1.8 | 0.9 | 25 | 53.6 | 301 | 1.6 | 12.4 | 96.6 |
| Mishra-03                      | 2  | 65.9 | 56  | 1.8 | $\sqrt{2}$ | 10 | 5.4 | 17  | 1.6 | 12.1 | 96.7 |
| Modified Rosenbrock            | 0.4| 5.3 | 1   | 0.8 | 0.4 | 25 | 83.8 | 31  | N/A | 7.7 | 100 |
| Function         | BH | BH-S |
|------------------|----|------|
|                  | $\sigma$ | $\rho_c$ | $\epsilon_c$ | $\nu_1$ | $\sigma$ | $K$ | $\rho_s$ | $\epsilon_s$ | $\nu_1$ | $\nu_s$ | $P_s$ |
| Price 02         | 2  | 44.4 | 234 | 1.8 | 0.9 | 25 | 60.6 | 208 | N/A | 17.1 | 100 |
| Rana             | 200 | 5.5 | 13 | 17.6 | 5 | 75 | 20.5 | 18 | 1.8 | 224.5 | 98.5 |
| Rosenbrock       | 0.2 | 99.3 | 149 | 0.6 | 2 | 10 | 100 | 695 | N/A | N/A | 0 |
| Schwefel-07      | 10 | 4.1 | 38 | 4 | 7 | 25 | 61.9 | 275 | N/A | 181.6 | 100 |
| Styblinski Tang  | 1  | 99.6 | 537 | 1.2 | 1 | 10 | 100 | 840 | N/A | 8.7 | 99.8 |
| Function  | BH-H | BH-S | \( \sigma \) | \( \sigma_c \) | \( \epsilon \) | \( \epsilon_s \) | \( v_1 \) | \( v_s \) | \( P_s \) |
|-----------|------|------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| Whitley   | 0.4  | 86.4 | 0.7         | 0.7         | 50          | 31.8        | 21          | 0.6         | 17          | 37.7        |
| Zirilli   | 0.2  | 99.9 | 0.6         | \( \sqrt{2} \) | 10          | 9.7         | 987         | 1.9         | 5.3         | 49.4        |
References

1. Andricioaei, I., Straub, J., Voterr, A.: Smart darting Monte Carlo. J. Chem. Phys. 114(16), 6994–7000 (2001)
2. Atchade, Y.F., Rosenthal, J.S.: On adaptive Markov chain Monte Carlo algorithms. Bernoulli 11(5), 815–828 (2005)
3. Baritompa, W., Dür, M., Hendrix, E.M., Noakes, L., Pullan, W., Wood, G.R.: Matching stochastic algorithms to objective function landscapes. J. Global Optim. 31(4), 579–598 (2005)
4. Brooks, S., Gelman, A., Jones, G., Meng, X.: Handbook of Markov Chain Monte Carlo. CRC Press, USA (2011)
5. Duan, Q., Kroese, D.: Splitting for optimization. Comput. Op. Res. 73, 119–131 (2016)
6. Gamerman, D., Lopes, H.: Markov chain Monte Carlo, stochastic simulation for bayesian inference, 2nd edn. Chapman & Hall/CRC, Boca Raton (2006)
7. Gavana, A.: Global optimization benchmarks and AMPGO. Retrieved April 20, 2021, from http://infinity77.net/global_optimization/test_functions.html, (2013)
8. Gehrke, R.: First-principles basin-hopping for the structure determination of atomic clusters. Freie Universität Berlin (2009). (PhD thesis)
9. Huang, R., Bi, L., Li, J., Wen, Y.: Basin hopping genetic algorithm for global optimization of ptco clusters. J. Chem. Inform. Model. 60, 2219 (2020)
10. Jain, P., Agogino, A.M.: Global optimization using the multistart method. J. Mech. Des. 115(4), 770–775 (1993)
11. Jamil, M., Yang, X.S.: A literature survey of benchmark functions for global optimisation problems. Int. J. Math. Model. Numer. Optim. 4(2), 150 (2013)
12. Kan, A.R., Timmer, G.T.: Stochastic global optimization methods part I: clustering methods. Math. Program. 39(1), 27–56 (1987)
13. Kan, A.R., Timmer, G.T.: Stochastic global optimization methods part II: multi level methods. Math. Program. 39(1), 57–78 (1987)
14. Kennedy, J., Eberhart, R.: Particle swarm optimization. In Proceedings of ICNN’95 - International Conference on Neural Networks, 4, 1942–1948, (1995)
15. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. Science 220(4598), 671–680 (1983)
16. Lan, S., Streets, J., Shahbaba, B.: Wormhole Hamiltonian Monte Carlo. In Proceedings of the 28th AAAI Conference on Artificial Intelligence. 1953–1959 (2014)
17. Leary, R.H.: Global optimization on funneling landscapes. J. Global Optim. 18(4), 367–383 (2000)
18. Liu, D., Nocedal, J.: On the limited memory BFGS method for large scale optimization. Math. Program. 45, 503–528 (1989)
19. Locatelli, M.: On the multilevel structure of global optimization problems. Comput. Optim. Appl. 30(1), 5–22 (2005)
20. Locatelli, M., Schoen, F.: Global optimization: theory, algorithms, and applications. SIAM (2013)
21. Martí, R.: Multi-start methods. In: Handbook of Metaheuristics, pp. 355–368. Kluwer Academic Publishers, Netherlands (2003)
22. Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., Teller, E.: Equation of state calculations by fast computing machines. J. Chem. Phys. 21(6), 1087–1092 (1953)
23. Moral, P.D., Doucet, A., Jasra, A.: Sequential Monte Carlo samplers. J. Royal Statist. Soc. Ser. B (Statist. Methodol.) 68(3), 411–436 (2006)
24. Moriarty, J., Vograinc, J., Zocca, A.: A Metropolis-class sampler for targets with non-convex support. Statist. Comput. 31(6), 1–16 (2021)
25. Olson, B., Hashmi, I., Molloy, K., Shehu, A.: Basin hopping as a general and versatile optimization framework for the characterization of biological macromolecules. Adv. Artif. Intell. 2012, 1–19 (2012)
26. Palei, M., Behler, J.: A flexible and adaptive grid algorithm for global optimization utilizing basin hopping Monte Carlo. J. Chem. Phys. 152, 094109 (2020)
27. Pompe, E., Holmes, C., Łatuszyński, K.: A framework for adaptive MCMC targeting multimodal distributions. Ann. Statist. 48(5), 2930 (2020)
28. Roberts, G.O., Rosenthal, J., et al.: General state space Markov chains and MCMC algorithms. Prob. Surv. 1, 20–71 (2004)
29. Rondina, G., da Silva, J.: Revised basin-hopping Monte Carlo algorithm for structure optimization of clusters and nanoparticles. J. Chem. Inform. Model 53(9), 2282–2298 (2013). (PMID: 23957311)
30. Rubinstein, R.: The cross-entropy method for combinatorial and continuous optimization. Methodol. Comput. Appl. Probab. 1(2), 127–190 (1999)
31. Schoen, F.: Stochastic techniques for global optimization: a survey of recent advances. J. Global Optim. 1(3), 207–228 (1991)
32. Schumer, M., Steiglitz, K.: Adaptive step size random search. IEEE Trans. Autom. Control 13(3), 270–276 (1968)
33. Sminchisescu, C., Welling, M.: Generalized Darting Monte Carlo. In M. Meila and X. Shen, editors, Proceedings of the 11th International Conference on Artificial Intelligence and Statistics, volume 2, pages 516–523. PMLR, 21–24 (2007)
34. Sminchisescu, C., Welling, M., Hinton, G.: A mode-hopping MCMC sampler. Technical report, CSRG-478, University of Toronto, (2003)
35. Surjanovic, S., Bingham, D.: Virtual library of simulation experiments: test functions and datasets. Retrieved March 9 (2021). http://www.sfu.ca/~ssurjano/egg.html
36. Tjelmeland, H., Hegstad, B.: Mode jumping proposals in MCMC. Scand. J. Statist. 28(1), 205–223 (2001)
37. Wales, D., Doye, J.: 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)
38. Zhigljavsky, A., Zilinskas, A.: Stochastic global optimization. Springer, Berlin (2007)

**Publisher’s Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.