Hopping between distant basins

**Maldon Goodridge · John Moriarty · Jure Vogrinc ·
Alessandro Zocca**

March 17, 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, stochastic optimisation, skipping sampler, rare events, Markov chains

Mathematics Subject Classification (2020) 65K10 · 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 stochastic 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 perturbed 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

Maldon Goodridge
School of Mathematics, Queen Mary, University of London, London, E1 4NS, UK, E-mail: ahw493@qmul.ac.uk

John Moriarty
School of Mathematics, Queen Mary, University of London, London, E1 4NS, UK E-mail: j.moriarty@qmul.ac.uk

Jure Vogrinc
Department of Statistics, University of Warwick, Coventry, CV4 7AL, UK E-mail: Jure.Vogrinc@warwick.ac.uk

Alessandro Zocca
Department of Mathematics, Vrije Universiteit Amsterdam, 1081HV, Netherlands, E-mail: a.zocca@vu.nl

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, 27, 16, 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 3.1 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: Section 2 introduces the algorithms, empirical results are presented and discussed in Section 3, and Section 4 concludes.

2 The BH-S algorithm

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

$$\min \quad f(\boldsymbol{x}) \quad \text{s.t.} \ \boldsymbol{x} \in D := \prod_{i=1}^{d} [l_i, u_i], \tag{2.1}$$

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 predefined 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.

Algorithm 1: Basin hopping

1 Generate a random initial state $Y_0 \in D$;
2 $X_0 = $ `LocalMinimisation`(Y_0);
3 $n = 0$;
4 **while** *Stopping criterion for $\{X_j\}_{j \leq n}$ is not satisfied:* **do**
5 $Y_n = $ `RandomPerturbation`(X_n);
6 $U_n = $ `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**

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.,

$$C_n := \{x \in D : f(x) \leq f(X_n)\} \subset \mathbb{R}^d. \tag{2.2}$$

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 Figure 2.1.
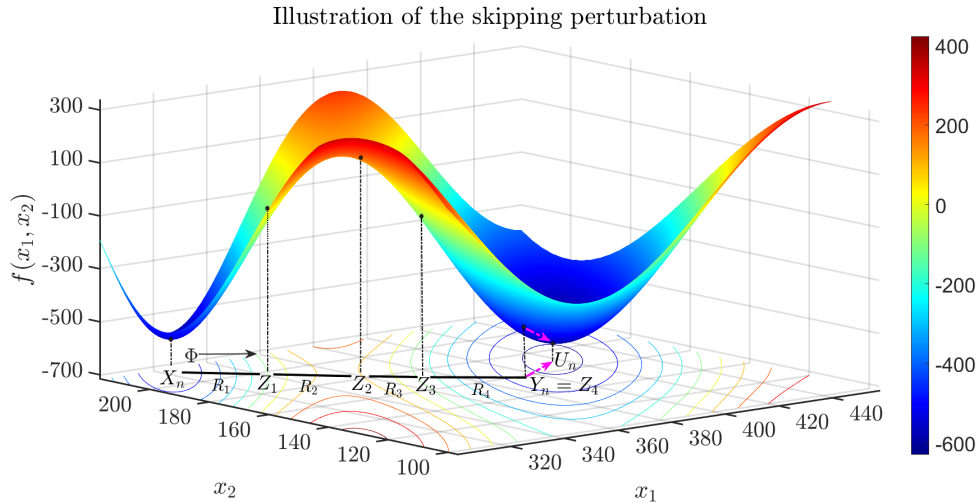
Fig. 2.1: Illustration of the skipping perturbation. Each state $X_n \neq X_0$ is a local minimum. All updates are in the direction $\Phi = \frac{Z_1 - X_n}{||Z_1 - X_n||}$ of the initial perturbation Z_1. If Z_1 does not have lower energy than X_n, a further independent random displacement R_2 is added in the direction Φ. This is repeated until a lower energy state is encountered, here at Z_4 (or until skipping is halted). The skipping perturbation is therefore $Y_n = Z_4$. The `LocalMinimisation` step maps Y_n to the local minimum U_n.

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

$$q_{r|\varphi}(r|\varphi) := \frac{q_{r,\varphi}(r,\varphi)}{q_\varphi(\varphi)}$$

the *conditional jump density*, i.e., the conditional density of the radial part r given the direction φ.

One could obviously pick a proposal density q that is radially symmetric, since in this case the conditioning is not required. However, there are also other canonical choices for q in which we can explicitly write the conditional density and efficiently sample from it. For example, in the case of a Gaussian proposal density $q \sim \mathcal{N}(0, \Sigma)$ for some $d \times d$ covariance matrix Σ, given a direction φ, each increment $|Z_{j+1} - Z_j|$ follows a generalised gamma distribution with density

$$q_{r|\varphi}(r|\varphi) = \frac{(\varphi^T \Sigma^{-1} \varphi)^{d/2}}{2^{d/2-1} \Gamma(\frac{d}{2})} \, r^{d-1} \, e^{-(\varphi^T \Sigma^{-1} \varphi)\frac{r^2}{2}} \, .$$

To construct the skipping perturbation, set $Z_0 = X_n$ and draw a random direction $\Phi \in \mathbb{S}^{d-1}$ from the directional density q_φ. A sequence of i.i.d. distances $R_1, R_2, \ldots$ is then drawn from the conditional jump density $q_{r|\Phi}$, defining a sequence of modified perturbations $\{Z_k\}_{k \geq 1}$ on $\mathbb{R}^d$ by

$$Z_{k+1} := Z_k + \Phi R_{k+1}, \qquad k = 0, 1, \ldots$$

Since this modification of the BH perturbation is more likely to generate states Z_k lying outside the optimisation domain D, we apply periodic boundary conditions.

If $Z_k \in C_n$ for some $k \leq K$, where K is a pre-defined maximum number of steps called the *halting index*, then we set $Y_n = Z_k$ in Algorithm 1 and continue to the `LocalMinimisation` and acceptance steps. Alternatively if $Z_k \notin C_n$ for all $k \leq K$ 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 σ, although other choices are possible (see the discussion in Section 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

    Input : State $X_n \in \mathbb{R}^d$
    Output: Randomly perturbed state $Y_n \in \mathbb{R}^d$
1 Set $Z_0 = X_n$;
2 Generate an initial perturbation W distributed according to the density $w \mapsto q(w - X_n)$;
3 Calculate the direction

$$\Phi = \frac{(W - X_n)}{\|W - X_n\|} ;$$

    Set $k = 1$ and $Z_1 := W$;
4 **while** $f(Z_k) > f(X_n)$ **and** $k < K$ **do**
5   Generate an independent distance increment R distributed as $\|W - X_n\|$ given Φ ;
6   Set $Z_{k+1} = Z_k + \Phi R$;
7   Increase k by one ;
8 **end**
9 Set $Y_n := Z_k$;

---

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 Section 3.6.

In Section 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 Section 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

$$q \sim \mathcal{N}(\mathbf{0}, \sigma^2 \cdot I_d), \tag{3.1}$$

where I_d is the $d \times d$ identity matrix and the parameter σ 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 $\mathcal{G}$ 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 $\mathcal{G}$,
– **Efficiency,** defined as the number of runs terminating in $\mathcal{G}$.

We write ρ_c and ρ_s for the effectiveness of the BH and BH-S algorithms respectively, while ϵ_c and ϵ_s denote their respective efficiencies. The BH and BH-S algorithms are individually tuned for each function by selecting σ 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 $\overline{J}$ of these perturbation distances is recorded. Then for each 300 second budget, the *expected mean jump distance* υ is the average $\upsilon := N^{-1} \sum_{n=1}^{N} \overline{J}^{(n)}$, where N is the number of runs realised within the execution time budget. For the BH-S algorithm, υ 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 υ_1 and υ_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 ρ_s/ρ_c and relative efficiency ϵ_s/ϵ_c are plotted for each benchmark energy landscape in Figure 3.1.
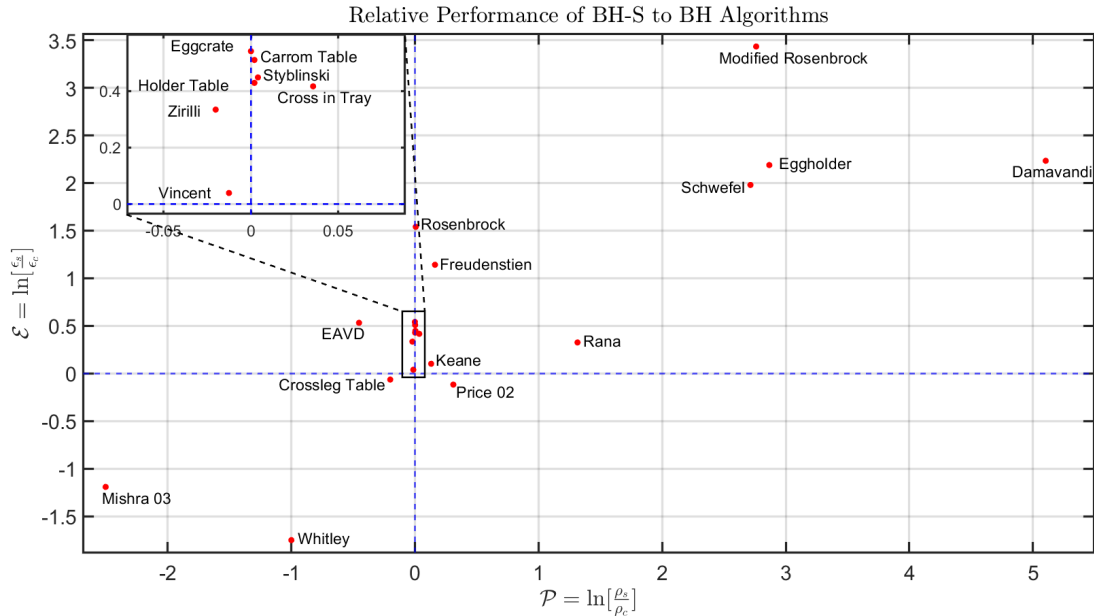
Fig. 3.1: Scatterplot of relative efficiency $\mathcal{E} = \ln(\epsilon_s/\epsilon_c)$ versus relative effectiveness $\mathcal{P} = \ln(\rho_s/\rho_c)$ for the BH and BH-S algorithms on benchmark energy landscapes

Landscapes in the first quadrant of Figure 3.1 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 Figure 3.1 represent cases where the BH-S algorithm is both less reliable and less efficient than BH. As explored more extensively later in Section 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 3.1 displays a positive correlation between relative efficiency and relative effectiveness. However for several landscapes (which lie near the vertical axis) the performance of BH and BH-S cannot be clearly distinguished on the basis of effectiveness alone. As confirmed by the Appendix, this is typically because both algorithms have effectiveness close to 100%. Nevertheless the algorithms differ in their efficiency, with

Table 3.1: Execution time spent on the perturbation and local minimisation steps by the BH and BH-S algorithms for the test functions discussed in Sections 3.3–3.5, normalised by efficiency ϵ.

		BH		BH-S	
Location in Figure 3.1	Function	Perturbation	Local Minimisation	Skipping Perturbation	Local Minimisation
First quadrant (Section 3.3)	Egg-holder	0.72	6.76	0.73	0.19
	Modified Rosenbrock	0.46	13.53	0.22	0.10
Third quadrant (Section 3.4)	Mishra-03	0.02	1.75	1.74	3.21
	Whitley	0.01	0.68	4.92	1.58
Special cases (Section 3.5)	Rosenbrock	0.01	0.13	0.05	0.01
	Styblinksi	0.02	0.06	0.06	0.01

BH-S observed to be more efficient than BH for each such landscape. One surface also lies in each of the second and fourth quadrants.

Further exploratory analysis is provided in Table 3.1, 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 Section 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 3.1, for the Damavandi, Schwefel, Modified Rosenbrock and Egg-holder functions for which BH-S works well (cf. Figure 3.1), after normalisation the BH-S algorithm spent approximately the 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 Figure 3.1, 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 3.1, 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 Figure 3.2, in Sections 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 3.2 plots two landscapes from the first quadrant of Figure 3.1–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].
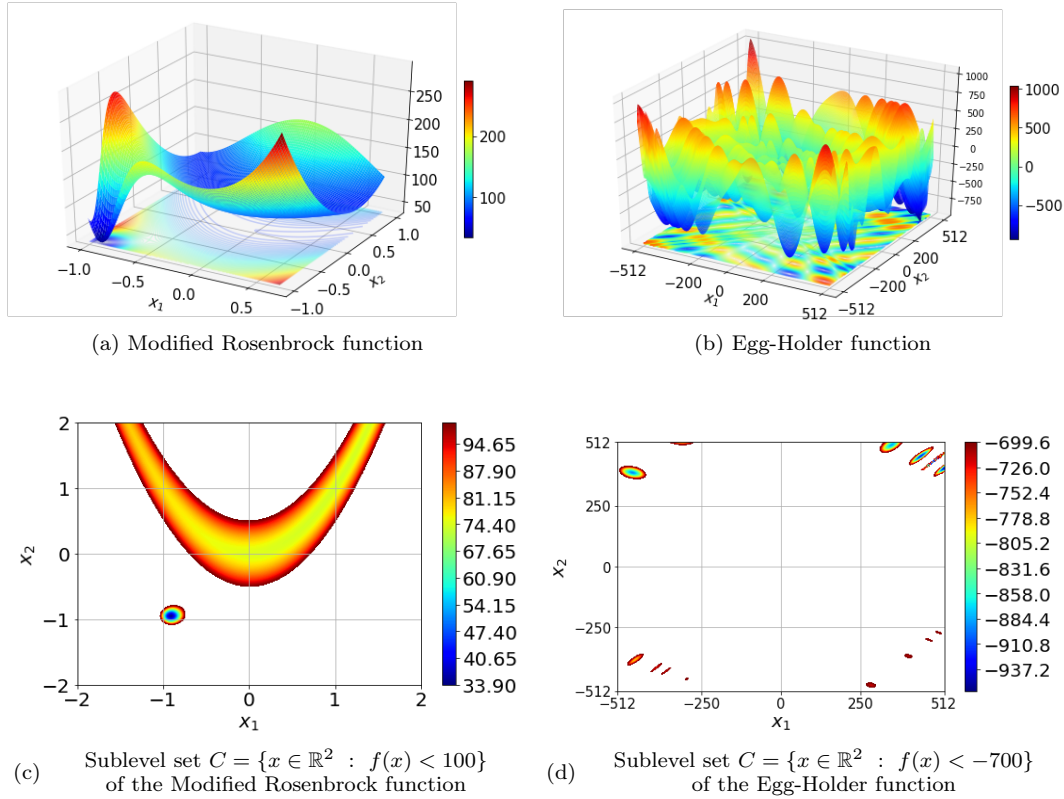
The Egg-Holder energy function is

(a) Modified Rosenbrock function

(b) Egg-Holder function

(c) Sublevel set $C = \{x \in \mathbb{R}^2 \; : \; f(x) < 100\}$ of the Modified Rosenbrock function

(d) Sublevel set $C = \{x \in \mathbb{R}^2 \; : \; f(x) < -700\}$ of the Egg-Holder function

Fig. 3.2: Examples of energy landscapes from the first quadrant of Figure 3.1

$$f(x) = -(x_2 + 47) \sin\left(\sqrt{\left|x_2 + \frac{x_1}{2} + 47\right|}\right) - x_1 \sin\left(\sqrt{|x_1 - (x_2 + 47)|}\right),$$

and we take the domain $D = [-512, 512]^2$, with global minimum at $x^* = (512, 404.2319)$ [11].

Observing Figure 3.2a, 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 σ, which would require *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 (2.2). From Figure 3.2c, 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 that basin provided that the straight line issuing from X_0 in the initial direction Φ in Algorithm 2 intersects it. In particular, this ability is robust to the choice of standard deviation σ provided that the halting index K is chosen appropriately (see the discussion on tuning in Section 3.7 below).

Figure 3.2b illustrates that the Egg-Holder function has multiple basins, many of which have near-global minima. Figure 3.2d 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 Figure 3.2b, the Egg-Holder function has shallower basins distributed throughout its domain. As discussed in Section 3.2 these provide an indirect, although potentially computationally expensive, route for BH to cross between the four groups of Figure 3.2d.

Table 3.2: effectiveness ρ and efficiency ϵ of the BH and BH-S algorithms, with and without the application of periodic boundary conditions, for the test functions discussed in Sections 3.3–3.5.

Location in Figure 3.1	Function	Boundary Condition	ρ_c	ϵ_c	ρ_s	ϵ_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

However between groups the Euclidean distance is large, creating the same challenge for BH as with the modified Rosenbrock function: even for optimally chosen σ, which would require *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 Figure 3.2d provided the initial direction Φ 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 σ.

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 Figure 3.1 and Table 5.1. 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 3.2 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 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 3.2 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 5.1 in the Appendix that the expected mean jump distances v_s and v_c (defined in Section 3.1) typically satisfy $v_s >> v_c$ for landscapes in the first quadrant of Figure 3.1. 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 Figure 3.1 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 3.3 plots two landscapes from the third quadrant of Figure 3.1, on which the BH algorithm outperforms BH-S, each with two sublevel sets above the global minimum value.
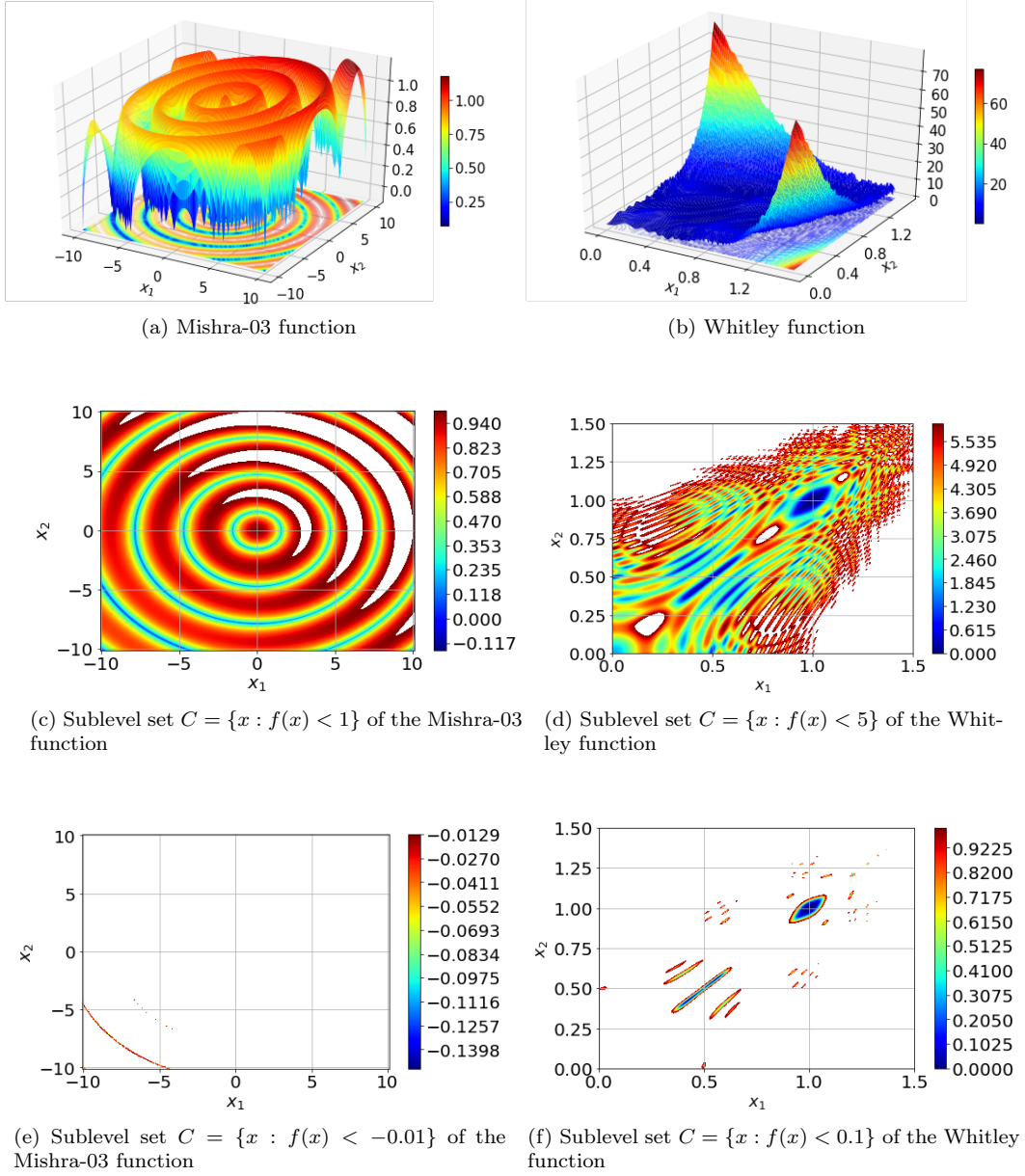

(a) Mishra-03 function

(b) Whitley function

(c) Sublevel set $C = \{x : f(x) < 1\}$ of the Mishra-03 function

(d) Sublevel set $C = \{x : f(x) < 5\}$ of the Whitley function

(e) Sublevel set $C = \{x : f(x) < -0.01\}$ of the Mishra-03 function

(f) Sublevel set $C = \{x : f(x) < 0.1\}$ of the Whitley function

Fig. 3.3: Examples of energy landscapes favouring the BH algorithm

The Mishra-03 function $f : \mathbb{R}^2 \to \mathbb{R}$ is given by

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

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

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

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

From Figure 3.3a, the Mishra-03 function is highly irregular and has many basins which appear almost point-like. Figure 3.3e confirms that the situation outlined in Section 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 σ. In particular, it outperforms BH-S on this landscape.

Similarly, by observing Figure 3.3d, 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 σ. As with Mishra-03, however, from Figure 3.3f 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 Section 3.8 we provide a discussion on how this alternating perturbation can be implemented.

3.5 Special cases

It was noted in Section 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 $f_d : \mathbb{R}^d \to \mathbb{R}$, 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 3.4a 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 Φ to intersect that basin. As Φ 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 Figure 3.4a, 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 $d \leq 7$.
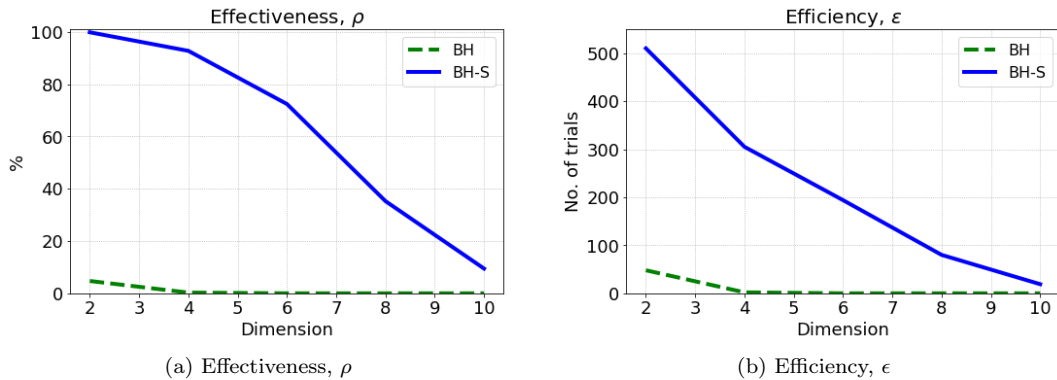
(a) Effectiveness, ρ (b) Efficiency, ϵ

Fig. 3.4: 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.

3.7 Adapting parameter values to the landscape

Both BH and BH-S have the parameter σ, 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 σ and also of the BH-S halting index K. To facilitate this discussion we restrict attention to the two-dimensional Egg-Holder function.

Figure 3.5 plots the effectiveness and efficiency of both BH and BH-S as σ 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 σ should not be very small (≤ 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 Figure 3.5 that both the effectiveness and efficiency of the BH algorithm increase approximately linearly within this range as σ increases. As discussed in Section 3.3, this reflects the fact that as σ increases, direct transitions between the four groups of deepest basins become more likely. In contrast, and again confirming the discussion in Section 3.3, both the efficiency and effectiveness of BH-S appear to be rather robust to the choice of σ.

Figure 3.6 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 σ. This suggests that K should not be too small, and the plot of efficiency in Figure 3.6b 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 σ is large enough that the initial perturbation regularly falls outside the basin of the current state X_n. Having selected σ, 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.

(a) Effectiveness, ρ

(b) Efficiency, ϵ

(c) Total perturbation steps conducted during the 300s execution time budget.
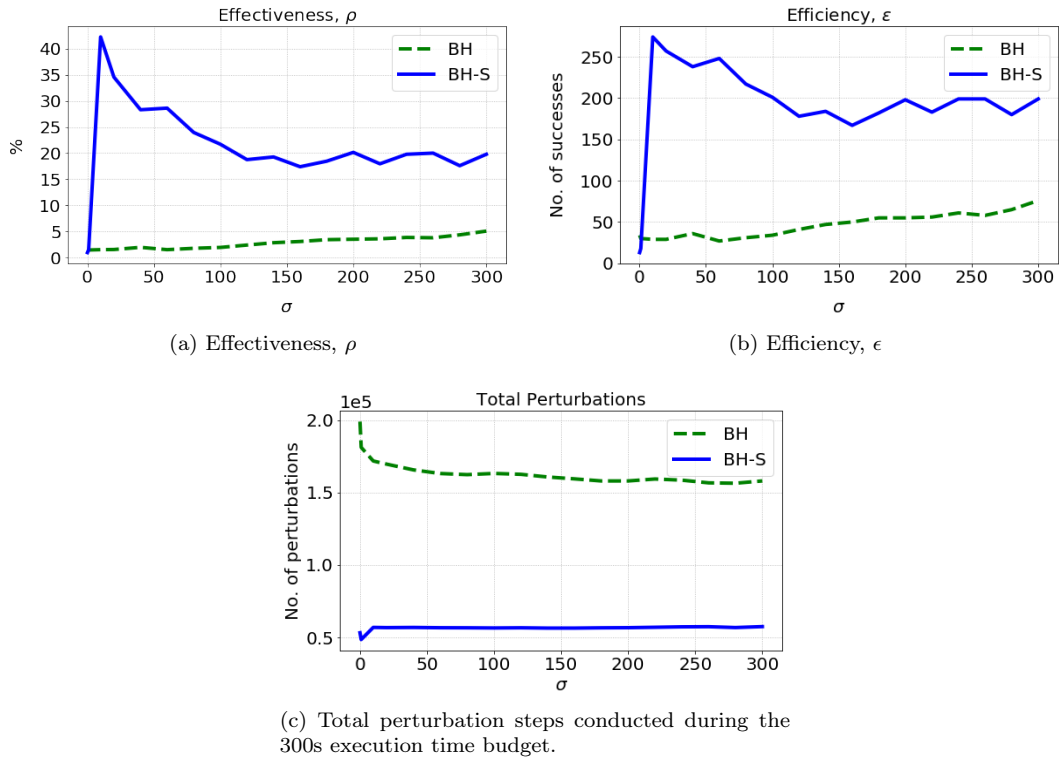
Fig. 3.5: Comparison of individually tuned BH-S and BH performances on the Egg-holder function. Set-up: execution time budget of 300 seconds; stopping criteria: 50 perturbations; the halting index for skipping perturbation is set to $K = 25$ for all simulations.
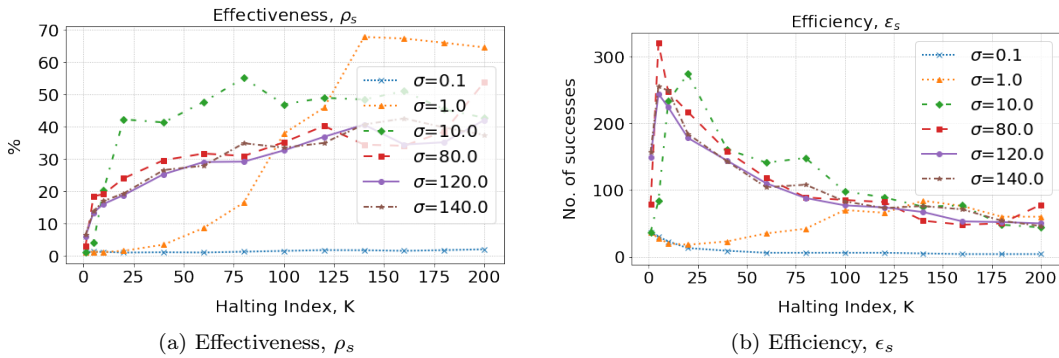

(a) Effectiveness, ρ_s

(b) Efficiency, ϵ_s

Fig. 3.6: Performance and efficiency results for the BH-S applied to the Egg-holder function for various combinations of K and σ. An execution time budget of 300s was applied to all simulations.

Figure 3.7 confirms these guidelines in higher dimensions, by plotting the BH-S effectiveness and efficiency in dimensions up to 10 as σ varies with the fixed choice $K = 50$. It confirms that these performance metrics are relatively robust to the value of σ, provided that σ 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 Section 3.4 for the monotonic BH-S algorithm by regularly including non-monotonic BH steps. Figure 3.8 plots the effectiveness and efficiency metrics for this hybrid algorithm on various landscapes, as the ratio between BH-S and BH steps varies.

(a) Effectiveness, ρ_s (b) Efficiency, ϵ_s
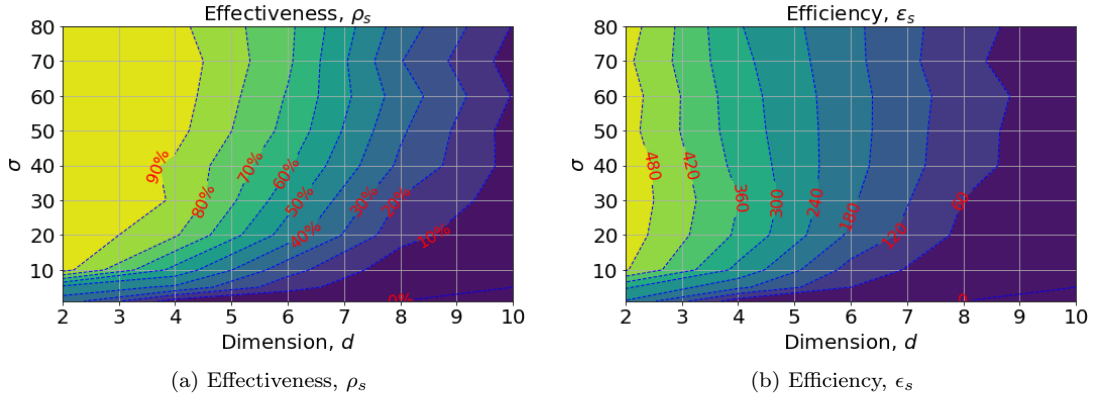
Fig. 3.7: Performance of the BH-S algorithm on the Schwefel-07 function for different combinations of domain dimension d and perturbation variance σ. Note: the halting index was set to $K = 50$ with an execution time budget of 300s for all simulations.

(a) Effectiveness, ρ (b) Efficiency, ϵ
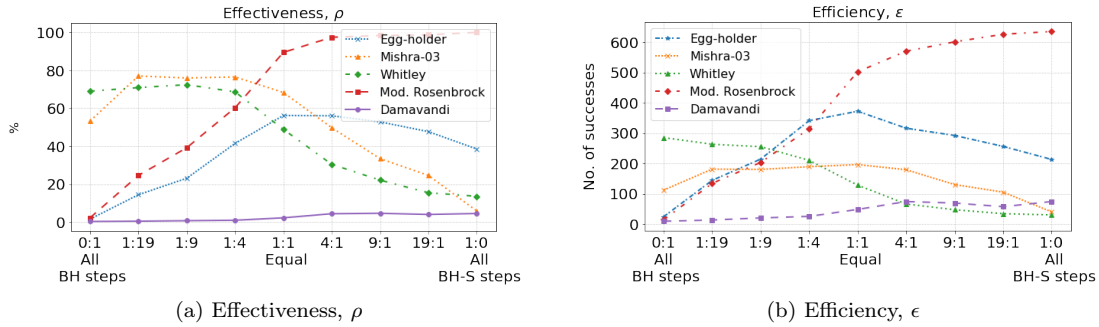
Fig. 3.8: Performance of the hybrid algorithm with varying proportions of BH to BH-S steps.

It can be seen that for the Mishra-03 and Whitley functions of Section 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 Section 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 σ, 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.

5 Declarations

5.1 Funding

5.2 Conflicts of interest/Competing Interests

The authors declare there are no conflicts of interest when presenting this research.

5.3 Availability of data and material

Not applicable.

5.4 Code Availability

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

References

1. I. Andricioaei, J. Straub, and A. Voter. Smart darting Monte Carlo. *The Journal of Chemical Physics*, 114(16):6994–7000, 2001.
2. Y. F. Atchade and J. S. Rosenthal. On adaptive Markov chain Monte Carlo algorithms. *Bernoulli*, 11(5):815–828, 2005.
3. W. Baritompa, M. Dür, E. M. Hendrix, L. Noakes, W. Pullan, and G. R. Wood. Matching stochastic algorithms to objective function landscapes. *Journal of Global Optimization*, 31(4):579–598, 2005.
4. S. Brooks, A. Gelman, G. Jones, and X. Meng. *Handbook of Markov Chain Monte Carlo*. CRC press, 2011.
5. Q. Duan and D. Kroese. Splitting for optimization. *Computers & Operations Research*, 73:119–131, 2016.
6. D. Gamerman and H. Lopes. *Markov chain Monte Carlo, stochastic simulation for bayesian inference*. Chapman & Hall/CRC, Boca Raton, 2nd edition, 2006.
7. A. Gavana. Global optimization benchmarks and AMPGO. Retrieved April 20, 2021, from http://infinity77.net/global_optimization/test_functions.html, 2013.
8. R. Gehrke. *First-principles basin-hopping for the structure determination of atomic clusters*. PhD thesis, Freie Universität Berlin, 2009.
9. R. Huang, L. Bi, J.and Li, and Y. Wen. Basin hopping genetic algorithm for global optimization of ptco clusters. *Journal of chemical information and modeling*, 2020.
10. P. Jain and A. M. Agogino. Global optimization using the multistart method. *Journal of Mechanical Design*, 115(4):770–775, Dec. 1993.
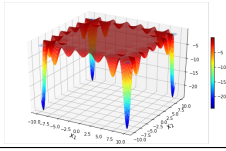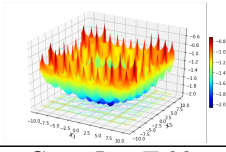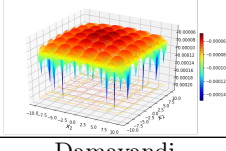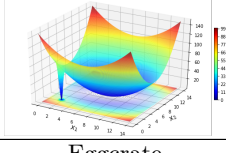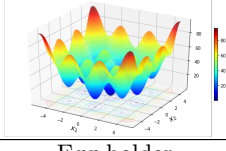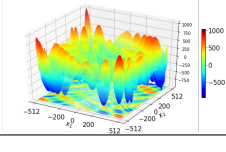
11. M. Jamil and X. S. Yang. A literature survey of benchmark functions for global optimisation problems. *International Journal of Mathematical Modelling and Numerical Optimisation*, 4(2):150, 2013.

12. A. R. Kan and G. T. Timmer. Stochastic global optimization methods part I: Clustering methods. *Mathematical programming*, 39(1):27–56, 1987.

13. A. R. Kan and G. T. Timmer. Stochastic global optimization methods part II: Multi level methods. *Mathematical Programming*, 39(1):57–78, 1987.

14. J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks*, volume 4, pages 1942–1948 vol.4, 1995.

15. S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

16. S. Lan, J. Streets, and B. Shahbaba. Wormhole Hamiltonian Monte Carlo. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, pages 1953–1959, 2014.

17. R. H. Leary. Global optimization on funneling landscapes. *Journal of Global Optimization*, 18(4):367–383, 2000.

18. D. Liu and J. Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45:503–528, 1989.

19. M. Locatelli. On the multilevel structure of global optimization problems. *Computational Optimization and Applications*, 30(1):5–22, 2005.

20. M. Locatelli and F. Schoen. *Global optimization: theory, algorithms, and applications*. SIAM, 2013.

21. R. Martí. Multi-start methods. In *Handbook of Metaheuristics*, pages 355–368. Kluwer Academic Publishers, 2003.

22. N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953.

23. P. D. Moral, A. Doucet, and A. Jasra. Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 68(3):411–436, 2006.

24. J. Moriarty, J. Vogrinc, and A. Zocca. A Metropolis-class sampler for targets with non-convex support. *Statistics and Computing*, 31(6), Sept. 2021.

25. B. Olson, I. Hashmi, K. Molloy, and A. Shehu. Basin hopping as a general and versatile optimization framework for the characterization of biological macromolecules. *Advances in Artificial Intelligence*, 2012:1–19, Dec. 2012.

26. M. Paleico and J. Behler. A flexible and adaptive grid algorithm for global optimization utilizing basin hopping Monte Carlo. *Journal of Chemical Physics*, 2020.

27. E. Pompe, C. Holmes, and K. Łatuszyński. A framework for adaptive MCMC targeting multimodal distributions. *The Annals of Statistics*, 48(5), Oct. 2020.

28. G. O. Roberts, J. Rosenthal, et al. General state space Markov chains and MCMC algorithms. *Probability surveys*, 1:20–71, 2004.

29. G. Rondina and J. Da Silva. Revised basin-hopping Monte Carlo algorithm for structure optimization of clusters and nanoparticles. *Journal of Chemical Information and Modeling*, 53(9):2282–2298, 2013. PMID: 23957311.

30. R. Rubinstein. The cross-entropy method for combinatorial and continuous optimization. *Methodology and computing in applied probability*, 1(2):127–190, 1999.

31. F. Schoen. Stochastic techniques for global optimization: A survey of recent advances. *Journal of Global Optimization*, 1(3):207–228, 1991.

32. M. Schumer and K. Steiglitz. Adaptive step size random search. *IEEE Transactions on Automatic Control*, 13(3):270–276, June 1968.

33. C. Sminchisescu and M. Welling. 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 Mar 2007.

34. C. Sminchisescu, M. Welling, and G. Hinton. A mode-hopping MCMC sampler. Technical report, CSRG-478, University of Toronto, 2003.

35. S. Surjanovic and D. Bingham. Virtual library of simulation experiments: test functions and datasets. Retrieved March 9, 2021, from http://www.sfu.ca/~ssurjano/egg.html.

36. H. Tjelmeland and B. Hegstad. Mode jumping proposals in MCMC. *Scandinavian Journal of Statistics*, 28(1):205–223, 2001.

37. D. Wales and J. Doye. Global optimization by basin-hopping and the lowest energy structures of Lennard-Jones clusters containing up to 110 atoms. *Journal of Physical Chemistry A*, 1997.

38. A. Zhigljavsky and A. Zilinskas. *Stochastic global optimization*, volume 9. Springer Science & Business Media, 2007.
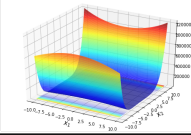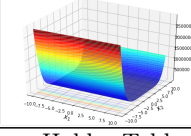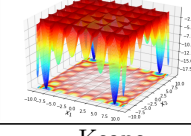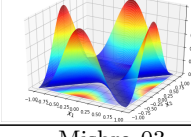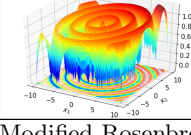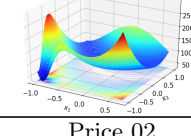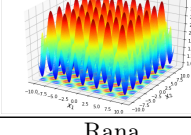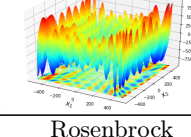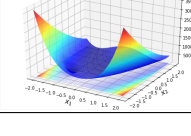
Appendix

Table 5.1 records the results for all landscapes in Figure 3.1. For each landscape, the parameters of BH and BH-S were chosen following the approach of Section 3.7. The following notation is used:

- ρ_c and ρ_s are the effectiveness of BH and BH-S respectively;
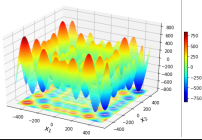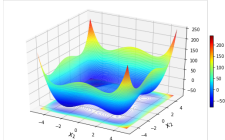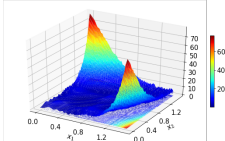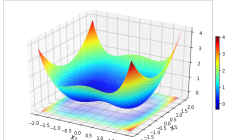- ϵ_c and ϵ_s are the efficiency of BH and BH-S respectively;
- υ_1 is the expected mean jump distance among random walk steps;
- υ_s is the expected mean jump distance among skipping transitions, i.e., when $k > 1$;
- $\mathbb{P}_s$ is the probability that, conditional on the BH-S perturbation being accepted, skipping had occurred ($k > 1$);
- ν_1 and ν_s are the expected mean jump distances among random walk steps ($k = 1$) and skipping steps ($k > 1$), respectively.

Table 5.1: Performance metrics for the BH and BH-S algorithms on all landscapes in Figure 3.1.

		BH					BH-S				
Function	σ	ρ_c	ϵ_c	ν_1	σ	K	ρ_s	ϵ_s	ν_1	ν_s	$\mathbb{P}_s$
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

	BH				BH-S						
Function	σ	ρ_c	ϵ_c	ν_1	σ	K	ρ_s	ϵ_s	ν_1	ν_s	$\mathbb{P}_s$
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
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

Function	BH					BH-S					
	σ	ρ_c	ϵ_c	ν_1	σ	K	ρ_s	ϵ_s	ν_1	ν_s	$\mathbb{P}_s$
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
Whitley	0.4	86.4	121	0.7	0.7	50	31.8	21	0.6	17	37.7
Zirilli	0.2	99.9	707	0.6	$\sqrt{2}$	10	97.9	987	1.9	5.3	49.4