Acta Applicandae Mathematicae (2024) 190:2
https://doi.org/10.1007/s10440-024-00639-0

Swarm-Based Optimization with Random Descent

Eitan Tadmor[1] · **Anil Zenginoğlu**[2]

Received: 16 November 2023 / Accepted: 17 February 2024 / Published online: 1 March 2024

Abstract
We extend our study of the swarm-based gradient descent method for non-convex optimization, (Lu et al., Swarm-based gradient descent method for non-convex optimization, 2022, arXiv:2211.17157), to allow random descent directions. We recall that the swarm-based approach consists of a swarm of agents, each identified with a position, $\mathbf{x}$, and mass, m. The key is the transfer of mass from high ground to low(-est) ground. The mass of an agent dictates its step size: lighter agents take larger steps. In this paper, the essential new feature is the choice of direction: rather than restricting the swarm to march in the steepest gradient descent, we let agents proceed in randomly chosen directions centered around — but otherwise different from — the gradient direction. The random search secures the descent property while at the same time, enabling greater exploration of ambient space. Convergence analysis and benchmark optimizations demonstrate the effectiveness of the swarm-based random descent method as a multi-dimensional global optimizer.

Keywords Optimization · Gradient descent · Swarming · Backtracking · Convergence analysis

Mathematics Subject Classification 90C26 · 65K10 · 92D25

1 Introduction. The Importance of Random Marching Directions

In this work we extend our study of swarm-based approach for non-convex optimization, [14], with the aim of finding minimizer(s) of a loss function, $\operatorname{argmin}_{\mathbf{x}\in\Omega} F(\mathbf{x})$, over an ambient bounded set $\Omega \subset \mathbb{R}^d$. Classical iterative algorithms for numerical optimization employ a single agent which explores the ambient space by successively improving the position of approximate optimize(s), e.g., [2, 16, 17] and the references therein or the more recent

Dedicated to Shi Jin for many years of friendship

✉ E. Tadmor
tadmor@umd.edu

A. Zenginoğlu
anil@umd.edu

1 Department of Mathematics and Institute for Physical Science & Technology, University of Maryland, College Park, MD, USA

2 Institute for Physical Science & Technology, University of Maryland, College Park, MD, USA

[9, 11] etc. Unlike those single-agent iterations, the swarm-based methods use a *crowd of coordinated agents* — the swarm, to explore Ω, e.g., [3–5, 7, 8, 10, 14, 19, 22, 23]. Here we follow the swarm-based approach introduced in [14], in which the swarm consists of N agents, each is identified with a position $\mathbf{x}_i^n$, and an independent mass (or weight), m_i^n,

$$\mathbf{x}_i^n = \mathbf{x}_i(t^n) \in \Omega \subset \mathbb{R}^d, \quad m_i^n = m_i(t^n) \in (0,1], \qquad i = 1, 2, \ldots, N.$$

Thus, the distinctive feature of our swarm-based iterations is the fact that they are embedded in the larger space, $(\mathbf{x}_i^n, m_i^n) \in \Omega \times [0,1] \subset \mathbb{R}^{d+1}$: the additional mass-parametrization is the essential platform which enables proper coordination of agents, in order to improve the overall configuration of the swarm in its search for an optimizer.

The basic step takes the form

$$\mathbf{x}_i^{n+1} = \mathbf{x}_i^n - h_i^n \mathbf{p}_i^n, \qquad i = 1, 2, \ldots, N.$$

It reflects the move of an agent from its current position, $\mathbf{x}_i^n$, in direction $\mathbf{p}_i^n$ with step-size h_i^n. In [14] we advocated the use of gradient direction, $\mathbf{p}_i^n = \nabla F(\mathbf{x}_i^n)$. By properly adjusting the step-size which takes into account the weights of all other agents in the crowd, $h_i^n = h_i\big(\mathbf{x}_i^n, \{m_j^n\}_{1\leqslant j\leqslant N}\big)$ (this is where the communication between agents enters), one is able to secure the all important *descent property*, [14, Eq. (5.5)]

$$F(\mathbf{x}_i^n - h_i^n \mathbf{p}_i^n) \leqslant F(\mathbf{x}_i^n) - \lambda_i h_i^n |\nabla F(\mathbf{x}_i^n)|^2, \qquad \mathbf{p}_i^n = \nabla F(\mathbf{x}_i^n);$$

here λ_i are the descent amplitudes, depending on the weights $\lambda_i = \lambda_i\big(\{m_j^n\}_{1\leqslant j\leqslant N}\big) \in (0,1)$. In this work, we abandon the use of the gradient direction, $\mathbf{p}_i^n = \nabla F(\mathbf{x}_i^n)$, and instead focus on the descent property as the sole guidance in our swarm-based iterations. This allows us to choose from a large cone of directions which still secure the descent property. By randomly choosing proper directions $\mathbf{p}_i^n$, which are still compatible with the descent property, we significantly increase the heterogeneity of the swarm-based method in exploring larger portion of the ambient space, while keeping the overall decent property.

Our swarm-based method ends up with an interplay between positions and weights which takes the schematic description

$$\begin{aligned} \big\{\{m_j^n\}_{1\leqslant j\leqslant N}, F(\mathbf{x}_i^n)\big\} &\mapsto m_i^{n+1}, \\ \big\{m_i^{n+1}, \mathbf{x}_i^n\big\} &\mapsto \mathbf{x}_i^{n+1}. \end{aligned}$$

The method repeatedly transfers mass from high to lower ground while on the way, driving agents to smaller (lower) loss values; in particular, $\{\min_i F(\mathbf{x}_i^n)\}$ forms a decreasing sequence in time, which ideally approaches the global minimizer in the region explored by these agents,

$$\mathbf{x}_i^n \overset{n\to\infty}{\longrightarrow} \operatorname*{argmin}_{\mathbf{x}\in\Omega} F(\mathbf{x}).$$

The last statement applies to certain sub-sequence, $\{\mathbf{x}_{i_n}^n\}_{n>n_0}$, which is made precise in the main convergence results of theorems 3.2 and 3.3 below.

A detailed description of this two-stage swarm-based mechanism now follows.

Mass transfer. In the first stage, positions change the distribution of mass: each agent with mass m_i^n transfers a fraction of its mass, $\eta_i^n m_i^n$, to the current global minimizer posi-

tioned at $\mathbf{x}_{i_n}$ where $i_n := \text{argmin}_i\, F(\mathbf{x}_i^n)$.

$$\begin{cases} m_i^{n+1} &= m_i^n - \eta_i^n m_i^n, \qquad i \neq i_n \\ m_{i_n}^{n+1} &= m_{i_n}^n + \sum_{i\neq i_n} \eta_i^n m_i^n, \end{cases} \qquad \eta_i^n := \Big(\frac{F(\mathbf{x}_i^n) - F_{\min}^n}{F_{\max}^n - F_{\min}^n}\Big)^q \in (0,1]. \tag{1.1}$$

The fraction of mass transfer, $\left(\frac{F(\mathbf{x}_i^n)-F_{\min}^n}{F_{\max}^n-F_{\min}^n}\right)^q$, is determined by the *relative height* of each agent, relative to the global extremes,[1] $F_{\min}^n = \min_j F(\mathbf{x}_j^n)$ and $F_{\max}^n = \max_j F(\mathbf{x}_j^n)$, and depending on user-choice of a mass transfer parameter, $q \geqslant 1$. The higher q is, the more tamed is the transfer of mass. A systematic study reported in Sect. 4.2 below reveals a dramatic improvement when increasing the mass transfer parameter $q = 2, 4, 8$.

Observe that while the total mass is conserved, say $\sum_i m_i^n = 1$, individual masses are redistributed from high to lower ground — the higher the agent, the larger fraction of its mass will be lost in favor of the agent at the lowest ground. In fact, the highest agent in each iteration is eliminated: to be precise, the worst performing agents are eliminated whenever $1 - \eta_i^n = \mathcal{O}(\epsilon) \ll 1$. This is an aggressive 'survival of the fittest' protocol, so that after N iterations the swarm consists of a single agent which should be in the best position to approach the minimum of the space explored so far by the swarm. We note in passing that one can adopt a more flexible protocol which allows the worst (highest) agents to survive a few iterations before elimination; this flexibility would improve the overall success rates of the swarm at the expense of efficiency.

In particular, the dynamic adjustment of masses in (1.1) can be interpreted as a particular case of *alignment dynamics*, with 'aggressive' protocol in which agents steer towards the *minimal* heading, $m_-^n = \min_i m_i^n$. Instead one may consider a more tamed alignment towards an average heading, as originated in [20], see e.g., [21] and the references therein. In this context we refer to the stochastic-based Consensus Based Optimization, [3, 6, 19, 22], steering towards a properly weighted convex combination, $\bar{m}^n = \sum_j \theta_j^n \mathbf{x}_j^n$ with weights $\theta_j^n = exp(-\alpha F(\mathbf{x}_j^n))/\big(\sum_k exp(-\alpha F(\mathbf{x}_k^n))\big)$, which in turn is driven to a global minimum by letting $\alpha \gg 1$. We note that a main novelty in our approach is the use of masses $\{m_i^n\}$ in (1.1) as *independent variables*, which evolve alongside the dynamics of positions $\{\mathbf{x}_i^n\}$ outlined in (1.2), (1.4a)–(1.4c) below. Indeed, carrying out the perspective of alignment dynamics, the masses provide the essential added dimension as a platform for communication among the agents of the swarm.

Stepping in descent direction — a random choice approach. In the second stage, the distribution of mass affects the change of positions,

$$\mathbf{x}_i^{n+1} = \mathbf{x}_i^n - h_i^n \mathbf{p}_i^n. \tag{1.2}$$

The driving force behind the protocol for choosing the direction, $\mathbf{p} = \mathbf{p}_i^n$, and the step size, $h = h_i^n$, is to secure the following descent property, depending on the relative mass $\widetilde{m}_i^{n+1}$ and a descent parameter $\lambda < 1$,

$$F(\mathbf{x}_i^n - h\mathbf{p}_i^n) \leqslant F(\mathbf{x}_i^n) - \frac{1}{2}\lambda \widetilde{m}_i^{n+1} h |\nabla F(\mathbf{x}_i^n)|^2, \qquad \widetilde{m}_i^{n+1} = \frac{m_i^{n+1}}{\max_i m_i^{n+1}}, \quad \lambda < 1. \tag{1.3}$$

[1] To prevent vanishing denominator in the extreme case $F_{\max} = F_{\min}$, we adjust (1.1) with a small ϵ-correction, $\eta_i^n := \left(\frac{F(\mathbf{x}_i^n)-F_{\min}^n}{F_{\max}^n-F_{\min}^n+\epsilon}\right)^q$.

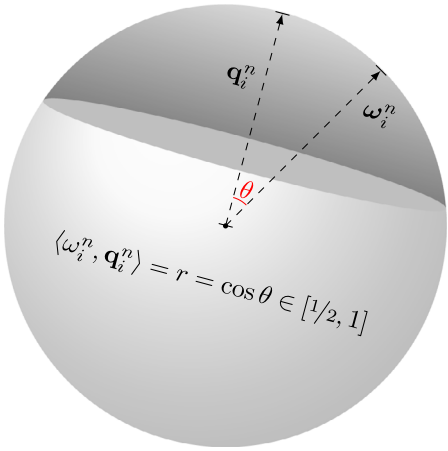

Fig. 1 $\mathbf{q}_i^n$ is the gradient orientation — the unit vector along the gradient direction, $\nabla F(\mathbf{x}_i^n)$, and the unit vector, $\boldsymbol{\omega}_i^n$, is determined by a randomly chosen point on a spherical cap centered around $\mathbf{q}_i^n$ (shown as the shaded part of the sphere)

We recall that the choice of the gradient direction, $\mathbf{p}_i^n = \nabla F(\mathbf{x}_i^n)$, secures the sharper steepest descent property,

$$F(\mathbf{x}_i^n - h\mathbf{p}_i^n)_{|\mathbf{p}_i^n = \nabla F(\mathbf{x}_i^n)} \leqslant F(\mathbf{x}_i^n) - \lambda \widetilde{m}_i^{n+1} h |\nabla F(\mathbf{x}_i^n)|^2,$$

which was the basis for the swarm-based gradient descent (SBGD) method we introduced in [14]. The purpose of this work is to extend the SBGD method by allowing a larger set of descent directions: the emphasis is no longer on the steepest descent along the gradient direction but instead, allowing a more effective exploration of the ambient space using a *random choice of directions*, $\{\mathbf{p}_i^n\}$, that still maintains (half the steepest) descent property. This implies that the swarm, stepping in other than the gradient direction, will explore a larger portion of the ambient space, which in turn leads to a more effective search, and proved to be particularly relevant in high-dimensional optimizations, see the numerical simulations reported in Sect. 4. We refer to this new version as the Swarm-Based Random Descent (SBRD) method.

We provide below a detailed, self-contained description of the SBRD. The stepping protocol is based on the choice of direction and step-size. The direction $\mathbf{p}_i^n$, is determined by its *orientation*, $\boldsymbol{\omega}_i^n$,

$$\mathbf{p}_i^n = |\nabla F(\mathbf{x}_i^n)| \boldsymbol{\omega}_i^n, \qquad \boldsymbol{\omega}_i^n \in \mathbb{S}^{d-1}, \tag{1.4a}$$

relative to the orientation of the gradient, $\mathbf{q}_i^n = \frac{\nabla F(\mathbf{x}_i^n)}{|\nabla F(\mathbf{x}_i^n)|} \in \mathbb{S}^{d-1}$, so that

$$\left\langle \boldsymbol{\omega}_i^n, \mathbf{q}_i^n \right\rangle = r, \qquad \mathbf{q}_i^n := \frac{\nabla F(\mathbf{x}_i^n)}{|\nabla F(\mathbf{x}_i^n)|}. \tag{1.4b}$$

Here, r is randomly chosen number from a uniform distribution in an interval dictated by the relative mass,

$$r \in \mathcal{U}\Big[{}^1\!/_2(1 + \widetilde{m}_i^{n+1}), 1 \Big]. \tag{1.4c}$$

This means that the orientation of $\mathbf{p}_i^n$ lies in a spherical cap centered around $\mathbf{q}_i^n$. The 'opening' of the corresponding spherical cone, see Fig. 1, $\theta := \arccos\left(\frac{1}{2}(1 + \widetilde{m}_i^{n+1})\right)$. It is larger

for lighter agents, and coincides with the gradient direction, $\nabla F(\mathbf{x}_i^n)$, for the heaviest agent where $\widetilde{m}_i^{n+1} = 1$. The protocol for randomly selecting $\boldsymbol{\omega}_i^n$ subject to (1.4b) is outlined in Sect. 2.1 below.

Choosing the step size — a backtracking protocol. It follows that the new position, $\mathbf{x}^{n+1}(h) = \mathbf{x}_i^n - h\mathbf{p}_i^n$ — viewed as a function of the step size h, satisfies the desired descent property, at least for small enough h. Indeed, (1.4a)–(1.4c) implies

$$\langle \mathbf{p}_i^n, \nabla F(\mathbf{x}_i^n)\rangle = r|\nabla F(\mathbf{x}_i^n)|^2 \geqslant \frac{1}{2}(1+\widetilde{m}_i^{n+1})|\nabla F(\mathbf{x}_i^n)|^2. \tag{1.5}$$

Hence, if we let $L := \max_{\mathbf{x}\in\Omega} \|D^2F(\mathbf{x})\|$ with $L < \infty$ serves as a Lipschitz bound[2] of ∇F, then for every $\lambda < 1$ and $h < 1/L$, there holds

$$\begin{aligned} F(\mathbf{x}_i^{n+1}(h)) &\leqslant F(\mathbf{x}_i^n) - h\langle \mathbf{p}_i^n, \nabla F(\mathbf{x}_i^n)\rangle + \frac{h^2}{2}L|\nabla F(\mathbf{x}_i^n)|^2 \\ &\leqslant F(\mathbf{x}_i^n) - \frac{1}{2}\big(1+\widetilde{m}_i^{n+1} - Lh\big)h|\nabla F(\mathbf{x}_i^n)|^2 \\ &< F(\mathbf{x}_i^n) - \frac{1}{2}\lambda\widetilde{m}_i^{n+1}h|\nabla F(\mathbf{x}_i^n)|^2. \end{aligned} \tag{1.6}$$

Thus, we recover (1.3) for any $h < 1/L$.[3] In particular step size of order $\lesssim 1/L$ need not necessary be very small to enforce the descent property. We note, however, that since we have no access to the Lipschitz bound L, we therefore do not have an effective protocol for computing a step size which secures (1.6), beyond making the generic statement that it holds for 'sufficiently small' h. In fact, $h < 1/L$ need not be small and we are interested in a protocol that identifies the *largest* h for which (1.6) holds (observe that the larger h is, the larger is the descent bound quoted in (1.6)). To this end we use a *backtracking protocol* outlined in Algorithm 2 below. The backtracking algorithm produces a time step, $h = h_i^n$, depending on the position of the agent $\mathbf{x}_i^n$, and its relative mass $\widetilde{m}_i^{n+1}$,

$$h_i^n = h(\mathbf{x}_i^n, \lambda\widetilde{m}_i^{n+1}).$$

It secures the *lower* bound $h_i^n \geqslant \frac{\gamma}{L}$ for some $\gamma < 1$, so that using (1.6) we finally end up with the descent property of the form,

$$F(\mathbf{x}_i^{n+1}) \leqslant F(\mathbf{x}_i^n) - \frac{\gamma}{2L}\lambda\widetilde{m}_i^{n+1}|\nabla F(\mathbf{x}_i^n)|^2, \qquad \mathbf{x}_i^{n+1} = \mathbf{x}_i^n - h_i^n\mathbf{p}_i^n.$$

Remark 1.1 This should be compared with the descent property of SBGD method restricted to the steepest descent direction $\mathbf{p}_i^n = \nabla F(\mathbf{x}_i^n)$, for which we have, [14, Proposition 5.2],

$$F(\mathbf{x}_i^{n+1}) \leqslant F(\mathbf{x}_i^n) - \frac{2\gamma}{L}\big(1-\lambda\widetilde{m}_i^{n+1}\big)\lambda\widetilde{m}_i^{n+1}|\nabla F(\mathbf{x}_i^n)|^2.$$

Thus, our stepping protocol retains at least half of the steepest descent, while gaining greater heterogeneity in space exploration. In particular, while heavier agents are still restrained by

[2] In fact, one can use a Lipscitz bound localized to the neighborhood of $\mathbf{x}_i^n$ that is being visited by the SBRD iterations, but since this neighborhood is not quantified we address the global Lipschitz bound $L = \max_{\alpha,\beta,\mathbf{x}\in\Omega}\left|\frac{\partial^2 F(\mathbf{x})}{\partial x_\alpha \partial x_\beta}\right|$.

[3] In fact, a slightly larger threshold, $h < \frac{1+\widetilde{m}_i^{n+1}(1-\lambda)}{L}$, still allows (1.6) to hold.

Table 1 Success rates of SBRD vs. SBGD for global optimization of the d-dimensional Ackley function using N agents based on $m = 1000$ runs of uniformly generated initial data, $\mathbf{x}_i^0 \in [-3, 3]^d$. Backtracking parameters are $\lambda = 0.2$ and $\gamma = 0.9$ (see algorithm 2). Boldfaced numbers emphasize the cases where SBRD outperforms SBGD by more than 1%. The randomization provided by SBRD becomes essential beyond the critical dimension $d = 13$

	$N = 10$		$N = 25$		$N = 50$		$N = 100$	
d	SBRD	SBGD	SBRD	SBGD	SBRD	SBGD	SBRD	SBGD
12	13.7%	26.7%	55.5%	96.2%	88.3%	100.0%	99.2%	100.0%
13	8.8%	9.2%	49.9%	65.5%	82.1%	95.6%	98.1%	99.9%
14	**3.0%**	1.7%	**42.4%**	22.3%	**77.9%**	51.0%	**96.1%**	85.4%
15	1.3%	0.4%	**35.9%**	2.7%	**70.2%**	10.6%	**90.5%**	23.7%
16	0.3%	0.0%	**23.6%**	0.1%	**60.6%**	0.8%	**85.2%**	2.2%
17	0.1%	0.0%	**14.1%**	0.0%	**50.8%**	0.1%	**79.1%**	0.4%
18	0.0%	0.0%	**8.8%**	0.0%	**37.3%**	0.0%	**65.5%**	0.0%
19	0.0%	0.0%	**2.0%**	0.0%	**16.8%**	0.0%	**48.2%**	0.0%
20	0.0%	0.0%	0.7%	0.0%	**5.1%**	0.0%	**21.3%**	0.0%

smaller time steps, lighter agents are now allowed to take larger time steps from a richer set of directions which are aligned with — but otherwise different from, the gradient direction. This 'greedy' exploration of the ambient space by lighter agents, increases their likelihood of encountering a new neighborhood with a better minimum, which may place one of them as the new heaviest minimizer and so on.

1.1 Why Randomization Is Important

We compare the swarm-based method

$$\mathbf{x}_i^{n+1} = \mathbf{x}_i^n - h(\mathbf{x}_i^n, \lambda \widetilde{m}_i^{n+1})\mathbf{p}_i^n, \qquad i = 1, 2, \ldots, N, \tag{1.7}$$

in two scenarios: with the gradient direction for SBGD, $\mathbf{p}_i^n = \nabla F(\mathbf{x}_i^n)$ and with the randomized direction for SBRD, $\mathbf{p}_i^n = |\nabla F(\mathbf{x}_i^n)|\boldsymbol{\omega}_i^n$ in (1.4a)–(1.4c). The same backtracking protocol was implemented in both cases. The advantage of randomization in exploring larger regions becomes apparent in SBRD when the number of agents is larger than the dimension of the search space, $N > d$. The results recorded in Table 1 for the Ackley function show that SBRD optimization outperforms SBGD optimization in higher dimensions.

More can be found in numerical simulations of several benchmark problems presented in Sect. 4.

Algorithm 1 Random descent direction $\mathbf{p}_i^n$ for agent $\mathbf{x}_i^n$ with relative mass $\widetilde{m}_i^{n+1}$

Set $\mathbf{q}_i^n = \frac{\nabla F(\mathbf{x}_i^n)}{|\nabla F(\mathbf{x}_i^n)|}$
Choose random r such that $\frac{1}{2}(1+\widetilde{m}_i^n) < r < 1$
Set random vector $\mathbf{Y} \in \mathbb{R}^{d-1}$ with $Y(i) \sim \mathcal{N}(0,1)$ so that $\mathbf{Y}/|\mathbf{Y}| \in \mathbb{S}^{d-2}$
for $i = 1$ to $d-1$ **do**
  Set $X(i) = \sqrt{1-r^2}\frac{Y(i)}{|\mathbf{Y}|}$
end for
Set $X(d) = r$ so $\mathbf{X} = (X(1), \ldots, X(d-1), X(d)) \in \mathbb{S}^{d-1}$
if $1 - \mathbf{q}_i^n(d) \neq 0$ **then**
  Set $\mathbf{v}_i^n = \mathbf{q}_i^n - \mathbf{z}$ where $\mathbf{z} := (0, \ldots, 0, 1)$ is the north pole of $\mathbb{S}^{d-1}$
  Set $\boldsymbol{\omega}_i^n = \mathbf{X} - 2\frac{\langle \mathbf{v}_i^n, \mathbf{X}\rangle}{|\mathbf{v}_i^n|^2}\mathbf{v}_i^n$ % Simplification: $|\mathbf{v}_i^n|^2 = 2\big(1 - q_i^n(d)\big)$
else
  Set $\boldsymbol{\omega}_i^n = \mathbf{X}$
end if
Set $\mathbf{p}_i^n = |\nabla F(\mathbf{x}_i^n)|\boldsymbol{\omega}_i^n$

2 Swarm-Based Random Descent (SBRD). Implementation of Algorithm

The SBRD iterations are summarized in (2.1).

$$\left\{\begin{array}{l}
\left\{\begin{array}{ll} m_i^{n+1} &= m_i^n - \eta_i^n m_i^n, \quad i \neq i_n := \text{argmin}_i\, F_i^n \\ m_{i_n}^{n+1} &= m_{i_n}^n + \sum_{i \neq i_n} \eta_i^n m_i^n, \end{array}\right\} \eta_i^n := \left(\dfrac{F_i^n - F_{\min}^n}{F_{\max}^n - F_{\min}^n}\right)^q \\
\quad \widetilde{m}_i^{n+1} := \dfrac{m_i^{n+1}}{m_+^{n+1}}, \quad m_+^{n+1} := \max_i m_i^{n+1} \\
\left\{\begin{array}{lll} \mathbf{p}_i^n &:= \mathbf{p}_i^n(\mathbf{x}_i^n, \widetilde{m}_i^{n+1}) & \left\{\begin{array}{l} \text{Choose a random } r \in \mathcal{U}\big[\frac{1}{2}(1+\widetilde{m}_i^{n+1}), 1\big]; \\ \text{Algorithm 1 computes } \mathbf{p}_i^n \text{ such that} \\ \big\langle \mathbf{p}_i^n, \nabla F(\mathbf{x}_i^n)\big\rangle = r|\nabla F(\mathbf{x}_i^n)|^2 \end{array}\right. \\ h_i^n &:= h(\mathbf{x}_i^n, \lambda\widetilde{m}_i^{n+1}) & \text{Backtracking protocol in Algorithm 2} \\ \mathbf{x}_i^{n+1} &= \mathbf{x}_i^n - h_i^n \mathbf{p}_i^n, & \end{array}\right.
\end{array}\right. \tag{2.1}$$

The first part encodes the mass transfer from high to low ground in terms of a communication protocol, that dictates mass transition factors, $\{\eta_i^n\}$. The second part encodes the stepping in a descent direction, $h_i^n \mathbf{p}_i^n$, based on two mass-dependent procedures:

(i) a random choice of the descent direction, $\mathbf{p}_i^n = \mathbf{p}_i^n(\mathbf{x}_i^n, \widetilde{m}_i^{n+1})$, whose orientation is aligned within a random opening away from the orientation of the gradient $\nabla F(\mathbf{x}_i^n)/|\nabla F(\mathbf{x}_i^n)|$; and

(ii) a backtracking strategy for adjusting the step size, $h_i^n = h\big(\mathbf{x}_i^n, \lambda\widetilde{m}_i^{n+1}\big)$, which secures the desired descent property. Observe that both the direction and step size are adjusted to the position and the relative mass of a given agent.

These procedures are summarized in the following pseudo-codes.

2.1 A Protocol for Random Choice of the Descent Direction

Algorithm 1 picks a random orientation lying in the spherical cap of the unit sphere, $\boldsymbol{\omega}_i^n \in \mathbb{S}^{d-1}$, centered around the gradient orientation, $\mathbf{q}_i^n = \frac{\nabla F(\mathbf{x}_i^n)}{|\nabla F(\mathbf{x}_i^n)|}$, and then sets the descent

direction $\mathbf{p}_i^n = |\nabla F(\mathbf{x}_i^n)|\boldsymbol{\omega}_i^n$. To this end, we proceed in two steps. First, sampling a randomly chosen point, $\mathbf{X} = \big(X(1), \ldots, X(d-1), X(d)\big) \in \mathbb{S}^{d-1}$, in the spherical cap centered around the north pole, $\mathbf{z} = (0, 0, \ldots, 1)$,

$$X(i) = \begin{cases} \sqrt{1-r^2}\dfrac{Y(i)}{|\mathbf{Y}|} & Y(i) \sim \mathcal{N}(0,1),\ i = 1, 2, \ldots d-1, \\ r, & i = d. \end{cases}$$

Note that $\mathbf{Y} = \big(\sum_i Y^2(i)\big)^{-1/2}\big(Y(1), \ldots, Y(d-1)\big)$ is a random point (with normally distributed components) on $\mathbb{S}^{d-2}$ and therefore $\mathbf{X}$ above is the projection of that random $\mathbf{Y}$ onto the spherical cap of $\mathbb{S}^{d-1}$ dictated by r. Here r is a randomly chosen parameter from a uniform distribution in $\frac{1}{2}(1 + \widetilde{m}_i^{n+1}) < r < 1$; thus, the spherical cap, shown as the shaded area in Fig. 1, has an opening angle of $\theta = \arccos(r)$, ranging from $\theta = 60^\circ$ for lightest agents to the gradient orientation, $\theta = 0^\circ$, for the heaviest agent. In the second step, Algorithm 1 uses the unitary (Householder) reflection which reflects the north pole $\mathbf{z}$ to $\mathbf{q}_i^n$

$$\mathbb{P}_i^n = \mathbb{I} - 2\frac{\mathbf{v}_i^n(\mathbf{v}_i^n)^\top}{|\mathbf{v}_i^n|^2}, \qquad \mathbf{v}_i^n := \mathbf{q}_i^n - \mathbf{z},$$

and then reflects $\mathbf{X}$ into the desired $\boldsymbol{\omega}_i^n := \mathbb{P}_i^n\mathbf{X}$, see Fig. 1,

2.2 Backtracking — a Protocol for Time Stepping

The direction $\mathbf{p}_i^n$ computed in Algorithm 1 is partially aligned with $\nabla F(\mathbf{x}_i^n)$ so that (1.5) holds. Once the direction $\mathbf{p}_i^n$ is set, the new position $\mathbf{x}_i^{n+1}(h) = \mathbf{x}_i^n - h\mathbf{p}_i^n$ is viewed as a function of the step size h, and the objective is to select an appropriate step size, $h_i^n = h\big(\mathbf{x}_i^n, \lambda\widetilde{m}_i^{n+1}\big)$, which ensures the corresponding descent bound (1.6),

$$F(\mathbf{x}_i^{n+1}) \leqslant F(\mathbf{x}_i^n) - \frac{1}{2}\lambda\widetilde{m}_i^{n+1}h_i^n|\nabla F(\mathbf{x}^n)|^2, \qquad \mathbf{x}_i^{n+1} = \mathbf{x}_i^n - h_i^n\mathbf{p}_i^n, \quad 0 < \lambda < 1. \tag{2.2}$$

A proper strategy for choosing such step size is based on the classical backtracking line search, [17, §3], which is a computational realization of the well-known Wolfe conditions [1, 24]. Recall that by Taylor's expansion (1.6), the desired bound holds for any sufficiently small step size, $h_i^n \ll 1$. In fact, any step size $h < 1/L$, which need not be small, will suffice for the descent property, except that we do not have apriori access to the value of L. Our aim, therefore, is to choose a relatively large step size h, that even if not optimally tuned with $1/L$, it is still large enough to enforce the descent term $\frac{1}{2}\lambda\widetilde{m}_i^{n+1}h_i^n|\nabla F(\mathbf{x}_i^n)|^2$. To this end, one employs a dynamic adjustment, starting with a relatively large $h = h_0$ (say – $h_0 = 1$) for which one expects

$$F\big(\mathbf{x}_i^n - h\mathbf{p}_i^n\big) > F(\mathbf{x}^n) - \frac{1}{2}\lambda\widetilde{m}_i^{n+1}h|\nabla F(\mathbf{x}_i^n)|^2,$$

and then successively shrink the step size, $h \to \gamma h$, using a shrinkage factor $0 < \gamma < 1$, until the descent condition (2.2) is fulfilled. Adjusting the shrinkage parameter γ requires careful consideration of the trade-off between the cost of a refined $\gamma \sim 1$ vs. improved performance with a crude $\gamma \ll 1$.

The pseudo-code for computing the SBRD steps based on backtracking line search is given in Algorithm 2 below.

Algorithm 2 Backtracking line search

Set the shrinkage parameter, $\gamma \in (0,1)$

Set the relative mass $\widetilde{m}_i^{n+1} = \frac{m_i^{n+1}}{m_+^{n+1}}$

Initialize the step size $h = h_0$.

while $F\big(\mathbf{x}_i^n - h\mathbf{p}_i^n\big) > F(\mathbf{x}_i^n) - \frac{1}{2}\lambda\widetilde{m}_i^{n+1}h|\nabla F(\mathbf{x}_i^n)|^2$ **do**

$h \leftarrow \gamma h$.

end while

Set $h_i^n = h\big(\mathbf{x}_i^n, \lambda\widetilde{m}_i^{n+1}\big) \leftarrow h$

A stepping protocol for a non-convex optimization is required to strike a balance between small steps in the vicinity of a potential minimizer and larger steps which avoid being trapped in local basins of attraction. The backtracking protocol achieves such a balance by adjusting the step size of each agent according to its relative mass, $\widetilde{m}_i^{n+1}$

$$h_i^n = h\big(\mathbf{x}_i^n, \lambda\widetilde{m}_i^{n+1}\big) \qquad \widetilde{m}_i^{n+1} := \frac{m_i^{n+1}}{m_+^{n+1}}, \quad m_+^{n+1} := \max_i m_i^{n+1}, \quad 0 < \lambda < 1, \tag{2.3}$$

where $h(\mathbf{x}_i^n, \cdot)$ is a decreasing function of the relative mass $\lambda\widetilde{m}_i^{n+1}$. Thus, our mass-dependent backtracking is an adaptive protocol: it adapts itself from small time steps in the steepest gradient direction for heavier agents which lead the swarm, to larger steps in randomly chosen directions (that may differ from the steepest descent) for lighter agents which are the explorers of the swarm, exploring the ambient space.

The descent property. The backtracking Algorithm 2 yields a step size $h_i^n = h(\mathbf{x}_i^n, \lambda\widetilde{m}_i^{n+1})$ with a lower bound $h_i^n > \frac{\gamma}{L}$ with L denoting a Lipschitz bound of ∇F which is assumed to exists, $L = \max_{\mathbf{x}\in\Omega} \|D^2F(\mathbf{x})\| < \infty$. Indeed, this can be argued by contradiction: if $\frac{h_i^n}{\gamma} \leqslant \frac{1}{L}$ then by (1.5) we would have,

$$\begin{aligned} F\Big(\mathbf{x}_i^n - \frac{h_i^n}{\gamma}\mathbf{p}_i^n\Big) &\leqslant F(\mathbf{x}^n) - \frac{h_i^n}{\gamma}\langle \mathbf{p}_i^n, \nabla F(\mathbf{x}_i^n)\rangle + \frac{L}{2}\big(\frac{h_i^n}{\gamma}\big)^2|\mathbf{p}_i^n)|^2 \\ &\leqslant F(\mathbf{x}_i^n) - \Big(\frac{1+\widetilde{m}_i^n}{2} - \frac{L}{2}\frac{h_i^n}{\gamma}\Big)\frac{h_i^n}{\gamma}|\nabla F(\mathbf{x}_i^n)|^2 \\ &\leqslant F(\mathbf{x}_i^n) - \frac{1}{2}\lambda\widetilde{m}_i^{n+1}\frac{h_i^n}{\gamma}|\nabla F(\mathbf{x}_i^n)|^2, \quad \lambda < 1. \end{aligned}$$

But this contradicts the fact that the backtracking iterations fail to satisfy such inequality with step size h_i^n/γ, since according to Algorithm 2, $F(\mathbf{x}_i^n - (h_i^n/\gamma)\mathbf{p}_i^n) > F(\mathbf{x}_i^n) - \frac{1}{2}\lambda\widetilde{m}_i^{n+1}(h_i^n/\gamma)|\nabla F(\mathbf{x}_i^n)|^2$ (in fact, the largest step size that succeeds in securing reverse inequality is with time step h_i^n, $h_i^n < h_i^n/\gamma$). This contradiction confirms that $h_i^n > \frac{\gamma}{L}$, which in turn enables us to convert the descent bound (2.2) into a precise descent property,

$$F(\mathbf{x}_i^{n+1}) \leqslant F(\mathbf{x}_i^n) - \frac{1}{2}\lambda\widetilde{m}_i^{n+1}h_i^n|\nabla F(\mathbf{x}_i^n)|^2 \leqslant F(\mathbf{x}_i^n) - \frac{\gamma}{2L}\lambda\widetilde{m}_i^{n+1}|\nabla F(\mathbf{x}_i^n)|^2 \tag{2.4}$$

The descent property we obtain is constrained by an additional factor of $\frac{1}{4}$ compared to the standard version of SBGD that relies on the gradient direction [14, Proposition 5.2]. However, the randomization of the descent direction brings the advantage of allowing lighter

Algorithm 3 Swarm-Based Random Descent Method

Set the parameters: $tolm$, $tolmerge$, $tolres$, and $nmax$
Set the number of agents, N, and the mass transfer parameter, $q \geqslant 1$
Randomly generate initial positions: $\mathbf{x}_1^0, \ldots, \mathbf{x}_N^0$
Set initial mass for all agents: $m_1^0 = \cdots = m_N^0 = 1/N$
for $n = 0, 1, 2, \ldots, nmax$ **do**
  Merge agents if their distance $< tolmerge$
  Set the index of the optimal agent: $i_n = \operatorname{argmin}_i F(\mathbf{x}_i^n)$
  Set $F_{\min} = F(\mathbf{x}_{i_n}^n)$ and $F_{\max} = \max_i F(\mathbf{x}_i^n)$
  for $i = 1, \ldots, N$ **and** $i \neq i_n$ **do**
    if $m_i^n < 1/N \cdot tolm$ **then**
      Set $m_i^{n+1} = 0$
      Reduce the number of active agents: $N \leftarrow N - 1$
    else
      Set $m_i^{n+1} = m_i^n - \eta_i^n m_i^n$ where $\eta_i^n = \left(\frac{F(\mathbf{x}_i^n) - F_{\min}^n}{F_{\max}^n - F_{\min}^n} \right)^q$
    end if
  end for
  Set $m_{i_n}^{n+1} = m_{i_n}^n + \sum_{i \neq i_n} \eta_i^n m_i^n$
  Set $m_+ = \max_i m_i^{n+1}$
  for $i = 1, \ldots, N$ **do**
    Compute relative masses $\widetilde{m}_i^{n+1} = \frac{m_i^{n+1}}{m_+}$
    Compute a random descent direction: $\mathbf{p}_i^n$ (using Algorithm 1)
    Compute the step size: $h = h(\mathbf{x}_i^n, \lambda \widetilde{m}_i^{n+1})$ (using Algorithm 2)
    Update position: $\mathbf{x}_i^{n+1} = \mathbf{x}_i^n - h\mathbf{p}_i^n$
  end for
  if $|\mathbf{x}_i^{n+1} - \mathbf{x}_i^n| \leqslant tolres$ **then**
    break
  end if
end for

agents to explore a wider range of directions. As we will see later, this exploration leads to substantial improvements in the optimization process in high dimensions.

It is important to note that heavy agents still adhere to the steepest descent along the gradient direction. The spherical cone of random directions is narrower for heavier agents. In fact, the heaviest agent strictly follows the steepest descent with $\mathbf{p}_i^n = \nabla F(\mathbf{x}_i^n)$, eliminating the need for a random choice at this particular point.

2.3 SBRD Pseudocode

The pseudocode of the SBRD method is presented in Algorithm 3. The initial setup involves N randomly distributed agents $\mathbf{x}_1^0, \ldots, \mathbf{x}_N^0$, associated with initial masses $m_1^0, \ldots, m_N^0$. Initially, all agents are assigned equal masses, $m_j^0 = 1/N$, $j = 1, \ldots, N$. At each iteration, the agent positioned at $\mathbf{x}_{i_n} = \operatorname{argmin}_{\mathbf{x}_i^n} F(\mathbf{x}_i^n)$ attains the minimal value, while the other agents transfer part of their masses to that minimizer $\mathbf{x}_{i_n}$. Then all the agents are updated with the gradient descent method using the direction obtained in (1.4b) and step size in (2.3).

We use three tolerance factors:

· *tolm*: If an agent's mass falls below this threshold, the agent is eliminated, and its remaining mass is transferred to the optimal agent at $\mathbf{x}_{i_n}$.

· *tolmerge*: Agents that are sufficiently close to each other, i.e., their distance is below this threshold, are merged into a new agent. The masses of the merged agents are combined into the newly generated agent.

· *tolres*: The iterations terminate when the descent of the minimizer between two consecutive iterations falls below this threshold.

3 Convergence and Error Analysis

The study of convergence and error estimates for the SBRD method requires quantifying the behavior of F. Here we emphasize that the required smoothness properties of F are only sought in the region explored by the SBRD iterations. We assume that there exists a *bounded* region, $\Omega \ni \mathbf{x}_i^n$ for all agents. Since the SBRD allows light agents to explore the ambient space with large step size (starting with h_0), we do not have an apriori bound on Ω; in particular, the footprint of the SBRD crowd $\mathrm{conv}_i\{\mathbf{x}_i^n\}$ may expand well beyond its initial convex hull $\mathrm{conv}_i\{\mathbf{x}_i^0\}$. The expansion of the initial convex hull is an essential feature of the algorithm that allows the agents to find minima outside their initial range, demonstrated in the numerical experiments with shifted initial data domains such as in Table 5.

We consider the class of loss functions, $F \in C^2(\Omega)$, with Lipschitz bound $L = \max_\Omega \|D^2 F\| < \infty$,

$$|\nabla F(\mathbf{x}) - \nabla F(\mathbf{y})| \leqslant L|\mathbf{x} - \mathbf{y}|, \quad \forall \mathbf{x}, \mathbf{y} \in \Omega. \tag{3.1}$$

3.1 Convergence to a Band of Local Minima

Our next proposition provides a precise quantitative description for the convergence of the SBRD method. The convergence is determined by the time series of SBRD minimizers, $\{\mathbf{X}_-^n\}$,

$$\mathbf{X}_-^n = \mathbf{x}_{i_n}^n, \qquad i_n := \operatorname*{argmin}_i F(\mathbf{x}_i^n). \tag{3.2a}$$

We shall also need the time series of its heaviest agents, $j_n := \mathrm{argmax}_i\, m_i^n$; to this end, we let $\mathbf{X}_+^n$ denote the *parent* of the heaviest agent at $t = t^{n+1}$

$$\mathbf{X}_+^n = \mathbf{x}_{j_{n+1}}^n, \qquad j_{n+1} := \operatorname*{argmax}_i m_i^{n+1}. \tag{3.2b}$$

The interplay between minimizers and the communication of masses leads to a gradual mass shift from higher ground to the minimizers. Eventually, the two sequences coincide when the SBRD minimizers gain enough mass to assume the role of heaviest agents. Finally, we introduce the scaling $M = \max_j F(\mathbf{x}_j^0) - F(\mathbf{x}^*)$ where $\mathbf{x}^*$ is the global minimum. Since $F(\mathbf{x}_i^n)$ are decreasing, we conclude that the SBRD iterations remain within that range, namely

$$F(\mathbf{x}_i^n) - F(\mathbf{x}_j^n) \leqslant M, \qquad \forall n, j, \quad M := \max F(\mathbf{x}_i^0) - F(\mathbf{x}^*). \tag{3.3}$$

Proposition 3.1 *Consider the SBRD iterations* (2.1) *with random-based search direction,* $\mathbf{p}_i^n$, *determined by Algorithm* 1, *and with a step-size* (2.3), $h_i^n = h\big(\mathbf{x}_i^n, \lambda \widetilde{m}_i^{n+1}\big)$, *determined by backtracking line search of Algorithm* 2.

Let $\{\mathbf{X}^n_-\}_{n\geqslant 0}$ *and* $\{\mathbf{X}^n_+\}_{n\geqslant 0}$ *denote the time sequence of SBRD minimizers and, respectively, (parent of) heaviest agents outlined in* (3.2a)–(3.2b) *Then, there exists a constant,* $C = C(\gamma, L, M, \lambda)$ *given in* (3.10) *below, such that we have summability of gradients*

$$\sum_{n=0}^{\infty} \delta_n^2 \cdot \min\left\{1, \delta_n^{2q}\right\} < C \min_i F(\mathbf{x}_i^0), \qquad \delta_n := \min\{|\nabla F(\mathbf{X}^n_+)|, |\nabla F(\mathbf{X}^n_-)|\}. \tag{3.4}$$

Here, $q \geqslant 1$ *is the mass transfer parameter in* (1.1).

Proof Our purpose is to find a lower bound on the relative masses, $\widetilde{m}_i^{n+1} = \frac{m_i^{n+1}}{m_{j_{n+1}}^{n+1}}$, which will dictate the descent property of the different agents according to (2.4). Observe that for the heaviest agent, $i = j_{n+1}$, (2.4) with $\widetilde{m}_{j_{n+1}}^{n+1} = 1$ implies

$$F(\mathbf{x}^{n+1}_{j_{n+1}}) \leqslant F(\mathbf{X}^n_+) - \frac{\gamma}{2L}\lambda|\nabla F(\mathbf{X}^n_+)|^2, \qquad \mathbf{X}^n_+ = \mathbf{x}^n_{j_{n+1}}. \tag{3.5}$$

We distinguish between two scenarios. The first is a canonical scenario in which the minimizing agent at $t = t^n$ coincides with the heaviest agent at time t^{n+1}, namely, when $i_n = j_{n+1}$, or $\mathbf{X}^n_- = \mathbf{X}^n_+$. Then (3.5) implies

$$F(\mathbf{X}^{n+1}_-) \leqslant F(\mathbf{x}^{n+1}_{j_{n+1}}) \leqslant F(\mathbf{X}^n_-) - \frac{\gamma}{2L}\lambda|\nabla F(\mathbf{X}^n_-)|^2, \qquad \mathbf{X}^n_- = \mathbf{X}^n_+. \tag{3.6}$$

The inequality on the left follows since $\mathbf{X}^{n+1}_- = \mathbf{x}^{n+1}_{i_{n+1}}$ is the global minimizer at t^{n+1}.

Next, we consider the second scenario $i_n \neq j_{n+1}$, that is — when the mass of the minimizer $m^{n+1}_{i_n}$ did not yet 'catch-up' the position as the heaviest agent so that $\widetilde{m}^{n+1}_{i_n} = \frac{m^{n+1}_{i_n}}{m^{n+1}_{j_{n+1}}} < 1$. Yet, we claim that the descent property associated with the relative mass $\widetilde{m}^{n+1}_{i_n}$ cannot be arbitrarily small. We consider two sub-cases, depending on the size of $F(\mathbf{X}^n_+) - F(\mathbf{X}^n_-)$.

Case (i). Assume $F(\mathbf{X}^n_+) - F(\mathbf{X}^n_-) \leqslant \frac{\gamma}{4L}\lambda|\nabla F(\mathbf{X}^n_+)|^2$. Appealing to (3.5) we find

$$F(\mathbf{X}^{n+1}_-) \leqslant F(\mathbf{x}^{n+1}_{j_{n+1}}) \leqslant F(\mathbf{X}^n_+) - \frac{\gamma\lambda}{2L}|\nabla F(\mathbf{X}^n_+)|^2 \leqslant F(\mathbf{X}^n_-) - \frac{\gamma\lambda}{4L}|\nabla F(\mathbf{X}^n_+)|^2. \tag{3.7}$$

The inequality on the left follows since $\mathbf{X}^{n+1}_-$ is the global minimizer at t^{n+1}; the middle inequality quotes (3.5) and the last inequality follows from our assumption.

Case (ii). Finally, we remain with the case

$$F(\mathbf{X}^n_+) - F(\mathbf{X}^n_-) > \frac{\gamma}{4L}\lambda|\nabla F(\mathbf{X}^n_+)|^2.$$

We claim that in this case,

$$\widetilde{m}^{n+1}_{i_n} > \frac{1}{M^2}\big(F(\mathbf{X}^n_+) - F(\mathbf{X}^n_-)\big)^2 \geqslant \Big(\frac{\gamma\lambda}{4ML}\Big)^q |\nabla F(\mathbf{X}^n_+)|^{2q}. \tag{3.8}$$

Indeed, since agent j_{n+1} is not the minimizer at time $t = t^n$, namely $j_{n+1} \neq i_n$, then it had to shed a portion of its mass, $m^n_{j_{n+1}} - \eta^n_+ m^n_{j_{n+1}} \to m^{n+1}_{j_{n+1}}$, which was transferred to the minimizer $m^{n+1}_{i_n} \leftarrow m^n_{i_n} + \cdots + \eta^n_+ m^n_{j_{n+1}}$. Thus, the loss of mass by heavy agent

$$m^{n+1}_{j_{n+1}} = m^n_{j_{n+1}} - \eta^n_+ m^n_{j_{n+1}}, \quad \eta^n_+ = \Big(\frac{F(\mathbf{x}^n_{j_{n+1}}) - F(\mathbf{x}^n_{i_n})}{\max_j F(\mathbf{x}^n_j) - F(\mathbf{x}^n_{i_n})}\Big)^q \geqslant \frac{1}{M^q}\big(F(\mathbf{X}^n_+) - F(\mathbf{X}^n_-)\big)^q.$$

was *gained* by the minimizer agent, $i = i_n$. Therefore, the relative mass of that minimizer is at least as large as claimed in (3.8)

$$\widetilde{m}_{i_n}^{n+1} = \frac{m_{i_n}^{n+1}}{m_{j_{n+1}}^{n+1}} > \frac{\eta_+^n}{1-\eta_+^n} \geqslant \frac{1}{M^q}\big(F(\mathbf{X}_+^n) - F(\mathbf{X}_-^n)\big)^q \geqslant \Big(\frac{\gamma\lambda}{4ML}\Big)^q |\nabla F(\mathbf{X}_+^n)|^{2q}.$$

The descent property (2.4) together with (3.8) imply

$$\begin{aligned} F(\mathbf{X}_-^{n+1}) &\leqslant F(\mathbf{x}_{i_n}^{n+1}) \leqslant F(\mathbf{x}_{i_n}^n) - \frac{\gamma}{2L}\lambda \widetilde{m}_{i_n}^{n+1} |\nabla F(\mathbf{x}_{i_n}^n)|^2 \\ &\leqslant F(\mathbf{X}_-^n) - \frac{\gamma\lambda}{2L}\Big(\frac{\gamma\lambda}{4ML}\Big)^q |\nabla F(\mathbf{X}_+^n)|^{2q} \cdot |\nabla F(\mathbf{X}_-^n)|^2. \end{aligned} \tag{3.9}$$

Combining (3.6), (3.7) and (3.9) we find

$$\begin{aligned} &F(\mathbf{X}_-^{n+1}) \\ &\quad\leqslant F(\mathbf{X}_-^n) - \frac{1}{C}\min\big\{|\nabla F(\mathbf{X}_-^n)|^2, |\nabla F(\mathbf{X}_+^n)|^2, |\nabla F(\mathbf{X}_+^n)|^{2q} \cdot |\nabla F(\mathbf{X}_-^n)|^2\big\} \\ &\quad\leqslant F(\mathbf{X}_-^n) - \frac{1}{C}\delta_n^2 \min\big\{1, \delta_n^{2q}\big\}, \qquad C = \frac{4L}{\gamma\lambda} \cdot \max\Big\{2, \Big(\frac{4ML}{\gamma\lambda}\Big)^q\Big\}. \end{aligned} \tag{3.10}$$

The desired bound (3.4) follows by a telescoping sum. □

The summability bound (3.4) implies that eventually, for large enough $n > N_0$, the minimizers and (parent of) heaviest SBRD agents, $\delta_n < 1$ and hence

$$\sum_{n>N_0}^{\infty} \min\{|\nabla F(\mathbf{X}_+^n)|, |\nabla F(\mathbf{X}_-^n)|\}^{2(q+1)} \leqslant C \min_i F(\mathbf{x}_i^0).$$

It follows that there exist sub-sequences, $\mathbf{X}^{n_\alpha} \in \{\mathbf{X}_+^n\}_{n\geqslant N_0} \cup \{\mathbf{X}_-^n\}_{n\geqslant N_0}$, satisfying the Palais-Smale condition, [18], $F(\mathbf{X}^{n_\alpha}) \leqslant \max_i F(\mathbf{x}_i^0)$ while $\nabla F(\mathbf{X}^{n_\alpha}) \overset{\alpha\to\infty}{\longrightarrow} 0$. Arguing along [14, Theorem 5.4] we summarize by stating the following.

Theorem 3.2 *Let* $\{\mathbf{X}^n\}_{n\geqslant N_0} := \{\mathbf{X}_+^n\}_{n\geqslant N_0} \cup \{\mathbf{X}_-^n\}_{n\geqslant N_0}$ *denote the combined time sequence of SBRD minimizers/heaviest agents,* (3.2a)–(3.2b)*. Then there exist one or more sub-sequences,* $\{\mathbf{X}^{n_\alpha},\ \alpha = 1, 2, \dots, \}$*, that converge to a band of local minima with equal heights,*

$$\mathbf{X}^{n_\alpha} \overset{\alpha\to\infty}{\longrightarrow} \mathbf{X}_\alpha^* \ \text{ such that } \nabla F(\mathbf{X}_\alpha^*) = 0, \ \text{ and } F(\mathbf{X}_\alpha^*) = F(\mathbf{X}_\beta^*) \tag{3.11}$$

In particular,in the generic case that F admits only distinct local minima in Ω*, namely — different local minima have different heights, then the whole sequence* $\mathbf{X}^n$ *converges to a local minimum.*

Proof Since we assume the sequence $\{\mathbf{X}_-^n\}$ is bounded in Ω, it has a converging sub-sequences. Take *any* such converging sub-sequence $\mathbf{X}_-^{n_\alpha} \to \mathbf{X}_\alpha^* \in \Omega$. By (3.4), $\nabla F(\mathbf{X}_-^{n_\alpha}) \to 0$ for all sub-sequences, and hence $\mathbf{X}_\alpha^*$ are local minimizers, $\nabla F(\mathbf{X}_\alpha^*) = 0$. Moreover, since $F(\mathbf{X}_-^n)$ is a decreasing, all $F(\mathbf{X}_\alpha^*)$ must have the same 'height'. The collection of equi-height minimizers $\{\mathbf{X}_\alpha^* \ \big|\ F(\mathbf{X}_\alpha^*) = F(\mathbf{X}_\beta^*)\}$ is the limit-set of $\{\mathbf{X}_-^n\}$. □

Moreover, for analytic F's, we can quantify the convergence *rate* (3.11). To this end we use Lojasiewicz inequality, [12, 13], which guarantees that each critical point of analytic F has "flatness" of some fixed order $\beta \in (1, 2]$ in the sense that there exists a neighborhood $\mathcal{N}_* \ni \mathbf{x}^*$ surrounding $\mathbf{x}^*$, an exponent β and a constant $\mu > 0$ such that

$$\mu |F(\mathbf{x}) - F(\mathbf{x}^*)| \leqslant |\nabla F(\mathbf{x})|^\beta, \qquad \forall \mathbf{x} \in \mathcal{N}_*. \tag{3.12}$$

Theorem 3.3 *Consider an analytic loss function F with minimal flatness $\beta \in (1, 2]$, such that the Lipschitz bound (3.1) holds. Let $\{\mathbf{X}_-^n\}_{n\geqslant 0}$ denote the time sequence of SBRD minimizers, (2.1), (2.3), with converging sub-sequence, $\{\mathbf{X}_-^{n_\alpha}\}_\alpha$, outlined in Theorem 3.2. Then, there exists a constant, $C = C(\gamma, \lambda, \mu)$, such that*

$$F(\mathbf{X}_-^{n_\alpha}) - F(\mathbf{X}_\alpha^*) \lesssim \Big(\frac{C}{n_\alpha}\Big)^{\beta'}, \qquad \beta' = \frac{\beta}{2(q+1)-\beta}, \quad \beta \in (1, 2). \tag{3.13}$$

Observe that as 'flatness', increases, β decreases the polynomial decay in (3.13). A more careful analysis which we omit,[4] allows to replace the factor $(q + 1)$ by q, in which case, (3.13) with $q = 1$, $\beta = 2$ implies exponential convergence.

Proof We summarize the different statements of descent properties in (3.6), (3.7) and (3.9), writing

$$F(\mathbf{X}_-^{n+1}) \leqslant F(\mathbf{X}_-^n) - \frac{1}{C}|\nabla F(\mathbf{X}_\pm^n)|^{2(q+1)}, \qquad n > N_0,$$

where $\mathbf{X}_\pm^n = \operatorname{argmin}_{\mathbf{X}_\pm^n}\{|\nabla F(\mathbf{X}_+^n)|, |\nabla F(\mathbf{X}_-^n)|\}$. We focus on the converging sub-sequence $\{\mathbf{X}_-^{n_\alpha}\}$,

$$F(\mathbf{X}_-^{n_\alpha+1}) - F(\mathbf{X}_\alpha^*) \leqslant F(\mathbf{X}_-^{n_\alpha}) - F(\mathbf{X}_\alpha^*) - \frac{1}{C}|\nabla F(\mathbf{X}_\pm^{n_\alpha})|^{2(q+1)}. \tag{3.14}$$

Using Lojasiewicz bound (3.12), and the fact that $F(\mathbf{X}_+^n) \geqslant F(\mathbf{X}_-^n)$, we find

$$|\nabla F(\mathbf{X}_\pm^{n_\alpha})|^\beta \geqslant \mu |F(\mathbf{X}_\pm^{n_\alpha}) - F(\mathbf{X}_\alpha^*)| \geqslant \mu |F(\mathbf{X}_-^{n_\alpha}) - F(\mathbf{X}_\alpha^*)|. \tag{3.15}$$

Combining (3.14), (3.15), we conclude that the error, $E_{n_\alpha} := F(\mathbf{X}_-^{n_\alpha}) - F(\mathbf{X}_\alpha^*)$, satisfies

$$E_{n_\alpha+1} \leqslant E_{n_\alpha} - \frac{1}{C}(\mu E_{n_\alpha})^{\frac{2(q+1)}{\beta}}, \qquad \mathbf{X}_-^{n_\alpha} \in \mathcal{N}_\alpha.$$

The solution of this Riccati inequality yields

$$F(\mathbf{X}_-^{n_\alpha}) - F(\mathbf{X}_\alpha^*) \leqslant \left\{ |\min_i F(\mathbf{x}_i^0) - F(\mathbf{X}_\alpha^*)|^{-1/\beta'} + \frac{1}{C}\mu^{\frac{2(q+1)}{\beta}} n_\alpha \right\}^{-\beta'}, \quad \beta' = \frac{\beta}{2(q+1)-\beta}.$$

and (3.13) follows. □

[4] Requires to eliminate case (ii) in the proof of proposition 3.1; consult [14].

4 Numerical Results

Initially, the agents are placed at random positions, $\{\mathbf{x}_i^0\}$ with equi-distributed masses $\{m_i^0 = 1/N\}$. Masses are transferred from the high to the lowest ground at each iteration. Since we implement a "survival of the fittest" protocol in which the agent with the worst (=highest) configuration is eliminated, the swarm size decreases, one agent at a time, until only the heaviest agent remains. Our choice for the time-stepping protocol, $h\big(\mathbf{x}, \lambda\widetilde{m}\big)$, is the *backtracking line search* outlined in §2.2, which is weighted by the relative masses, $\widetilde{m}_i^{n+1}$. The backtracking enforces a descent property for the SBRD iterations $\mathbf{x}_i^n$, and the parameter, $\lambda \in (0, 1)$, dictates how much the descent property holds in the sense that (1.3) is fulfilled.

We illustrate the performance of the multi-dimensional SBRD algorithm, (2.1), (1.4a)–(1.4c), in several benchmark test cases [15]. The results are based on $k = 1000$ runs of uniformly generated initial data in a hypercube. Backtracking parameters in Algorithm 2 are $\lambda = 0.2$ and $\gamma = 0.9$ and $h_0 = 1$. The parameters in Algorithm 3 are $tolm = 10^{-4}$, $tolmerge = 10^{-3}$, $tolmax = 10^{-4}$ and $nmax = 200$.

We use the *success rate* among the k independent simulations to evaluate the solution's quality. We consider a simulation to be successful if $\mathbf{x}_{SOL}$ is within the d-dimensional ball of the global minimum: $|\mathbf{x}^* - \mathbf{x}_{SOL}| \leqslant 0.1$. This condition ensures that the approximate solution lies in the basin of attraction of the global minimizer. In Sect. 4.1 we fix the mass transfer parameter $q = 2$; the effect of increasing $q = 4, 8$ is discussed in Sect. 4.2.

4.1 Examples of SBRD with Mass Transfer Parameter $q = 2$

Extensive comparisons of the gradient-based deterministic SBGD were performed in [14]. In this paper, we focus on the impact of randomization on success rates in comparison to SBGD. We consider four benchmarks using the Ackley, Rastrigin, Rosenbrock, Styblinski-Tang objective functions in d-dimensions.

The **Ackley** function

$$F_{\text{Ackley}}(\mathbf{x}) = -20\exp\Big\{-\frac{0.2}{\sqrt{d}}\Big\{\sum_{i=1}^{d} x_i^2\Big\}^{1/2}\Big\} - \exp\Big\{\frac{1}{d}\sum_{i=1}^{d}\cos(2\pi x_i)\Big\} + 20 + e, \tag{4.1}$$

and the **Rastrigin** function

$$F_{\text{Rstgin}}(\mathbf{x}) = 10d + \sum_{i=1}^{d}\Big\{x_i^2 - 10\cos(2\pi x_i)\Big\}. \tag{4.2}$$

have their global minimum at the origin, $\mathbf{x}^* = 0$. The **Rosenbrock** function

$$F_{\text{Rsnbrk}}(\mathbf{x}) = \sum_{i=1}^{d-1}\big(100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2\big), \tag{4.3}$$

has its global minimum at $\mathbf{x}^* = (1, \ldots, 1)$. And finally, the **Styblinski-Tang** function

$$F_{\text{ST}}(\mathbf{x}) = \frac{1}{2}\sum_{i=1}^{d}(x_i^4 - 16x_i^2 + 5x_i), \tag{4.4}$$

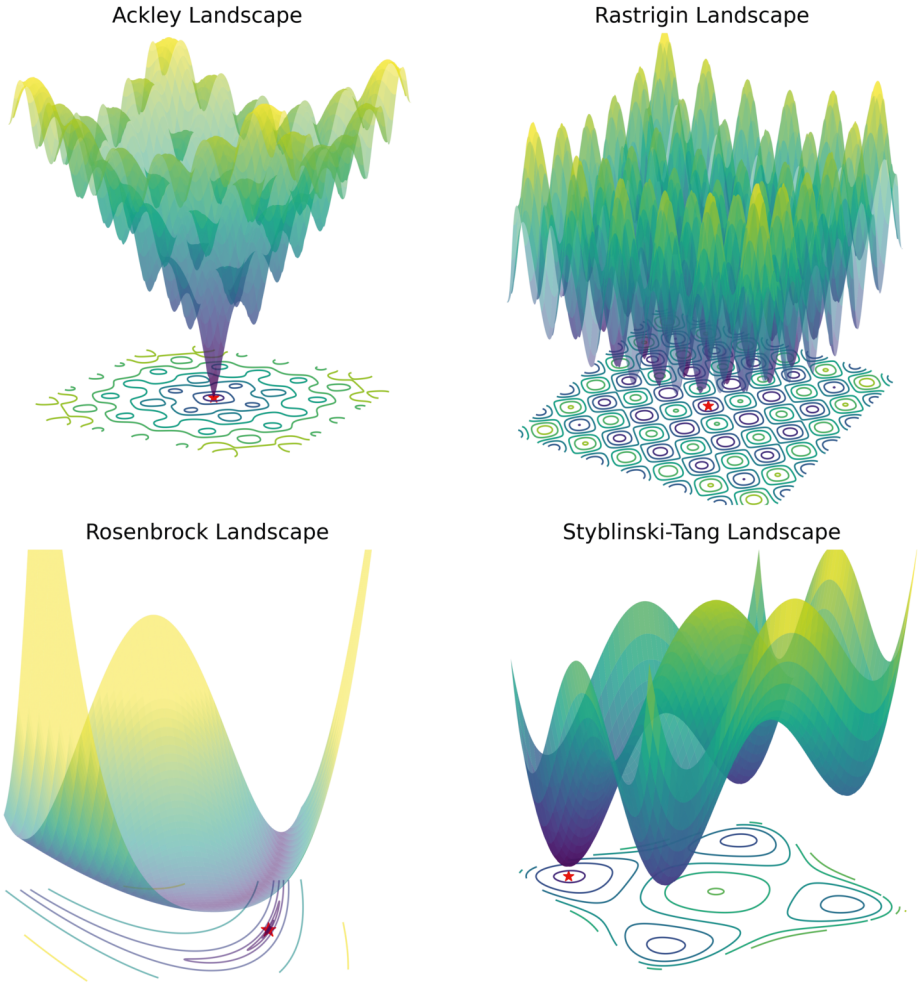

Fig. 2 Two-dimensional landscapes for the test functions Ackley (4.1), Rastrigin (4.2), Rosenbrock (4.3), and Styblinski-Tang (4.4) with a contour plot on the bottom and a red star indicating the global minimum (Color figure online).

has its global minimum at $\mathbf{x}^* = (-2.903534, \ldots, -2.903534)$. The two-dimensional landscapes of these benchmark examples are shown in Fig. 2.

Tables 2, 3 and 4 show the advantage of SBRD over SBGD for Rastrigin, Rosenbrock and Styblinski-Tang functions. We bold-face the success rate of SBRD in the tables when the advantage of SBRD over SBGD is at least 1%. Observe that the advantage of randomization is only relevant in higher dimensions where SBGD has a very low success rate. We recall that the same improved success rate of SBRD over SBGD was already recorded for the Ackley test function in Table 1. This observation remains valid when the range of initial agents for Ackley test function lies outside the neighborhood of its global minimum; this is documented in Table 5.

Figures 3 and 4 provide additional information about the 'inner working' of the SBRD dynamics. Figure 3 shows how the SBRD toggles between minimizers and heaviest agents:

Table 2 Success rates of SBRD vs. SBGD for global optimization of the d-dimensional Rastrigin function (4.2)

	$N = 10$		$N = 25$		$N = 50$		$N = 100$	
d	SBRD	SBGD	SBRD	SBGD	SBRD	SBGD	SBRD	SBGD
2	**31.9%**	28.0%	**96.8%**	67.8%	**100.0%**	95.3%	100.0%	100.0%
3	5.2%	5.6%	**17.6%**	13.6%	**57.9%**	28.6%	**92.4%**	52.0%
4	0.3%	1.0%	2.2%	3.9%	**7.2%**	5.7%	**17.9%**	11.4%
5	0.1%	0.0%	0.2%	0.4%	0.8%	0.4%	**3.2%**	1.2%
6	0.0%	0.0%	0.0%	0.0%	0.0%	0.1%	0.2%	0.4%

Table 3 Success rates for the optimization of the Rosenbrock function (4.3) with initial agents $\mathbf{x}_i^0 \in [-2.048, 2.048]^d$

	$N = 10$		$N = 25$		$N = 50$		$N = 100$	
d	SBRD	SBGD	SBRD	SBGD	SBRD	SBGD	SBRD	SBGD
2	**12.0%**	10.3%	**52.1%**	18.7%	**92.7%**	39.4%	**99.2%**	56.7%
3	2.4%	2.2%	8.1%	9.6%	27.2%	33.9%	**82.6%**	71.0%
4	2.3%	2.1%	3.5%	3.0%	**9.4%**	3.9%	**27.0%**	6.5%
5	1.1%	0.8%	1.3%	1.6%	**5.9%**	3.2%	**10.2%**	6.1%
6	0.5%	0.6%	1.1%	1.2%	1.6%	1.7%	**5.1%**	2.6%

Table 4 Success rates for the optimization of the Styblinski-Tang function (4.4) with initial agents $\mathbf{x}_i^0 \in [-3, 3]^d$

	$N = 10$		$N = 25$		$N = 50$		$N = 100$	
d	SBRD	SBGD	SBRD	SBGD	SBRD	SBGD	SBRD	SBGD
2	**97.0%**	92.8%	100.0%	99.9%	100.0%	100.0%	100.0%	100.0%
4	29.5%	35.3%	**83.7%**	79.0%	**99.2%**	97.4%	100.0%	99.9%
6	7.8%	10.4%	28.5%	32.5%	54.5%	55.4%	**86.3%**	83.2%
8	2.2%	2.5%	7.6%	9.7%	13.7%	18.7%	**36.7%**	35.4%
10	0.4%	0.6%	2.6%	3.2%	5.9%	6.0%	10.2%	12.5%
12	0.1%	0.2%	0.5%	0.8%	1.3%	2.2%	2.9%	3.8%

the loss function decays rapidly for the minimizers. Heavy agents, however, may arise due to merging multiple agents near local minima. The mass of such agents is then slowly transferred to the minimizers with a better minimum. Figure 4 demonstrates the difference between the randomized direction and the gradient direction.

Table 5 Same as Table 1 except the range of initial agents with $\mathbf{x}_i^0 \in [-3,-1]^d$ does not contain the global minimum of the Ackley function

	$N=10$		$N=25$		$N=50$		$N=100$	
d	SBRD	SBGD	SBRD	SBGD	SBRD	SBGD	SBRD	SBGD
12	2.8%	3.7%	39.3%	60.1%	74.8%	96.2%	94.5%	99.9%
14	0.3%	0.0%	**19.6%**	0.9%	**51.3%**	2.0%	**81.3%**	9.9%
16	0.0%	0.0%	**2.7%**	0.0%	**21.9%**	0.0%	**47.4%**	0.0%
18	0.0%	0.0%	0.0%	0.0%	0.7%	0.0%	**7.3%**	0.0%

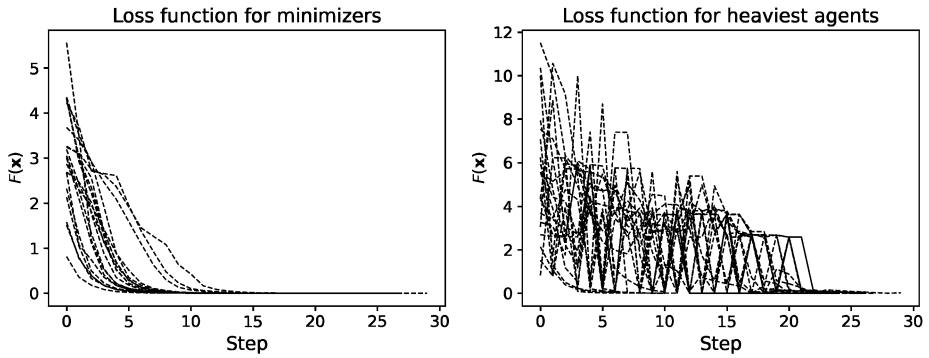

Fig. 3 Loss functions for minimizers and heaviest agents as defined in (3.2a)–(3.2b) and (3.2b) during optimization of the two-dimensional Ackley function for 20 simulations with $N = 50$ agents. Note the different scales on the y-axis between the minimizers and heaviest agents

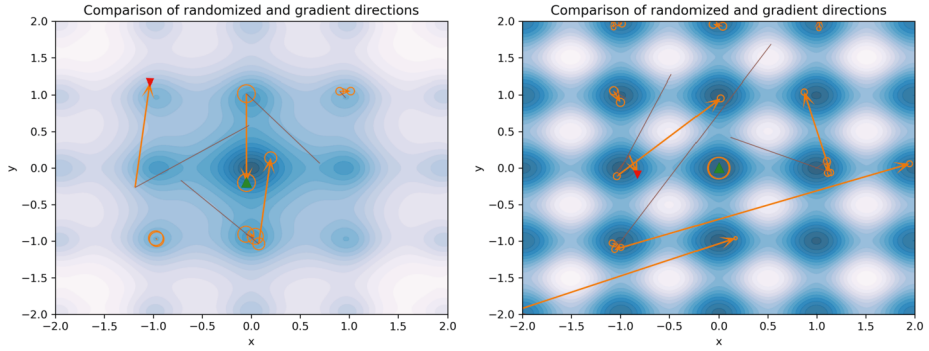

Fig. 4 A comparison of random descent, $-\mathbf{p}_i^n$ (orange arrows), and gradient descent, $-\nabla F(\mathbf{x}_i^n)$ (brown lines), during a simulation for the optimization of Ackley function (left) and the Rastrigin function (right). The green triangle is the current minimizer; the upside-down red triangle is the worst agent. The angles between the two directions and the step sizes are larger for lighter agents. Random descent is a better alternative to gradient descent for some agents and worse for others (Color figure online).

Table 6 Success rate of SBRD vs. SBGD for Ackley test function with mass transfer parameter $q = 4$

	$N = 10$		$N = 25$		$N = 50$		$N = 100$	
d	SBRD	SBGD	SBRD	SBGD	SBRD	SBGD	SBRD	SBGD
12	13.4%	24.6%	71.1%	97.1%	100.0%	100.0%	100.0%	100.0%
14	**3.8%**	1.3%	**60.0%**	22.0%	**100.0%**	49.9%	**100.0%**	84.1%
16	0.3%	0.0%	**38.3%**	0.1%	**95.0%**	0.8%	**100.0%**	1.6%
18	0.0%	0.0%	**16.3%**	0.0%	**79.7%**	0.0%	**99.6%**	0.0%
20	0.0%	0.0%	**1.4%**	0.0%	**25.1%**	0.0%	**74.5%**	0.0%

Table 7 Success rate of SBRD vs. SBGD for Ackley test function with mass transfer parameter $q = 8$

	$N = 10$		$N = 25$		$N = 50$		$N = 100$	
d	SBRD	SBGD	SBRD	SBGD	SBRD	SBGD	SBRD	SBGD
12	13.0%	20.7%	75.8%	93.3%	100.0%	99.8%	100.0%	100.0%
14	**4.2%**	1.1%	**66.9%**	15.3%	**100.0%**	39.4%	**100.0%**	77.8%
16	0.1%	0.0%	**38.4%**	0.1%	**99.8%**	0.8%	**100.0%**	1.4%
18	0.0%	0.0%	**14.6%**	0.0%	**87.3%**	0.0%	**100.0%**	0.0%
20	0.0%	0.0%	**1.0%**	0.0%	**30.7%**	0.0%	**84.7%**	0.0%

4.2 SBRD with Higher Order Mass Transition $q > 2$

In this section we revisit the benchmark examples with different mass transfer parameter q,

$$\begin{cases} m_i^{n+1} &= m_i^n - \eta_i^n m_i^n, \qquad i \neq i_n \\ m_{i_n}^{n+1} &= m_{i_n}^n + \sum_{i \neq i_n} \eta_i^n m_i^n, \end{cases} \tag{4.5}$$

$$\eta_i^n := \left(\frac{F(\mathbf{x}_i^n) - F_{\min}^n}{F_{\max}^n - F_{\min}^n} \right)^q \in (0, 1],$$

We find that increasing q in the mass transfer protocol (4.5), improves the success rate of SBRD. Previously, we found $q = 2$ to be an optimal choice for SBGD. However, as shown in Tables 6 and 7 for the Ackley test function, higher $q = 4$ and respectively $q = 8$, has a dramatic effect in improving the success rate of SBRD over SBGD. Randomization favors higher transfer parameter q. Indeed, increasing q enforces smaller amounts of mass transfer in (4.5) so that SBRD becomes more 'egalitarian': both the heavier leading agents and the lighter exploring agents are allowed more time (iterations) to settle or to explore, and hence the rate of change for mass configuration of the swarm become smaller. In particular, this allows a more effective exploration of the random-based descent, improving the overall performance of SBRD. This is demonstrated in Table 8.

Table 8 Success rates of SBRD vs. SBGD for global optimization of the d-dimensional objective functions with mass transfer parameter $q = 4$ and $q = 8$. this is to be compared with the corresponding success rate using mass transfer parameter $q = 2$, reported in the Tables 1, 2, 3 and 4

	$N = 10$		$N = 25$		$N = 50$		$N = 100$	
d	$q = 8$	$q = 4$	$q = 8$	$q = 4$	$q = 8$	$q = 4$	$q = 8$	$q = 4$
Ackley								
14	**4.2%**	3.8%	**66.9%**	60.0%	100.0%	100.0%	100.0%	100.0%
16	0.1%	0.3%	38.4%	38.3%	**99.8%**	95.0%	100.0%	100.0%
18	0.0%	0.0%	14.6%	16.3%	**87.3%**	79.7%	100.0%	99.6%
20	0.0%	0.0%	1.0%	1.4%	**30.7%**	25.1%	**84.7%**	74.5%
Rastrigin								
2	**42.5%**	37.4%	99.0%	98.8%	100.0%	100.0%	100.0%	100.0%
3	6.2%	6.5%	**32.4%**	29.0%	**80.1%**	74.3%	**99.1%**	98.0%
4	0.8%	1.0%	4.7%	4.9%	**14.3%**	11.5%	**35.2%**	30.3%
5	0.2%	0.1%	1.1%	0.9%	**3.0%**	1.7%	**4.8%**	3.7%
Rosenbrock								
3	2.6%	2.8%	**12.0%**	9.2%	**53.7%**	45.4%	**94.0%**	92.0%
4	2.2%	2.4%	**6.6%**	5.5%	**24.0%**	16.7%	**63.7%**	60.9%
5	0.9%	1.1%	2.4%	1.9%	**11.8%**	7.0%	**37.4%**	28.5%
6	0.5%	0.5%	1.3%	1.1%	**6.1%**	4.3%	**18.1%**	13.5%
Styblinski								
6	9.3%	9.3%	**42.4%**	38.2%	73.1%	72.9%	96.0%	95.8%
8	2.6%	2.9%	**12.8%**	10.3%	**31.2%**	30.4%	**60.3%**	58.4%
10	0.8%	0.5%	4.2%	3.5%	**11.4%**	10.4%	**23.5%**	20.8%
12	0.2%	0.1%	1.1%	1.2%	3.6%	3.3%	7.4%	8.1%

Funding Research was supported in part by ONR grant N00014-2112773.

Declarations

Competing Interests The authors declare no competing interests.

References

1. Armijo, L.: Minimization of functions having Lipschitz continuous first partial derivatives. Pac. J. Math. **16**(1), 1–3 (1966)
2. Boyd, S.P., Vandenberghe, L.: Convex Optimization. Cambridge University Press, Cambridge (2004)
3. Carrillo, J.A., Choi, Y.-P., Totzeck, C., Tse, O.: An analytical framework for consensus-based global optimization method. Math. Models Methods Appl. Sci. **28**(06), 1037–1066 (2018)
4. Carrillo, J.A., Jin, S., Li, L., Zhu, Y.: A consensus-based global optimization method for high dimensional machine learning problems. ESAIM Control Optim. Calc. Var. **27**, S5 (2021)
5. Grassi, S., Huang, H., Pareschi, L., Qiu, J.: Mean-field particle swarm optimization. In: Weizhu Bao, B.P., Markowich, P.A., Tadmor, E. (eds.) Modeling and Simulation for Collective Dynamics, pp. 127–194. World Scientific, Singapore (2023)
6. Ha, S.-Y., Jin, S., Kim, D.: Convergence of a first-order consensus-based global optimization algorithm. Math. Models Methods Appl. Sci. **30**(12), 2417–2444 (2020)
7. Ha, S.-Y., Jin, S., Kim, D.: Convergence and error estimates for time-discrete consensus-based optimization algorithms. Numer. Math. **147**(2), 255–282 (2021)

8. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of ICNN'95-International Conference on Neural Networks, vol. 4, pp. 1942–1948. IEEE, Los Alamitos (1995)
9. Kingma, D.P., Ba Adam, J.: A method for stochastic optimization (2017). ArXiv preprint arXiv:1412.6980
10. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. Science **220**(4598), 671–680 (1983)
11. Liu, H., Tian, X.: An adaptive gradient method with energy and momentum. Ann. Appl. Math. **38**(2), 183–222 (2022)
12. Łojasiewicz, S.: Ensembles Semi-Analytiques. IHES Notes (1965)
13. Łojasiewicz, S.: Sur la géométrie semi-et sous-analytique. In: Annales de l'institut Fourier, vol. 43, pp. 1575–1595 (1993)
14. Lu, J., Tadmor, E., Zenginoglu, A.: Swarm-based gradient descent method for non-convex optimization (2022). ArXiv preprint arXiv:2211.17157
15. Momin, J., Yang, X.-S.: A literature survey of benchmark functions for global optimisation problems. Int. J. Math. Model. Numer. Optim. **4**(2), 150–194 (2013)
16. Nocedal, J., Wright, S.J.: Numerical Optimization. Springer, Berlin (1999)
17. Nocedal, J., Wright, S.J.: Conjugate Gradient Methods. Springer, Berlin (2006)
18. Palais, R.S., Smale, S.: A generalized Morse theory. Bull. Am. Math. Soc. **70**, 165–172 (1964)
19. Pinnau, R., Totzeck, C., Tse, O., Martin, S.: A consensus-based model for global optimization and its mean-field limit. Math. Models Methods Appl. Sci. **27**(01), 183–204 (2017)
20. Reynolds, C.W.: Flocks, herds and schools: a distributed behavioral model. In: Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques, pp. 25–34 (1987)
21. Tadmor, E.: On the mathematics of swarming: emergent behavior in alignment dynamics. Not. Am. Math. Soc. **68**(4), 493–503 (2021)
22. Totzeck, C.: Trends in consensus-based optimization. In: Bellomo, N., Carrillo, J.A., Tadmor, E. (eds.) Active Particles, vol. 3, pp. 201–226. Springer, Berlin (2022)
23. Van Laarhoven, P.J., Aarts, E.H.: Simulated annealing. In: Simulated Annealing: Theory and Applications, pp. 7–15. Springer, Berlin (1987)
24. Wolfe, P.: Convergence conditions for ascent methods. SIAM Rev. **11**(2), 226–235 (1969)