ESAIM: COCV 27 (2021) S5
https://doi.org/10.1051/cocv/2020046

**ESAIM: Control, Optimisation and Calculus of Variations**
www.esaim-cocv.org

# A CONSENSUS-BASED GLOBAL OPTIMIZATION METHOD FOR HIGH DIMENSIONAL MACHINE LEARNING PROBLEMS

José A. Carrillo[1], Shi Jin[2,*], Lei Li[2] and Yuhua Zhu[3]

**Abstract.** We improve recently introduced consensus-based optimization method, proposed in [R. Pinnau, C. Totzeck, O. Tse, S. Martin, *Math. Models Methods Appl. Sci.* **27** (2017) 183–204], which is a gradient-free optimization method for general non-convex functions. We first replace the isotropic geometric Brownian motion by the component-wise one, thus removing the dimensionality dependence of the drift rate, making the method more competitive for high dimensional optimization problems. Secondly, we utilize the random mini-batch ideas to reduce the computational cost of calculating the weighted average which the individual particles tend to relax toward. For its mean-field limit – a nonlinear Fokker-Planck equation – we prove, in both time continuous and semi-discrete settings, that the convergence of the method, which is exponential in time, is guaranteed with parameter constraints *independent* of the dimensionality. We also conduct numerical tests to high dimensional problems to check the success rate of the method.

**Mathematics Subject Classification.** 60H35, 70F10.

Received October 12, 2019. Accepted July 15, 2020.

## 1. Introduction

Our main goal in this work is developing gradient-free optimization methods to the following classical unconstrained optimization problem

$$x^* = \operatorname*{argmin}_{x\in\mathbb{R}^d} L(x)\,, \tag{1.1}$$

in *high dimensions*, where the target function, not necessarily convex, is assumed to be a continuous function defined on $\mathbb{R}^d$ achieving a unique global minimum. Target functions defined on subsets of $\mathbb{R}^d$ can be extended to the whole space recasting the corresponding optimization problem in the form (1.1). Moreover, we can assume without loss of generality that the target function is positive, *i.e.*, $L(x^*) > 0$, by lifting $L$ by a suitable constant.

*Keywords and phrases:* Global optimization, high dimensional optimization, consensus-based optimization, random batch method.

[1] Mathematical Institute, University of Oxford, Oxford OX2 6GG, UK.
[2] School of Mathematical Sciences, Institute of Natural Sciences, MOE-LSC, Shanghai Jiao Tong University, Shanghai 200240, PR China.
[3] Department of Mathematics, Stanford University, California 94305, USA.

* Corresponding author: shijin-m@sjtu.edu.cn

Important examples of target functions stem from machine learning and artificial intelligence applications. Typical neural training networks lead to optimization for functions of the following form

$$L(x) = \frac{1}{n}\sum_{i=1}^{n} \ell(f(x,\hat{x}_i),\hat{y}_i) =: \frac{1}{n}\sum_{i=1}^{n} \ell_i(x), \tag{1.2}$$

where $x$ is the set of parameters defining the model, $(\hat{x}_i,\hat{y}_i)_{i=1}^n$ constitute the training data set, the function $f(x,\hat{x})$ defines the neural network that one wants to learn, and the function $\ell(f,y)$ is the loss function measuring the distance between the prediction $f(x,\hat{x}_i)$ and the observations $\hat{y}_i$.

For such optimization problems, gradient-based methods have been dominating. Nevertheless, in general, most gradient-based methods have problems dealing with functions that have large noise or non-differentiable functions. They are also not designed to handle multi-modal problems or discrete and mixed discrete-continuous design variables [2]. More specifically in machine learning problems, it has been proved that as the deep neural network gets deeper, the gradient tends to explode or vanish [5, 21]. Besides, it will be easily influenced by the geometry of the landscape [38].

On the other hand, there are also gradient-free methods such as Nelder-Mead (NM) method [41], genetic algorithm (GA) [23], simulated annealing (SA) [34], particle swarm optimization (PSO) [18, 32], etc.. The NM method is a direct search method based on function comparison; GA is inspired by genetic evolution and are commonly used to generate high-quality solutions to optimization; SA is a probabilistic technique for approximating the global optimum, which is often used in discrete search space; PSO is used to model the flocking behavior of birds, and also found to be a good optimization method.

One of such gradient-free methods is the concensus-based optimization (CBO) method, established in [11, 42, 45]. This is a method based on an interacting particle system, along the line of consensus based models [3, 6, 12, 13, 15, 20, 35, 40, 44]. This particle system consists of $N$-particles, labeled as $X^j, j = 1,\dots,N$, that tend to relax toward their weighted average, and in the meantime also undergo fluctuation with a multiplicative noise:

$$\mathrm{d}X^j = -\lambda(X^j - \bar{x}^*)H^\epsilon(L(X^j) - L(\bar{x}^*))\,\mathrm{d}t + \sigma|X^j - \bar{x}^*|\mathrm{d}W^j, \tag{1.3}$$

where $\bar{x}^*$ is the weighted average of the positions of the particles according to

$$\bar{x}^* = \frac{1}{\sum_{j=1}^N e^{-\beta L(X^j)}}\sum_{j=1}^{N} X^j e^{-\beta L(X^j)}. \tag{1.4}$$

The function $H^\epsilon$ is a regularization of the Heaviside function introduced by the authors with the objective that the particles will drift only if at their positions the cost value is higher than the average of all particles. Later the authors in [11] considered this model without the Heaviside cutoff for the convenience of analysis. The diffusion given in (1.3) is associated with $|X^j - \bar{x}^*|$ and this yields convergence conditions and methods depending on the dimension $d$, see ([11], Thm. 4.1). The motivation for this type of diffusion is that one wants to explore the landscape of the cost function $L(x)$ if one is far from consensus, but when consensus forms one wants the noise to decrease, and eventually to disappear, in order to stabilize the results towards the target $x^*$. This decrease of the temperature is a common feature with simulated annealing [24, 26, 37].

Here, $e^{-\beta L(x)}$ is the Gibbs distribution corresponding to $L(x)$. The motivation for this choice comes from statistical mechanics: the cost function $L(x)$ corresponds to a potential in which particles move by steepest descent modulated by Brownian noise with $\beta$ being the inverse of the temperature leading to this invariant measure. In this way, the smaller the value of the temperature is, the larger the weight of the normalized Gibbs measure for the agents is on the minimum value of the cost function $L(x)$. The quantitative formulation of this intuition is given by the Laplace principle [4, 17, 39], a classical asymptotic method for integrals, recalled here

for reader's sake: for any probability measure $\rho \in \mathcal{P}(\mathbb{R}^d)$ compactly supported with $x_* \in \mathrm{supp}(\rho)$, then

$$\lim_{\beta\to\infty}\left(-\frac{1}{\beta}\log\left(\int_{\mathbb{R}^d} e^{-\beta L(x)}\mathrm{d}\rho(x)\right)\right) = L(x^*) > 0. \tag{1.5}$$

Therefore, if $L$ attains its minimum at a single point $x^* \in \mathrm{supp}(\rho)$, then the suitably normalized measure $e^{-\beta L(x)}\rho$ assigns most of its mass to a small region around $x^*$ and hence we expect it approximates a Dirac distribution $\delta_{\bar{x}^*}$ for large $\beta \gg 1$. Consequently, the first moment of the normalized measure $e^{-\beta L(x)}\rho$, and thus, the discrete counterpart average $\bar{x}^*$, should provide a good estimate of the point at which the global minimum is attained, $x^* = \operatorname{argmin} L$. Furthermore, the convergence rate toward the global minimum is *exponential* in time. However, to guarantee the convergence of this method, *the drift rate $\lambda$ depends on the dimension parameter $d$*, which makes the particles move away from its global equilibrium more easily for high dimension problems in which $\mathrm{d} \gg 1$, such as those raising from machine or deep learning problems.

In this paper, we improve the above CBO algorithms in two ways. First, we replace the isotropic Brownian motion term $\sigma|X^j - \bar{x}^*|\mathrm{d}W^j$, which is added equally in all dimensions, by its component-wise counterpart. For its mean-field limit equation, in both time continuous and a time semi-discrete settings, we prove that this removes the $d$-dependence constraint on $\lambda$. Secondly, we utilize the random mini-batch ideas, an essential ingredient in stochastic gradient descent (SGD) method [8, 9, 43], and also introduced recently in [31] for interacting particle systems, that reduces the computational cost in calculating $\bar{x}^*$ from $O(N)$ to $O(1)$, or $O(nN)$ to $O(1)$ in the case of (1.2), and thus it reduces the overall computation cost of CBO. The noise introduced by the random selection of the mini-batches also adds extra stochasticity which makes the particles more likely to escape the local equilibrium, thus enhances the success rate of the algorithm toward the global minimum, even by one order of magnitude in some cases as shown in Section 4.2.

We point out that more recently, in [19] the consensus of particles, and convergence toward the global minimum under dimension-independent conditions on parameters and initial data, are established for the particle system (2.2), without using the mean-field limit.

The paper is organized as follows. We present our algorithm and its continuous model in Section 2. We prove in Section 3 that in the continuous and a semi-discrete in time settings the mean-field limit of the algorithm will converge to the global minimum exponentially fast, with a constraint on $\lambda$ that is independent of $d$. We give several numerical experiments in Section 4 to verify the performance and efficiency of the algorithm.

## 2. A new consensus based optimization method

Our first new observation over the CBO model (1.3) is that if one uses the *component-wise geometric Brownian motion*, which we shall clarify soon, the dimension dependence in the convergence estimates ([11], Thm. 4.1) can be dramatically reduced. To illustrate these ideas, let us fix $\bar{x}^* = a$ to be a constant vector and consider solely the effect of the diffusion term. Let us consider a shifted second moment for (1.3) in the case of $H \equiv 1$ to obtain

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbb{E}|X-a|^2 = -2\lambda\mathbb{E}|X-a|^2 + \sigma^2\sum_{i=1}^{d}\mathbb{E}|X-a|^2 = (-2\lambda + \sigma^2 d)\mathbb{E}|X-a|^2.$$

Clearly, if the particles are to form consensus, one needs $2\lambda > \mathrm{d}\sigma^2$. Now consider the SDE with component-wise geometric Brownian motion

$$\mathrm{d}X = -\lambda(X-a)\,\mathrm{d}t + \sigma\sum_{k=1}^{d}(X-a)_k\mathrm{d}W_k\vec{e}_k, \tag{2.1}$$

where $(X-a)_k$ means the $k$th component of $X-a$, $\{W_k\}_{k=1}^d$ are independent standard Brownian motions, and $\vec{e}_k$ is the unit vector along the $k$th dimension. For the interacting particle system (2.1), one easily finds that

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbb{E}|X-a|^2 = -2\lambda\mathbb{E}|X-a|^2 + \sigma^2\sum_{i=1}^{d}\mathbb{E}(X-a)_i^2 = (-2\lambda+\sigma^2)\mathbb{E}|X-a|^2.$$

We only need $2\lambda > \sigma^2$ for the particles to concentrate. The restriction between $\lambda$ and $\sigma$ is *dimension $d$ insensitive* for the particles to concentrate (or form a consensus as the terminology in [42]).

Based on this observation, we now propose a modification to the CBO model in (1.3) together with an efficient algorithm for its computation by the random batch approach championed in [31]. We tweak the CBO method introduced in [11, 42] by considering the following model with diffusion corresponding to a *component-wise* geometric Brownian motion

$$\begin{aligned} \mathrm{d}X^j &= -\lambda(X^j-\bar{x}^*)\,\mathrm{d}t + \sigma\sum_{k=1}^{d}(X^j-\bar{x}^*)_k\mathrm{d}W_k^j\vec{e}_k\,, \\ \bar{x}^* &= \frac{1}{\sum_{j=1}^N e^{-\beta L(X^j)}}\sum_{j=1}^{N}X^j e^{-\beta L(X^j)}. \end{aligned} \tag{2.2}$$

Here $\bar{x}^*$ is the same as (1.4).

Compared to (1.3), this new model has a simpler and cleaner form, where we omit the Heaviside function as in [11] and use the component-wise geometric Brownian motion to replace their noise. From the viewpoint of opinion models in social sciences, the interacting particle system (2.2) may be thought as the society drifting according to some common sense opinion or command stemming from the individual parties. Let us remark that $\bar{x}^*$ can be chosen to be updated only at some discrete time points in practice without changing the spirit of the modeling.

As already commented, the usage of the diffusion according to *component-wise* geometric Brownian motion is largely due to its scalability in high dimension space. In fact, the assumptions and the results later in Theorem 3.2 for the particles to converge are all *independent of the dimensionality $d$* of the particle $X$. The model (1.3) on one hand requires $\sigma^2$ to be small or large $\lambda$ which reduces the exploration ability at the initial stage. Moreover, it will also put more severe constraint on the parameters, especially on the second moments of the initial condition of $X_0$. Besides, the use of the diffusivity in (2.2) allows the particles to explore each dimension with different rate, and possibly easier to find the optimizer. To summarize, we expect the optimization method (2.2) to be efficient for optimization problems where the dimensionality of the parameter is very high, such as those in deep learning compare to (1.3).

The computational cost of a straightforward numerical scheme to approximate the new continuous CBO method (2.2) is too high if we do (1.4) in every time step, especially for big data problems in the form of (1.2). Hence, our second novel approach is to apply the random batch method [31] to the interacting particle system (2.2), which leads to efficient random algorithms. See also [8, 9, 43] for the relevant mini-batch ideas used for SGD. These random algorithms can also be viewed as new models, which seem to be closer to opinion models in social sciences.

The random mini-batch strategy developed in [31] will be extended to two levels for the typical cost functions arising in machine learning such as (1.2). First, we calculate the empirical expectation $\hat{L}^j = \hat{L}(X^j)$ from a random subset of the training data set instead of the accurate $L^j$ as the objective value for the ensemble of particles $X^j$; second, we apply the mini-batch approach and update the reference $\bar{x}^*$ by a random subset of the particle ensemble instead of all particles. These two modifications allow us to do high dimensional optimization more efficiently. The mini-batch is done without replacement, that is, we do a random permutation and then select the mini-batch in order. Let us finally remark that this random choice of subsets of interacting particles

is very similar to Monte Carlo approaches to compute large averages, and it has been used to produce efficient algorithms for mean-field (kinetic) swarming models in [1, 14].

We introduce below the random algorithm to solve in practice our new CBO model (2.2) and (1.4).

**Algorithm 2.1.** Generate $\{X_0^j \in \mathbb{R}^d\}_{j=1}^N$ according to the same distribution $\rho_0$. Set the remainder set $\mathcal{R}_0$ to be empty. For $k = 0, 1, 2, \ldots,$ do the following:

**Step 1.** Concatenate $\mathcal{R}_k$ and a random permutation $\mathcal{P}_k$ of the indices $\{1, 2, \cdots, N\}$ to form a list $\mathcal{I}_k = [\mathcal{R}_k, \mathcal{P}_k]$. Pick $q = \lfloor \frac{N+|\mathcal{R}_k|}{M} \rfloor$ sets of size $M \ll N$ from the list $\mathcal{I}_k$ in order to get batches $B_1^k, B_2^k, \ldots, B_q^k$ and set the remaining indices to be $\mathcal{R}_{k+1}$. Here, $|\mathcal{R}_k|$ means the number of elements in $\mathcal{R}_k$.

**Step 2.** For each $B_\theta^k$ $(\theta = 1, \ldots, q)$, do the following

(1) Calculate the function values (or approximated values) of $L$ at the location of the particles in $B_\theta^k$ by $L^j := L(X^j)$, $\forall j \in B_\theta^k$. If $L(x)$ is in the form (1.2) with $n \gg 1$, one then applies the random mini-batch idea again: generate a random index subset $A_\theta^k \subset \{1, \ldots, n\}$ with $|A_\theta^k| = m$, and approximate $L^j$ for all $j \in B_\theta^k$ by

$$\hat{L}^j := \hat{L}_\theta^k(X^j) = \frac{1}{m} \sum_{i \in A_\theta^k} \ell_i(X^j), \ \forall j \in B_\theta^k,$$

where $\hat{L}_\theta^k(x) := \frac{1}{m} \sum_{i \in A_\theta^k} \ell_i(x)$ is an unbiased approximation to $L(x) = \frac{1}{n} \sum_{i=1}^n \ell_i(x)$ defined in (1.2).

(2) Update $\bar{x}_{k,\theta}^*$ according to the following weighted average,

$$\bar{x}_{k,\theta}^* = \frac{1}{\sum_{j \in B_\theta^k} \mu_j} \sum_{j \in B_\theta^k} X^j \mu_j, \quad \text{with} \quad \mu_j = e^{-\beta L^j} \text{ or } e^{-\beta \hat{L}^j}. \tag{2.3}$$

(3) Update $X^j$ for $j \in \mathcal{J}_{k,\theta}$ as follows,

$$X^j \leftarrow X^j - \lambda \gamma_{k,\theta}(X^j - \bar{x}_{k,\theta}^*) + \sigma_{k,\theta} \sqrt{\gamma_{k,\theta}} \sum_{i=1}^d \vec{e}_i \left(X^j - \bar{x}_{k,\theta}^*\right)_i z_i^j, \quad z_i^j \sim \mathcal{N}(0,1), \tag{2.4}$$

where $\gamma_{k,\theta}$ is the learning rate chosen suitably and there are two options for $\mathcal{J}_{k,\theta}$:

$$\begin{aligned} &\textit{partial updates:} \quad \mathcal{J}_{k,\theta} = B_\theta^k, \\ &\textit{full updates:} \quad \mathcal{J}_{k,\theta} = \{1, \ldots, N\}. \end{aligned}$$

**Step 3.** Check the **Stopping criterion:**

$$\frac{1}{\mathrm{d}} \|\Delta \bar{x}^*\|_2^2 \leq \epsilon,$$

where $\|\cdot\|_2$ is the Euclidean norm and $\Delta x^*$ is the difference between two most recent $\bar{x}_{k,\theta}^*$. If this is not satisfied, repeat Steps 1–2.

Note again that $\left(X^j - \bar{x}_{k,\theta}^*\right)_i$ represents for the $i$th component of the vector in (2.4). $\lambda$ is the drift rate, $\gamma_{k,\theta}$ is the learning rate, $\sigma_{k,\theta}$ is the noise rate and $z_i$ is a random variable following the standard normal distribution. Note that we add $\sqrt{\gamma_{k,\theta}}$ on purpose to be consistent with the time-continuous model (2.2). These parameters can be different from step to step in practice, as often used in machine learning and optimization. In practice,

one often chooses $\gamma_{k,\theta}$ in a decreasing fashion satisfying $\sum_k \gamma_{k,\theta} = \infty$. In our experiments, we fix $\gamma_{k,\theta} \equiv \gamma$ to be constant. In general, it needs to be chosen to satisfy a numerical stability condition. One stability result is given in [19]. Besides, decreasing $\sigma_{k,\theta}$ slowly corresponds to the famous simulated annealing algorithm in optimization [24, 26, 37].

**Remark 2.2.** The estimated value $\hat{L}$ of the objective function is especially efficient for problems of the form (1.2). Usually, to train a good model, one requires a large number of data, that is, $n \gg 1$. The computational cost would be high if one calculate $L(x)$ at each step for all particles. If we calculate $\hat{L}$ based on a small subset of the data, the computational cost will be largely saved. Besides, we will show later in the numerical experiments that using $\hat{L}$ can not only save computational cost, but also make the algorithm converge to the optimizer faster, due to stochasticity introduced by randomly selecting the mini-batches. This is an established concept for algorithms such as the SGD [8, 9].

**Remark 2.3.** An alternative way to update $\bar{x}^*$ is to let it equal to $\operatorname{argmin} \hat{L}_j$, that is,

$$\bar{x}_k^* = \operatorname*{argmin}_{X^j \in B_\theta^k} \hat{L}(X^j). \tag{2.5}$$

We will show that it numerically performs as good as the penalized average. We will leave the theoretical proof of this case for future study.

**Remark 2.4.** For updating $X^j$:

- There are two ways to introduce extra noises into the algorithm. One way is to let particles do geometric Brownian motion as in (2.4), another way is let particles do a Brownian motion only when $X^j$ stops moving forward. The reason why the second method also works is because we already introduced noise by using the estimated $\hat{L}^j$ and (2.3) using the randomly generated sets $B_k$, so the noise term in (2.4) is sometimes not necessary. We will show later in the numerical experiments that if we do not have the last term in (2.4) and just add a Brownian motion when $X^j$ stops moving forward, the performance is still good.
- In some optimization problem, since the landscape of the objective function is too complicated (for example, the MNIST data in Section 4.3), it cannot converge to the global minimizer at stoppting time. Therefore, when $\bar{x}^*$ stops updating, we record $\hat{L}(\bar{x}^*)$ at that step, and make all particles do an independent Brownian motion, *i.e.* for $\forall j$,

$$X^j \leftarrow X^j + \sigma_{k,\theta}\sqrt{\gamma} \sum_{i=1}^{d} \vec{e}_i z_i^j, \quad z_i^j \sim \mathcal{N}(0,1),$$

  then repeat the algorithm. We terminate the procedure if the recorded $\hat{L}(\bar{x}^*)$ is not decreasing any more.

## 3. Analysis of the mean-field limit models

The analysis of the computational model in Section 2 is quite challenging: analyzing the $N$-particle system, showing the existence of singular invariant measures quantifying the convergence towards them would be a fantastic breakthrough. In this section, we will consider the formal mean-field limit models ($N \to \infty$) of the interacting particle system (2.2), which makes the analysis possible even if working with high dimensional PDEs, as already shown in [11]. We remark that the rigorous proof of the mean-field limit is another open problem for these interacting particle systems due to the difficulty of managing the multiplicative noise term in (2.2). Depending on how we treat the time variable, one can write a time-continuous model and a semi-continuous mean-field models, as discussed below.

### 3.1. Time continuous model

Formally, taking $N \to \infty$ in the model (2.2) with full batch (or alternatively, $\gamma \to 0$ and $N \to \infty$ in Algorithm 2.1 with full batch), the mean field limit of the model is formally given by the following stochastic differential equation for $X = X(t)$:

$$\mathrm{d}X = -\lambda(X - \bar{x}^*)\mathrm{d}t + \sigma \sum_{i=1}^{d} \vec{e}_i (X - \bar{x}^*)_i \mathrm{d}W_i, \tag{3.1}$$

where

$$\bar{x}^* = \frac{\mathbb{E}(Xe^{-\beta L(X)})}{\mathbb{E}(e^{-\beta L(X)})}. \tag{3.2}$$

We refer to [13] and [7, 10, 22, 28, 29] for formal and rigorous discussions respectively of the results about mean-field models. The law $\rho(\cdot, t)$ of the process $X(t)$ in the high dimensional space $\mathbb{R}^d$ solving the nonlinear stochastic differential equation (3.1)-(3.2) follows the nonlinear Fokker-Planck equation

$$\partial_t \rho = \lambda \nabla \cdot ((x - \bar{x}^*)\rho) + \frac{1}{2}\sigma^2 \sum_{i=1}^{d} \partial_{ii}((x - \bar{x}^*)_i^2 \rho),$$

where

$$\bar{x}^* = \frac{\int_{\mathbb{R}^d} x e^{-\beta L(x)} \rho(x,t)\,\mathrm{d}x}{\int_{\mathbb{R}^d} e^{-\beta L(x)} \rho(x,t)\,\mathrm{d}x}.$$

We will prove that the stochastic process $X(t)$ will approach some point $\tilde{x}$, which is an approximation of $\operatorname{argmin} L(x)$. Our proof needs the following assumptions:

**Assumption 3.1.** We assume that $L_m := \inf L > 0$, without loss of generality, and that the cost function $L$ satisfies $c_L := \max(\|\max_i |\partial_{ii} L|\|_\infty, \|r(\nabla^2 L)\|_\infty) < \infty$.

Here, $\nabla^2 L$ represents the Hessian of $L$, and $r(\nabla^2 L)$ is the spectral radius while $\partial_{ii} L$ is the diagonal element of $\nabla^2 L$. Our assumption means that these two quantities should be of the same order. One sufficient condition is that all second derivatives of $L$ are bounded. Let us also define the following averaged quantities:

$$V(t) := \mathbb{E}|X - \mathbb{E}X|^2 \qquad \text{and} \qquad M_L(t) := \mathbb{E}e^{-\beta L(X)}. \tag{3.3}$$

Here, $V$ is the variance of the process $X$, while $M_L(t)$ is the total weight in the optimization. The following result gives the convergence of the continuous mean field model (3.1), whose proof is deferred to Appendix A following the blueprint in [11].

**Theorem 3.2.** *If $\beta, \lambda, \sigma$ and the initial distribution are chosen such that*

$$\begin{aligned} \mu &:= 2\lambda - \sigma^2 - 2\sigma^2 \frac{e^{-\beta L_m}}{M_L(0)} > 0, \\ \nu &:= \frac{2V(0)}{\mu M_L^2(0)} \beta e^{-2\beta L_m} c_L (2\lambda + \sigma^2) \le \frac{3}{4}, \end{aligned} \tag{3.4}$$

*then* $V(t) \to 0$ *exponentially fast and there exists* $\tilde{x}$ *such that* $\bar{x}^*(t) \to \tilde{x}$, $\mathbb{E}X \to \tilde{x}$ *exponentially fast. Moreover, it holds that*

$$\begin{aligned} L(\tilde{x}) &\leq -\frac{1}{\beta}\log M_L(0) - \frac{1}{2\beta}\log(1-\nu) \\ &\leq L_m + r(\beta) + \frac{\log 2}{\beta}, \end{aligned}$$

*where*

$$r(\beta) := -\frac{1}{\beta}\log M_L(0) - L_m \to 0, \ \beta \to \infty.$$

In the above result, we have used (3.4) and the Laplace principle (1.5) so that $r(\beta) \to 0$, $\beta \to \infty$. How well $L(\tilde{x})$ approximates $L_m$ depends on how well $-\frac{1}{\beta}\log M_L(0)$ approximates $L_m$. When $\beta$ is sufficiently large and $L$ has a unique global minimizer, together with some reasonable assumptions put on $L$ and the probability density, one acually has [25, 27, 36]

$$r(\beta) \leq \frac{\mathrm{d}}{2}\frac{\log(\beta/(2\pi))}{\beta} + \frac{C_L}{\beta}, \quad \beta \gg 1, \tag{3.5}$$

where $C_L$ does not depend on $d$. The convergence rate in Theorem 3.2 does not depend on $d$ but in the error term (3.5), there is linear $d$ dependence. For moderately high dimensional case, for example, in Section 4.2 when $d = 20$ and $\beta = O(10^2)$, one gets reasonably small error estimate. In theory, of course for any $d$, one can choose $\beta$ sufficiently large so that the error is still small. However, when $\beta$ is too large, $\exp(-\beta L)$ is near zero, and one loses lots of significant digits. On the other hand, when $\beta \gg d \gg 1$, the average (1.4) or (2.3) becomes

$$\bar{x}^* = \operatorname*{argmin}_{X^j} L(X^j),$$

which works similarly well as commented in Section 4.3, though one cannot prove a theorem. In summary, our method can be feasible in high dimensional problems.

Besides the largeness of $\beta$, how big $r(\beta)$ is also depends on the initial support of the law of $X_0$. If the probability that $X_0$ is near the minimizer is small, the approximation quality is poor. This means for the algorithm to work well, the particles should explore the surrounding area well so that there is some probability that the neighborhood of the minimizer can be visited.

The assumptions (3.4) basically require $\lambda, \beta$ to be large or $\sigma, V(0)$ to be small enough. Notice that the set of the parameters is not empty as one can control the initial variance $V(0)$. The assumption is restrictive in the above theorem, however this has to be understood as a proof of concept where other possible approaches may lead to improvements. In the first equation of (3.4), $M_L(0)$ is also some quantity of the order $e^{-\beta L_m}$, so it essentially means $\lambda \gtrsim \sigma^2$, which ensures that the variance of $X$ decays to zero. In fact, in the geometric Brownian motion, one needs $2\lambda > \sigma^2$ to guarantee the variance of $X$ to vanish as shown in Section 2. Under such assumption, all particles will converge with exponential rate to a point which is within $O(1/\beta)$ of the global minimum. Moreover, recall $e^{-2\beta L_m}/M_L^2(0) \geq 1$. Hence, in the original CBO interacting particle system and their mean-field counterpart in [11], a large $d$ dependence in $\mu$ is sensitive for $\lambda$. This means that our model is more feasible and adapted for high dimensional optimization problems.

### 3.2. The semi-discrete model

Let us consider the CBO method (2.2) where $\bar{x}^*$ is updated at only a number of discrete time points. Alternatively, in Algorithm 2.1, we let $\gamma$ fixed, take $N \to \infty$ and use the time continuous SDE to replace the discrete scheme at one iteration.

Define

$$t_k := k\gamma. \tag{3.6}$$

Then, one has the following semi-discrete model in the time-continuous setting, where a particle evolves according to the component geometric Brownian motion on interval $I_k := [t_{k-1}, t_k)$, and the references $\bar{x}^*$ is only updated on some discrete time points as

$$\mathrm{d}X = -\lambda(X - \bar{x}_k^*)\,\mathrm{d}t + \sigma \sum_{i=1}^{d} (X - \bar{x}_k^*)_i \mathrm{d}W_i \vec{e}_i\,, \quad t \in I_k\,, \tag{3.7}$$

where

$$\bar{x}_k^* = \frac{\int_{\mathbb{R}^d} x \exp(-\beta L(x))\,\rho(x, t_{k-1})\,\mathrm{d}x}{\int_{\mathbb{R}^d} \exp(-\beta L(x))\,\rho(x, t_{k-1})\,\mathrm{d}x}.$$

Similarly, $\rho(\cdot, t_k)$ means the law of $X$ at time $t_k$. We again consider the mean and variance of the model $m(t) = \mathbb{E}X$ and $V(t) = \mathbb{E}|X - \mathbb{E}X|^2$. For this semi-discrete model, we have the following results, whose proof is given in Appendix B:

**Proposition 3.3.** *If the average sequence $\{\bar{x}_k^*\}$ is bounded and*

$$2\lambda > \sigma^2, \tag{3.8}$$

*then the total weight defined in (3.3) is bounded below*

$$M_L^* := \inf_k M_L(t_k) > 0.$$

*Moreover, if step size $\gamma$ also satisfies that*

$$e^{(-2\lambda+\sigma^2)\gamma} + \frac{e^{-\beta L_m}}{M_L^*}(e^{(-2\lambda+\sigma^2)\gamma} - e^{-2\lambda\gamma}) < 1, \tag{3.9}$$

*then $m(t_k) \to \bar{m}$ and $V(t_k) \to 0$. Consequently, $\bar{x}_k^* \to \bar{m}$ and the law of $X$ converges weakly to $\delta(x - \bar{m})$ (i.e. in the dual of $C_b(\mathbb{R}^d)$, the space of bounded continuous functions equipped with the supremum norm).*

**Remark 3.4.** Compared with Theorem 3.2, the choice of the parameters is much less restrictive. We only need (3.8) so that the variance can diminish to zero. However, we have assumed $\{\bar{x}_k^*\}$ to be bounded and condition (3.9).

**Remark 3.5.** In actual numerical experiments, the condition on $\sigma, \lambda$ is much loose than the theoretical condition.

To remove the assumption that $\bar{x}_k^*$ is bounded, one needs to estimate how $|\bar{x}_k^*|$ relies on the initial bounds of $|\bar{x}_0^*|$ so that the estimate can close up. For this discrete case, we have

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\mathbb{E}e^{-\beta L(X)} &= \beta\lambda\mathbb{E}(e^{-\beta L(X)}\nabla L(X)\cdot(X - \bar{x}_k^*)) \\ &\quad + \frac{1}{2}\sigma^2\mathbb{E}\left[e^{-\beta L(X)}\left(\beta^2\sum_i \partial_i L(X)^2(X - \bar{x}_k^*)_i^2 - \beta\sum_i (X - \bar{x}_k^*)_i^2\partial_{ii}L(X)\right)\right]\end{aligned}$$

The issue is that the first term is hard to control now. A possibility to overcome this difficulty is to study the fully discretized scheme, with the noise terms discretized using the Euler-Maruyama scheme. This will be explored elsewhere.

## 4. Numerical performance

We assume $\lambda = 1$ in all the numerical examples in Sections 4.1, 4.2 and 4.3.

Let us first comment on some practical implementation aspects of the Algorithm 2.1. The operator splitting to update all particles can be used in order to avoid overshooting. One can choose to implement the algorithm as

$$\hat{X_k^j} = \bar{x}_k^* + (X_k^j - \bar{x}_k^*)e^{-\lambda\gamma},$$
$$X_{k+1}^j = \hat{X_k^j} + \sigma\sqrt{\gamma}\sum_{i=1}^{d}\vec{e}_i\left(\hat{X_k^j} - \bar{x}_k^*\right)_i z_i^j,$$

where the first equation is the exact solution of the SDE $\mathrm{d}X^j = -\lambda(X^j - x^*)\mathrm{d}t$, from $t = k\gamma$ to $t = k(\gamma+1)$. By overshooting, we mean $\hat{X_k^j}$ oscillates around $\bar{x}_k^*$. This could bring instability as in the case of forward Euler in solving some stiff problems.

An alternative way to implement this step is to freeze $\bar{x}_k^*$ in a time-step interval, then the geometric Brownian motion can be solved exactly by

$$X_{k+1}^j = x_k^* + \sum_{i=1}^{d}\vec{e}_i(X_k^j - \bar{x}_k^*)_i \exp\left(\left(-\lambda - \frac{1}{2}\sigma^2\right)\gamma + \sigma\sqrt{\gamma}\, z_i^j\right) \tag{4.1}$$

In practical simulations, this is comparable with the above splitting approach in most cases.

Concerning the parameters in our CBO model (2.2), one can observe that by increasing $\beta$ and decreasing $\sigma$ as iterations accumulate, the accuracy and convergence speed of the results will be improved. The cooling strategy can be chosen to be similar to the annealing approach [24, 26, 37]. The intuition is that one decreases the temperature so that the system will cool down to the global minimum. Another practical strategy is to use larger $\sigma$ at early stages of the simulations for better exploration of the cost landscape, while use smaller $\sigma$ at later stages. For example, a possible strategy is to take

$$\sigma_k = \sigma_0/\log(k+1)$$

Decreasing $\sigma$ corresponds to decreasing the noise level. As it has been seen in the semi-discrete model (3.7), we need $2\lambda > \sigma^2$ for the particles to concentrate. Hence, this strategy allows us to use large $\sigma > \sqrt{2\lambda}$ in the early stage to explore the surrounding area well.

We now show the performance of Algorithm 2.1 for our CBO model (2.2) in three model cases: an optimization of a test function with large oscillations and wide local minima in one dimension, a neural network for the MNIST data set and an optimization of a test function with many local minima in high dimension.

### 4.1. Comparison with stochastic gradient descent (SGD) method

We first show an example where SGD can hardly find the global minimum, however, our method can easily find it. It has already been observed and proved that the geometry of the objective function will affect the performance of SGD method [16, 30, 33]. One of the reasons is that the expected exiting time for SGD to escape from a local minimum is exponentially proportional to the inverse of Hessian at the minimum, height of the

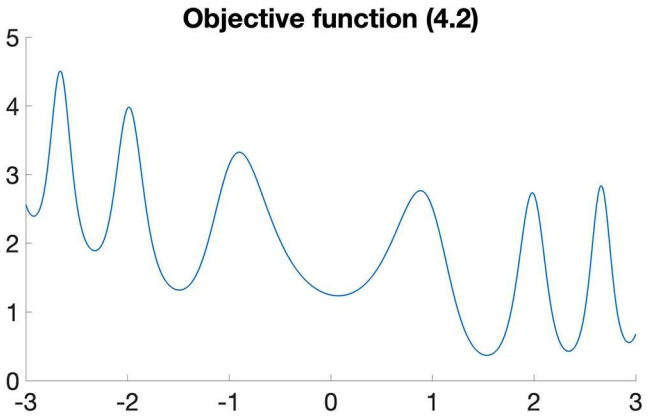


FIGURE 1. Objective function (4.2) with n = $10^4$.

basin and batch size. We construct the following optimization problem:

$$\begin{aligned} \ell(x,\hat{x}_i) &= e^{\sin(2x^2)} + \frac{1}{10}\left(x-\hat{x}_i-\frac{\pi}{2}\right)^2, \quad \hat{x}_i \sim \text{Normal}(0,0.1) \\ L(x) &= \frac{1}{n}\sum_{i=1}^{n}\ell(x,\hat{x}_i) \end{aligned} \tag{4.2}$$

The objective function $L(x)$ is plotted in Figure 1 for $n = 10^4$. It is easy to see that the global minimum is $x^* = \pi/2$.

SGD updates the parameter $x_k$ in the following way,

$$x_{k+1} = x_k - \frac{1}{m}\sum_{i\in b_k}\nabla_x \ell(x_k,\hat{x}_i),$$

where $b_k$ is an index set randomly drawn from $\{1,\dots,n\}$. We can see there are many local minima with different shapes. Especially, the height of all the basins are large, some of the local minimum is much flatter than the global minimum, in which case SGD tends to be trapped in those local minima. However, the geometry of the objective function has little influence on our method. We show the success rate of both methods in Figure 2 with the same initialization and variables $n, m, \gamma$. We consider one simulation is successful if, $|\bar{x}_k^* - x^*| < 0.25$ for our CBO or $|x_k - x^*| < 0.25$ for SGD, which means that our approximated minimizers is in one-half width of the global minimizer. For both methods, we run 100 simulations and each simulation we run $10^4$ steps. In addition, we initialize $X_0^j$ from uniform distribution in $[-3,3]$, and set

$$\gamma = 0.01, \quad n = 10^4, \quad m = 20.$$

Besides, for our method, we set

$$N = 100, \quad M = 20, \quad \sigma = 5, \quad \beta = 30,$$

and use the partial updates. For each simulation, the algorithm stops either when the stopping criterion in Step 3 of Algorithm 2.1 is satisfied with $\epsilon = 10^{-3}$, or it finishes $10^4$ steps.

| | SGD | Algorithm 2.1 |
|---|---|---|
| Success rate | 18% | 98% |

FIGURE 2. The table shows the success rate of SGD and Algorithm 2.1.

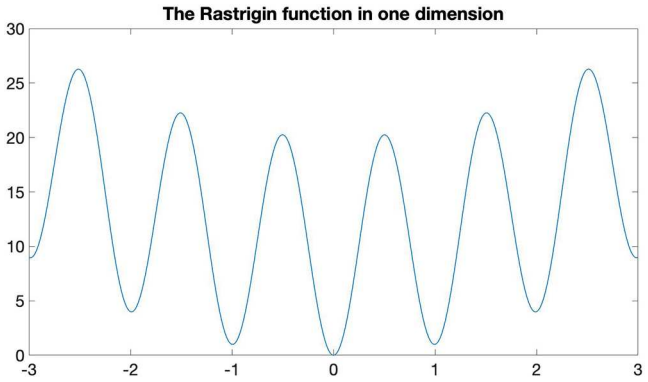


FIGURE 3. The Rastrigin function (4.3) with d = 1, B = C = 0.

From the table we can see that our method performs significantly better than SGD, the computational time for our method is a little longer than SGD though. Notice here $M = 20$, that means, there are only 20 particles interacting with each other in each step, which is also computationally efficient.

## 4.2. The Rastrigin function in $d = 20$

In this section, we compare our method with the one introduced in [42]. The goal is to find the global minimum of the Rastrigin function, which reads

$$L(x) = \frac{1}{\mathrm{d}} \sum_{i=1}^{d} \left[ (x_i - B)^2 - 10 \cos\left(2\pi (x_i - B)\right) + 10 \right] + C, \tag{4.3}$$

with $B = \operatorname{argmin} L(x)$ and $C = \min L(x)$.

Figure 3 shows the shape of the Rastrigin function $L(x)$ when $x \in \mathbb{R}^1, B = C = 0$ in (4.3). This illustration reveals that the local minima of this function are very close to the global minimum, so it is not easy for optimization algorithms to discern the location of the global minimum. The numerical experiments of [42] indicate that their method performs well in finding the global minimum of the *one*-dimensional and *twenty*-dimensional Ackley functions. However, compared to the Ackley function, the local minima of the Rastrigin function are much closer to the global minimum, so it is harder to find its global minimum. The performance of their method for $d = 20$ is not good enough for the same set of parameters, as shown in ([42], Tab. 2). The success rate in their numerical experiments for $N = 50, 100, 200$ is from 34% to 64%.

In our numerical experiments, we set $C = 0$ and $B = 0, 1, 2$. We initialize all the particles uniformly on the interval $[-3, 3]$. For the cases $B = 1, 2$, the minimizer is not at the center of the initialization, which increases the difficulty of converging towards it. Besides we use partial updates in Step 3 and set $\gamma = 0.01$ and $\beta = 30$. For each simulation, we run $10^4$ steps. Our results are shown in Figure 4. We display the success rate and averaged distance to the global minimum for 100 simulations. Notice that in ([42], Tab. 2), $v_f(T)$ corresponds to $\bar{x}_k^*$ in our paper if readers are interested in comparing performances. Here we consider one simulation is successful if

Rastrigin function in $d = 20$ with $\beta = 30$

| | N = 50, M = 40 $\sigma = 5.1$ | N = 100, M = 70 $\sigma = 5.1$ | N = 200, M = 100 $\sigma = 5.1$ |
|---|---|---|---|
| x* = 0, success rate | 97% | 99% | 98% |
| x* = 0, $\frac{1}{d}\mathbb{E}\left[\|x_T^* - x^*\|^2\right]$ | 5.6E-03 | 5.03E-04 | 9.71E-04 |
| x* = 1, success rate | 94% | 99% | 95% |
| x* = 1, $\frac{1}{d}\mathbb{E}\left[\|x_T^* - x^*\|^2\right]$ | 3.9E-03 | 4.95E-04 | 3E-03 |
| x* = 2, success rate | 97% | 100% | 92% |
| x* = 2, $\frac{1}{d}\mathbb{E}\left[\|x_T^* - x^*\|^2\right]$ | 3.0E-03 | 8.06E-06 | 4E-03 |
| Computing time saved | 22.03% | 30.11% | 36.14% |

FIGURE 4. This table shows the success rate and the error of our algorithm towards the global minimum for different Rastrigin functions with parameters leading to the global minimum being given by the constant vectors specified in each row. We also show the computational savings.

the final $\bar{x}_k^*$ is close to the global minimum $x^*$ in the sense that,

$$|(\bar{x}_k^*)_i - (x^*)_i| < 0.25, \quad \text{for all } i,$$

which means our result is in one-half width of the global minimizer and 0.25 is in order to keep consistency with [42]. Figure 4 shows that our method is not only more efficient, but also performs better in terms of finding the global minimizer. Notice that although condition (3.8) is violated, as we mentioned in Remark 3.5, the algorithm approaches the global minimum. Also we notice that the success rate becomes worse when $N = 200$. This is possibly due to random fluctuation, but in general, larger $N$ gives better results but computationally more expensive.

### 4.3. Experiments on MNIST data sets

In this section, we will run an optimization problem from machine learning, in order to show that our method also works for high dimensionality. The MNIST data is a set of pictures with grayscale numbers from 0 to 9. The input data $\hat{x}$ is a vector of dimension 728, which records the Grayscale of each pixel. The output data $\hat{y} \in \{\vec{e}_k\}_{k=1}^{10}$, where $\vec{e}_k$ is a vector of dimension 10 with only the $k$th element 1 and $\vec{e}_k$ represents that it is a picture of number $k-1$. We use the Neural Network without hidden layers to model this classification problem, the function defining the neural network is given by

$$f(x, \hat{x}) = a(ReLu(\theta\hat{x} + B)), \quad x = (\theta, B),$$

is a function depending on the parameter $x$ and mapping $\hat{x} \in \mathbb{R}^{728}$ to $\mathbb{R}^{10}$. Here $\theta \in \mathbb{R}^{10\times 728}$, $B \in \mathbb{R}^{10}$. $ReLu(x) = x\mathbb{1}_{x\geq 0} = ((x_i)_+)_{i\in\{1,\dots,728\}}$ is an activation function with $(r)_+$ being the positive part of the number $r$, while $a(x)$ is another activation function called *softmax*, which reads

$$a(\mathbf{x}) = \frac{e^{x_j}}{\sum_j e^{x_j}}.$$

So that the $j$th component of $f$ represents the probability of $\hat{x}$ being the image associated to number $j-1$. The objective function to be minimized is the cross entropy loss,

$$L(x) = \frac{1}{n}\sum_{i=1}^{n} \ell(f(x,\hat{x}_i),\hat{y}_i), \quad \ell(f,y) = -\sum_{k=1}^{10} \hat{y}_k \log(f_k), \tag{4.4}$$

where the observations belong to the subset $\hat{y} \in \{\vec{e}_k\}_{k=1}^{10}$.

In the setup of deep learning, one uses deep neural network to construct the model. As the neural network gets deeper or wider, the dimension $d$ of parameter $x$ becomes very large, $d \gg n \gg 1$, which results in potentially many local minima. So the goal here is not only finding the global minimum, but also the good global minimum. It is common practice to quantify the quality of the minimum by the test accuracy. We take the largest component as the prediction of our model at $\hat{x}$, that is,

$$g(x^*,\hat{x}) = \vec{e}_j, \quad \text{where } j = \underset{i}{\operatorname{argmax}}\, f(x^*,\hat{x})_i$$

and $f(x^*,\hat{x})_i$ means the $i$th component. We further define the test accuracy by

$$\text{accuracy}_{test}(x^*) = \frac{1}{p}\sum_{i=1}^{p} \mathbb{1}_{\{g(x^*,\hat{x}_i^{\text{test}})=\hat{y}_i^{\text{test}}\}}, \tag{4.5}$$

where $p$ is the size of the test set, number of data in the test set.

Let us first discuss how different elements in the algorithm influence the convergence rate. For all experiments, we use exactly the same initialization, which is drawn from standard normal distribution. Besides, we use the full updates in Step 3 and set the following values as reference case for our simulations,

$$N = 100, \quad M = 10, \quad n = 10^4, \quad m = 50, \quad \gamma = 0.1, \quad \sigma = \sqrt{0.1}, \quad \lambda = 1. \tag{4.6}$$

Here $N$ is the number of total particles, $M$ is the batch size used to update $\bar{x}^*$; $n$ is the number of total training data, $m$ is the batch size used to calculate the estimated objective function $\hat{L}_j$; $\gamma, \sigma, \lambda$ are the learning rate, the noise rate and drift rate respectively. As mentioned in Remark 2.4, we allow all the particles to do an independent Brownian motion with variance $\sigma$ when $\bar{x}^*$ stops updating and then the algorithm repeats until stabilization.

In both Figures 5–7, the x-axis represents for the number of epoch. Here one epoch is equal to $n/m$ steps, which means we go through the whole training data set: 200 steps for $m = 50$ and 1 step for $m = n$. The y-axis represents the test accuracy as defined in (4.5) over $p = 10^4$ data sets. The test data set elements $(x_i^{test}, y_i^{test})_{i=1}^{10000}$ are all different from the training data set elements we used in the objective function (4.4).

In Figure 5 we compare the performance over the neural network of the reference solution with the parameters in (4.6) with respect to other set of parameters. The main general observations inferred are:

- Using estimated value $\hat{L}^j$ is better than using the exact value $L_j$. The small batch $m = 50$ not only saves 99.5% of the computational cost when calculating $L^j$, the increase rate of the accuracy is even better than the full batch.
- Using more particles in the interaction, larger $M$, to compute the average $\bar{x}^*$ increases the overall performance as expected. However, we have to point out that the case where $M = 10$ save 40% of computing time compared to the case $M = 100$.
- Our method with 100 particles and 10 interacting particles at each step already gives good accuracy. We will show later in Figure 7 that as the number of particles goes to $10^3$, the accuracy will be comparable to SGD. However, compared to the performance of $N = 100$ and $N = 500$, the accuracy improved slowly as the number of particles becomes larger.

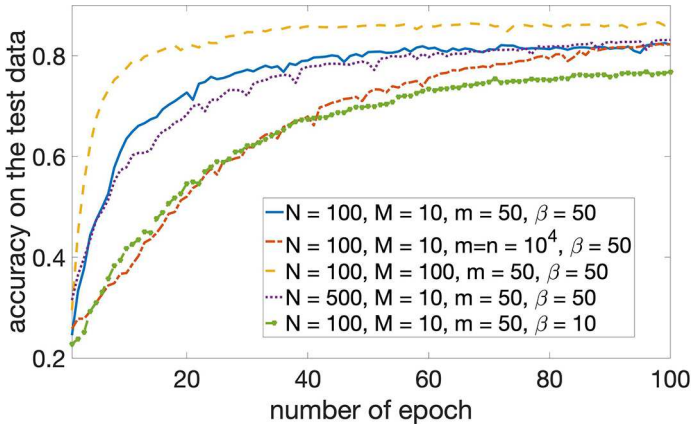


FIGURE 5. Comparison of the performance between the reference solution with parameter in (4.6) and first set of data with different parameters as explained in the inlet.

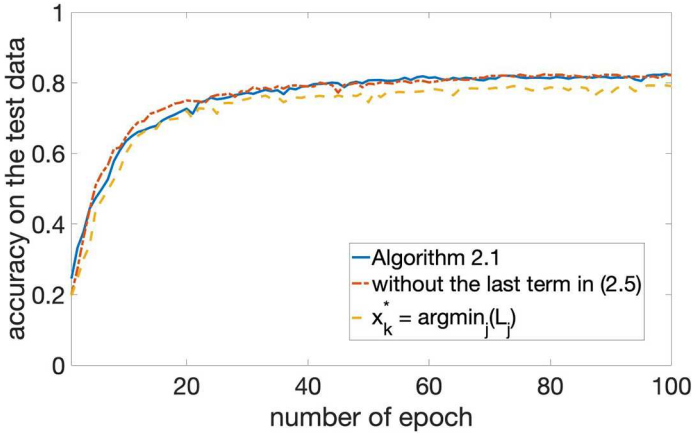


FIGURE 6. Comparison of the performance between our new CBO algorithm and its variants.

– Larger $\beta$ corresponds to faster convergence rate. As $\beta$ becomes larger, it achieves better accuracy faster. However, $\beta$ cannot be too large, otherwise $\mu^j$, defined as in (2.3), is smaller than the minimal threshold positive value for the computer, which makes $\bar{x}^*$ infinity.

In Figure 6 we compare our results with parameters (4.6) with the simulations using the variations of the new CBO model (2.2) discussed in Section 2. We use the variant of the CBO model without the noise term and the variant of the CBO model using the minimal value of the cost (2.5) over the agents rather than the average $\bar{x}^*$. We deduce the following general observations:

– A similar behavior in the performance with or without the last term in (2.4) is obtained. Since there is already stochasticity involved when calculating $\hat{L}^j$ and selecting random subsets of particles, it is usually not necessary to add extra noise when updating the positions of particles for these machine learning problems. However, for other optimization problems like the numerical experiment in Section 4.2, the geometric Brownian noise seems necessary to avoid clogging in local minima.
– Using argmin $\hat{L}(X^j)$ to update $x_k^*$ has also a similar performance.

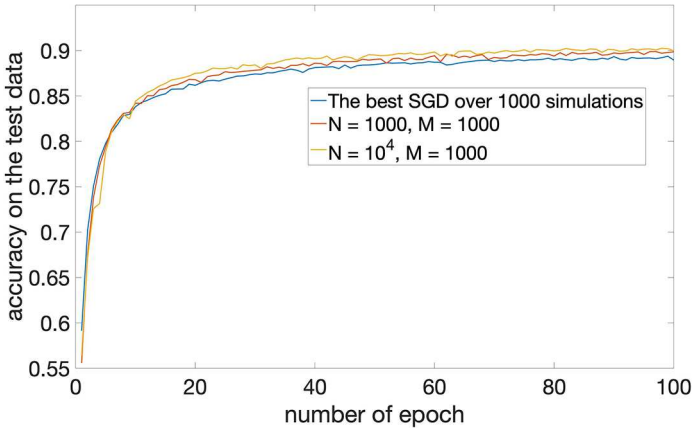


FIGURE 7. Comparison of our new CBO algorithm and SGD.

In Figure 7, we compare our results with stochastic gradient descent method. We use the same data set and neural network structure. We set the learning rate equal to 0.1, which is the same as $\gamma$ in our new method. To make a fair comparison, we run 1000 simulations of SGD with different initializations follow from standard normal distributions, which is also the same initialization as the proposed method. We plot the best one among all SGD simulations. We can see that our method with 1000 particles, which have the same computational cost, is slightly better than the best SGD over 1000 simulations. Besides, if we use $N = 10^4$, the test accuracy could be improved to around 90%. Therefore, our method can potentially get comparable accuracy with SGD in some settings.

As a concluding remark, we have shown in our numerical examples that, two alternative numerical methods, one where only random batch is involved without the diffusion term, one where $\bar{x}_k^*$ is directly equal to the argmin of the particles' value, performs as well as our method. However, the theoretical proof will be left for future study.

## 5. Conclusion

We improve the gradient-free optimization method upon [42], to make it effective for high dimensional optimization problems. We show in Theorem 3.2 and Proposition 3.3 that because of the component-wise geometric Brownian motion, the mean field limit of the method always converges to its good approximation of the global minimum with all the parameters independent of dimensionality. We show in Section 4.3 that for the MNIST data with two layers Neural Network, which is a 7290-dimensional optimization problem, only with 100 particles, it can already achieve 82% accuracy. In another example 4.2, our method has a significant higher success rate in finding the global minimum of the Rastrigin function compared to the practical implementation of the original method introduced in [42].

There are still lots of open problems in this direction, among them:

- Theoretical study for the method. Our theorems have not involved the random batch on particles and random batch on the data set. How the random batch method will affect the process of finding the global minimum will be the object of future studies.
- Stability condition for the method and criteria of choosing the optimal variables, such as, $M, N, \beta, \sigma, \gamma$ remain to be understood.

## Appendix A. Proof of Theorem 3.2

To prove this theorem, we need some preparation.

**Lemma A.1.** *$V(t)$ and $M(t)$ satisfy the following:*

$$\frac{\mathrm{d}}{\mathrm{d}t}V(t) \leq -\left(2\lambda - \sigma^2 - \sigma^2\frac{e^{-\beta L_m}}{M_L(t)}\right)V(t), \tag{A.1a}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}M_L^2(t) \geq -2\beta c_L(2\lambda+\sigma^2)e^{-2\beta L_m}V(t). \tag{A.1b}$$

*Proof.* By Itô's calculus, it holds that

$$\mathrm{d}|X-\mathbb{E}X|^2 = 2(X-\mathbb{E}X)\cdot\mathrm{d}X - 2(X-\mathbb{E}X)\cdot\mathrm{d}\mathbb{E}X + \sigma^2\sum_i (X-\bar{x}^*)_i^2\mathrm{d}t.$$

Hence, one can deduce that

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}V(t) &= -2\lambda\mathbb{E}\Big[(X-\mathbb{E}X)\cdot(X-\bar{x}^*)\Big] + \sigma^2\mathbb{E}|X-\bar{x}^*|^2\\ &= -2\lambda V(t) + \sigma^2\mathbb{E}|X-\bar{x}^*|^2.\end{aligned}$$

Here, we used the fact $\mathbb{E}(X-\mathbb{E}X)\cdot(\mathbb{E}X-\bar{x}^*)=0$. Moreover, we get $\mathbb{E}|X-\bar{x}^*|^2 = V(t)+|\mathbb{E}X-\bar{x}^*|^2$. By Jensen's inequality, for any $a\in\mathbb{R}^d$,

$$|a-\bar{x}^*|^2 \leq \frac{\mathbb{E}_{X_1\sim X}|a-X_1|^2 e^{-\beta L(X_1)}}{M_L}, \tag{A.2}$$

where $X_1\sim X$ means $X_1$ has the same distribution as $X$. Therefore,

$$\mathbb{E}|X-\bar{x}^*|^2 \leq V(t) + e^{-\beta L_m}\frac{V(t)}{M_L(t)}, \tag{A.3}$$

and (A.1a) follows. We remark that if $L$ a constant, the right hand side of (A.3) is $2V(t)$. However, in this case, $\bar{x}^*=\mathbb{E}X$ and the upper bound should be $V(t)$ instead of $2V(t)$. The reason is that Jensen's inequality (A.2) is lost for this case.

Analogously, by Itô's calculus, one can infer that

$$\begin{aligned}\mathrm{d}\mathbb{E}e^{-\beta L(X)} =& -\beta\mathbb{E}e^{-\beta L(X)}\nabla L(X)\cdot\mathrm{d}X\\ &+\frac{1}{2}\sigma^2\sum_i\mathbb{E}\Big[e^{-\beta L(X)}(X-\bar{x}^*)_i^2(\beta^2(\partial_i L)^2-\beta\partial_{ii}L(X))\Big]\,\mathrm{d}t =: (I_1+I_2)\mathrm{d}t.\end{aligned}$$

By definition and Assumption 3.1, one obtains

$$I_1 = \beta\lambda\mathbb{E}e^{-\beta L(X)}(\nabla L(X)-\nabla L(\bar{x}^*))\cdot(X-\bar{x}^*) \geq -\beta\lambda e^{-\beta L_m}c_L\mathbb{E}|X-\bar{x}^*|^2$$

where $\mathbb{E}e^{-\beta L(X)}\nabla L(\bar{x}^*)\cdot(X-\bar{x}^*)=0$ is used. For $I_2$, one has

$$I_2 \geq \frac{1}{2}\sigma^2(-\beta c_L)e^{-\beta L_m}\mathbb{E}|X-\bar{x}^*|^2.$$

Hence, we conclude that

$$\frac{\mathrm{d}M_L}{\mathrm{d}t} \geq -\beta c_L e^{-\beta L_m}\left(\lambda + \frac{1}{2}\sigma^2\right)\mathbb{E}|X - \bar{x}^*|^2. \tag{A.4}$$

Using the same estimate as in (A.2), one finds

$$\mathbb{E}|X - \bar{x}^*|^2 \leq V(t) + \frac{e^{-\beta L_m}}{M_L(t)}V(t) \leq 2\frac{e^{-\beta L_m}}{M_L(t)}V(t). \tag{A.5}$$

Inserting (A.5) into the differential inequality (A.4), the desired estimate follows. □

*Proof of Theorem 3.2.* Define

$$T := \sup\left\{t : M_L(s) \geq \frac{1}{2}M_L(0), \text{ for all } s \in [0,t]\right\}.$$

Clearly, $T > 0$. Assume that $T < \infty$. Then, for $t \in [0,T]$, by the assumption on $\mu$ in (3.4), one can deduce that

$$2\lambda - \sigma^2 - \sigma^2\frac{e^{-\beta L_m}}{M_L(t)} \geq 2\lambda - \sigma^2 - 2\sigma^2\frac{e^{-\beta L_m}}{M_L(0)} = \mu > 0.$$

Consequently, one has

$$\frac{\mathrm{d}V}{\mathrm{d}t} \leq -\mu V(t).$$

and thus

$$V(t) \leq V(0)\exp(-\mu t).$$

Hence, by the assumption on $\nu$ in (3.4),

$$\begin{aligned} M_L^2(t) &\geq M_L^2(0) - 2\beta c_L(2\lambda + \sigma^2)e^{-2\beta L_m}V(0)\int_0^t e^{-\mu s}\,\mathrm{d}s \\ &> M_L^2(0) - \frac{2V(0)\beta c_L(2\lambda+\sigma^2)e^{-2\beta L_m}}{\mu} \geq \frac{1}{4}M_L^2(0). \end{aligned}$$

This means that there exists $\delta > 0$ such that $M_L^2(t) \geq \frac{1}{4}M_L^2(0)$ in $[T, T+\delta)$ as well. This then contradicts with the definition of $T$. Hence, $T = \infty$. Consequently,

$$V(t) \leq V(0)\exp(-\mu t) \tag{A.6}$$

holds and

$$M_L(t) > \frac{1}{2}M_L(0) \tag{A.7}$$

for all $t > 0$. Using again Jensen's inequality (A.2) and (A.6)-(A.7), we infer that

$$|\mathbb{E}X - \bar{x}^*|^2 \leq \frac{\mathbb{E}_{X_1 \sim X}|X_1 - \bar{x}^*|^2 e^{-\beta L(X_1)}}{M_L(t)} \leq e^{-\beta L_m} \frac{V(t)}{M_L(t)} \leq C \exp(-\mu t).$$

Moreover, one has

$$|\frac{\mathrm{d}}{\mathrm{d}t}\mathbb{E}X| \leq \lambda \mathbb{E}|X - \bar{x}^*| \leq \lambda \sqrt{\mathbb{E}|X - \bar{x}^*|^2} \leq \lambda \sqrt{V(t) + |\mathbb{E}X - \bar{x}^*|^2} \leq C \exp(-\mu t/2).$$

Since the right-hand side is integrable on time, it follows that $\mathbb{E}X \to \tilde{x}$ for some $\tilde{x}$ and $\bar{x}^* \to \tilde{x}$, with exponential rate. Since $\mathbb{E}X \to \tilde{x}$ and $V(t) \to 0$, $M_L(t) \to e^{-\beta L(\tilde{x})}$. Hence, we deduce that

$$e^{-2\beta L(\tilde{x})} > M_L^2(0)(1 - \nu).$$

Therefore, we conclude that

$$L(\tilde{x}) < -\frac{1}{\beta} \log M_L(0) - \frac{1}{2\beta} \log(1 - \nu).$$

By the assumption on $\nu$ in (3.4), one thus has

$$L(\tilde{x}) < -\frac{1}{\beta} \log M_L(0) + \frac{\log 2}{\beta}.$$

By the Laplace principle (1.5), $r(\beta) = -\frac{1}{\beta} \log M_L(0) - L_m \to 0$. See more details in [11]. □

## Appendix B. Proof of Proposition 3.3

*Proof.* Since $\bar{x}_k^*$ is constant during the time interval, we find that the mean value $m(t)$ satisfies

$$\frac{\mathrm{d}}{\mathrm{d}t}(m(t) - \bar{x}_k^*) = -\lambda(m(t) - \bar{x}_k^*)$$

Therefore, we get $m(t_k) - \bar{x}_k^* = (m(t_{k-1}) - \bar{x}_k^*)e^{-\lambda\gamma}$. Hence, one obtains $m(t_k) = m(t_{k-1})e^{-\lambda\gamma} + \bar{x}_k^*(1 - e^{-\lambda\gamma})$. Consequently, it holds that

$$m(t_k) = m_0 e^{-\lambda k\gamma} + \sum_{\ell=0}^{k-1} (1 - e^{-\lambda\gamma}) \bar{x}_\ell^* e^{-(k-\ell)\lambda\gamma}.$$

If $\bar{x}_\ell^*$ is bounded, this summation converges, and the sum is controlled by $C \sup_k \|\bar{x}_k^*\|$ with $C$ independent of $\gamma$. By Itô's calculus, the second moment satisfies

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbb{E}|X - \bar{x}_k^*|^2 = (-2\lambda + \sigma^2)\mathbb{E}|X - \bar{x}_k^*|^2.$$

Since the variance is given by

$$V(t_k) = \mathbb{E}|X - \bar{x}_k^*|^2 - |m(t_k) - \bar{x}_k^*|^2,$$

we have

$$V(t_k) = V(t_{k-1})e^{(-2\lambda+\sigma^2)\gamma} + (e^{(-2\lambda+\sigma^2)\gamma} - e^{-2\lambda\gamma})|m(t_{k-1}) - \bar{x}_k^*|^2.$$

If $\{\bar{x}_k^*\}$ is bounded, then $m(t_k)$ is bounded as has been shown. Consequently, $V(t_k)$ is bounded uniformly. It then follows that there exists a compact set $K$ such that

$$\sup_k \int_{\mathbb{R}^d \setminus K} \mathrm{d}\rho(x, t_k) < 1/2.$$

Hence, we obtain that

$$M_L^* \geq \frac{1}{2} \inf_{x \in K} e^{-\beta L(x)} > 0.$$

Using (A.2), we have

$$|m(t_{k-1}) - \bar{x}_k^*|^2 \leq \frac{e^{-\beta L_m}}{M_L} V(t_{k-1}) \leq \frac{e^{-\beta L_m}}{M_L^*} V(t_{k-1}).$$

If $\gamma$ satisfies condition (3.9), then one sees easily that $V(t_k) \to 0$ and $|m(t_{k-1}) - x_k^*| \to 0$. Hence, it is clear that

$$\bar{m} := \lim_{k \to \infty} m(t_k)$$

exists.

Using Chebyshev's inequality, it is easy to see that for any $\epsilon > 0$, there exists $R > 0$ such that

$$\sup_{t \geq 0} \mathbb{E} 1_{X \in \mathbb{R}^d \setminus B(0,R)} \leq \epsilon.$$

For any test function $\varphi \in C_b$, we find $R > |\bar{m}|$, $\phi \in C_b^2(\mathbb{R}^d)$ such that $\|\phi\|_{C_b} \leq 2\|\varphi\|_{C_b}$, and that

$$\sup_{x \in B(0,R)} |\phi - \varphi| \leq \epsilon,$$

and

$$\sup_{t \geq 0} \mathbb{E} 1_{X \in \mathbb{R}^d \setminus B(0,R)} \leq \frac{\epsilon}{\|\varphi\|_{C_b}}.$$

Then, we deduce that

$$|\mathbb{E}\varphi(X) - \varphi(\bar{m})| \leq |\mathbb{E}\phi(X) - \varphi(\bar{m})| + \mathbb{E}|\phi(X) - \varphi(X)| \to |\phi(\bar{m}) - \varphi(\bar{m})| + \mathbb{E}|\phi(X) - \varphi(X)| \leq 2\epsilon.$$

Consequently, we have

$$\mathbb{E}e^{-\beta L(X)} \to \mathbb{E}e^{-\beta L(\bar{m})} > 0.$$

Hence, we conclude that

$$\inf_{t > 0} \mathbb{E}e^{-\beta L(X)} > 0.$$

Finally, using similar estimates as in the time continuous case, we obtain that

$$|\bar{x}_k^* - m(t_k)|^2 \to 0,$$

consistent with the fact that $\bar{x}_k^*$ is bounded. This finishes the proof. □

*Acknowledgements.* JAC was partially supported by EPSRC grant number EP/P031587/1 and the Advanced Grant Nonlocal-CPD (Nonlocal PDEs for Complex Particle Dynamics: Phase Transitions, Patterns and Synchronization) of the European Research Council Executive Agency (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 883363). SJ was partially supported by NSFC grants No. 11871297 and No. 31571071. LL was partially supported by NSFC grants No. 11901389 and No. 11971314, and the Shanghai Sailing Program 19YF1421300. We thank Dr. Doheon Kim for discussion on the convergence rate of the Laplace principle.

## References

[1] G. Albi and L. Pareschi, Binary interaction algorithms for the simulation of flocking and swarming dynamics. *Multisc. Model. Simul.* **11** (2013) 1–29.
[2] J.J. Alonso and J. Hicken, Introduction to multidisciplinary design optimization. In Vol.222 of *Aeronautics & Astronautics*. Standford University (2012).
[3] N. Bellomo, A. Bellouquid and D. Knopoff, From the microscale to collective crowd dynamics. *Multis. Model. Simul.* **11** (2013) 943–963.
[4] C.M. Bender and S.A. Orszag, Advanced Mathematical Methods for Scientists and Engineers. *International Series in Pure and Applied Mathematics*. McGraw-Hill (1978).
[5] Y. Bengio, P. Simard and P. Frasconi, Learning long-term dependencies with gradient descent is difficult. *IEEE Trans. Neural Netw.* **5** (1994) 157–166.
[6] A.L. Bertozzi, J. Rosado, M.B. Short and L. Wang, Contagion shocks in one dimension. *J. Stat. Phys.* **158** (2015) 647–664.
[7] F. Bolley, J.A. Cañizo and J.A. Carrillo, Stochastic mean-field limit: non-Lipschitz forces and swarming. *Math. Models Methods Appl. Sci.* **21** (2011) 2179–2210.
[8] L. Bottou, Online learning and stochastic approximations. *On-line Learn. Neural Netw.* **17** (1998) 142.
[9] S. Bubeck, Convex optimization: algorithms and complexity. *Found. Trends® Mach. Learn.* **8** (2015) 231–357.
[10] J.A. Carrillo, Y.-P. Choi and M. Hauray, The derivation of swarming models: mean-field limit and Wasserstein distances. Collective dynamics from bacteria to crowds, volume 553 of *CISM Courses and Lectures*. Springer, Vienna (2014) 1–46.
[11] J.A. Carrillo, Y.-P. Choi, C. Totzeck and O. Tse, An analytical framework for consensus-based global optimization method. *Math. Models Methods Appl. Sci.* **28** (2018) 1037–1066.
[12] J.A. Carrillo, M. Fornasier, J. Rosado and G. Toscani, Asymptotic flocking dynamics for the kinetic Cucker-Smale model. *SIAM J. Math. Anal.* **42** (2010) 218–236.
[13] J.A. Carrillo, M. Fornasier, G. Toscani and F. Vecil, Particle, kinetic, and hydrodynamic models of swarming. In Mathematical modeling of collective behavior in socio-economic and life sciences, *Modelling and Simulation in Materials Science and Engineering*. Birkhäuser Boston, Inc., Boston, MA (2010) 297–336.
[14] J.A. Carrillo, L. Pareschi and M. Zanella, Particle based gPC methods for mean-field models of swarming with uncertainty. *Commun. Comput. Phys.* **25** (2018) 508–531.
[15] F. Cucker and S. Smale, On the mathematics of emergence. *Jpn. J. Math.* **2** (2007) 197–227.
[16] X. Dai and Y. Zhu, Towards theoretical understanding of large batch training in stochastic gradient descent. Preprint arXiv:1812.00542 (2018).
[17] A. Dembo and O. Zeitouni, Vol. 38 of Large deviations techniques and applications. Springer Science & Business Media (2009).
[18] R. Eberhart and J. Kennedy, Particle swarm optimization. In Proc. of *IEEE International Conference on Neural Networks* **4** (1995) 1942–1948.
[19] S.-Y. Ha, S. Jin and D. Kim, Convergence of a first-order consensus-based global optimization algorithm. Preprint arXiv:1910.08239 (2019).
[20] S.-Y. Ha and E. Tadmor, From particle to kinetic and hydrodynamic descriptions of flocking. *Kinetic Related Models* **1** (2008) 415–435.
[21] B. Hanin, Which neural net architectures give rise to exploding and vanishing gradients? In *Adv. Neural Inf. Process. Syst.* (2018) 582–591.
[22] M. Hauray and P.-E. Jabin, $N$-particles approximation of the Vlasov equations with singular potential. *Arch. Ratl. Mech. Anal.* **183** (2007) 489–524.
[23] J.H. Holland, Genetic algorithms. *Sci. Am.* **267** (1992) 66–73.
[24] R.A. Holley, S. Kusuoka and D.W. Stroock, Asymptotics of the spectral gap with applications to the theory of simulated annealing. *J. Funct. Anal.* **83** (1989) 333–347.
[25] L.C. Hsu, A theorem on the asymptotic behavior of a multiple integral. *Duke Math. J.* **15** (1948) 623–6323.

[26] C.-R. Hwang and S.-J. Sheu, Large-time behavior of perturbed diffusion Markov processes with applications to the second eigenvalue problem for Fokker-Planck operators and simulated annealing. *Acta Appl. Math.* **19** (1990) 253–295.
[27] T. Inglot and P. Majerski, Simple upper and lower bounds for the multivariate Laplace approximation. *J. Approx. Theory* **186** (2014) 1–11.
[28] P.-E. Jabin, A review of the mean field limits for Vlasov equations. *Kinetic Related Models* **7** (2014) 661–711.
[29] P.-E. Jabin and Z. Wang, Quantitative estimates of propagation of chaos for stochastic systems with $W^{-1,\infty}$ kernels. *Invent. Math.* **214** (2018) 523–591.
[30] S. Jastrzebski, Z. Kenton, D. Arpit, N. Ballas, A. Fischer, Y. Bengio and A. Storkey, Three factors influencing minima in SGD. Preprint arXiv:1711.04623 (2017).
[31] S. Jin, L. Li and J.-G. Liu, Random batch methods (RBM) for interacting particle systems. *J. Comput. Phys.* **400** (2020) 108877.
[32] J. Kennedy, Swarm intelligence, handbook of nature-inspired and innovative computing. Springer (2006) 187–219.
[33] N.S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy and P.T.P. Tang, On large-batch training for deep learning: generalization gap and sharp minima. In International Conference on Learning Representations (2017).
[34] S. Kirkpatrick, C. Daniel Gelatt and M.P. Vecchi, Optimization by simulated annealing. *Science* **220** (1983) 671–680.
[35] T. Kolokolnikov, J.A. Carrillo, A. Bertozzi, R. Fetecau and M. Lewis, Emergent behaviour in multi-particle systems with non-local interactions [Editorial]. *J. Phys. D* **260** (2013) 1–4.
[36] J.P. McClure and R. Wong, Error bounds for multidimensional Laplace approximation. *J. Approx. Theory* **37** (1983) 372–390.
[37] P.J.M. van Laarhoven and E.H.L. Aarts, Simulated annealing: theory and applications. D. Reidel Publishing Co., Dordrecht (1987) 37.
[38] S. Liu, D. Papailiopoulos and D. Achlioptas, Bad global minima exist and SGD can reach them. Preprint arXiv:1906.02613 (2019).
[39] P.D. Miller, Applied asymptotic analysis. American Mathematical Society (2006).
[40] S. Motsch and E. Tadmor, Heterophilious dynamics enhances consensus. *SIAM Rev.* **56** (2014) 577–621.
[41] J.A. Nelder and R. Mead, A simplex method for function minimization. *Comput. J.* **7** (1965) 308–313.
[42] R. Pinnau, C. Totzeck, O. Tse and S. Martin, A consensus-based model for global optimization and its mean-field limit. *Math. Models Methods Appl. Sci.* **27** (2017) 183–204.
[43] H. Robbins and S. Monro, A stochastic approximation method. *Ann. Math. Stat.* (1951) 400–407.
[44] G. Toscani, Kinetic models of opinion formation. *Commun. Math. Sci.* **4** (2006) 481–496.
[45] C. Totzeck, R. Pinnau, S. Blauth and S. Schotthófer, A numerical comparison of consensus-based global optimization to other particle-based global optimization schemes. *Proc. Appl. Math. Mech.* **18** (2018) e201800291.