# Consensus based optimization with memory effects: random selection and applications

Giacomo Borghi\*

Sara Grassi<sup>†</sup>

Lorenzo Pareschi<sup>†</sup>

March 21, 2023

#### Abstract

In this work we extend the class of Consensus-Based Optimization (CBO) metaheuristic methods by considering memory effects and a random selection strategy. The proposed algorithm iteratively updates a population of particles according to a consensus dynamics inspired by social interactions among individuals. The consensus point is computed taking into account the past positions of all particles. While sharing features with the popular Particle Swarm Optimization (PSO) method, the exploratory behavior is fundamentally different and allows better control over the convergence of the particle system. We discuss some implementation aspects which lead to an increased efficiency while preserving the success rate in the optimization process. In particular, we show how employing a random selection strategy to discard particles during the computation improves the overall performance. Several benchmark problems and applications to image segmentation and Neural Networks training are used to validate and test the proposed method. A theoretical analysis allows to recover convergence guarantees under mild assumptions on the objective function. This is done by first approximating the particles evolution with a continuous-in-time dynamics, and then by taking the mean-field limit of such dynamics. Convergence to a global minimizer is finally proved at the mean-field level.

**Keywords**: consensus-based optimization, stochastic particle methods, memory effects, random selection, machine learning, mean-field limit

### Contents

1	Intr	oduction	2
<b>2</b>	Con	sensus-based optimization with memory effects	4
	2.1	Particles update rule	4
	2.2	Random selection strategy	5
	2.3	Comparison with CBO and PSO	6

\*RWTH Aachen University, Institute for Geometry and Applied Mathematics, Aachen, Germany (borghi@eddy.rwth-aachen.de)

<sup>†</sup>University of Ferrara, Department of Mathematics and Computer Science & Center for Modelling Computing and Statistics, Ferrara, Italy (sara.grassi@unife.it, lorenzo.pareschi@unife.it)

3	Nur	merical results	8
	3.1	Tests on benchmark problems	8
	3.2	Applications	13
		3.2.1 Image segmentation	14
		3.2.2 Approximating functions with NN	16
		3.2.3 Application on MNIST dataset	17
<b>4</b>	The	eoretical analysis	21
	4.1	Mean-field approximation	21
	4.2	Convergence in mean-field law	23
	4.3	Random selection analysis	25
<b>5</b>	Con	nclusions	27
$\mathbf{A}$	Pro	ofs	28
	A.1	Notation and auxiliary lemmas	28
	A.2	Proof of Proposition 4.1	29
	A.3	Proof of Theorem 4.1	30
	A.4	Proof of Proposition 4.2 and Theorem 4.2 $\ldots$	33

### 1 Introduction

Meta-heuristic algorithms are recognized as trustworthy, easy to understand optimization methods which have been widely applied to several fields such as Machine Learning [30], path planning [31] and image processing [47], to name a few. Starting from a set of possible solutions, a meta-heuristic algorithm typically updates such set iteratively by combining deterministic and stochastic choices, often inspired by natural phenomena. Exploration of the search space and exploitation of the current knowledge are the two fundamental mechanisms driving the algorithm iteration [48]. Examples of established meta-heuristic algorithms are given by Genetic Algorithm (GA) [19, 44], Simulated Annealing (SA) [27], Particle Swarm Optimization (PSO) [26] and Differential Evolution (DE) [42]. We refer to [23] for a complete literature review.

Consensus-Based Optimization (CBO) is a class of gradient-free meta-heuristic algorithms inspired by consensus dynamics among individuals. After its introduction [36] it has gained popularity among the mathematical community due to its robust mathematical framework [5, 11,18,21]. In CBO algorithms, a population of particles concentrates around a consensus point given by a weighted average of the particles position. In the computation of such consensus point, more importance is given to those particles attaining relatively low values of the objective function by means of the Gibbs distribution. The exploration mechanism is introduced by randomly perturbing the particles positions at each iteration. Particles which are close to the consensus point are subject to small perturbations, while those that are far from it display a more exploratory behavior.

In this work, following the recent analysis in [16], we study a Consensus-Based Optimization algorithm with Memory Effects (CBO-ME) where the consensus point is computed among the whole history of the particles positions and not just among the positions of the current iteration, as in the original CBO method. This is done by keeping track of the best position found so far by each particle, and by computing the consensus point among these "personal" bests. While sharing common elements with PSO, such as the convergence mechanism to a promising point and the presence of personal bests, CBO-ME differs in the way the exploration mechanism is implemented. Indeed, in CBO-ME, as in CBO algorithms, the stochastic behavior is given by adding Gaussian noise to the particles dynamics and can be tuned independently on the exploitation mechanisms, leading to a better control over the particles convergence. Therefore, while in classical PSO methods it is the balance between local best and global best that governs the optimization strategy, in CBO methods it is the balance between exploration and exploitation mechanisms that determines the choice of parameters. We recall that a generalization of PSO methods that allows leveraging the same flexibility in searching the global minimum as in CBO algorithms has been recently presented in [16].

Many real-life problems, especially those regarding Machine Learning, require to optimize a large number of parameters. Therefore, it essential to design fast algorithms to save computational time and memory. This is a major weakness of swarm-based methods, which require a set of particles to minimize the problem, unlike gradient-based methods that can work on a single particle trajectory. For methods based on a collection of particles, existing algorithms can be improved by discarding particles whenever the system has a prominent exploitative behavior. This is sometimes referred as "natural selection strategy" in the DE literature [29, 42] and aims to discard the non-promising solutions. Inspired by particle simulations techniques where it is important to preserve the particles probability distribution, we examine a "random selection strategy" where particles are discarded randomly based on the local consensus achieved. We will discuss such implementation aspects by testing CBO-ME against high-dimensional learning problems and theoretically analyze the impact of the random selection strategy on the system. In particular, we prove that if the full particle system is expected to converge towards a solution to the minimization problem, so will the reduce one, provided a sufficient number of particles remains active. Note that, such analysis can be generalized to other particle dynamics and may be of independent interest.

Owing to the convergence analysis of CBO algorithms [5, 11, 12, 21] and recent analysis of PSO [16, 22] we are able to prove convergence of the algorithm under mild assumption on the objective function. This is done by first approximating the algorithm with a continuous-intime dynamics and secondly by giving a probabilistic description to the particles system. By assuming propagation of chaos [43], particles are considered to behave independently according to the same law. This allows to reduce the possible large system of equations to a single partial differential equation: the so-called mean-field model. Such model is then analyzed to recover convergence guarantees under precise assumption on the objective function. Developed in the field of statistical physics, this approach has shown be fruitful in studying particle-based metaheuristic algorithms [11,12,22]. We note that convergence in mean-field law was recently proved in [39] in an independent work.

The rest of the paper is organized as follows. Section 2 is devoted to the introduction of the CBO-ME algorithm with random selection and comparison with CBO methods without memory effects as well as PSO. In Section 3, validate the proposed method against several benchmark problems and two Machine Learning tasks. Theoretical convergence guarantees and analysis of the random selection strategy are summarized in Section 4. Some final remarks are given in

Section 5. Technical details of the theoretical analysis are given in Appendix A.

### 2 Consensus-based optimization with memory effects

In this section, we present the Consensus-Based Optimization algorithm with Memory Effects (CBO-ME) to solve problems of the form

$$x^* \in \operatorname*{argmin}_{x \in \mathbb{R}^d} \mathcal{F}(x),$$
 (2.1)

where  $\mathbb{R}^d$ ,  $d \in \mathbb{N}$  is the, possibly large, search domain for the continuous function  $\mathcal{F} \in C(\mathbb{R}^d, \mathbb{R})$ . We will do so by also highlighting similarities and differences between classical CBO methods and PSO algorithms.

### 2.1 Particles update rule

At each iteration step k and for every particle i = 1, ..., N, we store its position  $x_i^k$  and its best position found so far  $y_i^k, \mathcal{F}(y_i^k) = \min_{h \le k} \mathcal{F}(x_i^h)$ . The best positions are used to compute a consensus point

$$\overline{y}^{\alpha,k} = \sum_{i=1}^{N} \omega_i^k y_i^k \quad \text{with} \quad \omega_i^k = \frac{e^{-\alpha \mathcal{F}(y_i^k)}}{\sum_{j=1}^{N} e^{-\alpha \mathcal{F}(y_j^k)}}$$
(2.2)

which approximates the global best solution  $\overline{y}^{\infty,k}$  among all particles and all times for  $\alpha > 1$ . Indeed, thanks to the choice of the weights  $\omega_i^k$ , we have that

$$\overline{y}^{\alpha,k} \longrightarrow \overline{y}^{\infty,k} := \operatorname{argmin}\{\mathcal{F}(y_1^k), \dots, \mathcal{F}(y_N^k)\}$$

as  $\alpha \to \infty$ , provided that there is only one global best position among  $\{y_1^k, \ldots, y_N^k\}$ . Such approximation was first introduced for CBO methods [36] as it leads to more amenable theoretical analysis, but it also allows for more flexibility. Indeed, relatively small values of  $\alpha$  can be used at the beginning of the computation to promote exploration. Large values of  $\alpha$ , on the other hand, lead to better exploitation of the computed solutions and to higher accuracy. We note that the weights used in (2.2) correspond in statistical mechanics to the Boltzmann-Gibbs distribution associated with the energy  $\mathcal{F}$ . In this context,  $\alpha$  plays the role of the inverse of the system temperature T and the limit  $\alpha \to \infty$  corresponds to  $T \to 0$ .

Once the consensus point  $\overline{y}^{\alpha,k}$  is computed, the particle positions are then updated according to the law

$$x_i^{k+1} = x_i^k + \lambda \left( \overline{y}^{\alpha,k} - x_i^k \right) + \sigma \left( \overline{y}^{\alpha,k} - x_i^k \right) \otimes \theta_i^k$$
(2.3)

with  $\theta_i^k \in \mathbb{R}^d$  randomly sampled from the normal distribution  $(\theta_i^k \sim \mathcal{N}(0, \mathbf{I}_d))$  and where  $\otimes$  is the component-wise product.

The update rule is characterized by a deterministic component of strength  $\lambda \in (0, 1)$  promoting concentration around the consensus point  $\overline{y}^{\alpha,k}$  and a stochastic component of strength  $\sigma > 0$ promoting exploration of the search space. As the latter depends on the difference  $(\overline{y}^{\alpha,k} - x_i^k)$ , the random behavior is stronger for particles which are far from the consensus point, whereas it is weaker for those that are close to it. Also, such exploration resembles an anisotropic diffusive behavior in which every coordinate direction is explored at a different rate. This approach was first proposed in [6] in the context of CBO methods and has been proved to suffer less from the curse of dimensionality with the respect to the originally proposed isotropic diffusion given by  $\sigma \|\overline{y}^{\alpha,k} - x_i^k\|_2 \theta_i^k$  with  $\theta_i^k$  being again a normally distributed d-dimensional vector [6].

### 2.2 Random selection strategy

When the particle system concentrates around the consensus point, showing a mostly exploitative behavior, we employ a particle selection strategy. Discarding particles introduces additional stochasticity to the system, while reducing the computational cost. Following the approach suggested in [9], we check the evolution of the system variance to decide how many particles to (eventually) discard.

For a given set of particles  $\mathbf{z} = \{z_i\}_{i \in J}$ , the system variance is given by

$$\operatorname{var}(\mathbf{z}) := \frac{1}{|J|} \sum_{j \in J} \|z_j - \mathbf{m}(\mathbf{z})\|_2^2 \quad \text{with} \quad \mathbf{m}(\mathbf{z}) := \frac{1}{|J|} \sum_{i \in J} z_i \,, \tag{2.4}$$

where |J| indicates the cardinality of J, that is, the number of particles in this context.

Let  $I_k \subseteq \{1, \ldots, N\}$  be the set of active particles at step k and  $N_k = |I_k|$ . To decide how many particles to select, we compare the variance of the particle system before the position update (2.3),  $\mathbf{x}^k = \{x_i^k\}_{i \in I_k}$  and after it,  $\tilde{\mathbf{x}}^{k+1} = \{x_i^{k+1}\}_{i \in I_k}$ . Then, the number  $N_{k+1}$  of particles we select for the next iteration is given by

$$\tilde{N}_{k+1} = \left\lfloor N_k \left( 1 + \mu \frac{\operatorname{var}(\tilde{\mathbf{x}}^{k+1}) - \operatorname{var}(\mathbf{x}^{k+1})}{\operatorname{var}(\mathbf{x}^{k+1})} \right) \right\rfloor$$

$$N_{k+1} = \min \left\{ \max \left\{ \tilde{N}_{k+1}, N_{\min} \right\}, N_k \right\}$$
(2.5)

with  $\lfloor z \rfloor$  being the integer part of a number z and  $N_{\min} \in \mathbb{N}$  the smallest amount of particles we allow to have. If  $N_{k+1} < N_k$ , a subset  $I_{k+1} \subset I_k$ ,  $|I_{k+1}| = N_{k+1}$ , of particles is randomly selected to continue the computation. The parameter  $\mu \in [0, 1]$  regulates the mechanism: for  $\mu = 0$  there is no particle discarding, while for  $\mu = 1$  the maximum number of particles is discarded if the variance is decreasing. As we will see in Section 3, this random selection mechanism reduces the computational time without affecting the algorithm performance. We will also theoretically analyze this aspect in Section 4.3, where we show that convergence properties are preserved.

As stopping criterion, we keep a counter n on how many times  $\|\overline{y}^{\alpha,k+1} - \overline{y}^{\alpha,k}\|_2$  is smaller than a certain tolerance  $\delta_{\text{stall}} > 0$ . If this happens for more than a given  $n_{\text{stall}}$  number of times in a row, we assume the particle system has found a solution and stop the computation. A maximum number of iteration  $k_{\text{max}}$  representing the computational budget is also given. The proposed CBO-ME is summarized in Algorithm 1.

**Remark 2.1.** In the meta-heuristic literature, particles are usually discarded depending on their objective value, in a way that particles with high objective value are more likely to be discarded [29, 42]. The proposed strategy does not add a further heuristic strategy but simply cut down the algorithm complexity. Also, convergence properties are in this way expected to be preserved.

We note that, on the other hand, there is no straightforward way to both generate particles and preserve the particle system distribution at the same time.

Algorithm 1: Consensus-Based Optimization with Memory Effects (CBO-ME)

**Input:**  $\mathcal{F}, N_0, N_{\min}, k_{\max}, \lambda, \sigma, \alpha, n_{\text{stall}}$  and  $\delta_{\text{stall}}$ ; 1 Inizialize  $N_0$  particle positions  $x_0^i, i = 1, \ldots, N$ ; **2**  $y_i^0 \leftarrow x_i^0$  for all  $i = 1, ..., N_0$ ; **3** Compute  $\overline{y}^{\alpha,0}$  according to (2.2); 4  $k \leftarrow 0, n \leftarrow 0;$ 5 while  $k < k_{\text{max}}$  and  $n < n_{\text{stall}}$  do for i = 1 to  $N_k$  do 6  $\theta_i^k \sim \mathcal{N}(0, \mathbf{I}_d);$ 7 Compute  $x_i^{k+1}$  according to (2.3); 8  $\begin{array}{c} \mathbf{if} \ \mathcal{F}(x_i^{k+1}) < \mathcal{F}(y_i^k) \ \mathbf{then} \\ \mid \ y_i^{k+1} \leftarrow x_i^{k+1}; \end{array}$ 9 10else  $\mathbf{11}$  $| y_i^{k+1} \leftarrow y_i^k;$ 12end  $\mathbf{13}$ end 14 Compute  $\overline{y}^{\alpha,k+1}$  according to (2.2);  $\mathbf{15}$ if  $\|\overline{y}^{\alpha,k+1} - \overline{y}^{\alpha,k}\|_2 < \delta_{\text{stall}}$  then 16  $n \leftarrow n+1;$  $\mathbf{17}$ else 18  $n \leftarrow 0;$ 19  $\mathbf{20}$ end Compute  $N_{k+1}$  according to (2.5);  $\mathbf{21}$ if  $N_{k+1} < N_k$  then  $\mathbf{22}$ Randomly discard  $N_{k+1} - N_k$  particles;  $\mathbf{23}$  $k \leftarrow k + 1;$  $\mathbf{24}$ 25 end 26 return  $\overline{y}^{\alpha,k}, \mathcal{F}(\overline{y}^{\alpha,k})$ 

### 2.3 Comparison with CBO and PSO

What distinguishes CBO-ME from plain CBO, see e.g [6, 36], is clearly the introduction of the best positions  $\{y_i^k\}_{i=1}^N$  and the fact that the consensus point is calculated among them and not just among the particle positions  $\{x_i^k\}_{i=1}^N$  at that given time k. Indeed, the classical CBO update rule without memory effects (and with anisotropic diffusion and projection step) is given by

$$x_i^{k+1} = x_i^k + \lambda \left( \overline{x}^{\alpha,k} - x_i^k \right) + \sigma \left( \overline{x}^{\alpha,k} - x_i^k \right) \otimes \theta_i^k$$
(2.6)

where  $\overline{x}^{\alpha,k}$  is defined consistently with (2.2) (by substituting  $y_i^k$  with  $x_i^k$ ). As we will see in the numerical tests, the use of memory effects improves the algorithm performance.

Since alignment towards personal bests  $y_i^k$  and towards the global best  $\overline{y}^{\infty,k}$  are also the fundamental building blocks of PSO algorithms, we highlight now the main differences and similarities between PSO and CBO-ME. For completeness, we recall the canonical PSO method, see e.g. [38], using the notation of (2.3) for easier comparison

$$\begin{cases} x_i^{k+1} &= x_i^k + v_i^{k+1} \\ v_i^{k+1} &= wv_i^k + C_1 \left( y_i^k - x_i^k \right) \otimes \widehat{\theta}_{i,1}^k + C_2 \left( \overline{y}^{\infty,k} - x_i^k \right) \otimes \widehat{\theta}_{i,2}^k \end{cases}$$
(2.7)

where  $v_i^k$  are the particles velocities,  $w, C_1, C_2 > 0$  are the algorithm parameters and  $\theta_{i,1}^k, \theta_{i,2}^k$ are uniformly sampled from  $[0, 1]^d$ ,  $(\hat{\theta}_{i,1}^k, \hat{\theta}_{i,2}^k) \sim \text{Unif}([0, 1]^d)$ . Several variants and improvements have been proposed starting from the above dynamics, but a complete review is beyond the scope of this paper and we refer to the recent survey [49] for more references.

We are interested in highlighting the main differences between (2.3) and (2.7) regarding the stochastic components: in CBO-ME deterministic and stochastic steps are de-coupled and tuned by two different parameters ( $\lambda$  and  $\sigma$ ), while in PSO they are coupled. Indeed, in (2.7), deterministic and stochastic components are both controlled by the same parameter:  $C_1$  in the case of personal best dynamics and  $C_2$  for the global best one. By splitting the term  $C_2 (\bar{y}^{\infty,k} - x_i^k) \hat{\theta}_{i,2}^k$  into a deterministic step and a zero-mean term we obtain

$$C_2\left(\overline{y}^{\infty,k} - x_i^k\right) \otimes \widehat{\theta}_{i,2}^k = \frac{C_2}{2}\left(\overline{y}^{\infty,k} - x_i^k\right) + \frac{C_2}{2}\left(\overline{y}^{\infty,k} - x_i^k\right) \otimes \theta_{i,2}^k \tag{2.8}$$

with  $\theta_{i,2}^k = 2\hat{\theta}_{i,2}^k - 1$ ,  $\theta_{i,2}^k \sim \text{Unif}([-1,1]^d)$ . Suggested in [16], such rewriting highlights how increasing the alignment strength towards the global best (by increasing  $C_2$ ) necessary increases the stochasticity of the system as well. In (2.3) and (2.6), on the other hand, one is allowed to tune the exploration and exploitation behaviors separately, by either changing parameter  $\lambda$  or  $\sigma$ .

Clearly, CBO-ME also differs from PSO due to its first-order dynamics. Having the aim of resembling birds flocking, the first PSO algorithm [26] was proposed as a second-order dynamics. The inertia weight w, introduced later in [41], became an essential parameter to prevent early convergence of the swarm and to increase the global exploration behavior, especially at the beginning of the computation, see e.g. [33, 41] and reviews [20, 38, 49] for more references. We note that several other strategies have proposed to improve PSO exploration behavior, see, for example, [53]. As already mentioned, in CBO methods convergence and exploration are de-coupled and can be tuned separately. Therefore, to keep the algorithm more amenable to theoretical analysis, we consider a simpler first-order dynamics. We note that a CBO dynamics with inertia mechanism was proposed in [7].

Similarly, we found the contribution given by the personal best alignment non-essential and difficult to tune. Thus, the lack of alignment towards personal best in (2.3). Replacing alignment towards personal best with Gaussian noise was also suggested in [50] where authors proposed the Accelerated PSO (APSO) algorithm. Further studied in [13, 51], APSO also allows to decouple the stochastic component from the deterministic one and the noise is heuristically tuned

Name	<b>Objective function</b> $\mathcal{F}(x)$	Search space	$x^*$	$\mathcal{F}(x^*)$
Ackley	$-20 \exp\left(-0.2\sqrt{\frac{1}{d}\sum_{i=1}^{d} (x_i)^2}\right) - \exp\left(\frac{1}{d}\sum_{i=1}^{d} \cos\left(2\pi(x_i)\right)\right) + 20 + e$	$[-32, 32]^d$	$(0,\ldots,0)$	0
Griewank	$1 + \sum_{i=1}^{d} \frac{(x_i)^2}{4000} - \prod_{i=1}^{d} \cos\left(\frac{x_i}{i}\right)$	$[-600, 600]^{d}$	$(0,\ldots,0)$	0
Rastrigin	$10d + \sum_{i=1}^{d} \left[ (x_i)^2 - 10\cos\left(2\pi(x_i)\right) \right]$	$[-5.12, 5.12]^{d}$	$(0,\ldots,0)$	0
Rosenbrock	$1 - \cos\left(2\pi\sqrt{\sum_{i=1}^{d} (x_i)^2}\right) + 0.1\sqrt{\sum_{i=1}^{d} (x_i)^2}$	$[-5, 10]^{d}$	(1,, 1)	0
Salomon	$1 - \cos\left(2\pi\sqrt{\sum_{i=1}^{d} (x_i)^2}\right) + 0.1\sqrt{\sum_{i=1}^{d} (x_i)^2}$	$[-100, 100]^{d}$	$(0,\ldots,0)$	0
Schwefel 2.20	$\sum_{i=1}^{ ext{d}}  x_i $	$[-100, 100]^{d}$	$(0,\ldots,0)$	0
XSY random	$\sum_{i=1}^{ ext{d}} \eta_i  x_i ^i, \qquad \eta_i \sim  ext{Unif}([0,1])$	$[-5, 5]^{d}$	$(0,\ldots,0)$	0
XSY 4	$\left(\sum_{i=1}^{\mathrm{d}} \sin^2(x_i) - e^{-\sum_{i=1}^{\mathrm{d}} (x_i)^2}\right) e^{-\sum_{i=1}^{\mathrm{d}} \sin^2 \sqrt{ x_i }}$	$[-10, 10]^{d}$	$(0,\ldots,0)$	-1

**Table 1:** Considered benchmark test functions for global optimization. For each function, the corresponding search space and global solution is given.

to decrease during the computation as in Simulated Annealing [27]. In CBO methods, the noise strength automatically adapts as it depends on the distance from the consensus point, which is also different for every particle. For completeness, we note that many other variants of PSO have been proposed to include different explorative behaviors, see e.g. Chaotic PSO [32].

# 3 Numerical results

Having discussed the fundamental features of the CBO dynamics with memory effects, we now validate Algorithm 1 and compare its performance with plain CBO and PSO. We will test the methods against several benchmark optimization problems and analyze the impact of the random selection technique on the convergence speed. We also employ Algorithm 1 to solve problems arising from applications, such as image segmentation and training of Machine Learning architectures for function approximation and image classification.

### 3.1 Tests on benchmark problems

We test the proposed algorithm against different optimization problems, by considering 8 benchmark objective functions, see e.g. [24], which we report in Table 1 for completeness. The search space dimension is set to d = 20 and the location of the global best  $x^*$  is known.

As in plain CBO methods, we expect the most important parameters to be  $\lambda$  and  $\sigma$ , governing the balance between the exploitative behavior and the explorative one. In particular, we are interested in the algorithm performance as we change the ratio between  $\lambda$  and  $\sigma$ . Therefore, in the first experiment we fix  $\lambda = 0.01$ , while considering different values of  $\sigma$ . The parameter  $\alpha$  is adapted during the computation: starting form  $\alpha_0 = 10$ , it increases according to the law

$$\alpha = \alpha_0 \cdot k \cdot \log_2(k) \,. \tag{3.1}$$



Figure 1: Optimization on benchmark functions using CBO-ME. Behavior of the expectation error and fitness value for different values of  $\sigma$ . Here  $\lambda = 0.01$  and  $\alpha$  is adaptive, with  $\alpha_0 = 10$ . The particle population is N = 200. Gray bands (of values [0.70, 1.05] for error and [0.65, 1] for fitness) show the range in which the minima of the different benchmark functions fall. The dotted line marks the visually estimated pseudo-optimal value  $\sigma = 0.8$ . Results are averaged on 250 runs and obtained with  $k_{\text{max}} = 10^4$  iterations, without stopping criterion.

Fig. 1 shows the accuracy and the objective value reached for  $\sigma \in [0, 2]$  after  $k_{\text{max}} = 10^4$  algorithm iterations with N = 200 particles, and without random selection. The optimal value for  $\sigma$  is clearly problem-dependent, but we note that the optimal values for the problems considered all fall within a relative small range (underlined in gray in Fig. 1).

From Fig. 1 we infer that a good value for all benchmark problems considered is given by  $\sigma = 0.8$ . Using this value, we now compare CBO-ME, with plain CBO and the standard PSO (with and without alignment towards personal best) for different population sizes N = 50, 100, 200. We keep the random selection mechanism off by setting  $\mu = 0$  and use the same previously chosen parameters when memory effects are used. For plain CBO, without memory effects, we set  $\sigma = \sqrt{2}/2 \approx 0.71$  and same  $\lambda$ . Concerning PSO, we use the solver provided by the MATLAB Global Optimisation Toolbox (particleswarm), changing the maximum number of iterations and the stall condition to the one used for CBO methods, to make the results comparable. The remaining parameters are kept as described in the relative documentation [35]. We set  $k_{\text{max}} = 10^4$ ,  $\delta_{\text{stall}} = 10^{-4}$  and consider a run successful when either

$$\|\overline{y}^{\alpha,k} - x^*\|_{\infty} < 0.1 \quad \text{or} \quad |\mathcal{F}(\overline{y}^{\alpha,k}) - \mathcal{F}(x^*)| < 0.01.$$

$$(3.2)$$

Table 2 reports success rate, final error given by  $\|\overline{y}^{\alpha,k} - x^*\|_{\infty}$ , objective function value  $\mathcal{F}(\overline{y}^{\alpha,k})$  and total number of iterations, averaged over 250 runs. In addition to the classic PSO method, where the acceleration coefficients are chosen to be equal  $C_1 = C_2 = 1.49$ , Table 2 also shows the results when only the alignment towards global best is considered in PSO ( $C_1 = 0$ ).

While CBO already manages to find the global minimizer in most of the problems considered, we note that it sometimes fails when Rastrigin, Rosenbrock or XSY random functions are optimized. CBO-ME outperforms CBO in these objective functions. Standard PSO in many cases fails to solve the problem, see e.g. Rastrigin, Salomon or XSY 4 functions. PSO success rate is also lower among all problems, with the exception of the Schwefel 2.20 benchmark problem. Considering only global adjustment seems to show advantages except in the case of Ackley where setting  $C_1 = 0$  decreases the success rate or, in the case of XSY 4, Salomon or Rastrigin, where convergence is not achieved even for  $C_1 = 0$ . Consensus methods, however, perform better in terms of both success rate and speed up. In addition, for most problems, the population size N seems not to play a significant role in the algorithms performance. This further motivates the introduction of the random selection strategy described in the Section 2.1 in order to save computational costs.

In the third experiment, we test the proposed random selection mechanism (2.5) for different values of the parameter  $\mu$ . We recall that with  $\mu = 0$  we have no particles removal, while as  $\mu$  increases, more particles are likely to be discarded when the system variance decreases. The initial population is set to  $N_0 = 200$ , while the minimum number of particles to  $N_{\min} = 10$ . Results are reported in Tables 3 and 4 in terms of: success rate, error, objective value, weighted number of iterations, given by

$$w_{\text{iter}} = \sum_{k=0}^{\kappa_{\text{end}}} \frac{N_k}{N_0} \,, \tag{3.3}$$

and percentage of Computational Time Saved (CTS). Results show that relative large values of  $\mu$  allow to reach fast convergence without affecting the algorithm performance. In our experiments, the Rastrigin problem allows for larger values of  $\mu$ , while the Rosenbrock one seems to be more sensitive to the selection mechanism with respect to the other objectives. This justifies the different values of  $\mu$  considered in Tables 3 and 4. In both cases, a suitable value of  $\mu$  reduces the computational time with almost no impact in terms of accuracy.

Fig.s 2 and 3 show error and fitness value as a function of the number of fitness evaluations during the algorithm computation for Ackley and Rastrigin problems, respectively. Several values of  $\mu$  are considered to display how the random selection mechanism affects the convergence speed. Initial particle population is set to  $N_0 = 10^4$  and particles evolve for  $k_{\text{max}} = 10^4$  iterations. We note how convergence speed increases as  $\mu$  increases.

	CBO $(\sigma = \sqrt{2}/2)$			CBO-ME ( $\sigma = 0.8$ )		PSO			PSO $(C_1 = 0)$				
		N = 50	N = 100	N = 200	N = 50	N = 100	N = 200	N = 50	N = 100	N = 200	N = 50	N = 100	N = 200
Ackley	Rate	99.8%	100.0%	100.0%	100.0%	100.0%	100.0%	17.1%	41.2%	54.3%	4.2%	16.2%	40.1%
	Error	4.22e-06	2.14e-06	3.55e-06	2.42e-06	1.89e-06	1.56e-06	6.17e-09	8.86e-11	2.01e-12	2.23e-08	1.80e-10	8.65e-13
	F	1.18e-04	5.81e-05	7.30e-05	1.54e-04	4.96e-05	4.99e-05	6.24e-09	7.65e-11	1.94e-12	2.06e-08	1.70e-10	8.01e-13
	Iterations	912.3	718.1	623.2	977.2	703.3	622.2	501.8	424.2	341.3	502.3	421.2	321.2
Griewank	Rate	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	46.0%	48.6%	55.3%	50.0%	58.7%	78.0%
	Error	2.20e-02	2.21e-02	2.24e-02	2.13e-02	2.16e-02	2.25e-02	7.34e-02	1.56e-02	9.45e-03	1.17e-01	1.10e-01	8.96e-02
	F	5.26e-02	5.31e-02	5.47e-02	4.95e-02	5.15e-02	5.82e-02	3.23e-03	4.11e-03	3.78-03	3.73e-03	3.71e-03	2.90e-03
	Iterations	922.1	723.3	633.2	911.2	778.3	635.4	512.2	403.1	399.2	432.1	391.2	311.2
Rastrigin	Rate	12.1%	34.3%	62.7%	23.2%	69.7%	89.1%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	Error	1.28e-04	1.83e-04	2.34e-04	9.73e-05	1.27e-04	1.76e-04	-	-	-	-	-	-
	F	4.51e-06	9.03e-06	1.46e-05	2.54e-06	4.31e-06	8.28e-06	-	-	-	-	-	-
	Iterations	1083.0	933.7	819.8	1007.6	922.5	769.9	10000.0	10000.0	10000.0	10000.0	10000.0	10000.0
Rosenbrock	Rate	65.3%	86.7%	100.0%	70.1%	94.2%	100.0%	9.3%	22.6%	36.6%	46.7%	60.7%	76.7%
	Error	1.84e-02	2.43e-02	1.42e-02	3.60e-02	4.01e-02	1.82e-02	6.19e-04	2.56e-04	1.67e-04	4.44e-02	4.45e-02	4.46e-02
	F	6.13e-03	7.57e-03	2.40e-03	1.26e-02	1.42e-02	2.65e-03	3.80e-02	3.76e-02	2.56e-02	2.56e-03	8.95e-04	3.71e-04
	Iterations	5773.2	5423.2	5233.1	5933.2	4956.2	4155.2	4822.2	3823.2	3026.3	5924.2	3834.1	2933.3
Schwefel 2.20	Rate	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%
	Error	5.79e-06	8.23e-07	2.44e-07	8.42e-06	1.03e-06	2.76e-07	8.34e-10	1.97e-12	4.58e-14	1.68e-07	3.41e-10	8.03e-14
	F	1.04e-03	2.15e-04	8.36e-05	1.50e-03	3.12e-04	9.37e-05	1.94e-09	6.36e-12	1.52e-13	2.44e-07	6.48e-10	2.46e-13
	Iterations	822.2	682.2	622.1	655.2	544.2	455.2	491.2	434.2	399.1	578.2	467.2	423.2
Salomon	Rate	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	Error	3.12e-02	2.14e-02	1.87e-02	5.28e-02	4.49e-02	3.91e-02	-	-	-	-	-	-
	F	3.14e-01	2.15e-01	1.88e-01	2.44e-01	1.86e-01	1.91e-01	-	-	-	-	-	-
	Iterations	10000.0	10000.0	10000.0	8872.2	9021.2	5356.5	10000.0	10000.0	10000.0	10000.0	10000.0	10000.0
XSY random	Rate	52.3%	81.7%	92.6%	100.0%	100.0%	100.0%	3.2%	17.1%	31.2%	100.0%	100.0%	100.0%
	Error	2.64e-02	1.62e-02	9.80e-03	3.06e-02	1.86e-02	1.15e-02	2.25e-01	9.56e-02	8.42e-02	6.23e-02	5.12e-02	2.34e-02
	F	6.95e-08	3.54e-08	2.13e-08	2.21e-06	4.85e-08	3.17e-08	3.35e-04	2.28e-04	1.34e-04	8.22e-04	4.11e-04	3.45e-04
	Iterations	10000.0	10000.0	10000.0	10000.0	10000.0	10000.0	10000.0	10000.0	10000.0	10000.0	10000.0	10000.0
XSY 4	Rate	27.2%	89.3%	100.0%	25.2%	91.2%	100.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	Error	8.10e-01	7.12e-01	7.89e-01	8.01e-01	7.55e-01	6.17e-01	-	-	-	-	-	-
	F	4.79e-07	3.78e-07	3.46e-07	1.58e-06	8.56e-07	5.43e-07	-	-	-	-	-	-
	Iterations	10000.0	10000.0	10000.0	9733.2	9531.1	8733.2	10000.0	10000.0	10000.0	10000.0	10000.0	10000.0

**Table 2:** Comparison between classical CBO, CBO-ME and standard PSO with and without alignment towards personal best on benchmark problems. The solver particleswarm available in the MATLAB Global Optimisation Toolbox was used for the results concerning the PSO method. Optimal choice of parameters, different for each method, are used for the CBO algorithms. Same stopping criterion and definition of success, see (3.2), were used. Performance metric considered: success rate (see (3.2)), error  $\|\overline{y}^{\alpha,k} - x^*\|_{\infty}$ , fitness value  $\mathcal{F}(\overline{y}^{\alpha,k})$  and number of iterations. Results are averaged over 250 runs.

		$\mu = 0$	$\mu = 0.05$	$\mu = 0.1$	$\mu = 0.2$
Ackley	Rate	100.0%	100.0%	100.0%	100.0%
	Error	1.89e-06	2.78e-06	6.12e-06	2.29e-05
	$\mathcal{F}$	9.21e-05	5.77e-05	2.12e-04	5.12e-04
	$w_{\mathrm{iter}}$	688.2	502.3	387.2	178.2
	CTS	-	31.3%	52.1~%	70.2%
Griewank	Rate	100.0%	100.0%	100.0%	100.0%
	Error	2.12e-02	2.13e-02	2.18e-02	2.21e-02
	$\mathcal{F}$	5.80e-02	5.12e-02	5.90e-02	5.21e-02
	$w_{\mathrm{iter}}$	634.2	400.1	202.3	191.2
	CTS	-	31.3%	58.0%	71.6%
Schwefel 2.20	Rate	100.0%	100.0%	100.0%	100.0%
	Error	2.16e-07	8.89e-07	8.12e-07	2.34e-08
	$\mathcal{F}$	9.11e-05	3.02e-05	1.23e-05	3.22e-05
	$w_{\text{iter}}$	465.2	360.1	320.2	191.1
	CTS	-	24.7%	33.2%	62.1%
Salomon	Rate	100.0%	100.0%	100.0%	100.0%
	Error	4.13e-02	3.37e-02	2.77e-02	1.69e-02
	$\mathcal{F}$	4.21e-01	4.22e-01	4.10e-01	3.67e-01
	$w_{\text{iter}}$	2455.1	1551.1	1242.3	892.3
	CTS	-	38.2%	50.2%	66.2%
XSY random	Rate	100.0%	100.0%	100.0%	100.0%
	Error	1.54e-02	8.34e-02	8.90e-02	9.23e-02
	$\mathcal{F}$	6.34e-07	2.05e-05	6.34e-05	2.33e-04
	$w_{\text{iter}}$	10000.0	2821.3	1921.7	1167.2
	CTS	-	70.2%	85.3%	89.7%
XSY 4	Rate	100.0%	100.0%	100.0%	100.0%
	Error	5.37e-01	3.90e-01	1.55e-01	1.67e-01
	$\mathcal{F}$	1.19e-05	6.23e-06	3.67e-06	3.99e-06
	$w_{\text{iter}}$	8945.1	3967.3	1923.4	1055.7
	CTS	-	50.2%	69.3%	85.6%

**Table 3:** CBO-ME algorithm with random selection of particles tested against different benchmark functions with different values of  $\mu$ , which regulates the random selection mechanism. The system is initialized with  $N_0 = 200$  particles and  $\sigma = 0.8$ . Performance metric considered: success rate (see (3.2)), error  $\|\overline{y}^{\alpha,k} - x^*\|_{\infty}$ , fitness value  $\mathcal{F}(\overline{y}^{\alpha,k})$ , weighted iteration (3.3), and Computational Time Saved (CTS). Results are averaged over 250 runs.

		$\mu = 0$	$\mu = 0.1$	$\mu = 0.2$	$\mu = 0.5$
Rastrigin	Rate	100.0%	100.0%	100.0%	100.0%
	Error	9.22e-05	7.76e-05	3.54e-05	1.34e-05
	$\mathcal{F}$	2.90e-06	2.99e-06	1.45e-06	1.12e-06
	$w_{\mathrm{iter}}$	1150.3	720.6	250.5	106.3
	CTS	-	39.2%	78.9%	92.3%
		$\mu = 0$	$\mu = 0.01$	$\mu = 0.02$	$\mu = 0.05$
Rosenbrock	Rate	100.0%	100.0%	99.4%	99.0%
	Error	2.12e-02	2.21e-02	1.78e-02	1.45e-02
	$\mathcal{F}$	4.22e-03	5.67e-03	4.12e-03	4.45e-03
	$w_{\text{iter}}$	3189.3	840.3	350.3	102.3
	CTS	-	75.3%	90.2%	92.4%

**Table 4:** CBO-ME algorithm with particle reduction tested against Rastrigin and Rosenbrock functions with an higher diffusion parameter  $\sigma = 1.1$  and for different values of  $\mu$ , which regulates the random selection mechanism. The system is initialized with  $N_0 = 200$  particles. Performance metric considered: success rate (see (3.2)), error ( $||\overline{y}^{\alpha,k} - x^*||_{\infty}$ ), fitness value  $\mathcal{F}(\overline{y}^{\alpha,k})$ , weighted iteration (3.3), and Computational Time Saved (CTS).



**Figure 2:** Optimization of Ackley function for different values of the random selection parameter  $\mu$ , where the initial particle population is  $N_0 = 10^4$ . We report error (on the left) and fitness values (on the right) as the number of function evaluations increases. Parameters are set as  $\lambda = 0.01$ ,  $\sigma = 0.8$ ,  $\alpha$  adaptive starting from  $\alpha_0 = 10$  and following the law  $\alpha = \alpha_0 \cdot k \cdot \log_2(k)$ . Results are averaged over 250 runs.



**Figure 3:** Optimization of Rastrigin function for different values of the random selection parameter  $\mu$  where the initial particle population is  $N_0 = 10^4$ . We report error (on the left) and fitness values (on the right) as the number of function evaluations increases. Parameters are set as  $\lambda = 0.01$ ,  $\sigma = 1.1$ ,  $\alpha$  adaptive starting from  $\alpha_0 = 10$  and following the law  $\alpha = \alpha_0 \cdot k \cdot \log_2(k)$ . Results are averaged over 250 runs.

### 3.2 Applications

In this section, we propose some applications of the proposed optimization algorithm. First we consider a image segmentation problem using multi-thresholding, then we use the CBO-ME to train a Neural Network (NN) architecture to approximate functions and perform image classification on the MNIST database of handwritten digits.

#### 3.2.1 Image segmentation

To perform image segmentation, we use a threshold detection technique, namely, the multidimensional Otsu algorithm [34,46] in order to compare the results to similar optimization algorithm, such as the Modified PSO in [45].

In the Otsu algorithm, every pixel of the image is assigned to one of the possible L grayscale values. We denote with  $\eta_i$  the number of pixel with gray level  $i, 1 \leq i \leq L$  and  $N_{pix} = \sum_{i=1}^{L} \eta_i$  the total number of pixels [34]. We consider an extension of Otsu's technique to the multidimensional case [46] to test capabilities of method. Assuming we want to optimize the choice of d thresholds, we require d + 1 classes of different gray-scales  $(C_0, \ldots, C_d)$  with relative probabilities of occurrence classes defined as

$$\omega_0(l_1) = \sum_{i=1}^{l_1} p_i , \quad \dots, \quad \omega_d(l_d) = \sum_{i=l_d+1}^{L} p_i, \quad p_i = \frac{\eta_i}{N_{pix}}$$

and classes mean levels

$$\mu_0(l_1) = \frac{\sum_{i=1}^{l_1} i p_i}{\omega_0}, \quad \dots, \quad \mu_d(l_d) = \frac{\sum_{i=l_{d+1}}^{L} i p_i}{\omega_d},$$

The optimal thresholds  $(\hat{l}_1, \ldots, \hat{l}_d)$  are those that satisfy  $\hat{l}_1 < \cdots < \hat{l}_d$  and maximise

$$f(l_1, \dots l_d) = \sum_{i=1}^d \omega_i(l_i) \mu_i^2(l_i) \,. \tag{3.4}$$

For the experiment, we chose d = 5 thresholds and compare the segmentation performed by Otsu's method, solved with both standard PSO and CBO-ME, with segmentation obtained by dividing the grayscale into d + 1 uniformly spaced intervals. For PSO, we use to the default parameters in the particleswarm function in the MATLAB Global Optimisation Toolbox, while for CBO-ME we used optimal parameters found in Section 3.1 and exploit the random selection technique to speed up the algorithm.

We report the results on two sample images, Fig.s 4 and 5. We fix  $k_{\text{max}} = 10^3$  and average results over 250 runs. As in [3], we evaluate multi-thresholding segmentation through the Peak Signal to Noise Ratio (*PSNR*) computed as:

$$PSNR = 20 \cdot \log_{10} \left(\frac{255}{RMSE}\right)$$

where RMSE is the Root Mean-Squared Error, defined as

$$RMSE = \sqrt{\frac{1}{N_{pix}} \sum_{i=1}^{N_{row}} \sum_{j=1}^{N_{col}} [I(i,j) - S(i,j)]^2}$$

where  $N_{pix} = N_{row} \cdot N_{col}$ , *I* is the original image and *S* is the associated segmented image. The higher the value of *PSNR* is, the greater the similarity between the clustered image and the original image is. From Fig.s 4,5, we note that the most accurate segmentation on details is obtained by the CBO-ME method. This is quantitatively confirmed by the *PSNR* values reported in Table 5.



(a) Original (b) Standard segmentation (c) Otsu seg. (PSO) (d) Otsu seg. (CBO-ME)

Figure 4: Image segmentation of darkhair woman image  $(256 \times 256 \text{ pixels})$  with standard segmentation and Otsu segmentation solved respectively by PSO (c) and by CBO-ME (d). Results are averaged over 250 runs, with an initial population of  $N_0 = 10^3$  particles.



**Figure 5:** Image segmentation of lake image  $(256 \times 256 \text{ pixels})$  with standard segmentation and Otsu segmentation solved respectively by PSO (c) and by CBO-ME (d). Results are averaged over 250 runs, with an initial population of  $N_0 = 10^3$  particles.

	cameraman	lake	lena	peppers	woman darkhair
Standard segmentation	22.83	21.72	24.35	27.24	25.33
Otsu segmentation (PSO)	34.62	32.33	38.19	38.03	37.14
Otsu segmentation (CBO-ME)	37.22	35.44	38.72	38.28	39.57

**Table 5:** PSNR values obtained for 5 sample images known in literature. We compared the Otsu segmentation solved by the proposed CBO-ME method with the classical PSO method and with equispaced thresholding segmentation. Experiments are performed with d = 5 thresholds.



**Figure 6:** Approximating smooth function  $u_1$  (3.7) using a network with n = 50 and m = 3. We initially use  $N_0 = 500$  particles and we set  $\lambda = 0.01$ ,  $\sigma = 0.8$ . Parameter  $\alpha$  is adaptive, starting from  $\alpha_0 = 10$ .

#### 3.2.2 Approximating functions with NN

In this section, we use the proposed CBO-ME algorithm to train a NN architecture into approximating a function  $u: I \to \mathbb{R}, I \subset \mathbb{R}$  with low regularity. As in [7], we use a fully-connected NN with *m* layers

$$f(x;\theta) = (L_m \circ \dots L_2 \circ L_1)(x) \tag{3.5}$$

where each layer is given by

$$L_i = \sigma(W^i x + b^i)$$

with  $\sigma(x) = 1/(1 + \exp(-x))$  being the component-wise sigmoid function. We use internal layers of dimension n, so  $W^1 \in \mathbb{R}^{n \times 1}, b^1 \in \mathbb{R}, W^m \in \mathbb{R}^{1 \times n}, b^m \in \mathbb{R}^d$  and  $W^i \in \mathbb{R}^{n \times n}$  for all  $i = 2, \ldots, m-1$ . In (3.5), all DNN parameters are collected in  $\theta = \{W^i, b^i\}_{i=1}^m$ .

As loss function which need to be minimized, we consider the  $L^2$ -norm between the target function u and its NN approximation  $f(\cdot; \theta)$ 

$$\mathcal{F}(\theta) := \|f(\cdot;\theta) - u\|_{L^2(I)}.$$
(3.6)

Again, similarly to [7], we test the method against the following two functions:

$$u_1(x) = \sin(2\pi x) + \sin(8\pi x^2) \tag{3.7}$$

$$u_2(x) = \begin{cases} 1 & \text{if } x < -\frac{7}{8}, -\frac{1}{8} < x < \frac{1}{8}, x > \frac{7}{8}, \\ -1 & \text{if } \frac{3}{8} < x < \frac{5}{8}, -\frac{5}{8} < x < -\frac{3}{8}, \\ 0 & \text{otherwise.} \end{cases}$$
(3.8)

We note that  $u_1$  is smooth, while  $u_2$  is discontinuous. Parameters of the CBO-ME algorithm have been set to  $\lambda = 0.01, \sigma = 0.8$ , as in the previous sections. Parameter  $\alpha$  is adapted during the computation as in (3.1) and random selection mechanism is used. We employ m = 3 layers with internal dimension n = 50. Results are displayed in Fig.s 6 and 7. We note that convergence is slower for the discontinuous step function  $u_2$ .



**Figure 7:** Approximating non-smooth  $u_2$  (3.8) function using a network with n = 50, m = 3. We initially use  $N_0 = 500$  particles and we set  $\lambda = 0.01$ ,  $\sigma = 0.8$ . Parameter  $\alpha$  is adaptive, starting from  $\alpha_0 = 10$ .

### 3.2.3 Application on MNIST dataset

We now employ the proposed algorithm to train a NN architecture to solve a image classification task. We will consider the MNIST dataset [28] composed of handwritten digits in gray scale with  $28 \times 28$  pixels. For better comparability with CBO methods without memory effects, we follow the experiment settings used in the literature [4, 6, 12, 39], which we summarize below.

We consider a 1-layer NN where input images  $x \in \mathbb{R}^{28 \times 28}$  are first vectorized  $x \mapsto \text{vec}(x) \in \mathbb{R}^{784}$  and then processed through a fully-connected layer with parameters  $\theta = \{W, b\}$ , where  $W \in \mathbb{R}^{10 \times 784}, b \in \mathbb{R}^{10}$ . That is, the network is given by

$$f^{\text{SNN}}(x;\theta) = \text{softmax}\left(\text{ReLU}\left(W\text{vec}(x)+b\right)\right),\tag{3.9}$$

where  $\operatorname{ReLU}(z) = \max\{z, 0\}$  (component-wise) and  $\operatorname{softmax}(z) = (e^{z_1}, \ldots, e^{z_n})/(\sum_i e^{z_i})$  are the commonly used activation functions. During the training, batch regularization is performed after ReLU is applied in order to speed up convergence. Given a training set  $\{(x^m, \ell^m)\}_{m=1}^M$ ,  $x_m \in \mathbb{R}^{28 \times 28}, \ell_m \in \{0, 1\}^{10}$  made of M image-label tuples, we train the model by minimizing the categorical cross-entropy loss

$$\mathcal{F}(\theta) = \frac{1}{M} \sum_{m=1}^{M} \left( -\sum_{i=1}^{10} \ell_i^m \log(f_i(x^m, \theta)) \right) \,. \tag{3.10}$$

The entire training set is made of 60,000 images, 6,000 per class, but we divide it in batches of size M = 120, and consider a different batch at each algorithm iteration to evaluate (3.10).

The initial population of  $N_0$  particles is sampled from the standard normal distribution  $\mathcal{N}(0, \mathbf{I}_d)$  and we employ the particle reduction strategy given by (2.5) with  $N_{\min} = 100$ . Differently from previous experiments, though, we compute the new number of particles  $N_{k+1}$  based on the variance of the personal bests  $\{y_i^k\}_{i\in N_k}$ , rather then considering the particle positions  $\{x_i^k\}_{i\in N_k}$ . This is because, in this application, the variance of the particle positions shows an oscillatory behavior (see Fig. 9).

Following the mini-batch approach suggested in [6], at each algorithm iteration we divide the particle population in mini-batches of size  $n_N = 20$  (the last one being eventually smaller) and independently perform the update within the different mini-batches. Particles are re-ordered after each update step, so that the mini-batches always vary during the computation.

Fig. 8a, shows the algorithm performance when  $N_0 = 1000$  particles are initially used and random selection is performed with different parameters  $\mu$ . We note that, in terms of accuracy and loss, the performance is comparable. The random selection strategy sensibly reduces the number of function evaluations needed, especially when the particle system has already formed consensus. The number of particles per iteration is displayed in Fig. 9, further showing how the computational complexity of an update step decays during the computation.

We also compare the algorithm performance when different initial population sizes  $N_0 = 500, 1000, 2000$  are considered, with same random selection strength  $\mu = 0.1$ . In this case, the best balance between computational cost and accuracy is given by  $N_0 = 1000$ . We note in particular how starting with a larger population size of 2000 particles leads to a marginal improvement of the algorithm performance, while requiring a much higher number of loss evaluations.

In the last experiment, we compare algorithms CBO-ME and CBO without memory effects. We also consider a third heuristic proposed in [6] and further tested in [12]. In this CBO variant, no memory is employed, but, whenever  $\|\overline{x}^{\alpha,k+1} - \overline{x}^{\alpha,k}\|_{\infty} \leq \delta$ , particles are randomly perturbed before the CBO iteration, by adding Gaussian noise:

$$x_i^k \leftarrow x_i^k + \sigma \tilde{\theta}_i^k \quad \text{with} \quad \tilde{\theta}_i^k \sim \mathcal{N}(0, \mathbf{I}_d) \,.$$

$$(3.11)$$

In our experiment, we also consider the above variant with  $\delta = 10^{-5}$  and  $\overline{x}^{\alpha,k}$  computed among the whole particle system.

Fig. 10 illustrates the performance of the different algorithms for three different choices of parameter  $\alpha$ : increasing, fixed to  $\alpha = 50$  and fixed to  $\alpha = 5 \cdot 10^4$ . The drift parameter is set to  $\lambda = 0.1$  while we set  $\sigma = \sqrt{0.1}$  for CBO-ME and CBO without random perturbations and  $\sigma = \sqrt{0.04}$  for CBO when random noise is added. Initial populations of  $N_0 = 1000$  with selection parameter  $\mu = 0.2$  are used when there is no additional noise, while we use  $N_0 = 250$  and no particle selection when we perform random perturbations. This is motivated by the fact that the particle variance, whenever noise is added, shows an oscillatory behavior which is not compatible with the mechanism of random selection.

We note that CBO-ME performs better when lower values of  $\alpha$  are used, while the effect of memory is reduced for larger values of  $\alpha$ . A low value of sigma, together with random perturbations, typically slow down the convergence of the particle system.

- **Remark 3.1.** In CBO literature, an additional parameter  $\Delta t$  is typically used to define step size  $\lambda = \tilde{\lambda} \Delta t$  and the diffusion strength  $\sigma = \tilde{\sigma} \sqrt{\Delta t}$ , for some  $\tilde{\lambda}, \tilde{\sigma}$ . This is because the particles update rule is interpreted as a numerical scheme solving a time-continuous dynamics, as we will see in the next Section. We decided here to avoid using  $\Delta t$  for better comparability with PSO algorithms. We note how choosing, for instance,  $\lambda = 0.1, \sigma = \sqrt{0.1}$ is equivalent to the parameters choice  $\tilde{\lambda} = 1, \tilde{\sigma} = 1$  with  $\Delta t = 0.1$ .
  - Experiments show how both CBO and CBO-ME converges towards a solution in less than an epoch. This is coherent with other population-based algorithms, such as Ensemble Kalman Filter [52]. Moreover, we note how adding noise during the computation sensibly reduces the convergence speed, see Fig. 10.



(a) Different random selection parameters  $\mu$ , same initial population size  $N_0 = 1000$ 



(b) Different initial population sizes  $N_0$ , same random selection parameter  $\mu = 0.1$ 

**Figure 8:** Performance of CBO-ME algorithm in training a shallow NN for MNIST classification. Experiment with different combinations of random selection parameter  $\mu$  and initial population sizes are considered. Plots on the left display accuracy as a function of the amount of training data considered. On the right, the loss is displayed as a function of the number of loss evaluations. Clearly, when less particles are employed (either due to large  $\mu$  or to small  $N_0$ ), fewer loss evaluations are needed. The average number of particles is denoted by  $N_{avg}$ . Algorithm parameters are set to  $\lambda = 0.1, \sigma = \sqrt{0.1}, \alpha = 5 \cdot 10^4$ 



**Figure 9:** Population statistics during the training of shallow NN with CBO-ME method. Experiments with different random selection parameters  $\mu$  and initial population sizes  $N_0 = 1000$ . Algorithm parameters are set to  $\lambda = 0.1, \sigma = \sqrt{0.1}, \alpha = 5 \cdot 10^4$ 



**Figure 10:** Performance comparison between different consensus algorithms in training of shallow NN. We consider the proposed CBO-ME algorithm, the plain CBO algorithm and the CBO algorithm with random perturbation (3.11), as proposed in [6]. For the first two algorithms we set  $\sigma = \sqrt{0.1}$ , while for the third one  $\sigma = \sqrt{0.04}$ . We set  $\lambda = 0.1$  and consider three different strategies for  $\alpha$ :  $\alpha_k = 5 \cdot k$ ,  $\alpha_k = 50$  and  $\alpha_k = 5 \cdot 10^4$ 

### 4 Theoretical analysis

A strength of CBO algorithms lays on the possibility of theoretically analyze the particle system by relying on a mean-field approximation of the dynamics. We will illustrate in this section how to formally derive such approximation and present the main theoretical result regarding the convergence of the mean-field particle system towards a solution to (2.1), in case of no selection mechanism. Next, we will study the impact of the random selection strategy on the convergence properties of the algorithm. Technical details are left to Appendix A.

### 4.1 Mean-field approximation

First, we note that a simple update rule for the personal bests  $y_i^k$  is given by

$$y_i^{k+1} = y_i^k + \frac{1}{2} \left( x_i^{k+1} - y_i^k \right) S(x_i^{k+1}, y_i^k), \quad \text{with} \quad S(x, y) = 1 + \text{sign} \left( \mathcal{F}(y) - \mathcal{F}(x) \right).$$
(4.1)

As in [16], we approximate it for  $\beta \gg 1$  as

$$y_i^{k+1} = y_i^k + \frac{\nu}{2} \left( x_i^{k+1} - y_i^k \right) S^\beta(x_i^{k+1}, y_i^k) , \qquad (4.2)$$

with  $S^{\beta}(x, y)$  being a continuous approximation of S(x, y) as  $\beta \to \infty$ . By choosing  $\nu = 1$  we get (4.1) with the only difference of having  $S^{\beta}$  instead of S. As for  $\overline{y}^{\alpha,k}$  with respect to  $\overline{y}^{\infty,k}$ , this is needed to make the update rule easier to handle mathematically, but it does have an impact on the performance for large values of  $\beta$ . We note that alternative ways of modeling the memory mechanisms have been suggested in the literature of PSO, see, for instance, [1] where fractional order calculus is used.

With the aim of deriving a continuous-in-time reformulation of the particle update rules (2.3) and (4.2), we introduce a single parameter  $\Delta t > 0$  which controls the step length of all involved update mechanisms. By performing the rescaling

$$\lambda \leftarrow \lambda \Delta t \,, \quad \sigma \leftarrow \sigma \sqrt{\Delta t} \,, \quad \nu \leftarrow \nu \Delta t$$

we get the update rules

$$\begin{cases} x_i^{k+1} &= x_i^k + \lambda \Delta t \left( \overline{y}^{\alpha,k} - x_i^k \right) + \sigma \sqrt{\Delta t} \left( \overline{y}^{\alpha,k} - x_i^k \right) \otimes \theta_i^k \\ y_i^{k+1} &= y_i^k + (\nu \Delta t/2) \left( x_i^{k+1} - y_i^k \right) S^\beta(x_i^{k+1}, y_i^k) \end{cases}$$
(4.3)

which differ from the original formulation (2.3), (4.1) only due to the use of  $S^{\beta}$  instead of S.

As already noted in [16], the iterative process (4.3) corresponds to an Euler-Maruyama scheme applied to a system of Stochastic Differential Equations (SDEs). Indeed, (4.3) corresponds to a discretization of the system

$$\begin{cases} dX_t^i = \lambda \left( \overline{y}^{\alpha}(\overline{\rho}_t^N) - X_t^i \right) dt + \sigma \left( \overline{y}^{\alpha}(\overline{\rho}_t^N) - X_t^i \right) \otimes dB_t^i \\ dY_t^i = \nu (X_t^i - Y_t^i) S^{\beta}(X_t^i, Y_t^i) dt \end{cases}$$

$$\tag{4.4}$$

where, for convenience, we underlined above the dependence of the consensus point on the empirical distribution  $\overline{\rho}_t^N = \sum_i \delta_{Y_t^i}$  ( $\delta_y$  being the Dirac measure at  $y \in \mathbb{R}^d$ ) by using

$$\overline{y}^{\alpha}(\rho) := \frac{\int y e^{-\alpha \mathcal{F}(y)} d\rho(y)}{\int e^{-\alpha \mathcal{F}(y)} d\rho(y)},$$
(4.5)

defined for any Borel probability measure  $\rho$  over  $\mathbb{R}^d$  ( $\rho \in \mathcal{P}(\mathbb{R}^d)$ ). In this way, we generalized the definition introduced in (2.2) to any  $\rho \in \mathcal{P}(\mathbb{R}^d)$ , provided the above integrals exists. In (4.4), the random component of the dynamics is now described by N independent Wiener processes  $(B_t^i)_{t>0}$ . As before, we supplement the system with initial conditions  $X_0^i \sim \rho_0, Y_0^i = X_0^i$  for some  $\rho_0 \in \mathcal{P}(\mathbb{R}^d)$ .

The continuous-in-time description (4.4) already simplifies the analytical analysis of the optimization algorithm, but still pays the price of a possible large number  $\mathcal{O}(N)$  of equations. This issue is typically addressed by assuming that for large populations N, particles become indistinguishable from one another and start behaving, in some sense, as a unique system. More precisely, let  $F^N(t) \in \mathcal{P}(\mathbb{R}^{(2d)N})$  denote the joint probability distribution of N tuples  $(X_t^i, Y_t^i)$ . We assume propagation of chaos [43] for large  $N \gg 1$ , that is, we assume that the joint probability distribution decomposes as  $F^N(t) = f(t)^{\otimes N}$  for some  $f(t) \in \mathcal{P}(\mathbb{R}^{2d})$ . System (4.4) becomes independent on the index i and hence every particle moves according to the mono-particle process

$$\begin{cases} d\overline{X}_t = \lambda(\overline{y}^{\alpha}(\overline{\rho}_t) - \overline{X}_t) \, dt + \sigma \left(\overline{y}^{\alpha}(\overline{\rho}_t) - \overline{X}_t\right) \otimes d\overline{B}_t \\ d\overline{Y}_t = \nu(\overline{X}_t - \overline{Y}_t) S^{\beta}(\overline{X}_t, \overline{Y}_t) \, dt \end{cases}$$
(4.6)

where  $\overline{\rho}_t = \text{Law}(\overline{Y}_t)$ .

Assume  $(\overline{X}_t, \overline{Y}_t)$  are initially distributed according to  $f_0 = \rho_0^{\otimes 2}$ ; By applying Itô's formula we have that  $f(t) = \text{Law}(\overline{X}_t^i, \overline{Y}_t^i)$  satisfies in a weak sense

$$\partial_t f + \nabla_x \cdot \left(\lambda(\overline{y}^{\alpha}(\overline{\rho}) - x)f\right) + \nabla_y \cdot \left(\nu(x - y)S^{\beta}(x, y)f\right) = \frac{1}{2} \sum_{\ell=1}^{d} \partial_{x_{\ell}x_{\ell}}^2 \left(\sigma^2(\overline{y}^{\alpha}(\overline{\rho}) - x)_{\ell}^2 f\right) \quad (4.7)$$

with initial data  $\lim_{t\to 0} f(t) = f_0$ . Dynamics (4.6), or, equivalently, (4.7), corresponds to the mean-field approximation of the particle system (4.4) as  $N \to \infty$ . We remark that the above derivation has only been possible thanks to the approximations  $S \approx S^{\beta}$  and  $\overline{y}^{\infty} \approx \overline{y}^{\alpha}$  for large  $\alpha$  and  $\beta$ . Well-posedness of the system is also granted by such approximations, provided the objective function  $\mathcal{F}$  satisfies the following assumptions (proof details are given in Appendix A.2).

**Assumption 4.1.** The objective function  $\mathcal{F} : \mathbb{R}^d \to \mathbb{R}$  is bounded from below,  $\inf \mathcal{F} > -\infty$ , and there exist some constants  $L_{\mathcal{F}}, c_u, c_l, R_l > 0$  such that

$$|\mathcal{F}(x) - \mathcal{F}(x')| \le L_{\mathcal{F}} (||x||_2 + ||x'||_2) ||x - x'||_2$$
 for all  $x, x' \in \mathbb{R}^d$ ,

and

$$\begin{aligned} \mathcal{F}(x) &-\inf \mathcal{F} \le c_u (1 + \|x\|_2^2) & \text{for all} \quad x \in \mathbb{R}^d ,\\ \mathcal{F}(x) &-\inf \mathcal{F} \ge c_l \|x\|_2^2 & \text{for all} \quad x \in \mathbb{R}^d \|x\|_2 \ge R_l. \end{aligned}$$

**Proposition 4.1** (Existence of solution to (4.6)). Assume  $\mathcal{F}$  satisfies Assumption 4.1. There exists a process  $(\overline{X}, \overline{Y}) \in C([0, T], \mathbb{R}^d)$ , T > 0 satisfying (4.4) with initial conditions  $(\overline{X}_0, \overline{Y}_0)$  where  $\overline{X}_0 \sim \rho_0 \in \mathcal{P}_4(\mathbb{R}^d)$  and  $\overline{Y}_0 = \overline{X}_0$ .

Being mathematically tractable, we show next that the mean-field dynamics converges to a global solution to (2.1) if  $\mathcal{F}, S^{\beta}$  satisfy suitable assumptions.

### 4.2 Convergence in mean-field law

We start by enunciating the necessary assumptions to the convergence result.

Assumption 4.2. The objective function  $\mathcal{F} \in C(\mathbb{R}^d, \mathbb{R})$ , satisfies:

- A1 there exists uniquely  $x^* \in \mathbb{R}^d$  solution to (2.1);
- A2 there exist  $\eta, R_0 > 0$  and  $\gamma \in (0, \infty)$  such that

$$\mathcal{F}(x) - \inf \mathcal{F} \ge \eta \, \|x - x^*\|_{\infty}^{\gamma} \qquad \qquad \forall x \in \mathbb{R}^d, \, \|x - x^*\|_{\infty} \le R_0 \\ \mathcal{F}(x) - \inf \mathcal{F} \ge \eta \, R_0^{\gamma} \qquad \qquad \forall x \in \mathbb{R}^d, \, \|x - x^*\|_{\infty} > R_0$$

A3  $\mathcal{F}$  is convex in a (possibly small) neighborhood  $\{x \in \mathbb{R}^d : ||x - x^*||_{\infty} \leq R_1\}$  of  $x^*$  for some  $R_1 < R_0$ .

Assumption 4.3 (Assumptions on  $S^{\beta}$ ). The function  $S^{\beta} \in C(\mathbb{R}^{2d}, [0, 2])$ , with  $\beta > 0$ 

A4 has the following structure

$$S^{\beta}(x,y) = 2\psi \left(\beta(\mathcal{F}(y) - \mathcal{F}(x))\right), \qquad (4.8)$$

with  $\psi \in C^1(\mathbb{R}, [0, 1])$  being a non-decreasing function with Lipschitz constant  $L_{\psi} = 1$ .

A5 The value  $S^{\beta}(x, y)$  is positive only when x is strictly better than y in terms of objective value  $\mathcal{F}$ :

$$S^{\beta}(x,y) \begin{cases} \geq 0 & \text{if } \mathcal{F}(x) < \mathcal{F}(y) \\ = 0 & \text{else.} \end{cases}$$

Assuming uniqueness of global minimum is a typical assumption for analysis of CBO methods [11,12] and it is due to the definition of the consensus point  $\overline{y}^{\alpha}$  (or  $\overline{x}^{\alpha}$  in the case without memory mechanism). Indeed, in presence of two global minima,  $\overline{y}^{\alpha}$  may be placed between them, no matter how large  $\alpha$  is. Assumption A1 ensure to avoid such situations. Furthermore, A2 also allows to give quantitative estimates on the difference between the global minimum and eventual local minima. In the literature, such property is known as *conditioning* [14]. Requirements A3 and A5 will be needed to ensure that if a personal best  $y_i^k$  enters such small neighborhood where  $\mathcal{F}$  is convex, it will not leave it for the rest of the computation. For an intuition of A2 and A3 we refer to Figure 11, where the Rastrigin function is considered.



**Figure 11:** Assumptions 4.2 illustrated for Rastrigin function. For example, such objective function satisfies A2 with  $\eta = 1$ ,  $\gamma = 1.8$ ,  $R_0 = 1.42$  and A3 with  $R_1 = 0.25$ .

**Theorem 4.1** (Convergence in mean-field law). Assume  $\mathcal{F}$  satisfies Assumption 4.2 and  $S^{\beta}$  satisfies Assumption 4.3 for some  $\beta > 0$  fixed. Let  $(\overline{X}_t, \overline{Y}_t)_{t\geq 0}$  be a solution to (4.6) for  $t \in [0, T^*]$ , with initial data  $\overline{X}_0 \sim \rho_0 \in \mathcal{P}_4(\mathbb{R}^d), \overline{Y}_0 = \overline{X}_0$  such that  $x^* \in \operatorname{supp}(\rho_0)$ .

Fix an accuracy  $\varepsilon > 0$ . If  $2\lambda > \sigma^2$ , the expected  $\ell_2$ -error satisfies

$$\min_{t \in [0,T^*]} \mathbb{E}\left[ \|\overline{X}_t - x^*\|_2^2 \right] \le \varepsilon$$
(4.9)

provided  $T^*, \alpha > 0$  are large enough.

We refer to Appendix A for a proof.

**Remark 4.1.** The mean-field mono-particle process (4.6) aims to approximate the algorithm iterative dynamics (4.3) for small time steps  $\Delta t \ll 1$  and large particle populations  $N \gg 1$ . Therefore, convergence of the algorithm dynamics towards the global solution  $x^*$  can be proven by coupling Theorem 4.1 with error estimates of such approximation.

For instance, assuming that all considered dynamics take place on a bounded set  $\mathcal{D}$  ensures that the error introduced by the continuous-in-time approximation will be of order  $\Delta t$  thanks to classical results on Euler-Maruyama schemes [37]. Likewise, considering a bounded dynamics allows to prove that the error introduced by the mean-field approximation is of order  $N^{-1}$  (see e.g. [10, Theorem 3.1], [11, Proposition 16]). If such error rate holds, consider  $\{(x_i^k, y_i^k)\}_{i=1}^N$  be given by (4.3),  $\{(X_t^i, Y_t^i)\}_{i=1}^N$  be a solution (4.4) and  $\{(\overline{X}_t^i, \overline{Y}_t^i)\}_{i=1}^N$  be N-copies of a solution to (4.6). Let  $t^* \in [0, T^*]$  being a time minimizing the mean-field error in (4.9). Altogether, one obtains the following error decomposition for  $k = |t^*/\Delta t|$ ,

$$\mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N}\|x_{i}^{k}-x^{*}\|_{2}^{2}\right] \leq C\left(\mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N}\|x_{i}^{k}-X_{t^{*}}^{i}\|_{2}^{2}\right] + \mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N}\|X_{t^{*}}^{i}-\overline{X}_{t^{*}}^{i}\|_{2}^{2}\right] + \mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N}\|\overline{X}_{t^{*}}^{i}-x^{*}\|_{2}^{2}\right]\right)$$
$$\leq C_{\mathrm{EM}}\Delta t + C_{\mathrm{MFA}}N^{-1} + \varepsilon$$

where  $C, C_{\rm EM}, C_{\rm MFA}$  are positive constant independent on  $N, \Delta t$ .

#### 4.3 Random selection analysis

In this section, we analytically investigate the impact of randomly discarding particles during the computation. We are particularly interested in tracking the distance between a particle system  $\{x_i^k, y_i^k\}_{i=1}^{N_0}$  evolving according to (4.3) where no particles are discarded, and a second system  $\{\hat{x}_i^k, \hat{y}_i^k\}_{i=1}^{N_0}$ ,  $|I_k| = N_k$  where  $N_k - N_{k+1}$  particles are discarded after update rule (4.3). Clearly, we require that  $N_{k+1} \leq N_k$  and  $I_{k+1} \subseteq I_k \subseteq I_0 = \{1, \ldots, N_0\}$  for all k. Similarly to the analysis carried out in [17, 18], we restrict to the simpler dynamics where, at every step k, the random variables  $\theta_i^k$  and  $\hat{\theta}_i^k$  used to generate such systems are the same for all particles:

$$\theta_i^k = \widehat{\theta}_j^k = \theta^k \sim \mathcal{N}(0, \mathbf{I}_d) \quad \text{for all} \quad i \in I_k, \ j \in I_0.$$
(4.10)

To compare particle systems with a different number of particles, we rely on their representation as empirical probability measures and the notion of 2-Wasserstein distance. For  $\{\hat{x}_i^k\}_{i\in I_k}$ and  $\{x_i^k\}_{i=1}^{N_0}$  we consider, respectively, the following probability measures

$$\rho_{N_k}^k := \frac{1}{N_k} \sum_{i \in I_k} \delta_{\widehat{x}_i^k} \quad \text{and} \quad \rho_{N_0}^k := \frac{1}{N_0} \sum_{i \in I_0} \delta_{x_i^k} \,. \tag{4.11}$$

Informally, the 2-Wasserstein distance  $W_2(\rho_{N_k}^k, \rho_{N_0}^k)$  quantifies the minimal effort needed to move the mass from distribution  $\rho_{N_k}^k$  into  $\rho_{N_0}^k$  [40]. Let  $w_{ij}$  denote the amount of mass leaving particle  $x_i^k$  and going into  $\hat{x}_i^k$ : the cost of such movement is assumed to be given by  $w_{ij} || x_i^k - \hat{x}_j^k ||_2^2$ . Therefore, if we indicate the set of all admissible couplings between the two discrete probability measures as

$$\Gamma(\rho_{N_k}^k, \rho_{N_0}^k) = \left\{ w \in \mathbb{R}^{N_0 \times N_k} : \sum_{j=1}^{N_k} w_{ij} = \frac{1}{N_0}, \sum_{i=1}^{N_0} w_{ij} = \frac{1}{N_k}, w_{ij} \ge 0, \forall i, j \right\},$$
(4.12)

the 2- Wasserstein distance is defined as

$$W_2(\rho_{N_k}^k, \rho_{N_0}^k) := \min_{w \in \Gamma(\rho_{N_k}^k, \rho_{N_0}^k)} \left( \sum_{i,j} w_{ij} \|x_i^k - \widehat{x}_j^k\|_2^2 \right)^{\frac{1}{2}}$$
(4.13)

1

see, for instance, [40, Section 6.4.1].

Before providing estimates on (4.12), let us present a more general result on the impact that the random selection strategy has on an arbitrary particle distribution.

**Proposition 4.2** (Stability of random selection procedure). Let  $\mathbf{z} = \{z_i\}_{i \in I}, |I| = N$  be an ensemble of particles and  $\{z_i\}_{j \in I_{sel}}$  with  $I_{sel} \subseteq I, |I| = N_{sel}$  a random sub-set of such ensemble. Consider the associated empirical distributions  $\mu_N$  and  $\mu_{N_{sel}}$  (defined consistently to (4.11)).

It holds

$$\mathbb{E}\left[W_2^2(\mu_N, \mu_{N_{\text{sel}}})\right] \le 2\operatorname{var}(\mathbf{z}) \,\frac{N - N_{\text{sel}}}{N - 1}\,,\tag{4.14}$$

where the expectation is taken with respect to the random selection of  $I_{sel}$ .

The proof is provided Appendix A.4. We note how the system variance  $var(\mathbf{z})$  enters the error estimate due to the randomness of the selection, similar to the Law of Large Number error for random variables. In particular, the smaller the particles variance is, the closer the reduced particle system will be to the original distribution. This justifies the choice of  $N_{k+1}$  proposed in Section 2.2 where we are allowed to discard particles only if the system shows a contractive behavior, see (2.5).

By iteratively applying Proposition 4.2 and by using suitable stability estimates of dynamics (4.3), we are able to bound the error introduced by the random selection procedure as follows. Proof details are a given in Appendix A.4.

**Theorem 4.2.** Let  $\{x_i^k, y_i^k\}_{i=1}^{N_0}$  be constructed according to (4.3) were particles are not discarded, and  $\{\hat{x}_i^k, \hat{y}_i^k\}_{I_k}, |I_k| = N_k$  where  $N_k - N_{k+1}$  particles are discarded after update rule (4.3). Assume (4.10) is satisfied and consider the probability measures (4.11).

Under Assumptions 4.1 and 4.3, if  $\{x_i^k, y_i^k\}_{i=1}^{N_0}, \{\widehat{x}_i^k, \widehat{y}_i^k\}_{i\in I_k} \subset B_M(0)$  at all step k for some M > 0, it holds

$$\mathbb{E}\left[W_2^2\left(\rho_{N_k}^k, \rho_{N_0}^k\right)\right] \le C \max_{h=1,\dots,k} \operatorname{var}\left(\tilde{\mathbf{z}}^h\right) \frac{N_0 - N_k}{N_k - 1}$$
(4.15)

where  $C = C(\Delta t, \lambda, \sigma, \nu, \beta, \alpha, k, L_F, M)$  and  $\tilde{\mathbf{z}}^h = \{(\hat{x}^h_i, \hat{y}^h_i)\}_{i \in I_{h-1}}$  describes the particle system just before the random selection procedure at step  $h \leq k$ . The expectation is taken with respect to the sampling of  $\{\theta^h\}_{h=1}^k$  and with respect to the selection procedure.

We can directly apply the above result to relate the expected  $\ell_2$ -errors of the two particle system, which we define as

$$\operatorname{Err}(k) := \mathbb{E}\left[\frac{1}{N_0} \sum_{i \in I_0} \|x_i^k - x^*\|_2^2\right], \quad \operatorname{Err}_{\operatorname{sel}}(k) := \mathbb{E}\left[\frac{1}{N_k} \sum_{i \in I_k} \|\widehat{x}_i^k - x^*\|_2^2\right],$$

that is, the discrete counterpart of the mean-field error  $\mathbb{E}[\|\overline{X}_t^i - x^*\|_2^2]$  studied in Theorem 4.1. By definition of the 2-Wasserstein distance, we have

$$\operatorname{Err}(k) = \mathbb{E}\left[W_2^2(\rho_{N_0}^k, \delta_{x^*})\right]$$

for any solution  $x^*$  to (2.1), and the same holds of  $\operatorname{Err}_{sel}(k)$ . We then apply inequality

$$W_2^2(\rho_{N_k}^k, \delta_{x^*}) \le 2\left(W_2^2(\rho_{N_k}^k, \rho_{N_0}^k) + W_2^2(\rho_{N_0}^k, \delta_{x^*})\right)$$

to obtain the following estimate.

**Corollary 4.1.** Under the assumptions of Theorem 4.2, at all steps k, it holds

$$\operatorname{Err}_{\operatorname{sel}}(k) \le 2\left(\operatorname{Err}(k) + C \max_{h=1,\dots,k} \operatorname{var}(\tilde{\mathbf{z}}^h) \frac{N_0 - N_k}{N_k - 1}\right).$$
(4.16)

Before concluding the section, let us report some remarks concerning the theoretical results just presented.

#### Remark 4.2.

- Proof of Theorem 4.2 can be adapted to any other particle system with random selection, provided that the update rule is stable with respect to the 2-Wasserstein distance. In the proposed method, such stability was proved thanks to the approximation of the global best  $\overline{y}^{\infty,k}$  with  $\overline{y}^{\alpha,k}$  for  $\alpha \gg 1$  (see (2.2)) and S(x,y) with  $S^{\beta}(x,y)$  for  $\beta \gg 1$  in the personal best update (4.2).
- Quantitative estimates on the variance decay can be used, if available, to improve the error bound in Theorem 4.2, see also proof in Appendix A.4.
- The error introduced by a sub-sampling technique in a Monte Carlo integral approximation is expected to be of order

$$2\operatorname{var}(\mathbf{z}) \left(\frac{1}{N-1} - \frac{1}{N_{\operatorname{sel}} - 1}\right) = 2\operatorname{var}(\mathbf{z}) \frac{N - N_{\operatorname{sel}}}{(N-1)(N_{\operatorname{sel}} - 1)}, \quad (4.17)$$

see e.g. [25]. Therefore, an additional factor of order  $1/(N_{sel} - 1)$  seems to be missing in Proposition 4.2. We remark, though, that Proposition 4.2 does not concern the Monte Carlo approximation of an integral quantity, but rather consider the 2-Wasserstein distance between empirical probability measures. Numerical simulations suggest that estimates of order (4.17) do not hold on in this case, see Fig.12.

### 5 Conclusions

In this work, we studied a Consensus-Based Optimization algorithm with Memory Effects (CBO-ME) and random selection for single objective optimization problems of the form (2.1). While sharing common features with Particle Swarm Optimization (PSO) methods, CBO-ME differs on the way the particle system explore the search space. Its structure provides greater flexibility in balancing the exploration and exploitation processes. In particular, we implemented and analytically investigates a random selection strategy which allows to reduce the algorithm computational complexity, without affecting convergence properties and overall accuracy. This analysis is entirely general and, in perspective, applicable to other particle-based optimization methods as well. The convergence analysis to the global minimum is carried out by relying on a mean-field approximation of the particle system and error estimates are given under mild assumptions on the objective function. We compared CBO-ME against CBO without memory effects and PSO against several benchmark problem and showed how the introduction of memory effects and random selection improves the algorithm performance. Applications to image segmentation and machine learning problems are also reported.



Figure 12: Numerical validation of Proposition 4.2 with different dimensions d = 3, 10. N = 100 points are randomly, uniformly sampled over  $[0,1]^d$  to construct the empirical distribution  $\mu_N$  and  $N_{sel} \in [2, N - 1]$  are discarded to obtain  $\mu_{N_{sel}}$ . The experiment is repeated 500 times for all  $N_{sel}$  to obtain an approximation of  $\mathbb{E}\left[W_2^2(\mu_N, \mu_{N_{sel}})\right]$  (blue line). In red, estimate provided by Proposition 4.2 (RHS of (4.14)), in yellow the one given equation (4.17). Wasserstein distances are computed with the ot.emd function provided by the Python Optimal Transport library [8].

### A Proofs

### A.1 Notation and auxiliary lemmas

We will use the following notation. For any  $a \in \mathbb{R}$ , |a| indicates the absolute value. For a given vector  $b \in \mathbb{R}^d$ ,  $||b||_p$  indicates its *p*-norm,  $p \in [1, \infty]$ ;  $(b)_\ell$  its  $\ell$ -th component; while diag $(b) \in \mathbb{R}^{d \times d}$  is the diagonal matrix with elements of *b* on the main diagonal. Let  $a, b \in \mathbb{R}^d$ ,  $\langle a, b \rangle$  denotes the scalar product in  $\mathbb{R}^d$ . For a given closed convex set  $A \subset \mathbb{R}^d$ ,  $\mathcal{N}(A, x)$ ,  $\mathcal{T}(A, x)$  denote the Clarke normal and the tangential cone at  $x \in A$  respectively. The  $\ell^p$ -ball,  $p \in [1, \infty]$ , of radius *r* centered at  $x \in \mathbb{R}^d$  is indicated with  $B_r^p(x) = \{x \in \mathbb{R}^d \mid ||x||_p \leq r\}$ . All considered stochastic processes are assumed to take their realizations over the common probability space  $(\Omega, \overline{\mathcal{F}}, \mathbb{P})$ .  $\mathcal{P}(\mathbb{R}^d)$  is the set of Borel probability measures over  $\mathbb{R}^d$  and  $\mathcal{P}_q(\mathbb{R}^d) = \{\mu \in \mathcal{P}(\mathbb{R}^d) \mid \int ||x||_2^q d\mu < \infty\}$  which we equip with the Wasserstein distance  $W_q$ ,  $q \geq 1$ , see [40]. For a random variable  $X, X \sim \mu$ ,  $\mu \in \mathcal{P}(\mathbb{R}^d)$  indicates a sampling procedure such that  $\mathbb{P}(X \in A) = \mu(A)$  for any Borel set  $A \subset \mathbb{R}^d$ . With  $\text{Unif}(A) \in \mathcal{P}(\mathbb{R}^d)$  we denote the uniform probability measure over a bounded Borel set A. Throughout the computations, C will denote an arbitrary positive constant, whose value may vary from line to line. Dependence on relevant parameters or variables, will be underlined.

**Lemma A.1** ( [5, Lemma 3.2]). Let  $\mathcal{F}$  satisfy Assumption 4.1 and  $\rho_1, \rho_2 \in \mathcal{P}_4(\mathbb{R}^d)$  with

$$\int \|x\|_2^4 d\rho_1 \,, \quad \int \|x\|_2^4 d\rho_2 \le M$$

Then, the following stability estimate holds

$$\|\overline{y}^{\alpha}(\rho_1) - \overline{y}^{\alpha}(\rho_2)\|_2 \le C W_2(\rho_1, \rho_2)$$

for a constant  $C = C(\alpha, L_{\mathcal{F}}, M)$ .

**Lemma A.2.** Under Assumptions 4.1 and 4.3, for any  $x_1, x_2, y_1, y_2 \in B^2_M(0)$ , it holds

$$\|(x_1 - y_1)S^{\beta}(x_1, y_1) - (x_2 - y_2)S^{\beta}(x_2, y_2)\|_2 \le C (\|x_1 - y_1\|_2 + \|x_2 - y_2\|_2)$$

where  $C = C(\beta, L_{\mathcal{F}}, M)$ .

*Proof.* We note that function  $S^{\beta}$  is locally Lipschitz continuous thanks to the locally Lipschitz continuity of  $\mathcal{F}$  (Assumption 4.1) and the Lipschitz continuity of  $\psi$  (Assumption 4.3):

$$|S^{\beta}(x_1, y_1) - S^{\beta}(x_2, y_2)| = |2\psi \left(\beta(\mathcal{F}(y_1) - \mathcal{F}(x_1)) - 2\psi(\beta(\mathcal{F}(y_2) - \mathcal{F}(x_2))\right)|$$
  

$$\leq 2\beta |\mathcal{F}(y_1) - \mathcal{F}(x_1) - \mathcal{F}(y_2) + \mathcal{F}(x_2)|$$
  

$$\leq 4\beta L_{\mathcal{F}} M \left( ||x_1 - x_2||_2 + ||y_1 - y_2||_2 \right).$$

Next, we have

$$\begin{aligned} \|(x_1 - y_1)S^{\beta}(x_1, y_1) - (x_2 - y_2)S^{\beta}(x_2, y_2)\|_2 \\ &\leq \|(x_1 - y_1)S^{\beta}(x_1, y_1) - (x_2 - y_2)S^{\beta}(x_1, y_1)\|_2 \\ &+ \|(x_2 - y_2)S^{\beta}(x_1, y_1) - (x_2 - y_2)S^{\beta}(x_2, y_2)\|_2 \\ &\leq \|(x_1 - x_2 + y_2 - y_1)S^{\beta}(x_1, y_1)\|_2 + \|(x_2 - y_2)\left(S^{\beta}(x_1, y_1) - S^{\beta}(x_2, y_2)\right)\|_2 \\ &\leq 2\left(\|x_1 - x_2\|_2 + \|y_1 - y_2\|_2\right) + 2M|S^{\beta}(x_1, y_1) - S^{\beta}(x_2, y_2)| \\ &\leq C\left(\|x_1 - x_2\|_2 + \|y_1 - y_2\|_2\right) \end{aligned}$$

with  $C = C(\beta, L_F, M)$ , where we used the proved locally Lipischitz continuity of  $S^{\beta}$  in the last inequality.

### A.2 Proof of Proposition 4.1

*Proof of Proposition 4.1.* The proof is based on the Leray–Schauder fixed point theorem [15, Chapter 11], and we follow the proof of [5, Theorem 3.2].

**Step 1.** For any  $\xi \in C([0,T], \mathbb{R}^d)$  there exists a unique process  $(\widehat{X}_t, \widehat{Y}_t) \in C([0,T], \mathbb{R}^d)$  satisfying

$$d\widehat{X}_t = \lambda(\xi(t) - \widehat{X}_t) dt + \sigma(\xi(t) - \widehat{X}_t) \otimes d\widehat{B}_t$$
$$d\widehat{Y}_t = \nu(\widehat{X}_t - \widehat{Y}_t) S^\beta(\widehat{X}_t, \widehat{Y}_t) dt$$

with  $\operatorname{Law}(\widehat{X}_0) = \operatorname{Law}(\widehat{Y}_0) = \rho_0 \in \mathcal{P}(\mathbb{R}^d)$ , by locally Lipschitz continuity and linear growth of the coefficients (thanks to Lemma A.2). As a consequence, we also have that  $f(t) := \operatorname{Law}(\widehat{X}_t, \widehat{Y}_t)$  satisfies

$$\frac{d}{dt}\int\phi\,df(t) = \int\left(-\lambda\langle\nabla_x\phi,\xi(t)-x\rangle + \frac{1}{2}\sigma\sum_{\ell=1}^{d}\frac{\partial^2\phi}{\partial x_{\ell}^2}(\xi(t)-y)_{\ell}^2 - \nu S^\beta\langle\nabla_y\phi,y-x\rangle\right)df(t)$$

for all  $\phi \in C_b^2(\mathbb{R}^{2d})$  by applying Itô's formula. Therefore, let  $\overline{\rho}(t) = \text{Law}(\widehat{Y}_t)$ , we can set  $\mathcal{T}\xi := \overline{y}^{\alpha}(\overline{\rho}(\cdot)) \in C([0,T], \mathbb{R}^d)$  to define

$$\mathcal{T}: C([0,T],\mathbb{R}^{\mathrm{d}}) \to C([0,T],\mathbb{R}^{\mathrm{d}}).$$

**Step 2.** We prove now compactness of  $\mathcal{T}$ . Thanks to  $\rho_0 \in \mathcal{P}_4(\mathbb{R}^d)$  and standard results for SDEs (see [2, Chapter 7]) we have boundedness of the forth moments

$$\mathbb{E}\left[\|\widehat{X}_t\|_2^4 + \|\widehat{Y}_t\|_2^4\right] \le c_1 \left(1 + \mathbb{E}[\|\widehat{X}_0\|_2^4 + \|\widehat{Y}_0\|_2^4]e^{c_2t}\right)$$

for some  $c_1, c_2 > 0$ . Therefore, we can apply Lemma A.1 to obtain for any 0 < s < t < T,

$$\|\overline{y}^{\alpha}(\overline{\rho}(t)) - \overline{y}^{\alpha}(\overline{\rho}(s))\|_{2} \le CW_{2}\left(\overline{\rho}(t), \overline{\rho}(s)\right) \le \tilde{C}|t-s|^{1/2}$$

for some constants  $C, \tilde{C} > 0$ , from which Hölder continuity of  $t \mapsto \overline{y}^{\alpha}(\overline{\rho}(t))$  follows. Compactness of  $\mathcal{T}$  follows by

$$\mathcal{T}(C([0,T],\mathbb{R}^{\mathrm{d}})) \subset C^{0,\frac{1}{2}}([0,T],\mathbb{R}^{\mathrm{d}}) \hookrightarrow C([0,T],\mathbb{R}^{\mathrm{d}}).$$

Step 3. Consider  $\xi \in C([0,T], \mathbb{R}^d)$  satisfying  $\xi = \tau \mathcal{T}\xi$ , for  $\tau \in [0,1]$ . Thanks to [5, Lemma 3.3] and boundedness of second moments, we obtain compactness of the set

$$\{\xi \in C([0,T], \mathbb{R}^d) : \xi = \tau \mathcal{T}\xi, \tau \in [0,1]\}$$

and by Leray–Schauder fixed point theorem there exists a fixed point for the mapping  $\mathcal{T}$  and hence a solution to (4.6).

### A.3 Proof of Theorem 4.1

Having proved there exists a solution  $(\overline{X}_t, \overline{Y}_t)_{t \in [0,T]}$  to the mean-field process (4.6) we are here interested in studying the expected  $\ell_2$ -error given by

$$\mathbb{E}\|\overline{X}_t - x^*\|_2^2$$

where  $x^*$  is the unique solution to the minimization problem (2.1) (uniqueness is given by Assumption 4.2). We do so by means of the following quantitative version of the Laplace principle.

**Proposition A.1** (Quantitative Laplace principle [12, Proposition 1]). Let  $\rho \in \mathcal{P}(\mathbb{R}^d)$  be such that  $x^* \in \operatorname{supp}(\rho)$  and fix  $\alpha > 0$ . For any r > 0, define  $\mathcal{F}_r := \sup_{x \in B_r^{\infty}(x^*)} \mathcal{F}(x) - \mathcal{F}(x^*)$ .

Then, under Assumption 4.2, for any  $r \in (0, R_0]$  and q > 0 such that  $q + \mathcal{F}_r \leq \mathcal{F}_{\infty} := \eta R_0^{\gamma}$ , it holds

$$\|\overline{y}^{\alpha}(\rho) - x^*\|_2 \le \frac{\sqrt{\mathrm{d}}(q + \mathcal{F}_r)^{\gamma}}{\eta} + \frac{\sqrt{\mathrm{d}}\exp(-\alpha q)}{\rho(B_r^{\infty}(x^*))} \int \|x - x^*\|_2 \, d\rho(x). \tag{A.1}$$

We remark that RHS of (A.1) can be made arbitrary small by taking large values of  $\alpha$  and small values of q, r provided the integral is bounded.

**Lemma A.3.** Let  $(\overline{X}_t, \overline{Y}_t)_{t \in [0,T]}$  be a solution to (4.6) and initial data  $\overline{X}_0 = \overline{Y}_0$  and  $x^* \in \mathbb{R}^d$ . For any  $t \in [0,T]$ , it holds

$$\mathbb{E}[\|\overline{Y}_t - x^*\|_2^2] \le 2e^{\nu t} \sup_{s \in [0,t]} \mathbb{E}[\|\overline{X}_s - x^*\|_2^2].$$
(A.2)

*Proof.* Due to (4.6) and chain rule, it holds

$$\begin{split} \frac{d}{dt} \|\overline{Y}_t - x^*\|_2^2 &= 2\nu \langle \overline{Y}_t - x^*, \overline{X}_t - \overline{Y}_t \rangle S^{\beta}(\overline{X}_t, \overline{Y}_t) dt \\ &= 2\nu \langle \overline{Y}_t - x^*, \overline{X}_t - x^* \rangle S^{\beta}(\overline{X}_t, \overline{Y}_t) dt - 2\nu \|\overline{Y}_t - x^*\|_2^2 S^{\beta}(\overline{X}_t, \overline{Y}_t) dt \\ &\leq \nu \left( \|\overline{Y}_t - x^*\|_2^2 + \|\overline{X}_t - x^*\|_2^2 \right) dt \end{split}$$

By taking the expectation and applying Grönwall's inequality, we have

$$\mathbb{E}[\|\overline{Y}_t - x^*\|_2^2] \le \mathbb{E}[\|\overline{Y}_0 - x^*\|_2^2]e^{\nu t} + \int_0^t \mathbb{E}[\|\overline{X}_s - x^*\|_2^2]e^{\nu(t-s)}ds.$$

Estimate (A.2) can be obtained after noting that  $\mathbb{E}[\|\overline{Y}_0 - x^*\|_2^2] = \mathbb{E}[\|\overline{X}_0 - x^*\|_2^2]$  due to choice of the initial data, and by taking the supremum over all times  $s \in [0, t]$ .

To apply Proposition A.1 to all  $\overline{\rho}(t) = \text{Law}(\overline{Y}_t)$ , we need though to provide lower bounds on  $\overline{\rho}(t)(B_r^{\infty}(x^*))$  for any small radius r and times  $t \in [0, T]$ .

**Lemma A.4.** Let  $\overline{\rho}(t) = \text{Law}(\overline{Y}_t)$ , with  $\overline{Y}_t$  evolving according to (4.6) and  $\lim_{t\to 0} \overline{\rho}(t) = \rho_0$ with  $x^* \in \text{supp}(\rho_0)$ . Under Assumptions 4.2 and 4.3, it holds  $\overline{\rho}(t)(B_r^{\infty}(x^*)) \ge m_r > 0$ , for all  $t \in [0,T]$  and for all  $r \le R_0$ .

*Proof.* Let  $\delta = \eta \min\{R_1, r\}^{\gamma}$ , we start by proving that the mass in the set

$$L_{\delta} = \{ x \in \mathbb{R}^{d} \mid \mathcal{F}(x) \le \inf \mathcal{F} + \delta \}$$

is non-decreasing. We note that for this choice of  $\delta$ ,  $L_{\delta}$  is convex due to Assumption 4.2. Consider now  $(\Omega, \overline{\mathcal{F}}, \mathbb{P})$  to be the common probability space over which the considered processes take their realization and define  $\Omega_{\delta} = \{\omega : \overline{Y}_0(\omega) \in L_{\delta}\}$ . By Assumption 4.3,  $S^{\beta}(\overline{X}_t(\omega), \overline{Y}_t(\omega)) = 0$ whenever  $\overline{X}_t(\omega) \notin L_{\delta}$ . Therefore, it holds

$$\left\langle (\overline{X}_t(\omega) - \overline{Y}_t(\omega)) S^{\beta}(\overline{X}_t(\omega), \overline{Y}_t(\omega)), n(\overline{Y}_t(\omega)) \right\rangle \begin{cases} = 0 & \text{if } \overline{X}_t(\omega) \notin L_{\delta} \\ \leq 0 & \text{if } \overline{X}_t(\omega) \in L_{\delta} \end{cases} \text{ for } \overline{Y}_t(\omega) \in \partial L_{\delta} \end{cases}$$

for any  $n(\overline{Y}_t(\omega)) \in \mathcal{N}(L_{\delta}, x)$  from which follows that  $\overline{Y}_t(\omega)$  solves

$$\overline{Y}_t(\omega) = \overline{Y}_0(\omega) + \int_0^t \Pi_{\mathcal{T}(L_{\delta}, \overline{Y}_s(\omega))} \left( (\overline{X}_s(\omega) - \overline{Y}_s(\omega)) S^{\beta}(\overline{X}_s(\omega), \overline{Y}_s(\omega)) \right) ds$$

for all  $\omega \in \Omega_{\delta}$ . As a consequence, if  $\overline{Y}_0(\omega) \in L_{\delta}$ ,  $\overline{Y}_t(\omega) \in L_{\delta}$  for all  $t \ge 0$  and so

$$\overline{\rho}(t)(B_r^{\infty}(x^*)) = \mathbb{P}(\overline{Y}_t \in L_{\delta}) \ge \mathbb{P}(\overline{Y}_0 \in L_{\delta}) =: m_t$$

for all  $t \ge 0$ . We conclude by noting that  $m_r > 0$  since  $x^* \in \text{supp}(\rho_0)$ .

Next, we study the evolution of the error  $\mathbb{E}\|\overline{X}_t - x^*\|_2^2$  and, in particular, we try to bound it in terms of  $\|\overline{y}^{\alpha}(\overline{\rho}(s)) - x^*\|_2$  and  $\mathbb{E}\|\overline{X}_t - x^*\|_2$  itself for  $s \in [0, t]$ .

**Proposition A.2.** [12, Lemma 1] Let  $(\overline{X}_t, \overline{Y}_t) \in C([0, T], \mathbb{R}^{2d})$  satisfy (4.6) with initial datum  $\overline{X}_0 \sim \rho_0, \rho_0 \in \mathcal{P}_4(\mathbb{R}^d), \overline{Y}_0 = \overline{X}_0$  for some time horizon T > 0. Set  $\mathcal{V}(\rho(t)) := (1/2)\mathbb{E} \|\overline{X}_t - x^*\|_2^2$  with  $\rho(t) \in \text{Law}(\overline{X}_t)$ . For all  $t \in [0, T]$ , it holds

$$\frac{d}{dt}\mathcal{V}(\rho(t)) \leq -(2\lambda - \sigma^2)\mathcal{V}(\rho(t)) + \sqrt{2}(\lambda + \sigma^2)\sqrt{\mathcal{V}(\rho(t))} \|\overline{y}^{\alpha}(\overline{\rho}(t)) - x^*\|_2 + \frac{\sigma^2}{2}\|\overline{y}^{\alpha}(\overline{\rho}(t)) - x^*\|_2^2. \quad (A.3)$$

where  $\overline{\rho}(t) = \operatorname{Law}(\overline{Y}_t)$ .

*Proof of Theorem 4.1.* The above result, together with Lemma A.4, leads to the convergence in mean-field law of the dynamics towards the solution to (2.1). The proof can be carried out exactly as in [12, Theorem 12] and we summarize here the main steps for completeness.

For notational simplicity, we introduce  $\operatorname{Err}(t) := \mathbb{E}[\|\overline{X}_t - x^*\|_2^2]$ . We start by setting the time horizon  $T^* = -2\log(\varepsilon/\operatorname{Err}(0))/(2\lambda - \sigma^2)$ . We apply Proposition A.2 and, since  $\mathcal{V}(\rho(t)) = \operatorname{Err}(t)/2$ , we have for all  $t \in [0, T^*]$ 

$$\frac{d}{dt}\operatorname{Err}(t) \le -(2\lambda - \sigma^2)\operatorname{Err}(t) + (\lambda + \sigma^2)\sqrt{\operatorname{Err}(t)} \|\overline{y}^{\alpha}(\overline{\rho}(t)) - x^*\|_2 + \sigma^2 \|\overline{y}^{\alpha}(\overline{\rho}(t)) - x^*\|_2^2.$$

Let  $T \ge 0$  be given by

$$T := \sup \left\{ t \in [0, T^*] : \operatorname{Err}(t') > \varepsilon \text{ and } \|\overline{y}^{\alpha}(\overline{\rho}(t')) - x^*\|_2^2 < C(t') \quad \forall t' \in [0, t] \right\}$$

with

$$C(t) := \min\left\{\frac{1}{4}\frac{2\lambda - \sigma^2}{\lambda + \sigma^2}, \sqrt{\frac{1}{4}\frac{2\lambda - \sigma^2}{\sigma^2}}\right\}\sqrt{\operatorname{Err}(t)}.$$

For this particular choice of T, we have that for all  $t \in [0, T]$ 

$$\frac{d}{dt}\operatorname{Err}(t) \le -\frac{1}{2}(2\lambda - \sigma^2)\operatorname{Err}(t) \quad \Rightarrow \quad \operatorname{Err}(t) \le \operatorname{Err}(0) e^{-\frac{1}{2}(2\lambda - \sigma^2)t} \tag{A.4}$$

where we applied Grönwall's inequality. Now, we consider three possible scenarios.

**Case**  $T = T^*$ . By definition of  $T^*$  and decay estimate (A.4), we have  $\text{Err}(T^*) = \varepsilon$ .

Case  $T < T^*$  and  $\operatorname{Err}(T) = \varepsilon$ . Nothing to prove in this case.

**Case**  $T < T^*$  and  $\|\overline{y}^{\alpha}(\overline{\rho}(s)) - x^*\|_2^2 \ge C(T)$ . We will show that if  $\alpha$  is large enough, this case cannot occur. From Proposition A.1 and Lemma A.4 we have

$$\|\overline{y}^{\alpha}(\overline{\rho}(T)) - x^*\|_2 \le \frac{\sqrt{d}(q + \mathcal{F}_r)^{\gamma}}{\eta} + \frac{\sqrt{d}\exp(-\alpha q)}{m_r} \mathbb{E}[\|\overline{Y}_T - x^*\|_2]$$

Now, by continuity of  $\mathcal{F}$ , we can take q, r small small enough such that the first term on the right-hand side is strictly smaller than C(T)/2. Thanks to Lemma A.2 and bound (A.4), it holds

$$\mathbb{E}[\|\overline{Y}_T - x^*\|_2] \le \left(\mathbb{E}[\|\overline{Y}_T - x^*\|_2^2]\right)^{1/2} \le \sqrt{2}e^{\frac{1}{2}\nu T} (\sup_{t \in [0,T]} \operatorname{Err}(t))^{1/2} \le \sqrt{2}e^{\frac{1}{2}\nu T} \sqrt{\operatorname{Err}(0)} \,.$$

Therefore, we can take  $\alpha$  sufficiently large such that

$$\frac{\sqrt{\mathrm{d}}\exp(-\alpha q)}{m_r}\mathbb{E}[\|\overline{Y}_T - x^*\|_2] \le \frac{\sqrt{\mathrm{d}}\exp(-\alpha q)}{m_r}\sqrt{2}e^{\frac{1}{2}\nu T}\sqrt{\mathrm{Err}(0)} < \frac{C(T)}{2}, \quad (A.5)$$

from which follows

$$\|\overline{y}^{\alpha}(\overline{\rho}(T)) - x^*\|_2 < C(T) \,.$$

Therefore, we have a contradiction and we can conclude that this third case can be avoided by taking  $\alpha$  sufficiently large.

#### A.4 Proof of Proposition 4.2 and Theorem 4.2

We start by collecting a preliminary result.

**Lemma A.5.** Let  $\{x_{1,i}^k, y_{1,i}^k\}_{i=1}^{N_1}$  and  $\{x_{2,j}^k, y_{2,j}^k\}_{j=1}^{N_2}$  be two particle populations generated through update rules (4.3) with  $\theta_{1,i}^k = \theta_{2,j}^k = \theta^k$  for all i, j and  $k \in \mathbb{Z}_+$ . At any iteration step k and for any couple of indexes (i, j), it holds

$$\begin{split} \mathbb{E}\left[\|x_{1,i}^{k+1} - x_{2,j}^{k+1}\|_{2}^{2} + \|y_{1,i}^{k+1} - y_{2,j}^{k+1}\|_{2}^{2}\right] \leq \\ C\mathbb{E}\left[\|x_{1,i}^{k} - x_{2,j}^{k}\|_{2}^{2} + \|y_{1,i}^{k} - y_{2,j}^{k}\|_{2}^{2} + \|\overline{y}^{\alpha}(\overline{\rho}_{1}^{k}) - \overline{y}^{\alpha}(\overline{\rho}_{2}^{k})\|_{2}^{2}\right] \end{split}$$

where  $C = C(\Delta t, \lambda, \sigma, \nu, \beta)$  is a positive constant and  $\overline{\rho}_1^k, \overline{\rho}_2^k$  are the empirical distributions associated with  $\{y_{1,i}^k\}_{i=1}^{N_1}$  and  $\{y_{2,j}^k\}_{j=1}^{N_2}$  respectively.

*Proof.* For all  $k \in \mathbb{Z}_+$  and i, j

$$\begin{split} \mathbb{E} \|x_{1,i}^{k+1} - x_{2,j}^{k+1}\|_{2}^{2} &\leq \mathbb{E} \left\| x_{1,i}^{k} + \lambda \Delta t \left( \overline{y}^{\alpha}(\overline{\rho}_{1}^{k}) - x_{1,i}^{k} \right) + \sigma \sqrt{\Delta t} \left( \overline{y}^{\alpha}(\overline{\rho}_{1}^{k}) - x_{1,i}^{k} \right) \otimes \theta_{1,i}^{k} \\ &- \left( x_{2,j}^{k} + \lambda \Delta t \left( \overline{y}^{\alpha}(\overline{\rho}_{2}^{k}) - x_{2,j}^{k} \right) + \sigma \sqrt{\Delta t} \left( \overline{y}^{\alpha}(\overline{\rho}_{2}^{k}) - x_{2,j}^{k} \right) \otimes \theta_{2,j}^{k} \right) \right\|_{2}^{2} \\ &\leq 2\mathbb{E} \left\| \left( 1 - \lambda \Delta t - \sigma \sqrt{\Delta t} \operatorname{diag}(\theta^{k}) \right) \left( x_{1,i}^{k} - x_{2,j}^{k} \right) \right\|_{2}^{2} \\ &+ 2\mathbb{E} \left\| \left( \lambda \Delta t + \sigma \sqrt{\Delta t} \operatorname{diag}(\theta^{k}) \right) \left( \overline{y}^{\alpha}(\overline{\rho}_{1}^{k}) - \overline{y}^{\alpha}(\overline{\rho}_{2}^{k}) \right) \right\|_{2}^{2} \\ &\leq 2(1 + \lambda^{2} \Delta t^{2} + \sigma^{2} \Delta t) \mathbb{E} \| x_{1,i}^{k} - x_{2,j}^{k} \|_{2}^{2} \\ &+ 2(\lambda^{2} \Delta t^{2} + \sigma^{2} \Delta t) \mathbb{E} \| \overline{y}^{\alpha}(\overline{\rho}_{1}^{k}) - \overline{y}^{\alpha}(\overline{\rho}_{2}^{k}) \|_{2}^{2}, \end{split}$$
(A.6)

where we also used that  $\mathbb{E}[(\theta^k)_{\ell}^2] = 1$  for all  $\ell = 1, \dots, d$ . We now bound  $\|y_{1,i}^{k+1} - y_{2,j}^{k+1}\|_2^2$  as

$$\mathbb{E}\|y_{1,i}^{k+1} - y_{2,j}^{k+1}\|_2^2 \le \mathbb{E}\left\|y_{1,i}^k + (\nu\Delta t/2)\left(x_{i,1}^{k+1} - y_{1,i}^k\right)S^\beta(x_{1,i}^{k+1}, y_{1,i}^k)\right\|_2^2 \le \mathbb{E}\left\|y_{1,i}^k + (\nu\Delta t/2)\left(x_{1,i}^{k+1} - y_{1,i}^k\right)S^\beta(x_{1,i}^{k+1}, y_{1,i}^k)\right\|_2^2 \le \mathbb{E}\left\|y_{1,i}^k + (\nu\Delta t/2)\left(x_{1,i}^{k+1} - y_{1,i}^k\right)S^\beta(x_{1,i}^k, y_{1,i}^k)\right\|_2^2 \le \mathbb{E}\left\|y_{1,i}^k + (\nu\Delta t/2)\left(x_{1,i}^k + y_{1,i}^k\right)S^\beta(x_{1,i}^k, y_{1,i$$

$$-\left(y_{2,j}^{k}+(\nu\Delta t/2)\left(x_{2,j}^{k+1}-y_{2,j}^{k}\right)S^{\beta}(x_{2,j}^{k+1},y_{2,j}^{k})\right)\Big\|_{2}^{2}$$

$$\leq C\mathbb{E}\left[\|x_{i,1}^{k+1}-x_{j,2}^{k+1}\|_{2}^{2}+\|y_{i,1}^{k}-y_{j,2}^{k}\|_{2}^{2}\right]$$
(A.7)

where we used Lemma A.2 and  $C = C(\Delta t, \beta, \nu)$ . By combining (A.6) and (A.7) we get the desired estimate.

Next, we show how the particle update rule (4.3) is stable with respect to the 2-Wasserstein distance.

**Proposition A.3** (Stability of update rule (4.3)). Let  $\{x_{1,i}^k, y_{1,i}^k\}_{i=1}^{N_1}, \{x_{2,j}^k, y_{2,j}^k\}_{j=1}^{N_2} \subset B_M(0)$ , for some M > 0, be two particle populations generated through the update rules (4.3) with  $\theta_{1,i}^k = \theta_{2,j}^k = \theta^k$  for all i, j and  $k \in \mathbb{Z}_+$ . Let  $\mu_1^k, \mu_2^k \in \mathcal{P}(\mathbb{R}^{2d})$  the empirical probability measures defined as

$$\mu_1^k := \frac{1}{N_1} \sum_{i=1}^{N_1} \delta_{(x_{1,i}^k, y_{1,i}^k)} \,, \quad \mu_2^k := \frac{1}{N_2} \sum_{j=1}^{N_2} \delta_{(x_{2,j}^k, y_{2,j}^k)} \,.$$

Under Assumptions 4.1 and 4.3, it holds

$$\mathbb{E}\left[W_2^2(\mu_1^{k+1}, \mu_2^{k+1})\right] \le C_1 \mathbb{E}\left[W_2^2(\mu_1^k, \mu_2^k)\right],\,$$

where  $C_1 = C_1(\Delta, \lambda, \sigma, \nu, \alpha, \beta, L_F, M)$  is positive constant.

*Proof.* Let  $\mathbb{E}_{\theta^k}[\cdot]$  denote the expectation taken with respect to the sampling of  $\theta^k$  only and  $w \in \mathbb{R}^{N_1 \times N_2}$  be the optimal coupling between  $\mu_1^k, \mu_2^k$  (see (4.12) and (4.13)). Being w a sub-optimal coupling for  $\mu_1^{k+1}, \mu_2^{k+1}$ , it holds

$$\mathbb{E}_{\theta^{k}}[W_{2}^{2}(\mu_{1}^{k+1},\mu_{2}^{k+1})] \leq \mathbb{E}_{\theta^{k}}\sum_{i,j}w_{ij}\left(\|x_{1,i}^{k+1}-x_{2,j}^{k+1}\|_{2}^{2}+\|y_{1,i}^{k}-y_{2,j}^{k}\|_{2}^{2}\right) \\
\leq C\sum_{i,j}w_{ij}\left(\|x_{1,i}^{k}-x_{2,j}^{k}\|_{2}^{2}+\|y_{1,i}^{k}-y_{2,j}^{k}\|_{2}^{2}\right)+\|\overline{y}^{\alpha}(\overline{\rho}_{1}^{k})-\overline{y}^{\alpha}(\overline{\rho}_{2}^{k})\|_{2}^{2}$$

where we used the linearity of the expectation, estimates given by Lemma A.5 and, to take the last term out of the sum, the fact that  $\sum_{ij} w_{ij} = 1$ .

To estimate the distance between the two consensus points, we use Lemma A.1 and note that the coupling w is sub-optimal for  $\overline{\rho}_1^k, \overline{\rho}_2^k$  with respect to the optimal transport. By Lemma A.1, it follows

$$\|\overline{y}^{\alpha}(\overline{\rho}_{1}^{k}) - \overline{y}^{\alpha}(\overline{\rho}_{2}^{k})\|_{2}^{2} \leq CW_{2}^{2}(\overline{\rho}_{1}^{k}, \overline{\rho}_{2}^{k}) \leq C\sum_{i,j} w_{ij} \|y_{1,i}^{k} - y_{2,j}^{k}\|^{2}.$$

Therefore,

$$\mathbb{E}_{\theta^{k}}[W_{2}^{2}(\mu_{1}^{k+1},\mu_{2}^{k+1})] \leq C_{1}\sum_{i,j} w_{ij}\left(\|x_{1,i}^{k}-x_{2,j}^{k}\|_{2}^{2}+\|y_{1,i}^{k}-y_{2,j}^{k}\|_{2}^{2}\right) = C_{1}W_{2}^{2}(\mu_{1}^{k},\mu_{2}^{k}),$$

thanks to the optimality of w, with  $C_1 = C_1(\Delta, \lambda, \sigma, \nu, \alpha, \beta, L_F, M)$  being a positive constant. One can conclude by taking the expectation of the above inequality with respect to the sampling of  $\theta^h, h < k$ . We now quantify the impact of the particle discarding step.

Proof of Proposition 4.2. For notational simplicity, let us introduce  $z_i = (x_i, y_i) \in \mathbb{R}^{2d}$ . As in (4.13), the 2-Wasserstein distance is given by an optimal coupling between the full particle system  $\{z_i\}_{i \in I}$  and the reduced one  $\{z_j\}_{j \in I_{sel}}$ . We consider the following transportation of mass from  $\mu_N$  to  $\mu_{N_{sel}}$ : if particle *i* has not been discarded, all its mass remains in  $x_i$ , otherwise the mass is uniformly distributed among the selected particles to generate an admissible coupling  $w \in \mathbb{R}^{N \times N_{sel}}$ . This means that *w* is given by

$$w_{ij} = \begin{cases} 1/N & \text{if } j = i, i \in I_{\text{sel}} \\ 1/(N \cdot N_{\text{sel}}) & \text{if } i \in I \setminus I_{\text{sel}}, j \in I_{\text{sel}} \\ 0 & \text{else} \,. \end{cases}$$
(A.8)

We note that such coupling w satisfies the coupling conditions

$$\sum_{j \in I_{\text{sel}}} w_{ij} = \frac{1}{N} \qquad \sum_{i \in I} w_{ij} = \frac{1}{N_{\text{sel}}}, \quad \forall i \in I, j \in I_{\text{sel}}$$
(A.9)

and that this choice will be in general sub-optimal. Therefore, it holds

$$\begin{split} W_{2}^{2}(\mu_{N},\mu_{N_{\text{sel}}}) &\leq \sum_{i \in I, \, j \in I_{\text{sel}}} w_{ij} \|z_{i} - z_{j}\|_{2}^{2} \\ &= \frac{1}{N} \sum_{i \in I_{\text{sel}}} \|z_{i} - z_{i}\|_{2}^{2} + \frac{1}{N \cdot N_{\text{sel}}} \sum_{i \in I \setminus I_{\text{sel}}, \, j \in I_{\text{sel}}} \|z_{i} - z_{j}\|_{2}^{2} \\ &= \frac{1}{N \cdot N_{\text{sel}}} \sum_{i, j \in I} \|z_{i} - z_{j}\|_{2}^{2} \mathbf{1}_{i \in I \setminus I_{\text{sel}}} \mathbf{1}_{j \in I_{\text{sel}}} \end{split}$$

where  $\mathbf{1}_{i \in A} = 1$  if  $i \in A$  and  $\mathbf{1}_{i \in A} = 0$  if  $i \notin A$ .

Now, the probability of having  $i \in I \setminus I_{sel}$  is given by  $(N - N_{sel})/N$ , while the probability of having  $j \in I_{sel}$  (condition  $i \in I \setminus I_{sel}$ ) is given by  $N_{sel}/(N-1)$ . Hence, we have

$$\mathbb{E}\left[\mathbf{1}_{i \in I \setminus I_{\text{sel}}} \mathbf{1}_{j \in I_{\text{sel}}}\right] = \mathbb{P}\left[i \in I \setminus I_{\text{sel}}, j \in I_{\text{sel}}\right] = \frac{(N - N_{\text{sel}})N_{\text{sel}}}{N(N - 1)}$$

from which follows

$$\mathbb{E}\left[W_2^2(\mu_N, \mu_{N_{\text{sel}}})\right] \le \frac{1}{N \cdot N_{\text{sel}}} \sum_{i,j \in I} \|z_i - z_j\|_2^2 \mathbb{E}\left[\mathbf{1}_{i \in I \setminus I_{\text{sel}}} \mathbf{1}_{j \in I_{\text{sel}}}\right]$$
$$= \frac{1}{N \cdot N_{\text{sel}}} \cdot \frac{(N - N_{\text{sel}})N_{\text{sel}}}{N(N - 1)} \sum_{i,j \in I} \|z_i - z_j\|_2^2.$$

The desired estimates is then obtained by noting that the variance can be computed as  $\operatorname{var}(\mathbf{z}) = 1/(2N^2) \sum_{i,j \in I} ||z_i - z_j||_2^2$ , see definition (2.4).

Finally, we are ready to provide a proof of Theorem 4.2.

Proof of Theorem 4.2. Let  $\{(x_i^k, y_i^k)\}_{i \in I_k}, |I_k| = N_k$  be the sequence of particles generated by iteration (4.3) where additionally  $N_{k+1} - N_k$  particles are discarded after each step  $k \ge 0$ . We denote with  $\mu_{N_k}^k \in \mathcal{P}(\mathbb{R}^{2d})$  the empirical measure associated with such particle system given by

$$\mu_{N_k}^k = \frac{1}{N_k} \sum_{i \in I_k} \delta_{(x_i^k, y_i^k)}$$

We also introduce the measures  $\mu_{N_0}^k, k \ge 0$  corresponding to a particle system generated with the same initial conditions  $\mu_{N_0}^0$  but where no particle reduction occurs. Consistently, we define  $\mu_{N_k}^h, h > k$  to represent the particle system generated starting from  $\mu_{N_k}^k$ , after h - k iterations, with no random selection. The relation between such measures is summarized in the following diagram

 $\mu_{N_{i}}^{k}$ 

where  $\rightarrow$  indicates an iteration step (4.3) while  $\neg \rightarrow$  a particle reduction procedure. Therefore, we are interested in studying the distance between the main diagonal of such diagram  $\mu_{N_k}^k$ , corresponding to the system with particle reduction, and the first row  $\mu_{N_0}^k$  where particle reduction is never performed.

We note that the 2-Wasserstein distance between subsequent rows can be estimated thanks to Proposition A.3 and Proposition 4.2. Let  $\tilde{\mathbf{z}}^{h+1}$  denote the set of particles associated with the probability measure  $\mu_{N_h}^{h+1}$ , that is, the particle systems before the selection procedure (upper diagonal elements in scheme (A.10)). By first applying Proposition A.3 and, subsequently, Proposition 4.2 to  $\tilde{\mathbf{z}}^{h+1}$ , we obtain that for some constant C > 0

$$\mathbb{E}\left[W_{2}^{2}(\mu_{N_{k}}^{k},\mu_{N_{0}}^{k})\right] \leq C \sum_{h=0}^{k-1} \mathbb{E}\left[W_{2}^{2}\left(\mu_{N_{h}}^{k},\mu_{N_{h+1}}^{k}\right)\right]$$
$$\leq C \sum_{h=0}^{k-1} C_{1}^{k-h+1} \mathbb{E}\left[W_{2}^{2}\left(\mu_{N_{h}}^{h+1},\mu_{N_{h+1}}^{h+1}\right)\right]$$
$$\leq 2C \sum_{h=0}^{k-1} C_{1}^{k-h+1} \operatorname{var}\left(\tilde{\mathbf{z}}^{h+1}\right) \frac{N_{h}-N_{h+1}}{N_{h}-1}$$

$$\leq C_2 \max_{h=1,\dots,k} \operatorname{var}\left(\tilde{\mathbf{z}}^h\right) \frac{1}{N_k - 1} \sum_{h=0}^{k-1} N_h - N_{h+1}$$
$$= C_2 \max_{h=1,\dots,k} \operatorname{var}\left(\tilde{\mathbf{z}}^h\right) \frac{N_0 - N_k}{N_k - 1}$$

with  $C_2 = C_2(\Delta t, \lambda, \sigma, \nu, \beta, \alpha, k, M)$ . Finally, the desired estimate follows after noting that

$$W_2^2(\rho_{N_k}^k, \rho_{N_0}^k) \le W_2^2(\mu_{N_k}^k, \mu_{N_0}^k)$$

since  $||x_i^k - x_j^k||_2^2 \le ||(x_i^k, y_i^k) - (x_j^k, y_j^k)||_2^2$  for all couples of particles (i, j).

# Acknowledgments

This work has been written within the activities of GNCS group of INdAM (National Institute of High Mathematics). L.P. acknowledges the partial support of MIUR-PRIN Project 2017, No. 2017KKJP4X "Innovative numerical methods for evolutionary partial differential equations and applications". The work of G.B. is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) through 320021702/GRK2326 "Energy, Entropy, and Dissipative Dynamics (EDDy)" and SFB 1481 "Sparsity and Singular Structures". S.G. acknowledges the support of the ESF PhD Grant "Mathematical and statistical methods for machine learning in biomedical and socio-sanitary applications".

# References

- S. M. Abedi Pahnehkolaei, A. Alfi, and J. Tenreiro Machado. Analytical stability analysis of the fractional-order particle swarm optimization algorithm. *Chaos, Solitons & Fractals*, 155:111658, 2022.
- [2] L. Arnold. *Stochastic Differential Equations: Theory and Applications*. Wiley Interscience, first edition, 1974. Structure-preserving algorithms for ordinary differential equations.
- [3] S. Arora, J. Acharya, A. Verma, and P. K. Panigrahi. Multilevel thresholding for image segmentation through a fast statistical recursive algorithm. *Pattern Recognition Letters*, 29(2):119–125, 2008.
- [4] A. Benfenati, G. Borghi, and L. Pareschi. Binary interaction methods for high dimensional global optimization and machine learning. *Applied Mathematics & Optimization*, 86(1):9, June 2022.
- [5] J. A. Carrillo, Y.-P. Choi, C. Totzeck, and O. Tse. An analytical framework for consensusbased global optimization method. *Math. Models Methods Appl. Sci.*, 28(6):1037–1066, 2018.
- [6] J. A. Carrillo, S. Jin, L. Li, and Y. Zhu. A consensus-based global optimization method for high dimensional machine learning problems. *ESAIM: COCV*, 27:S5, 2021.

- [7] J. Chen, S. Jin, and L. Lyu. A consensus-based global optimization method with adaptive momentum estimation. *Communications in Computational Physics*, 31(4):1296–1316, 2022.
- [8] R. Flamary, N. Courty, A. Gramfort, M. Z. Alaya, A. Boisbunon, S. Chambon, L. Chapel, A. Corenflos, K. Fatras, N. Fournier, L. Gautheron, N. T. Gayraud, H. Janati, A. Rakotomamonjy, I. Redko, A. Rolet, A. Schutz, V. Seguy, D. J. Sutherland, R. Tavenard, A. Tong, and T. Vayer. POT: Python Optimal Transport. *Journal of Machine Learning Research*, 22(78):1–8, 2021.
- [9] M. Fornasier, H. Huang, L. Pareschi, and P. Sünnen. Consensus-based optimization on the sphere: Convergence to global minimizers and machine learning. J. Machine Learning Research, 22(237):1–55, 2021.
- [10] M. Fornasier, H. Huang, L. Pareschi, and P. Sünnen. Consensus-based optimization on hypersurfaces: Well-posedness and mean-field limit. *Mathematical Models and Methods in Applied Sciences*, 30(14):2725–2751, 2020.
- [11] M. Fornasier, T. Klock, and K. Riedl. Consensus-based optimization methods converge globally. arXiv:2103.15130, 2021.
- [12] M. Fornasier, T. Klock, and K. Riedl. Convergence of anisotropic consensus-based optimization in mean-field law. In J. L. Jiménez Laredo, J. I. Hidalgo, and K. O. Babaagba, editors, *Applications of Evolutionary Computation*, pages 738–754, Cham, 2022. Springer International Publishing.
- [13] A. H. Gandomi, G. J. Yun, X.-S. Yang, and S. Talatahari. Chaos-enhanced accelerated particle swarm optimization. *Communications in Nonlinear Science and Numerical Simulation*, 18(2):327–340, 2013.
- [14] G. Garrigos, L. Rosasco, and S. Villa. Convergence of the forward-backward algorithm: beyond the worst-case with the help of geometry. *Mathematical Programming*, June 2022.
- [15] D. Gilbarg and N. Trudinger. Elliptic Partial Differential Equations of Second Order. Classics in Mathematics. Springer Berlin Heidelberg, 2001.
- [16] S. Grassi and L. Pareschi. From particle swarm optimization to consensus based optimization: Stochastic modeling and mean-field limit. *Mathematical Models and Methods in Applied Sciences*, 31(08):1625–1657, 2021.
- [17] S.-Y. Ha, S. Jin, and D. Kim. Convergence of a first-order consensus-based global optimization algorithm. *Mathematical Models and Methods in Applied Sciences*, 30(12):2417–2444, 2020.
- [18] S.-Y. Ha, S. Jin, and D. Kim. Convergence and error estimates for time-discrete consensusbased optimization algorithms. *Numerische Mathematik*, 147(2):255–282, Feb 2021.
- [19] J. H. Holland. Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. MIT press, 1992.

- [20] E. H. Houssein, A. G. Gad, K. Hussain, and P. N. Suganthan. Major advances in particle swarm optimization: Theory, analysis, and application. *Swarm and Evolutionary Computation*, 63:100868, 2021.
- [21] H. Huang and J. Qiu. On the mean-field limit for the consensus-based optimization. Mathematical Methods in the Applied Sciences, 45(12):7814–7831, 2022.
- [22] H. Huang, J. Qiu, and K. Riedl. On the global convergence of particle swarm optimization methods. arXiv:2201.12460, 2022.
- [23] K. Hussain, M. N. Mohd Salleh, S. Cheng, and Y. Shi. Metaheuristic research: a comprehensive survey. Artificial Intelligence Review, 52(4):2191–2233, Dec 2019.
- [24] M. Jamil and X.-S. Yang. A literature survey of benchmark functions for global optimization problems. Int. Journal of Mathematical Modelling and Numerical Optimisation, 2(4):150– 194, 2013.
- [25] S. Jin, L. Li, and J.-G. Liu. Random Batch Methods (RBM) for interacting particle systems. Journal of Computational Physics, 400:108877, 2020.
- [26] J. Kennedy and R. Eberhart. Particle swarm optimization. In Proceedings of ICNN'95 -International Conference on Neural Networks, volume 4, pages 1942–1948 vol.4, 1995.
- [27] S. Kirkpatrick, C. D. Gelatt Jr, and M. P. Vecchi. Optimization by simulated annealing. science, 220(4598):671–680, 1983.
- [28] Y. LeCun and C. Cortes. MNIST handwritten digit database. 2010.
- [29] C. Li, Y. Liu, A. Zhou, L. Kang, and H. Wang. A fast particle swarm optimization algorithm with cauchy mutation and natural selection strategy. In L. Kang, Y. Liu, and S. Zeng, editors, *Advances in Computation and Intelligence*, pages 334–343, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.
- [30] J. Li, X. Dong, S. Ruan, and L. Shi. A parallel integrated learning technique of improved particle swarm optimization and bp neural network and its application. *Scientific Reports*, 12(1):19325, Nov 2022.
- [31] Y. Liang and L. Wang. Applying genetic algorithm and ant colony optimization algorithm into marine investigation path planning model. Soft Computing, 24(11):8199–8210, Jun 2020.
- [32] B. Liu, L. Wang, Y.-H. Jin, F. Tang, and D.-X. Huang. Improved particle swarm optimization combined with chaos. *Chaos, Solitons & Fractals*, 25(5):1261–1271, 2005.
- [33] A. Nickabadi, M. M. Ebadzadeh, and R. Safabakhsh. A novel particle swarm optimization algorithm with adaptive inertia weight. *Applied Soft Computing*, 11(4):3658–3670, 2011.
- [34] N. Otsu. A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1):62–66, 1979.

- [35] M. E. H. Pedersen. Good parameters for particle swarm optimization. Hvass Lab., Copenhagen, Denmark, Tech. Rep. HL1001, pages 1551–3203, 2010.
- [36] 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(1):183–204, 2017.
- [37] E. Platen. An introduction to numerical methods for stochastic differential equations. Acta Numerica, 8:197–246, 1999.
- [38] R. Poli, J. Kennedy, and T. Blackwell. Particle swarm optimization. Swarm intelligence, 1(1):33–57, 2007.
- [39] K. Riedl. Leveraging memory effects and gradient information in consensus-based optimization: On global convergence in mean-field law. arXiv.2211.12184, 2022.
- [40] F. Santambrogio. Optimal Transport for Applied Mathematicians. Birkhäuser, 2015.
- [41] Y. Shi and R. Eberhart. A modified particle swarm optimizer. In 1998 IEEE international conference on evolutionary computation proceedings. IEEE world congress on computational intelligence (Cat. No. 98TH8360), pages 69–73. IEEE, 1998.
- [42] R. Storn and K. Price. Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4):341–359, 1997.
- [43] A.-S. Sznitman. Topics in propagation of chaos. In P.-L. Hennequin, editor, Ecole d'Eté de Probabilités de Saint-Flour XIX — 1989, pages 165–251, Berlin, Heidelberg, 1991. Springer Berlin Heidelberg.
- [44] K.-S. Tang, K.-F. Man, S. Kwong, and Q. He. Genetic algorithms and their applications. IEEE signal processing magazine, 13(6):22–37, 1996.
- [45] D. Tian and Z. Shi. MPSO: Modified particle swarm optimization and its applications. Swarm and evolutionary computation, 41:49–68, 2018.
- [46] C.-Y. Tsai, T.-Y. Liu, and W.-C. Chen. A novel histogram-based multi-threshold searching algorithm for multilevel colour thresholding. *International Journal of Advanced Robotic* Systems, 9(5):223, 2012.
- [47] R. Tuli, H. N. Soneji, and P. Churi. Pixadapt: A novel approach to adaptive image encryption. Chaos, Solitons & Fractals, 164:112628, 2022.
- [48] M. Crepinšek, S.-H. Liu, and M. Mernik. Exploration and exploitation in evolutionary algorithms: A survey. ACM Comput. Surv., 45(3), jul 2013.
- [49] D. Wang, D. Tan, and L. Liu. Particle swarm optimization algorithm: an overview. Soft computing, 22(2):387–408, 2018.
- [50] X.-S. Yang. Nature-inspired optimization algorithms. Academic Press, 2020.

- [51] X.-S. Yang, S. Deb, and S. Fong. Accelerated particle swarm optimization and support vector machine for business optimization and applications. In S. Fong, editor, *Networked Digital Technologies*, pages 53–66, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.
- [52] A. Yegenoglu, K. Krajsek, S. D. Pier, and M. Herty. Ensemble kalman filter optimizing deep neural networks: An alternative approach to non-performing gradient descent. In G. Nicosia, V. Ojha, E. La Malfa, G. Jansen, V. Sciacca, P. Pardalos, G. Giuffrida, and R. Umeton, editors, *Machine Learning, Optimization, and Data Science*, pages 78–92, Cham, 2020. Springer International Publishing.
- [53] Y. Zhang and X. Kong. A particle swarm optimization algorithm with empirical balance strategy. *Chaos, Solitons & Fractals: X*, 10:100089, 2023.