A MULTISCALE CONSENSUS-BASED ALGORITHM FOR MULTI-LEVEL OPTIMIZATION

MICHAEL HERTY, YUYANG HUANG, DANTE KALISE, AND HICHAM KOUHKOUH

Abstract. A novel multiscale consensus-based optimization (CBO) algorithm for solving bi- and tri-level optimization problems is introduced. Existing CBO techniques are generalized by the proposed method through the employment of multiple interacting populations of particles, each of which is used to optimize one level of the problem. These particle populations are evolved through multiscale-in-time dynamics, which are formulated as a singularly perturbed system of stochastic differential equations. Theoretical convergence analysis for the multiscale CBO model to an averaged effective dynamics as the time-scale separation parameter approaches zero is provided. The resulting algorithm is presented for both bi-level and tri-level optimization problems. The effectiveness of the approach in tackling complex multi-level optimization tasks is demonstrated through numerical experiments on various benchmark functions. Additionally, it is shown that the proposed method performs well on min-max optimization problems, comparing favorably with existing CBO algorithms for saddle point problems.

1. Introduction

We are concerned with an important class of multi-level optimization problems where decision variables cascade throughout a hierarchy. As a prototypical example, we consider the bi-level problem formulated as

$$\begin{aligned} &\min_{x\in\mathbb{R}^n} \; F(x,y) \\ &\text{s.t.} \quad y \in \operatorname*{argmin}_{y\in\mathbb{R}^m} G(x,y), \end{aligned} \tag{1.1}$$

where $F, G : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ for some $n, m \in \mathbb{N}$. A lower-level problem $\min_y G(x,y)$ is embedded within an upper-level problem $\min_x F(x,y)$. This bi-level structure is ubiquitous in operations research and planning problems [72, 19, 70]. For example, in a decentralized supply chain with one manufacturer and one distributor, the optimization of the manufacturer's production decisions (e.g., production quantities, pricing strategies) at the upper level and the optimization of the distributor's distribution decisions (e.g., order quantities, transportation routes) at the lower level are interdependent, leading to a hierarchical structure characteristic of bi-level optimization problems [1, 4]. With rapid development of data analysis, bi-level optimization has become increasingly significant in the community of signal processing and machine learning. Recent applications includes image processing [26, 56, 25], coreset

Date: June 23, 2025.

2020 *Mathematics Subject Classification.* 65C35, 90C56, 90C26, 49M37, 93C70.

Key words and phrases. Consensus-Based Optimization, multiscale systems, singular perturbations, averaging principle, bi-level optimization.

The authors thank the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) for the financial support through 320021702 / GRK2326, and for the financial support through 442047500/SFB1481 within the projects B04 (Sparsity fördernde Muster in kinetischen Hierarchien), B05 (Sparsifizierung zeitabhängiger Netzwerkflußprobleme mittels diskreter Optimierung) and B06 (Kinetische Theorie trifft algebraische Systemtheorie). Support received funding from the European Union's Horizon Europe research and innovation programme under the Marie Sklodowska-Curie Doctoral Network Datahyking (Grant No. 101072546). DK is partially supported by the EPSRC Standard Grant EP/T024429/1. YH has been supported by a Roth Scholarship from Imperial College London.

selection [66, 18, 80] and robust training of deep neural networks [81, 78, 74, 77, 73]. For example, when training large-scale machine learning models, it is crucial to identify the most informative subset of data from a massive entire dataset. This process, known as coreset selection, involves selecting the most representative data samples to form the coreset as lower-level problem and minimizing the model training loss of the selected coreset as upper-level problem [76].

Bi-level optimization has gained high attention in tackling sequential decision-making processes owing to their inherent hierarchical structures. However, this also presents considerable challenges when solving it. Finding optimum for bi-level optimization is generally NP-hard, even for the simplest linear case [11]. For linear or convex bi-level problems, existing methods include Karush-Kuhn-Tucker conditions [39] and penalty function approaches [79]. The presence of non-convexity and high dimensionality of potential problems further increase the computational complexity [55]. Therefore, heuristic methods have been proposed to solve bi-level optimization problems, see for instance, Simulated Annealing [5], Particle Swarm Optimization [50] and Genetic Algorithms [57].

Beyond bi-level problems, multi-level problems have broader applicability in diverse areas, ranging from agricultural economics [20], design engineering [9], to optimal actuator placement [51, 28]. In particular, a tri-level optimization problem is formulated as

$$\begin{aligned} &\min_{x\in\mathbb{R}^n} F(x,y,r) \\ &\quad \text{s.t.} \quad y \in \operatorname*{argmin}_{y\in\mathbb{R}^m} G(x,y,r) \\ &\qquad \text{s.t.} \quad r \in \operatorname*{argmin}_{r\in\mathbb{R}^p} E(x,y,r). \end{aligned} \tag{1.2}$$

However, there is limited literature regarding optimization methods for multilevel problems due to the overwhelming complexity of the multilevel hierarchy. Existing methods include metaheuristic methods [67], fixed point type iterations [49] and gradient methods for convex problems [64, 65].

In this paper, we are interested in designing a variant of consensus-based optimization (CBO) algorithms for bi-level and tri-level optimization problems. The CBO method is a multi-particle derivative-free optimization method, originally designed for solving global non-convex optimization problems in high dimensions [59]. The rigorous convergence analysis of CBO in the mean-field limit has been thoroughly studied over recent years [21, 30, 32, 36, 38, 46, 69], in particular also in [44, 43, 45], and in the time-discrete setting [37]. Meanwhile, numerous extensions have been proposed, including applications in constrained optimization [16, 29, 6], multi-objective optimization [15], stochastic optimization [13, 14], large-scale machine learning [68] and variations with jump diffusion [52], with control [48], with truncated diffusion [31], etc.

Of particular interest, Huang, Qiu and Riedl [47] proposed a CBO model (hereafter denoted by `SP-CBO`) consisting of two interacting populations of particles to find saddle points for min-max optimization problems. Their problem can be written as

$$\min_x \max_y F(x,y). \tag{1.3}$$

It can be noted that this is a special case of bi-level optimization. Indeed, by choosing $G = -F$ in (1.1), then $y \in \operatorname{argmin} -F(x,y)$ is equivalent to $y \in \operatorname{argmax} F(x,y)$, and we have

$$\min_x \max_y F(x,y) \quad \Leftrightarrow \quad \left[\begin{aligned} &\min_x F(x,y) \\ &\quad \text{s.t.} \quad y \in \operatorname*{argmin}_y -F(x,y). \end{aligned} \right.$$

Contributions. Our contributions are twofold. Firstly, we introduce a multiscale CBO algorithm for the optimization problems (1.1) and (1.2). We also show how it can be adapted to multi-level optimization problems. The main idea relies on an interacting system of multiple populations of particles, each one aiming at optimizing one of the levels in (1.1), but running on different time scales.

Besides the novel application of CBO algorithm, the multiple scaling in time which our model exhibits could be of its own interest, and certainly also be adapted to various other problems and applications. See the discussion in §2.3.

Secondly, we show that this multiscale CBO algorithm can be simplified thanks to the particular structure of the dynamics, which is suitable for the application of the so-called averaging principle. We construct the resulting reduced dynamics, referred to as *effective* CBO, as it is customary in homogenization; see Section 3.2. Then we show the convergence of our multiscale CBO model to the effective CBO. Indeed this can be seen as a model reduction technique since the dimension of the resulting dynamics (or the number of particles) decreases. Ultimately, these ideas are put into action in `Algorithm 1` for bi-level problems (1.1), and in `Algorithms 2 & 3` for tri-level problems (1.2).

Organization. The rest of the paper is structured as follows. In Section 2, starting from the introduction of the standard CBO model, we construct the multiscale CBO model, discussing the main motivation and mathematical intuition behind it. Section 3 formalizes a modification of the model presented in the previous section, enhancing its theoretical and numerical suitability for multi-level optimization. Additionally, we present the corresponding algorithm. Section 4 generalizes the model to tackle multi-level optimization problems and presents an algorithm for tri-level optimization problems. Section 5 provides a numerical benchmark over a standardized class of tests, assessing the efficiency of our model. We conclude in Section 6 with a brief summary of the results and future perspectives. The appendix A contains the proof of Theorem 3.1.

2. The multiscale CBO method

Starting from the standard CBO method, in this section we introduce a new multiscale CBO algorithm for bi-level optimization problems of the form (1.1).

2.1. The standard CBO model (parametrized).

Let $x \in \mathbb{R}^n$ be fixed. The standard CBO methods for a minimization problem

$$\min_{y\in\mathbb{R}^m} G(x,y) \tag{2.1}$$

employs a group of $M \in \mathbb{N}$ particles $\{Y^j_{x,t}\}_{j=1}^M$, which evolves in time $t \in [0,T]$ with $T < +\infty$, according to the stochastic differential equations (SDEs):

$$\mathrm{d}Y^j_{x,t} = -\lambda_2 \left(Y^j_{x,t} - \mathbf{v}(\widehat{\rho}_{Y_{x,t}}, x)\right) \mathrm{d}t + \sigma_2 D\left(Y^j_{x,t} - \mathbf{v}(\widehat{\rho}_{Y_{x,t}}, x)\right) \mathrm{d}W^{Y,j}_{x,t}, \tag{2.2}$$

where the operator $D(\cdot) = \mathrm{diag}(\cdot)$ maps a d-dimensional vector v to a diagonal matrix in $\mathbb{R}^{d\times d}$ whose components are the elements of the vector v, and λ_2, σ_2 are positive constants. Let us denote $Y_{x,t} := (Y^1_{x,t}, \dots, Y^M_{x,t}) \in (\mathbb{R}^m)^M$, and $\widehat{\rho}_{Y_{x,t}}$ its corresponding empirical measure

$$\widehat{\rho}_{Y_{x,t}} = \frac{1}{M}\sum_{k=1}^M \delta_{Y^k_{x,t}}. \tag{2.3}$$

The weight function is given by

$$\varpi_\beta(x,y) = \exp(-\beta G(x,y)) \tag{2.4}$$

for some $\beta > 0$ constant. Then the consensus point is then defined as:

$$\mathbf{v}(\widehat{\rho}_{Y_{x,t}}, x) = \sum_{k=1}^M Y^k_{x,t} \frac{\varpi_\beta(x, Y^k_{x,t})}{\sum_{r=1}^M \varpi_\beta(x, Y^r_{x,t})} = \int_{\mathbb{R}^m} y \frac{\varpi_\beta(x,y)}{\left\|\varpi_\beta(x,\cdot)\right\|_{L^1(\widehat{\rho}_{Y_{x,t}})}} \mathrm{d}\widehat{\rho}_{Y_{x,t}}(y) \tag{2.5}$$

with $\left\|\varpi_\beta(x,\cdot)\right\|_{L^1(\widehat{\rho}_{Y_{x,t}})} = \int_{\mathbb{R}^m} \varpi_\beta(x,y)\,\mathrm{d}\widehat{\rho}_{Y_{x,t}}(y)$.

2.2. **The coupled CBO model.** Let us now consider N particles $\{X_\cdot^i\}_{i=1}^N$ in $\mathbb{R}^n$, where n, N are fixed integers. To each particle $X_\cdot^i \in \mathbb{R}^n$ with $1 \leq i \leq N$, we associate a set of M particles $(Y^1_{X^i,\cdot}, \dots, Y^M_{X^i,\cdot}) \in (\mathbb{R}^m)^M$, where $Y^j_{X^i,\cdot} \in \mathbb{R}^m$ for any $1 \leq j \leq M$. The dynamics of each $Y^j_{X^i,t}$ is governed by (2.2), where instead of x, we now have X^i, and evolves in time $t \in [0,T]$ with $T < +\infty$.

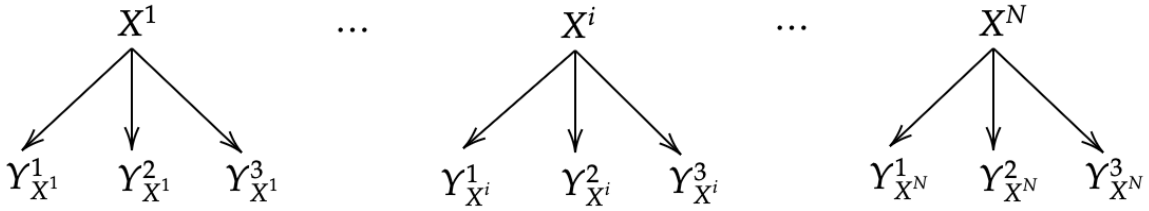


FIGURE 1. Illustration of the particle systems in coupled CBO model with $M = 3$ particles for Y-system.

The particles $X_\cdot^i$, with $1 \leq i \leq N$, are designed for the minimization problem

$$\min_{x \in \mathbb{R}^n} F\big(x, \xi(x)\big),$$

where $\xi(\cdot)$ is a function that *approximately* captures the minimization problem (2.1) in a way that we will soon make precise. These particles follow a CBO scheme

$$\mathrm{d}X_t^i = -\lambda_1 \left(X_t^i - z(\widehat{\rho}_{X_t}, \xi_t(X_t^i))\right) \mathrm{d}t + \sigma_1 D\left(X_t^i - z(\widehat{\rho}_{X_t}, \xi_t(X_t^i))\right) \mathrm{d}W_t^{X,i},$$

where $\lambda_1, \sigma_1 > 0$ are constants, $X_t := (X_t^1, \dots, X_t^N) \in (\mathbb{R}^n)^N$, and $\widehat{\rho}_{X_t}$ its corresponding empirical measure is

$$\widehat{\rho}_{X_t} = \frac{1}{N} \sum_{i=1}^N \delta_{X_t^i}.$$

The weights functions are given by

$$\omega_\alpha(x,y) = \exp(-\alpha F(x,y)), \tag{2.6}$$

for some $\alpha > 0$ constant. We can then define the consensus point as

$$z(\widehat{\rho}_{X_t}, \xi_t(X_t^i)) = \sum_{k=1}^N X_t^k \frac{\omega_\alpha(X_t^k, \xi_t(X_t^i))}{\sum_{r=1}^N \omega_\alpha(X_t^r, \xi_t(X_t^i))} = \int_{\mathbb{R}^n} x \frac{\omega_\alpha(x, \xi_t(X_t^i))}{\|\omega_\alpha(\cdot, \xi_t(X_t^i))\|_{L^1(\widehat{\rho}_{X_t})}} \mathrm{d}\widehat{\rho}_{X_t}(x).$$

The function $\xi_t(\cdot)$ is the coupling term between the x- and y-population of particles. It is given as follows

$$\xi_t(x) = \sum_{j=1}^M Y_{x,t}^j \frac{\varpi_\beta(x, Y_{x,t}^j)}{\sum_{\ell=1}^M \varpi_\beta(x, Y_{x,t}^\ell)} = \int_{\mathbb{R}^m} y \frac{\varpi_\beta(x,y)}{\|\varpi_\beta(x,\cdot)\|_{L^1(\widehat{\rho}_{Y_{x,t}})}} \mathrm{d}\widehat{\rho}_{Y_{x,t}}(y),$$

where we recall the weight function $\varpi_\beta(\cdot,\cdot)$ is defined in (2.4). This is in fact the consensus of the group of y-particles associated with each x-particle at time t. In other words, given a particle X_t^i, $1 \leq i \leq N$, the consensus point at time t for its corresponding population of y-particles $Y_{X^i,t} = (Y^1_{X^i,t}, \dots, Y^M_{X^i,t})$ is $\xi_t(X_t^i) \in \mathbb{R}^m$ defined by

$$\xi_t(X_t^i) = \sum_{j=1}^M Y_{X^i,t}^j \frac{\varpi_\beta(X_t^i, Y_{X^i,t}^j)}{\sum_{\ell=1}^M \varpi_\beta(X_t^i, Y_{X^i,t}^\ell)} = \int_{\mathbb{R}^m} y \frac{\varpi_\beta(X_t^i, y)}{\|\varpi_\beta(X_t^i,\cdot)\|_{L^1\left(\widehat{\rho}_{Y_{X^i,t}}\right)}} \mathrm{d}\widehat{\rho}_{Y_{X^i,t}}(y),$$

hence we have $\xi_t(X_t^i) = \mathbf{v}(\widehat{\rho}_{Y_{X^i,t}}, X_t^i)$ where we recall $\mathbf{v}$ is defined in (2.5). For simplicity of notation, we shall denote it by

$$\mathbf{v}_t^i := \mathbf{v}(\widehat{\rho}_{Y_{X^i,t}}, X_t^i).$$

This means the consensus point z takes the form

$$\begin{aligned} z(\widehat{\rho}_{X_t}, \xi_t(X_t^i)) &= \sum_{k=1}^N X_t^k \frac{\omega_\alpha(X_t^k, \mathbf{v}(\widehat{\rho}_{Y_{X^i,t}}, X_t^i))}{\sum_{r=1}^N \omega_\alpha(X_t^r, \mathbf{v}(\widehat{\rho}_{Y_{X^i,t}}, X_t^i))} \\ &= \int_{\mathbb{R}^n} x \frac{\omega_\alpha(x, \mathbf{v}(\widehat{\rho}_{Y_{X^i,t}}, X_t^i))}{\|\omega_\alpha(\cdot\,, \mathbf{v}(\widehat{\rho}_{Y_{X^i,t}}, X_t^i))\|_{L^1(\widehat{\rho}_{X_t})}} \,\mathrm{d}\widehat{\rho}_{X_t}(x) \\ &=: \mathbf{z}(\widehat{\rho}_{X_t}, \mathbf{v}_t^i). \end{aligned} \tag{2.7}$$

Therefore, the dynamics are given by

$$\mathrm{d}X_t^i = -\lambda_1 \left(X_t^i - \mathbf{z}(\widehat{\rho}_{X_t}, \mathbf{v}_t^i)\right) \mathrm{d}t + \sigma_1 D\left(X_t^i - \mathbf{z}(\widehat{\rho}_{X_t}, \mathbf{v}_t^i)\right) \mathrm{d}W_t^{X,i}, \tag{2.8}$$

and

$$\mathbf{v}_t^i = \sum_{j=1}^M Y_{X^i,t}^j \frac{\varpi_\beta(X_t^i, Y_{X^i,t}^j)}{\sum_{\ell=1}^M \varpi_\beta(X_t^i, Y_{X^i,t}^\ell)} = \int_{\mathbb{R}^m} y \frac{\varpi_\beta(X_t^i, y)}{\|\varpi_\beta(X_t^i, \cdot)\|_{L^1\left(\widehat{\rho}_{Y_{X^i,t}}\right)}} \mathrm{d}\widehat{\rho}_{Y_{X^i,t}}(y). \tag{2.9}$$

Note that the X-particles are interacting between each other, and each X-particle is also interacting with its corresponding Y-particles. But for any $1 \le i \le N$, the particles $\{Y_{X^i,\cdot}^j\}_{j=1}^M$ associated to $X_\cdot^i$ do not interact with the particles $\{Y_{X^k,\cdot}^j\}_{j=1}^M$ for any $k \neq i$.

2.3. **The multiscale CBO model.** The presence of a hierarchical structure in the bi-level optimization problem, or in the order of optimization for min-max problems, suggests that the x- and y-particles are not at the same decision level. Indeed, while the y-particles perform optimization, the x-particles remain frozen, waiting for the y-particles to reach a steady-state. Numerically, this means that for each element of the x-population, we update its corresponding y-population for several time steps before we update the x-population. In other words, the evolution of y-population is on a faster time-scale than the x-population. This distinction motivates the introduction of two time scales.

The multi-time scaling of such coupled systems of SDEs can be modelled with *singular perturbation*, that is, given some small parameter $0 < \varepsilon \ll 1$, we consider the following dynamics:

$$\begin{cases} \mathrm{d}X_t^i = -\lambda_1 \left(X_t^i - \mathbf{z}(\widehat{\rho}_{X_t}, \mathbf{v}_t^i)\right) \mathrm{d}t + \sigma_1 D\left(X_t^i - \mathbf{z}(\widehat{\rho}_{X_t}, \mathbf{v}_t^i)\right) \mathrm{d}W_t^{X,i} \\ \mathrm{d}Y_{X^i,t}^j = -\lambda_2 \dfrac{1}{\varepsilon}\left(Y_{X^i,t}^j - \mathbf{v}_t^i\right) \mathrm{d}t + \sqrt{\dfrac{1}{\varepsilon}}\sigma_2 D\left(Y_{X^i,t}^j - \mathbf{v}_t^i\right) \mathrm{d}W_{X^i,t}^{Y,j} \\ \text{Given } (X_0^i, Y_{i,0}^j), \qquad 1 \le j \le M, \quad 1 \le i \le N, \quad 0 \le t \le T, \end{cases} \tag{2.10}$$

where $\mathbf{v}_t^i$ defined in (2.9) is a function of $(X_t^i, Y_{X^i,t}^1, \dots, Y_{X^i,t}^M)$, and and $\mathbf{z}(\cdot\,,\cdot)$ is defined in (2.7).

The x-processes are referred to as the *slow dynamics*, while the y-processes are the *fast dynamics*. This denomination can be justified as follows. Let us change the time variable using $\tau = t/\varepsilon$. The system (2.10) becomes

$$\begin{cases} \mathrm{d}\widetilde{X}_\tau^i = -\varepsilon\,\lambda_1 \left(\widetilde{X}_\tau^i - \mathbf{z}(\widehat{\rho}_{\widetilde{X}_\tau}, \mathbf{v}_\tau^i)\right) \mathrm{d}\tau + \sqrt{\varepsilon}\,\sigma_1 D\left(\widetilde{X}_\tau^i - \mathbf{z}(\widehat{\rho}_{\widetilde{X}_\tau}, \mathbf{v}_\tau^i)\right) \mathrm{d}\widetilde{W}_\tau^{X,i} \\ \mathrm{d}\widetilde{Y}_{\widetilde{X}^i,\tau}^j = -\lambda_2 \left(\widetilde{Y}_{\widetilde{X}^i,\tau}^j - \mathbf{v}_\tau^i\right) \mathrm{d}\tau + \sigma_2 D\left(\widetilde{Y}_{\widetilde{X}^i,\tau}^j - \mathbf{v}_\tau^i\right) \mathrm{d}\widetilde{W}_{\widetilde{X}^i,\tau}^{Y,j} \\ \text{Given } (\widetilde{X}_0^i, \widetilde{Y}_{\widetilde{X}^i,0}^j), \qquad 1 \le j \le M, \quad 1 \le i \le N, \end{cases}$$

where we use the notation $\widetilde{Z}_\tau = Z_{\varepsilon\tau}$, for $Z_\cdot = X^i_\cdot, Y^j_{X^i,\cdot}$, and $\widetilde{W}^{X,i}_\tau = (1/\sqrt{\varepsilon})W^{X,i}_{\varepsilon\tau}$, $\widetilde{W}^{Y,j}_{i,\tau} = (1/\sqrt{\varepsilon})W^{Y,j}_{\widetilde{X}^i,\varepsilon\tau}$. Thus it can be seen that when ε is small, the processes $\widetilde{X}^i_\tau$ tend to be frozen compared to the evolution of $\widetilde{Y}^j_{\widetilde{X}^i,\tau}$. In other words, by the time the processes $\widetilde{X}^i_\tau$ make a significant change in their evolution, the processes $\widetilde{Y}^j_{\widetilde{X}^i,\tau}$ would have already reached their long-time steady state. This steady-state behavior should be captured by their invariant measure under suitable recurrence condition. The treatment (both in theory and practice) of two-scale dynamics of this type is a challenging task. This motivates the introduction of simplified dynamics obtained by passing to the limit $\varepsilon \to 0$ in (2.10). Indeed, in this case, one obtains a system consisting only of the *slow* component of (2.10), whose dynamics is averaged with respect to the invariant measure of the *fast* component, as it will be explained in more detail in the next section. This phenomenon is known as the *averaging principle* and is the core idea behind singular perturbations and homogenization. In the context of optimization, singular perturbations of SDEs have been used in [24] which was then extended to a more general setting in [7, 8] where a control parameter have been introduced, playing the role of a controlled learning rate. Therein, the singularly perturbed dynamics played the role of an approximation of the *Entropic (controlled) Stochastic Gradient Descent*. Another context in which multiscale methods and singularly perturbed dynamics (also referred to as fast–slow dynamics) arise is financial mathematics. We refer in particular to [34, 35] and to the book [33] with the numerous references therein. Indeed, it appears that many financial models suffer from a fast volatility which motivates the development of *averaging methods* in order to provide a reliable and trackable approximation of these financial signals. Similar problems motivated by physical phenomena are discussed in [40, Appendix 2], and in more details in the books [71, 58].

A particular advantage of the latter is that it also plays the role of a model reduction technique: from a system consisting of two components (the x- and y-particles), we obtain a system of a single component (the modified x-particles). On the other hand, a challenge associated to the averaging method is that it requires the (explicit) computation of the invariant measure of the *fast* process (the y-particles). In practice, we shall bypass this obstacle by substituting the average with respect to the invariant measure (which is an average in space) by the average with respect to time; see (3.6). This is possible due to ergodicity of the y-particles, as established by the celebrated Birkhoff's ergodic theorem (see, e.g., [58, equation (6.4.2)], and more generally [12, 53, 27]).

Although the averaging principle has been intensively studied since decades, to the best of our knowledge, the system described in (2.10) does not satisfy the assumptions in the very recent developments of the averaging principle (see for example [42, 41, 62, 54] and the references therein). Further modifications are thus needed, which is the object of the next section.

3. Implementation and convergence of the CBO model with effective dynamics

3.1. **The modified multiscale CBO model.** Before formalizing the modified multiscale CBO model, we shall introduce some notations. Given $R > 0$ fixed, we define for $\forall\, v = (v_k)_{1\le k\le d} \in \mathbb{R}^d$, two truncation functions

$$\phi_R(v) = \big(\min\{|v_k|, R\}\big)_{1\le k\le d} \in \mathbb{R}^d,$$

$$\psi_R(v) = \big(\psi_R(v)_k\big)_{1\le k\le d}, \quad \text{where} \quad \psi_R(v)_k = \begin{cases} v_k\,, & \text{if } |v_k| \le R \\ \dfrac{v_k}{|v_k|}R\,, & \text{if } |v_k| > R. \end{cases}$$

Then, we define a matrix-valued function

$$\forall\, v \in \mathbb{R}^d, \quad D^\delta_R(v) = \mathrm{diag}(\delta + \phi_R(v)) \in \mathbb{R}^{d\times d},$$

where as before, $\mathrm{diag}(\cdot)$ is the diagonal matrix whose components are the elements of the vector to which it is applied, e.g., the k-th element in the diagonal of $D^\delta_R(v)$ is $\delta + \min\{|v_k|, R\} > 0$. In

particular $D_R^\delta(\cdot)$ is a positive definite matrix. We shall keep the same notation $\phi_R, \psi_R, D_R^\delta$ regardless of the dimension d, as it will be clear from the context. We then define a matrix-valued function

$$\forall\, M_1, \dots, M_k \in \mathbb{R}^{d\times d}, \quad \mathbb{D}(M_1, \dots, M_k) = \mathrm{Diag}(M_1, \dots, M_k),$$

where $\mathrm{Diag}(\cdot)$ is the block-diagonal matrix whose diagonal blocks are the matrices $M_1, \dots, M_k$. In particular, we define

$$\forall\, v^1, \dots, v^k \in \mathbb{R}^d, \quad \mathbb{D}_R^\delta(v^1, \dots, v^k) = \mathbb{D}(D_R^\delta(v^1), \dots, D_R^\delta(v^k)).$$

This is a block-diagonal matrix in $\mathbb{R}^{dk\times dk}$. Each of its j-th block ($1 \le j \le k$) is a diagonal matrix whose diagonal components are $\delta + \min\{|v_i^j|, R\}$ with $1 \le i \le d$.

To keep the notation concise in what follows, let us denote the state vector of the particle system using the notation

$$\mathbb{X}^\varepsilon = \big(\mathbf{X}_i\big)_{1\le i\le N} \in \mathbb{R}^{nN}, \quad \text{where } \mathbf{X}_i \in \mathbb{R}^n,$$
$$\mathbb{Y}^\varepsilon = (\mathbf{Y}_i)_{1\le i\le N} \in \mathbb{R}^{mMN}, \quad \text{where } \mathbf{Y}_i = (Y^j_{\mathbf{X}_i})_{1\le j\le M} \in \mathbb{R}^{mM}, \quad \text{and } Y^j_{\mathbf{X}_i} \in \mathbb{R}^m.$$

The finite dimensional system of singularly perturbed SDEs is stated as follows:

$$\begin{cases} \mathrm{d}\mathbb{X}_t^\varepsilon = \mathscr{F}(\mathbb{X}_t^\varepsilon, \mathbb{Y}_t^\varepsilon)\,\mathrm{d}t + \mathscr{G}(\mathbb{X}_t^\varepsilon, \mathbb{Y}_t^\varepsilon)\,\mathrm{d}\mathbb{W}_t^1 \\ \mathrm{d}\mathbb{Y}_t^\varepsilon = \dfrac{1}{\varepsilon}\mathscr{B}(\mathbb{X}_t^\varepsilon, \mathbb{Y}_t^\varepsilon)\,\mathrm{d}t + \dfrac{1}{\sqrt{\varepsilon}}\mathscr{H}(\mathbb{X}_t^\varepsilon, \mathbb{Y}_t^\varepsilon)\,\mathrm{d}\mathbb{W}_t^2 \\ \mathbb{X}_0^\varepsilon = \mathbf{x}_0 \in \mathbb{R}^{nN}, \quad \text{and} \quad \mathbb{Y}_0^\varepsilon = \mathbf{y}_0 \in \mathbb{R}^{mMN}, \end{cases} \tag{3.1}$$

where $\mathbb{W}_\cdot^1$ and $\mathbb{W}_\cdot^2$ are nN, mMN-dimensional independent standard Brownian motions. We can now define the coefficients in (3.1). To do so, in the following subsections, we shall drop ε in the notation and simply write $(\mathbb{X}, \mathbb{Y})$ instead of $(\mathbb{X}^\varepsilon, \mathbb{Y}^\varepsilon)$.

3.1.1. *The drift $\mathscr{B}$ in $\mathbb{Y}$-process.* The drift term of the fast process $\{\mathbb{Y}_t\}_{t\ge 0}$ is given by

$$\mathscr{B}(\mathbb{X}, \mathbb{Y}) = \big(B(\mathbf{X}_i, \mathbf{Y}_i)\big)_{1\le i\le N}, \quad \text{where} \quad B(\mathbf{X}_i, \mathbf{Y}_i) = \big(b_i^j(\mathbf{X}_i, \mathbf{Y}_i)\big)_{1\le j\le M}$$

and

$$b_i^j(\mathbf{X}_i, \mathbf{Y}_i) = -\lambda_2\, \psi_{R_2}\left(Y^j_{\mathbf{X}_i} - \kappa \sum_{r=1}^M Y^r_{\mathbf{X}_i} \frac{\varpi_\beta\left(\mathbf{X}_i, Y^r_{\mathbf{X}_i}\right)}{\sum_{\ell=1}^M \varpi_\beta\left(\mathbf{X}_i, Y^\ell_{\mathbf{X}_i}\right)} \right) \in \mathbb{R}^m,$$

where κ is a positive constant that will be needed later in Theorem 3.1. It guarantees the validity of a recurrence condition which ultimately ensures ergodicity of the process, i.e. existence and uniqueness of its invariant measure (see point (iii) in the proof of Theorem 3.1 in Appendix A, and also Remark 3.1 below). Therefore we have

$$b_i^j(\mathbf{X}_i, \mathbf{Y}_i) \in \mathbb{R}^m, \quad B(\mathbf{X}_i, \mathbf{Y}_i) \in \mathbb{R}^{mM} \quad \text{and} \quad \mathscr{B}(\mathbb{X}, \mathbb{Y}) \in \mathbb{R}^{mMN}.$$

Note that, using ψ_{R_2}, the drift functions b_i^j take values in balls of radius $\lambda_2 R_2$.

3.1.2. *The diffusion $\mathscr{H}$ in $\mathbb{Y}$-process.* The diffusion term of the fast process $\{\mathbb{Y}_t\}_{t\ge 0}$ is given by

$$\mathscr{H}(\mathbb{X}, \mathbb{Y}) = \mathrm{Diag}\left(\{H(\mathbf{X}_i, \mathbf{Y}_i)\}_{1\le i\le N}\right),$$
$$\text{where} \quad H(\mathbf{X}_i, \mathbf{Y}_i) = \sigma_2\, \mathbb{D}_{R_2}^{\delta_2}\left(\{h_i^j(\mathbf{X}_i, \mathbf{Y}_i)\}_{1\le j\le M}\right),$$

and, similar to b_i^j, we have

$$h_i^j(\mathbf{X}_i, \mathbf{Y}_i) = Y^j_{\mathbf{X}_i} - \kappa \sum_{r=1}^M Y^r_{\mathbf{X}_i} \frac{\varpi_\beta\left(\mathbf{X}_i, Y^r_{\mathbf{X}_i}\right)}{\sum\limits_{\ell=1}^M \varpi_\beta\left(\mathbf{X}_i, Y^\ell_{\mathbf{X}_i}\right)} \in \mathbb{R}^m.$$

Therefore we have

$$h_i^j(\mathbf{X}_i, \mathbf{Y}_i) \in \mathbb{R}^m, \quad H(\mathbf{X}_i, \mathbf{Y}_i) \in \mathbb{R}^{mM \times mM}, \quad \text{and} \quad \mathscr{H}(\mathbb{X}, \mathbb{Y}) \in \mathbb{R}^{mMN \times mMN}.$$

Using $\mathbb{D}_{R_2}^{\delta_2}$, the diffusion functions H take values in balls of radius $\sigma_2(\delta_2 + R_2)$.

3.1.3. *The drift $\mathscr{F}$ in $\mathbb{X}$-process.* The drift term of the slow process $\{\mathbb{X}_t\}_{t\geq 0}$ is given by

$$\mathscr{F}(\mathbb{X}, \mathbb{Y}) = \big(F_i(\mathbb{X}, \mathbf{Y}_i)\big)_{1\leq i\leq N}$$

$$\text{and} \quad F_i(\mathbb{X}, \mathbf{Y}_i) = -\lambda_1 \, \psi_{R_1}\left(\mathbf{X}_i - \sum_{r=1}^{N} \mathbf{X}_r \frac{\omega_\alpha\left(\mathbf{X}_r, \frac{1}{\kappa}\mathbf{v}^i\right)}{\sum_{\ell=1}^{N} \omega_\alpha\left(\mathbf{X}_\ell, \frac{1}{\kappa}\mathbf{v}^i\right)}\right),$$

where we recall

$$\mathbf{v}^i = \sum_{j=1}^{M} Y_{\mathbf{X}_i}^j \frac{\varpi_\beta(\mathbf{X}_i, Y_{\mathbf{X}_i}^j)}{\sum_{\ell=1}^{M} \varpi_\beta(\mathbf{X}_i, Y_{\mathbf{X}_i}^\ell)}. \tag{3.2}$$

Therefore we have

$$F_i(\mathbb{X}, \mathbf{Y}_i) \in \mathbb{R}^n \quad \text{and} \quad \mathscr{F}(\mathbb{X}, \mathbb{Y}) \in \mathbb{R}^{nN}.$$

Note that, using ψ_{R_1}, the drift functions F_i take values in balls of radius $\lambda_1 R_1$.

3.1.4. *The diffusion $\mathscr{G}$ in $\mathbb{X}$-process.* The diffusion term of the slow process $\{\mathbb{X}_t\}_{t\geq 0}$ is given by

$$\mathscr{G}(\mathbb{X}, \mathbb{Y}) = \sigma_1 \, \mathbb{D}_{R_1}^{\delta_1}\left(\{G_i(\mathbb{X}, \mathbf{Y}_i)\}_{1\leq i\leq N}\right)$$

and, similar to F_i, we have

$$G_i(\mathbb{X}, \mathbf{Y}_i) = \mathbf{X}_i - \sum_{r=1}^{N} \mathbf{X}_r \frac{\omega_\alpha\left(\mathbf{X}_r, \frac{1}{\kappa}\mathbf{v}^i\right)}{\sum_{\ell=1}^{N} \omega_\alpha\left(\mathbf{X}_\ell, \frac{1}{\kappa}\mathbf{v}^i\right)}.$$

Therefore we have

$$G_i(\mathbb{X}, \mathbf{Y}_i) \in \mathbb{R}^n \quad \text{and} \quad \mathscr{G}(\mathbb{X}, \mathbb{Y}) \in \mathbb{R}^{nN \times nN}.$$

Using $\mathbb{D}_{R_1}^{\delta_1}$, the diffusion function $\mathscr{G}$ takes values in a unit ball of radius $\sigma_1(\delta_1 + R_1)$.

Remark 3.1. The following are some relevant observations regarding the proposed multiscale CBO model.

(1) Without loss of generality, one could consider $R_1 = R_2 =: R$, with $R < \infty$ sufficiently large, because the CBO algorithm works in practice over a compact domain, for example an Euclidean ball $\mathcal{B}_0(\bar{R})$ centered in 0 with radius $\bar{R}$, which contains the initial position of the particles. Therefore, if the truncation parameter R is chosen large enough, i.e., $R \gg \bar{R}$, the truncation function becomes $\phi_R(v) = (|v_k|)_{1\leq k\leq d}$ since the particles are unlikely to leave a neighborhood of $\mathcal{B}_0(\bar{R})$. This has been made precise in [10, Proposition 2.3]. It is also worth mentioning that a CBO algorithm with a truncated noise has been studied in [30].
(2) The parameters δ_1, δ_2 are chosen to be small constants. In particular, one could set $\delta_1 = \delta_2 =: \delta > 0$. They ensure the diffusion matrix remains positive definite, providing enough noise to prevent the dynamics from collapsing to one point very quickly. Numerically, this still allows the particles to converge to a small neighborhood around the global optimizer.

(3) In the present context, the parameter κ ensures ergodicity of the fast process, thus the validity of the averaging principle as we will show next. Besides that, using such parameter does not alter the convergence of the CBO algorithm, as it has recently been shown in [45, Corollary 3.4]. Moreover, it allows to prove a uniform-in-time mean-field limit [44], and to construct a self-interacting CBO [43].
(4) The initial condition being deterministic is necessary for our model in order to justify the limit $\varepsilon \to 0$, as we will do in the next section, and in regard to the existing literature on this topic. Nonetheless, it should be noted that, when implemented numerically, the CBO model is able to perform well with random initialization as it is customary done in CBO methods.

3.2. **The effective CBO and the averaging principle.** At the limit $\varepsilon \to 0$, we obtain *averaged (effective)* dynamics, where the dynamics associated to $\mathbb{Y}^\varepsilon$ disappear and the dynamics of $\mathbb{X}^\varepsilon$ is averaged with respect to the limiting (ergodic) behavior of $\mathbb{Y}^\varepsilon$. This yields an effective CBO model which corresponds to the macro-scale dynamics governed by

$$\mathrm{d}\hat{\mathbb{X}}_t = \overline{\mathscr{F}}(\hat{\mathbb{X}}_t)\,\mathrm{d}t + \overline{\mathscr{G}}(\hat{\mathbb{X}}_t)\,\mathrm{d}\mathbb{W}^1_t, \quad \hat{\mathbb{X}}_0 = \mathbf{x}_0 \in \mathbb{R}^{nN}. \tag{3.3}$$

The new *averaged* coefficients are given by

$$\overline{\mathscr{F}}(x) := \int \mathscr{F}(x,y)\,\Upsilon_x(\mathrm{d}y) \quad \text{and} \quad \overline{\mathscr{G}}(x) := \sqrt{\int \mathscr{G}(x,y)\mathscr{G}^*(x,y)\,\Upsilon_x(\mathrm{d}y)},$$

the two integrals are defined on $\mathbb{R}^{mMN} \ni y$, where y is a mute variable, and the averaged coefficients are functions of $x \in \mathbb{R}^{nN}$. Here, Υ_x is the unique invariant probability measure for the dynamics $\mathbb{Y}^x_\cdot$ solution to the *frozen* equation

$$\mathrm{d}\mathbb{Y}^x_t = \mathscr{B}(x,\mathbb{Y}^x_t)\,\mathrm{d}t + \mathscr{H}(x,\mathbb{Y}^x_t)\,\mathrm{d}\mathbb{W}^2_t, \quad \mathbb{Y}^x_0 = \mathbf{y}_0 \in \mathbb{R}^{mMN}, \tag{3.4}$$

where we have arbitrarily fixed $x \in \mathbb{R}^{nN}$. This corresponds to the dynamics of the fast process $\mathbb{Y}$, where we have frozen the slow process $\mathbb{X}$ to some value x. To avoid any confusion, let us once again explicitly rewrite the dynamics (3.3)

$$\mathrm{d}\hat{\mathbb{X}}_t = \int \mathscr{F}(\hat{\mathbb{X}}_t,y)\,\Upsilon_{\hat{\mathbb{X}}_t}(\mathrm{d}y)\,\mathrm{d}t + \sqrt{\int \mathscr{G}(\hat{\mathbb{X}}_t,y)\mathscr{G}^*(\hat{\mathbb{X}}_t,y)\,\Upsilon_{\hat{\mathbb{X}}_t}(\mathrm{d}y)}\;\mathrm{d}\mathbb{W}^1_t.$$

In other words, the system (3.1), whose dimension is $nN \times mMN$, reduces to (3.3), whose dimension is nN.

The convergence of the slow component $\mathbb{X}^\varepsilon$ in (3.1) to the process $\hat{\mathbb{X}}$ in (3.3) has been studied for decades, yet many open questions still persist. In our setting, we will show that the convergence is guaranteed with the recent results in [63] (see also [61, Theorem 2.5]), provided the positive constant κ in Section 3.1.1 is small. We shall denote by $C_b^{2+\theta}$, with $\theta \in (0,1)$, the set of bounded and C^2 functions whose second order derivatives are locally Hölder continuous with exponent θ.

Theorem 3.1. *Let $T > 0$ and $\theta \in (0,1)$. Then for $\kappa \in (0,1)$ small, the process $\mathbb{X}^\varepsilon$ in (3.1) converges to the process $\hat{\mathbb{X}}$ in (3.3) in the sense*

$$\sup_{t\in[0,T]} \left| \mathbb{E}\left[\varphi(\mathbb{X}^\varepsilon_t)\right] - \mathbb{E}[\varphi(\hat{\mathbb{X}}_t)] \right| \leqslant C_T\,\varepsilon^{\frac{\theta}{2}}, \quad \forall\,\varphi \in C_b^{2+\theta}(\mathbb{R}^{nN}),$$

where $C_T > 0$ is a constant independent of ε.

The proof of the theorem is postponed to Appendix A.

Remark 3.2. We can choose $\theta \in (0,2]$, provided that we smoothen the truncation functions (used in (3.1)) using mollifiers, as it is customary. In practice, this will not alter the numerical results. Yet we refrain from doing it in the present manuscript as it will make the presentation less readable.

3.3. **The algorithm.** We propose an algorithm which approximates the computation of the effective (averaged) dynamics $\hat{\mathbb{X}}_\cdot$ in (3.3). Without loss of generality, we assume the parameter R_1 of the truncation functions used in the drift $\mathscr{F}$ and in the diffusion $\mathscr{G}$ is sufficiently large, so that the truncation functions are in fact the identity. This will simplify the computation of the averaged quantities $\overline{\mathscr{F}}$ and $\overline{\mathscr{G}}$. Indeed, in this case, we have $\mathscr{F}(\mathbb{X},\mathbb{Y}) = \big(F_i(\mathbb{X},\mathbf{Y}_i)\big)_{1\le i\le N}$ and $F_i(\mathbb{X},\mathbf{Y}_i) = -\lambda_1\,\big(\mathbf{X}_i - \mathbf{z}(\mathbb{X},\kappa^{-1}\mathbf{v}^i)\big)$, where

$$\mathbf{z}(\mathbb{X},\kappa^{-1}\mathbf{v}^i) = \sum_{r=1}^N \mathbf{X}_r \frac{\omega_\alpha\left(\mathbf{X}_r,\frac{1}{\kappa}\mathbf{v}^i\right)}{\sum\limits_{\ell=1}^N \omega_\alpha\left(\mathbf{X}_\ell,\frac{1}{\kappa}\mathbf{v}^i\right)} \quad\text{and}\quad \mathbf{v}^i = \sum_{j=1}^M Y^j_{\mathbf{X}_i}\frac{\varpi_\beta(\mathbf{X}_i,Y^j_{\mathbf{X}_i})}{\sum\limits_{\ell=1}^M \varpi_\beta(\mathbf{X}_i,Y^\ell_{\mathbf{X}_i})}.$$

Its averaged analogue is obtained as $\overline{\mathscr{F}}(\mathbb{X}) = \big(\overline{F}_i(\mathbb{X})\big)_{1\le i\le N}$ and

$$\overline{F}_i(\mathbb{X}) = -\lambda_1\int\big(\mathbf{X}_i - \mathbf{z}(\mathbb{X},\kappa^{-1}\mathbf{v}^i)\big)\,\mathrm{d}\Upsilon^i_{\mathbb{X}} = -\lambda_1\left(\mathbf{X}_i - \int \mathbf{z}(\mathbb{X},\kappa^{-1}\mathbf{v}^i)\,\mathrm{d}\Upsilon^i_{\mathbb{X}}\right),$$

where we recall $\mathbf{v}^i \equiv \mathbf{v}^i(\mathbf{X}_i,\mathbf{Y}_i)$, and $\mathrm{d}\Upsilon_{\mathbb{X}}$ defined in the previous section is expressed as a product measure $\mathrm{d}\Upsilon_{\mathbb{X}}(y) = \bigotimes_{i=1}^N \Upsilon^i_{\mathbb{X}}(\mathrm{d}y_i)$ for $y\in\mathbb{R}^{mMN}$ and $y=(y_i)_{1\le i\le N}$. Therefore, the goal is to estimate the *averaged consensus* point given

$$\overline{\mathbf{z}}_i(\mathbb{X}) := \int_{\mathbb{R}^{mM}} \mathbf{z}(\mathbb{X},\kappa^{-1}\mathbf{v}^i(y_i))\,\Upsilon^i_{\mathbb{X}}(\mathrm{d}y_i) \tag{3.5}$$

for any given fixed vector $\mathbb{X}$. This is done using the relationship between the invariant measure and the averaged long-time behavior:

$$\int_{\mathbb{R}^{mM}} \mathbf{z}(\mathbb{X},\kappa^{-1}\mathbf{v}^i(\mathbf{X}_i,y_i))\,\Upsilon^i_{\mathbb{X}}(\mathrm{d}y_i) = \lim_{T\to+\infty}\frac{1}{T}\int_0^T \mathbf{z}(\mathbb{X},\kappa^{-1}\mathbf{v}^i(\mathbf{X}_i,\mathbf{Y}_{i,t}))\,\mathrm{d}t, \tag{3.6}$$

where $\mathbf{Y}_i$ solves (3.4) with $x=\mathbb{X}=(\mathbf{X}_i)_{1\le i\le N}$ is a fixed parameter therein (constant in time). See for example [58, equation (6.4.2)].

For the diffusion term, instead of computing $\overline{\mathscr{G}}$ as defined in the previous subsection, we shall approximate it with a diagonal matrix whose elements are the components of the vector

$$\sigma_1\left(\delta + \left|\mathbf{X}_i - \overline{\mathbf{z}}_i(\mathbb{X})\right|\right).$$

Here, we use the notation $|v| = (|v_i|)_{1\le i\le d}$, for any vector $v\in\mathbb{R}^d$, and the constant δ is a vector of the same dimension whose elements are all equal to δ.

Thus, at each iteration of the algorithm, we would like to retrieve a quantity that becomes closer to $\overline{\mathbf{z}}_i(\mathbb{X})$ as defined in (3.5), using the averaging in time as in the right-hand side of (3.6). We proceed in a similar fashion as [23, Algorithm 1], using a Forward–Looking Averaging (instead of a linear averaging), that is:

FLA: we compute $\overline{\mathbf{z}}_i(\mathbb{X})$ at time step $t+1$ as a convex combination of its value at the previous step t, and the value of $\mathbf{z}(\mathbb{X},\kappa^{-1}\mathbf{v}^i)$ at time step $t+1$; see the forward looking averaging (FLA) step in `Algorithm 1` below.

The FLA method is comparable to the exponentially weighted moving average [75, process Z_t in page 510], which is an example of the more general class of *moving average* methods as known in statistics.

Remark 3.3. Note that the computation of the invariant measure could be improved and/or realized differently, for example using the results in [3, 22]. In [3], the invariant measure is computed using a population of interacting particles. Using such approximation in our present setting would require the introduction of new interacting particles which would make the model less readable. In [22], the invariant measure is computed using a self-interacting particle. This approach is reminiscent to (FLA),

since both of them use the past information of the particle, but compute the average with different weights. Ultimately, we have chosen (FLA) as it computes directly the desired averaged quantity, rather than the invariant measure (as in the mentioned references).

In the following, we present the pseudo-code for multiscale CBO solving the bi-level optimization problem (1.1).

Algorithm 1: Multiscale CBO for bi-level optimization problems

Set parameters: γ, α, β, λ_1, λ_2, σ_1, σ_2, Δt, $\Delta\tau$, T_x, T_y, R_1, R_2, δ_1, δ_2 , κ ;

Initialize: $\mathbf{z}_0^i = X_0^i \sim \rho_0,\ i = 1,\dots,N$;

$Y_{i,0}^j \sim \mu_0,\ j = 1,\dots,M$, where $M \ll N$;

$t = 0$, $\mathcal{T}_x = T_x/\Delta t$, $\mathcal{T}_y = T_y/\Delta\tau$;

while $t \le \mathcal{T}_x$ **do**

  for $i = 1,\dots,N$ **do**

    $\tau = 0$;

    $\mathbf{v}_\tau^i = \sum_{j=1}^M Y_{X_i,\tau}^j \frac{\varpi_\beta(X_t^i, Y_{X_i,\tau}^j)}{\sum_{\ell=1}^M \varpi_\beta(X_t^i, Y_{X_i,\tau}^\ell)}$; `/* Initial consensus of` $\mathbf{Y}_{i,0}$ `*/`

    while $\tau \le \mathcal{T}_y$ **do**

      for $j = 1,\dots,M$ **do**

        sample $W_{X_i,\tau}^{Y,j}$ from $\mathcal{N}(0, \Delta\tau\,\mathbb{I}_m)$;

        $Y_{X_i,\tau}^j \leftarrow Y_{X_i,\tau}^j - \lambda_2\psi_{R_2}\left(Y_{X_i,\tau}^j - \kappa\,\mathbf{v}_\tau^i\right)\Delta\tau + \sigma_2 D_{R_2}^{\delta_2}\left(Y_{X_i,\tau}^j - \kappa\,\mathbf{v}_\tau^i\right) W_{X_i,\tau}^{Y,j}$;

      end

      $\mathbf{v}_\tau^i \leftarrow \sum_{j=1}^M Y_{X_i,\tau}^j \frac{\varpi_\beta(X_t^i, Y_{X_i,\tau}^j)}{\sum_{\ell=1}^M \varpi_\beta(X_t^i, Y_{X_i,\tau}^\ell)}$; `/* consensus of` $\mathbf{Y}_{i,\tau}$ `*/`

      $\mathbf{z}' \leftarrow \sum_{k=1}^N X_t^k \frac{\omega_\alpha(X_t^k, \frac{1}{\kappa}\mathbf{v}_\tau^i)}{\sum_{r=1}^N \omega_\alpha(X_t^r, \frac{1}{\kappa}\mathbf{v}_\tau^i)}$; `/* auxiliary variable */`

      $\mathbf{z}_t^i \leftarrow (1-\gamma)\mathbf{z}_t^i + \gamma\mathbf{z}'$; `/* FLA of` X_t^i`; depends on` i `*/`

      $\tau \leftarrow \tau + 1$;

    end

    sample $W_{i,t}^X$ from $\mathcal{N}(0, \Delta t\,\mathbb{I}_n)$;

    $X_t^i \leftarrow X_t^i - \lambda_1\psi_{R_1}\left(X_t^i - \mathbf{z}_t^i\right)\Delta t + \sigma_1 D_{R_1}^{\delta_1}\left(X_t^i - \mathbf{z}_t^i\right) W_{i,t}^X$;

  end

  $t \leftarrow t + 1$;

end

$(\bar{X}, \bar{\mathbf{v}}) \leftarrow \left(\frac{1}{N}\sum_{i=1}^N X_{\mathcal{T}_x}^i,\ \frac{1}{N}\sum_{i=1}^N \mathbf{v}_{\mathcal{T}_y}^i\right)$;

$X^* \leftarrow \sum_{r=1}^N X_{\mathcal{T}_x}^r \frac{\omega_\alpha\left(X_{\mathcal{T}_x}^r, \frac{1}{\kappa}\bar{\mathbf{v}}\right)}{\sum_{\ell=1}^N \omega_\alpha\left(X_{\mathcal{T}_x}^\ell, \frac{1}{\kappa}\bar{\mathbf{v}}\right)}$;

$Y^* \leftarrow \frac{1}{\kappa}\sum_{r=1}^N \mathbf{v}_{\mathcal{T}_y}^r \frac{\varpi_\beta\left(\bar{X}, \frac{1}{\kappa}\mathbf{v}_{\mathcal{T}_y}^r\right)}{\sum_{\ell=1}^N \varpi_\beta\left(\bar{X}, \frac{1}{\kappa}\mathbf{v}_{\mathcal{T}_y}^\ell\right)}$;

Return (X^*, Y^*) ; `/* Approximation of solution to (1.1) */`

In `Algorithm 1`, the weight function are defined according to the objective functions of upper and lower level optimization problems (see (2.4)-(2.6)), i.e.,

$$\varpi_\beta(x,y) = \exp(-\beta G(x,y)) \quad \text{and} \quad \omega_\alpha(x,y) = \exp(-\alpha F(x,y)).$$

Remark 3.4. It is important to note that for running `Algorithm 1`, we do not need to have as many y-particles as x-particles, in particular we have $M \ll N$. Moreover, it is not necessary to run the dynamics of the y-particles for a long time, that is, an approximate value of their consensus is already enough for the algorithm to perform well. Hence, we can choose $T_y \ll T_x$. This can be justified by the fact that *errors* in the computation of the consensus for the y-particles, can be tamed during the averaging procedure used for computing the consensus of the x-particles. On the other hand, the `for` loop can be computed in parallel, since the groups of particles $\mathbf{Y}_i = \{Y^1_{\mathbf{X}_i}, \cdots, Y^M_{\mathbf{X}_i}\}$ are independent from $\mathbf{Y}_k$ for any $k \neq i$.

Remark 3.5. Although Theorem 3.1 establishes the convergence of the model (3.1) to an averaged dynamics as $\varepsilon \to 0$ under the assumption $\kappa \in (0,1)$ small, numerical results suggest that this smallness assumption is not necessary for the convergence to hold. Indeed, in practice (see `Algorithm 1`), we implement an approximation of the averaging with respect to the invariant measure (see the discussion at the beginning of Section 3.3). In other words, we do not implement the *effective* (averaged) dynamics which is the one obtained at the limit $\varepsilon \to 0$, because this would require an explicit formula for the invariant measure which is difficult to construct (analytically). To bypass this difficulty, what we implement instead is an approximation of this limiting behavior. This approximation is obtained with a finitely–small $\varepsilon > 0$. Hence, there is no need to restrict the choice of the parameter κ in practice.

4. Multiscale CBO for multi-level optimization

4.1. **The general strategy.** The bi-level approach presented in the previous sections can be extended to tackle multi-level optimization problems of the form

$$
\begin{aligned}
&\min_{x_0 \in \mathbb{R}^{d_0}} \; F_0(x_0, y_1, \dots, y_n) \\
&\quad \text{s.t.} \;\; y_1 \in \operatorname*{argmin}_{y \in \mathbb{R}^{d_1}} F_1(x_0, y, y_2, \dots, y_n) \\
&\qquad \text{s.t.} \;\; y_2 \in \operatorname*{argmin}_{y \in \mathbb{R}^{d_2}} \; F_2(x_0, y_1, y, y_3, \dots, y_n) \\
&\qquad\qquad \ddots \qquad\qquad \ddots \\
&\qquad\qquad\quad \text{s.t.} \;\; y_n \in \operatorname*{argmin}_{y \in \mathbb{R}^{d_n}} \; F_n(x_0, \dots, y_{n-1}, y)
\end{aligned}
\tag{4.1}
$$

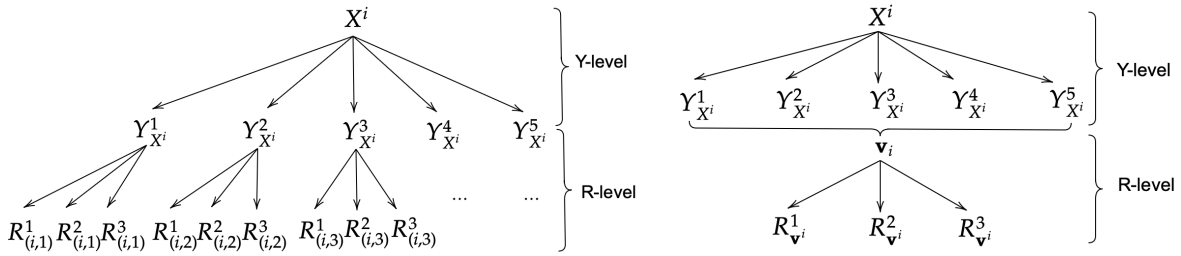
In this case, the multiscale CBO model would have as many scales as the number of levels in (4.1). Its system of interacting SDEs is analogous to (3.1), that is

$$
\begin{cases}
\mathrm{d}\mathbb{X}_t = \mathscr{F}(\mathbb{X}_t, \mathbb{Y}^1_t, \dots, \mathbb{Y}^n_t)\,\mathrm{d}t + \mathscr{G}(\mathbb{X}_t, \mathbb{Y}^1_t, \dots, \mathbb{Y}^n_t)\,\mathrm{d}\mathbb{W}^0_t \\
\mathrm{d}\mathbb{Y}^1_t = \dfrac{1}{\varepsilon}\mathscr{B}_1(\mathbb{X}_t, \mathbb{Y}^1_t, \dots, \mathbb{Y}^n_t)\,\mathrm{d}t + \dfrac{1}{\sqrt{\varepsilon}}\mathscr{H}_1(\mathbb{X}_t, \mathbb{Y}^1_t, \dots, \mathbb{Y}^n_t)\,\mathrm{d}\mathbb{W}^1_t \\
\qquad \vdots \\
\mathrm{d}\mathbb{Y}^n_t = \dfrac{1}{\varepsilon^n}\mathscr{B}_n(\mathbb{X}_t, \mathbb{Y}^1_t, \dots, \mathbb{Y}^n_t)\,\mathrm{d}t + \dfrac{1}{\sqrt{\varepsilon^n}}\mathscr{H}_n(\mathbb{X}_t, \mathbb{Y}^1_t, \dots, \mathbb{Y}^n_t)\,\mathrm{d}\mathbb{W}^n_t.
\end{cases}
\tag{4.2}
$$

The analysis can be performed in a cascade manner: First we consider $\mathbb{Y}^n_\cdot$ as being the only *fast* variable (while $\mathbb{X}_\cdot, \mathbb{Y}^1_\cdot, \dots, \mathbb{Y}^{n-1}_\cdot$ are frozen), this yields an averaged system made of the interacting SDEs $\hat{\mathbb{X}}_\cdot, \hat{\mathbb{Y}}^1_\cdot, \dots, \hat{\mathbb{Y}}^{n-1}_\cdot$ (these are analogously defined as in (3.3)). Then we repeat the process considering $\hat{\mathbb{Y}}^{n-1}_\cdot$ as being the only *fast* variable (while $\hat{\mathbb{X}}_\cdot, \hat{\mathbb{Y}}^1_\cdot, \dots, \hat{\mathbb{Y}}^{n-2}_\cdot$ are frozen), and so on.

4.2. **Tri-level optimization problems.** We consider the tri-level optimization problem (1.2) as an example of the multi-level problem. To generalize the algorithm in Section 3, we consider N particles $\{X^i_\cdot\}_{i=1}^N$ in $\mathbb{R}^n$, where n, N are fixed integers. To each particle $X^i_\cdot \in \mathbb{R}^n$ with $1 \leq i \leq N$, we associate

a set of M particles $(Y^1_{X^i,\cdot},\dots,Y^M_{X^i,\cdot}) \in (\mathbb{R}^m)^M$, where $Y^j_{X^i,\cdot} \in \mathbb{R}^m$ for any $1 \le j \le M$. Then, for every particle $Y^j_{X^i,\cdot} \in \mathbb{R}^m$ with $1 \le i \le N$ and $1 \le j \le M$, we associate a set of P particles $(R^1_{(i,j),\cdot},\dots,R^P_{(i,j),\cdot}) \in (\mathbb{R}^p)^P$, where $R^k_{(i,j),\cdot} \in \mathbb{R}^p$ for any $1 \le k \le P$. The subscript (i,j) refers to the dependency on both X^i and $Y^j_{X^i}$. This corresponds to `Approach 1` in Figure 2.



(A) `Approach 1` (B) `Approach 2`

FIGURE 2. Particle systems in tri-level optimization algorithm with $M = 5$ particles for Y-system and $P = 3$ particles for R-system. `Y-level` corresponds to the Y-particles solving the second level of optimization. In both approaches, they correspond to each fixed X-particle (here X^i). `R-level` corresponds to the R-particles solving the third level of optimization. In `Approach 1`, they correspond to each fixed Y-particle, whereas in `Approach 2`, they correspond to a representative (consensus) of the Y-particles.

However, in practice, this straightforward generalization is computationally expensive, since we need N particles in X-system, $N \times M$ particles in Y-system and $N \times M \times P$ particles in R-system. Therefore, for tri-level optimization problems, we propose `Approach 2` as illustrated in Figure 2. In the latter, we shall consider N particles $\{X^i_\cdot\}_{i=1}^N$ in $\mathbb{R}^n$, and M particles $(Y^1_{X^i,\cdot},\dots,Y^M_{X^i,\cdot}) \in (\mathbb{R}^m)^M$ associated to each particle $X^i_\cdot$ as for bi-level problems. Then, we compute the consensus point $\mathbf{v}^i$ of $\{Y^j_{X^i,\cdot}\}_{j=1}^M$, for $1 \le i \le N$, to which we associate a set of P particles $\{R^k_{\mathbf{v}^i,\cdot}\}_{k=1}^P \in (\mathbb{R}^p)^P$. The dynamics now only have N particles in X-system, $N \times M$ particles in Y-system and $N \times P$ particles in R-system.

The pseudo-code corresponding to `Approach 2` in Figure 2 is `Algorithm 3`, where the weight functions $\omega_i = \omega_i(x,y,r), i = 1,2,3$, for some $\alpha_1,\alpha_2,\alpha_3 > 0$ are

$$\omega_1 = \exp(-\alpha_1 F(x,y,r)),\ \omega_2 = \exp(-\alpha_2 G(x,y,r)),\ \omega_3 = \exp(-\alpha_3 E(x,y,r)).$$

Similar to `Algorithm 1` for bi-level problems, we can choose $M \ll N$, and $P \ll N$. Additionally, we set for simplicity $\kappa = 1$. We also introduce the notation $\mathbf{D} \equiv \mathbb{D}^\delta_Q$ and $\Psi \equiv \psi_Q$, which are the truncation functions defined in §3.1. Here, the truncation parameter (denoted therein by R) is now a fixed constant $Q > 0$. To gain more clarity in `Algorithm 3`, for $1 \le i \le N$ and $t \ge 0$ we shall use the notation

$$\begin{aligned} \mathbf{Y}_{i,t} &:= \mathbf{Y}_{\mathbf{X}^i,t} = (Y^1_{\mathbf{X}^i,t},\dots,Y^M_{\mathbf{X}^i,t}) \in (\mathbb{R}^m)^M, \\ \mathbf{R}_{i,t} &:= \mathbf{R}_{\mathbf{v}^i,t} = (R^1_{\mathbf{v}^i,t},\dots,R^P_{\mathbf{v}^i,t}) \in (\mathbb{R}^p)^P, \\ &\qquad \text{where } \mathbf{v}^i \text{ is the consensus of } \{Y^j_{\mathbf{X}^i,\cdot}\}_{j=1}^M \text{ in (3.2)}. \end{aligned} \tag{4.3}$$

As described in §4.1, we proceed in a cascade manner: We focus first on the lowest level (the R-particles) while keeping the other (X,Y)-particles fixed. We run the standard CBO algorithm for the R-particles. Then we move to the next higher level (the Y-particles). Here we must take the averaged consensus of the Y-particles with respect to the already computed consensus of the R-particles. At

this stage, the highest level (X-particles) is still fixed, while the consensus term in the Y-particles is averaged using the (FLA) method so that it includes the effect of the R-particles. Thus we are left with two populations: the X-particles which have remained fixed so far, and the Y-particles whose consensus terms have been averaged according to the (now, disappeared) R-particles. This concludes the first *averaging* method, and reduces the setting from a 3-scale to 2-scale dynamics. We can then continue as in `Algorithm 1`. This procedure is highlighted in `Algorithm 2` where the green lines correspond to what is different from `Algorithm 1`. This is expressed with symbols in `Algorithm 3`.

Algorithm 2: Multiscale CBO for tri-level optimization – in words.

Set parameters: γ, α_1, α_2, α_3, λ, σ, Δt, T_x, T_y, T_r, δ, $\kappa = 1$;
Initialize: $\mathbf{z}_0^i = X_0^i \sim \rho_0,\ i = 1, \dots, N$; $Y_{i,0}^j \sim \mu_0,\ j = 1, \dots, M$; $R_{i,0}^k \sim \gamma_0,\ k = 1, \dots, P$;
$t = 0$, $\mathcal{T}_x = T_x/\Delta t$, $\mathcal{T}_y = T_y/\Delta t$, $\mathcal{T}_r = T_r/\Delta t$;
while $t \leq \mathcal{T}_x$ **do**
  for $i = 1, \dots, N$ **do**
    Initialize time index ($\tau = 0$) for the Y-particles;
    Initialize consensus point $\mathbf{v}_0^i$ of $\mathbf{Y}_{i,0}$;
    while $\tau \leq \mathcal{T}_y$ **do**
      Initialize time index ($s = 0$) for the R-particles;
      Initialize consensus point $\mathbf{r}_0^i$ of $\mathbf{R}_{i,0}$;
      while $s \leq \mathcal{T}_r$ **do**
        Run CBO for the R-particles;
        Compute consensus point $\mathbf{r}_s^i$ of of $\mathbf{R}_{i,s}$;
        Update the *averaged* consensus of the Y-particles (using (FLA));
        Update time index of R-particles ($s \leftarrow s + 1$);
      end
      (Note that $s = \mathcal{T}_r$ here);
      Run CBO for the Y-particles;
      Compute consensus point $\mathbf{v}_s^i$ of $\mathbf{Y}_{i,s}$;
      Update the *averaged* consensus of the X-particles (using (FLA));
      Update time index of Y-particles ($\tau \leftarrow \tau + 1$);
    end
  end
  Run CBO for the X-particles;
  Update time index of X-particles ($t \leftarrow t + 1$);
end
Compute the arithmetic mean of $(X_{\mathcal{T}_x}^i)_{i=1}^N$, $(\mathbf{v}_{\mathcal{T}_y}^i)_{i=1}^N$, and $(\mathbf{r}_{\mathcal{T}_r}^i)_{i=1}^N$;
(The latter are needed to compute the weights in the consensus next);
Compute the consensus points of (X^*, Y^*, R^*);
Return (X^*, Y^*, R^*) ; `/* Approximation of solution to (1.2) */`

Recalling the notation in (4.3), the algorithm for tri-level optimization problem can be expressed as follows.

Algorithm 3: Multiscale CBO for tri-level optimization.

Set parameters: $\gamma,\ \alpha_1,\ \alpha_2,\ \alpha_3,\ \lambda,\ \sigma,\ \Delta t,\ T_x,\ T_y,\ T_r,\ \delta,\ \kappa = 1$;

Initialize: $\mathbf{z}_0^i = X_0^i \sim \rho_0,\ i = 1,\dots,N$; $Y_{i,0}^j \sim \mu_0,\ j = 1,\dots,M$; $R_{i,0}^k \sim \gamma_0,\ k = 1,\dots,P$; $t = 0$, $\mathcal{T}_x = T_x/\Delta t$, $\mathcal{T}_y = T_y/\Delta t$, $\mathcal{T}_r = T_r/\Delta t$;

while $t \le \mathcal{T}_x$ **do**

  for $i = 1,\dots,N$ **do**

    $\tau = 0$;

    $\mathbf{v}_\tau^i = \sum_{j=1}^M Y_{i,\tau}^j \frac{\omega_2(X_t^i, Y_{i,\tau}^j, \frac{1}{P}\sum_{k=1}^P R_{i,0}^k)}{\sum_{\ell=1}^M \omega_2(X_t^i, Y_{i,\tau}^\ell, \frac{1}{P}\sum_{k=1}^P R_{i,0}^k)}$; /* Consensus of $\mathbf{Y}_{i,0}$ */

    while $\tau \le \mathcal{T}_y$ **do**

      $s = 0$;

      $\mathbf{r}_s^i \leftarrow \sum_{k=1}^P R_{i,s}^k \frac{\omega_3(X_t^i, \mathbf{v}_\tau^i, R_{i,s}^k)}{\sum_{\ell=1}^M \omega_3(X_t^i, \mathbf{v}_{i,\tau}^M, R_{i,s}^\ell)}$; /* Consensus of $\mathbf{R}_{i,0}$ */

      while $s \le \mathcal{T}_r$ **do**

        $\mathbf{R}_{i,s} \leftarrow \mathbf{R}_{i,s} - \lambda\,\Psi\left(\mathbf{R}_{i,s} - \mathbf{r}_s^i\right)\Delta t + \sigma \mathbf{D}\left(\mathbf{R}_{i,s} - \mathbf{r}_s^i\right) W_{i,s}^{\mathbf{R}}$;

        $\mathbf{r}_s^i \leftarrow \sum_{k=1}^P R_{i,s}^k \frac{\omega_3(X_t^i, \mathbf{v}_\tau^i, R_{i,s}^k)}{\sum_{\ell=1}^P \omega_3(X_t^i, \mathbf{v}_\tau^i, R_{i,s}^\ell)}$; /* consensus of $\mathbf{R}_{i,s}$ */

        $\mathbf{v}' \leftarrow \sum_{j=1}^M Y_{i,\tau}^j \frac{\omega_2(X_t^i, Y_{i,\tau}^j, \mathbf{r}_s^i)}{\sum_{\ell=1}^M \omega_2(X_t^i, Y_{i,\tau}^\ell, \mathbf{r}_s^i)}$;

        $\mathbf{v}_\tau^i \leftarrow (1-\gamma)\mathbf{v}_\tau^i + \gamma \mathbf{v}'$; /* FLA of $\mathbf{Y}_{i,\tau}$ */

        $s \leftarrow s + 1$;

      end

      $\mathbf{Y}_{i,\tau} \leftarrow \mathbf{Y}_{i,\tau} - \lambda\,\Psi\left(\mathbf{Y}_{i,\tau} - \mathbf{v}_\tau^i\right)\Delta t + \sigma \mathbf{D}\left(\mathbf{Y}_{i,\tau} - \mathbf{v}_\tau^i\right) W_{i,\tau}^{\mathbf{Y}}$;

      $\mathbf{v}_\tau^i = \sum_{j=1}^M Y_{i,\tau}^j \frac{\omega_2(X_t^i, Y_{i,\tau}^j, \mathbf{r}_s^i)}{\sum_{\ell=1}^M \omega_2(X_t^i, Y_{i,\tau}^\ell, \mathbf{r}_s^i)}$; /* Note that $s = \mathcal{T}_r$ */

      $\mathbf{z}' \leftarrow \sum_{q=1}^N X_t^q \frac{\omega_1(X_t^q, \mathbf{v}_\tau^i, \mathbf{r}_s^i)}{\sum_{r=1}^N \omega_1(X_t^r, \mathbf{v}_\tau^i, \mathbf{r}_s^i)}$;

      $\mathbf{z}_t^i \leftarrow (1-\gamma)\mathbf{z}_t^i + \gamma \mathbf{z}'$; /* FLA of X_t^i; it depends on i */

      $\tau \leftarrow \tau + 1$

    end

  end

  $\mathbf{X}_t^i \leftarrow \mathbf{X}_t^i - \lambda\Psi\left(\mathbf{X}_t^i - \mathbf{z}_t^i\right)\Delta t + \sigma \mathbf{D}\left(\mathbf{X}_t^i - \mathbf{z}_t^i\right) W_{i,t}^{\mathbf{X}}$;

  $t \leftarrow t + 1$;

end

$(\bar{X},\ \bar{\mathbf{v}},\ \bar{\mathbf{r}}) \leftarrow \left(\frac{1}{N}\sum_{i=1}^N X_{\mathcal{T}_x}^i,\ \frac{1}{N}\sum_{i=1}^N \mathbf{v}_{\mathcal{T}_y}^i,\ \frac{1}{N}\sum_{i=1}^N \mathbf{r}_{\mathcal{T}_r}^i\right)$;

$X^* \leftarrow \sum_{k=1}^N X_{\mathcal{T}_x}^k \frac{\omega_1(X_{\mathcal{T}_x}^k, \bar{\mathbf{v}}, \bar{\mathbf{r}})}{\sum_{\ell=1}^N \omega_1(X_{\mathcal{T}_x}^\ell, \bar{\mathbf{v}}, \bar{\mathbf{r}})}$;

$Y^* \leftarrow \sum_{k=1}^N \mathbf{v}_{\mathcal{T}_y}^k \frac{\omega_2(\bar{X}, \mathbf{v}_{\mathcal{T}_y}^k, \bar{\mathbf{r}})}{\sum_{\ell=1}^N \omega_2(\bar{X}, \mathbf{v}_{\mathcal{T}_y}^\ell, \bar{\mathbf{r}})}$;

$R^* \leftarrow \sum_{k=1}^N \mathbf{r}_{\mathcal{T}_r}^k \frac{\omega_3(\bar{X}, \bar{\mathbf{v}}, \mathbf{r}_{\mathcal{T}_r}^k)}{\sum_{\ell=1}^N \omega_3(\bar{X}, \bar{\mathbf{v}}, \mathbf{r}_{\mathcal{T}_r}^\ell)}$;

Return (X^*, Y^*, R^*) ; /* Approximation of solution to (1.2) */

5. Numerical experiments

In this section, we first assess the performance of the multiscale CBO algorithm for bi-level and tri-level optimization problems. Then, we compare the performance of multiscale CBO methods with the CBO methods proposed by [47] in solving min-max optimization problems. Lastly, we conduct a sensitivity analysis of the `MS-CBO` parameters specifically for bi-level optimization problems. We shall construct multi-level optimization problems using the following benchmark functions:

(1) Ackley function [2]**:**

$$A(x) := -20\exp\left(-0.2\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}\right) - \exp\left(\frac{1}{d}\sum_{i=1}^{d}\cos\left(2\pi x_i\right)\right) + \exp(1) + 20.$$

(2) Rastrigin function [60]**:**

$$R(x) := \sum_{i=1}^{d} x_i^2 + 1.5(1-\cos\left(2\pi x_i\right)).$$

(3) Levy function:

$$\begin{aligned} L(x) :=& \sin^2\left(\pi\left(1+\frac{x_1}{4}\right)\right) + \sum_{i=1}^{d-1}\left(\frac{x_i}{4}\right)^2\left[1+10\sin^2\left(\pi\left(1+\frac{x_i}{4}\right)+1\right)\right] \\ &+ \left(\frac{x_d}{4}\right)^2\left[1+\sin^2\left(2\pi\left(1+\frac{x_i}{4}\right)\right)\right], \qquad d \geq 2. \end{aligned}$$

All the benchmarks have global minimizer $x^* = (0, \cdots, 0)$. In the following, we perform 100 Monte Carlo simulations to compute the success rate of the multiscale CBO algorithm and expectation of the error, which is defined as

$$\texttt{error} := \|X^* - x^*\|_2 + \|Y^* - y^*\|_2$$

with output (X^*, Y^*) for bi-level and min-max optimization algorithms, (x^*, y^*) is the true solution. For tri-level optimization problem, given the output (X^*, Y^*, R^*) of `Algorithm 3`, the error is defined analogously as

$$\texttt{error} := \|X^* - x^*\|_2 + \|Y^* - y^*\|_2 + \|R^* - r^*\|_2,$$

where (x^*, y^*, r^*) is the true solution of the tri-level optimization problem. We consider a run is successful if

$$\texttt{error} \leq 0.25, \tag{5.1}$$

following what has previously been done in [52, 59]. All numerical experiments were implemented using Matlab R2024b on a MacBook Pro equipped with an Apple M4 chip and 16 GB of RAM. Matlab codes for `MS-CBO` and the numerical tests presented in this paper are available in the GitHub repository:
https://github.com/AmberYuyangHuang/Multi-scale-CBO

5.1. **Bi-level optimization.** In this section, we consider the bi-level optimization problem (1.1). The particle dynamics are discretized using the Euler–Maruyama scheme. The parameters are chosen as

$$\begin{aligned} &\alpha = \beta = 10^{15}, \quad T_x = 50, \quad T_y = 0.5, \quad \Delta t = \Delta\tau = 0.1 \\ &N = 100,\ M = 25, \quad \lambda_1 = \lambda_2 = 1, \quad \sigma_1 = \sigma_2 = 2, \quad \gamma = 0.75, \\ &\delta_1 = \delta_2 = 10^{-5}, \quad R_1 = R_2 = 10, \quad \kappa = 1, \quad n = m = d \end{aligned} \tag{5.2}$$

The initial positions of x-particles and y-particles are sampled from $\rho_0 = \mathcal{U}[-1,3]^d$ and $\mu_0 = \mathcal{U}[-1,3]^d$, respectively.

Remark 5.1. Let us comment on the values in (5.2) for the choice of parameters.

(1) Although theoretical conditions in literature to establish convergence results for single-level anisotropic CBO methods suggest choosing $2\lambda > \sigma^2$ when $\kappa = 1$ in [29], or $\lambda > 16\sigma^2$ when $\kappa < 1$ in [45, Remark 2.6], it appears that in practice, one still need σ to be large enough. This is mainly due to the difference between the mean–field setting in which the convergence analysis is usually performed, and the particle system which is implemented in practice. Additionally, the hierarchical structure in multilevel optimization problems further requires increased exploration, motivating the practical choice of larger values for σ. The need for stronger diffusion has also been observed numerically in related work about saddle point problems [47] and min-max problems [17].
(2) As we have already mentioned in Remark 3.5, the parameter κ can be set to 1 in practice, since we do not implement the average system (3.3) that is obtained at the limit $\varepsilon \to 0$ (and which require $\kappa < 1$). What we do implement instead is an approximation of the averaging procedure which in practice uses (3.1), that is for ε small but positive.
(3) The choice of γ is rather based on empirical observations. It appears that the current consensus point has more accurate information than the previous seemingly because it was computed based on the previous knowledge of the position of the particles. Hence we attribute to the current consensus point a larger weight ($\gamma = 0.75$) than to the previous one ($1 - \gamma$) when computing the averaged consensus with the (FLA) method. This approach is motivated by the more general class of *moving average* methods as known in statistics. See e.g. [75].
(4) The choice of the parameter α is motivated by Laplace principle: Letting α and β go to infinity enhances the concentration properties of the probability density of the process, as it is analogous to Gibbs measure (where α, β would play the role of inverse temperature). Thus we fix it at a very large value to approximate the asymptotic regime (of low temperatures). In practice, the term evolving α and β are computed after a shift such as $\exp(-\alpha(F(x,y) - \min_x F))$ and $\exp(-\beta(G(x,y) - \min_y G))$ which better stabilizes the computations.

In Table 1 we display the numerical results obtained for bi-level optimization problems, `MS-CBO` refers to the multiscale CBO algorithm we propose, where `MS` stands for *multiscale*. The solutions for all the tests are at origin, except for the test (`ii`) whose solution is $x^* = y^* = (1, \cdots, 1)$.

We observe that `Algorithm 1` demonstrates satisfactory accuracy in solving bi-level optimization problems, more importantly when the objective functions are non-convex with multiple local minimizers.

5.2. **Tri-level optimization.** In this section, we consider the tri-level optimization problem (1.2) with $n = m = p = d$. The particles are sampled initially from $\rho_0 = \mu_0 = \gamma_0 = \mathcal{U}[-1, 3]^d$, and the parameters are chosen as

$$\alpha_1 = \alpha_2 = \alpha_3 = 10^{15}, \quad T_x = 50, \quad T_y = T_z = 0.5, \quad \Delta t = 0.1, \quad \kappa = 1$$

$$N = 100, \quad M = 50, \quad P = 25, \quad \lambda = 1, \quad \sigma = 2, \quad \gamma = 0.75, \quad \delta = 10^{-5}, \quad Q = 10.$$

The numerical results obtained for tri-level optimization problems are shown in Table 2. The optimal solution for **Test (A)** and **Test (B)** is $x^* = y^* = r^* = (0, \cdots, 0)$ and the optimal solution for **Test (C)** is $x^* = y^* = r^* = (1, \cdots, 1)$.

The results in Table 2 show that `Algorithm 3` is capable of finding solutions to tri-level optimization problems and achieving satisfactory accuracy.

5.3. **Min-max optimization.** In this section, we consider the min-max optimization problem (1.3), where $F : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, with $\mathcal{X} = \mathbb{R}^d$ and $\mathcal{Y} = \mathbb{R}^d$. We first consider the one dimensional setting to

$d = 10$		MS-CBO	
(i)	$F(x,y) = \sum_{i=1}^d x_i^2 + y_i^2$ $G(x,y) = \sum_{i=1}^d (x_i - y_i)^2$	success rate	100%
		$\mathbb{E}$[error]	1.394×10^{-4}
		running time (s)	1.625
(ii)	$F(x,y) = \sum_{i=1}^d (x_i - 1)^2 + (y_i - 1)^2$ $G(x,y) = \sum_{i=1}^d (x_i - y_i)^2$	success rate	100%
		$\mathbb{E}$[error]	1.353×10^{-4}
		running time (s)	1.616
(iii)	$F(x,y) = \sum_{i=1}^d x_i^2 + y_i^2 + 2x_iy_i$ $G(x,y) = \sum_{i=1}^d (x_i - y_i)^2$	success rate	100%
		$\mathbb{E}$[error]	1.425×10^{-3}
		running time (s)	1.890
(iv)	$F(x,y) = A(x) + A(y)$ $G(x,y) = \sum_{i=1}^d (x_i - y_i)^2$	success rate	100%
		$\mathbb{E}$[error]	1.333×10^{-4}
		running time (s)	2.682
(v)	$F(x,y) = R(x) + R(y) + 2\sum_{i=1}^d x_iy_i$ $G(x,y) = A(x - y)$	success rate	100%
		$\mathbb{E}$[error]	4.311×10^{-3}
		running time (s)	3.379
(vi)	$F(x,y) = L(x) + L(y)$ $G(x,y) = A(x - y)$	success rate	100%
		$\mathbb{E}$[error]	1.519×10^{-4}
		running time (s)	4.396

TABLE 1. The numerical results obtained with `Algorithm 1` for bi-level optimization problems (1.1), where $n = m = d = 10$.

visualize the behaviour of the MS-CBO algorithm. The parameters are chosen as

$$\alpha = \beta = 10^{15}, \quad T_x = 10,\ T_y = 2, \quad \Delta t = \Delta\tau = 0.1, \quad N = 20,\ M = 5, \quad \kappa = 1,$$
$$\lambda_1 = \lambda_2 = 1, \quad \sigma_1 = \sigma_2 = 1, \quad \gamma = 0.75, \quad \delta_1 = \delta_2 = 0.1, \quad R_1 = R_2 = 10.$$

The x- and y-particles are sampled initially from $\rho_0 = \mu_0 = \mathcal{U}[-4, 4]$.

In the following figures, the blue points are the initial configuration $(X_0^i, \mathbf{v}_0^i)_{i=1}^N$, recall the definition of $\mathbf{v}_t^i$ in (2.9), and the red points are the final position of particles $(X_{T_x}^i, \mathbf{v}_{T_y}^i)_{i=1}^N$. The illustration are shown in Figure 3.

Then, we further test the performance of MS-CBO by comparing with existing CBO methods, SP-CBO refers to the CBO algorithm proposed in [47], where SP stands for *saddle points*.

The test functions are defined as

Test (A)		
$F(x,y,r) = \sum_{i=1}^d x_i^2 + y_i^2$ $G(x,y,r) = L(x-y)$ $E(x,y,r) = L(r-y)$	success rate	100%
	$\mathbb{E}[\texttt{error}]$	1.181×10^{-3}
	running time (s)	20.352
Test (B)		
$F(x,y,r) = \sum_{i=1}^d x_i^2 + y_i^2 + (r_i - x_i)^2$ $G(x,y,r) = L(x-y)$ $E(x,y,r) = R(r-y)$	success rate	94%
	$\mathbb{E}[\texttt{error}]$	1.118×10^{-1}
	running time (s)	19.092
Test (C)		
$F(x,y,r) = \sum_{i=1}^d (x_i-1)^2 + (y_i-1)^2 + (r_i-1)^2$ $G(x,y,r) = \sum_{i=1}^d (y_i - x_i)^2$ $E(x,y,r) = \sum_{i=1}^d (r_i - y_i)^2$	success rate	100%
	$\mathbb{E}[\texttt{error}]$	2.110×10^{-4}
	running time (s)	9.384

TABLE 2. The numerical results obtained with `Algorithm 3` for tri-level optimization problems (1.2), where $n = m = p = d = 10$.

(a) Ackley function:

$$F(x,y) = A(x) - A(y),$$

(b) A non-separable (NS) Rastrigin function:

$$F(x,y) = R(x) - R(y) - 2\sum_{i=1}^d x_i y_i,$$

(c) Levy function:

$$F(x,y) = L(x) - L(y),$$

(d) A non-separable function:

$$F(x,y) = \sum_{i=1}^d x_i^2 - \sum_{i=1}^d y_i^2 - 2\sum_{i=1}^d x_i y_i.$$

The parameters of both `MS-CBO` and `SP-CBO` are chosen as in (5.2).

Note that in `SP-CBO`, the number of particles in both the X- and Y-systems are N and time horizon is chosen as $T = T_x = 50$. The numerical experiments for the `MS-CBO` algorithm are divided into two cases: when $\kappa = 1$ and when $\kappa = 1/\sqrt{M} - 0.01$. The numerical results of min-max optimization problems are shown in Table 3.

As mentioned in Remark 3.4 and Section 5.4, we implement the `for` loop of y-particles in parallel with $T_y \ll T_x$. Although the running time of `MS-CBO` is larger than the running time of `SP-CBO`, the advantage is at the level of the numerical complexity, since we have $\mathcal{O}(N^2)$ for `SP-CBO`, whereas `MS-CBO` is of $\mathcal{O}(NM)$ with $M \ll N$. This would be beneficial when the underlying dimension d grows, or when the structure of the optimization problem becomes more complex (e.g. multi-level problems) for

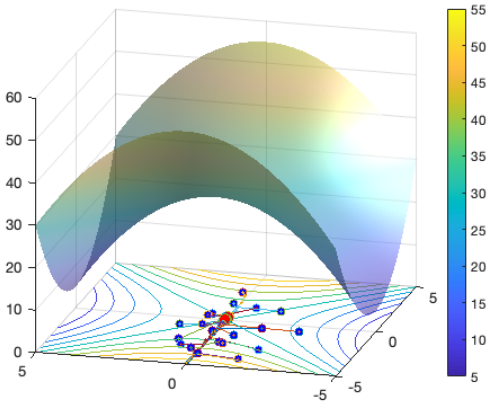


Figure 3.1 $F(x,y) = x^2 - y^2$

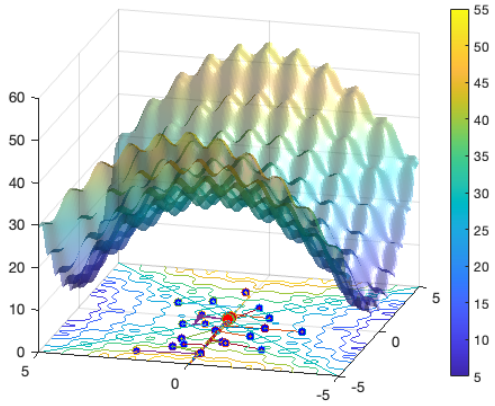


Figure 3.2 $F(x,y) = R(x) - R(y)$

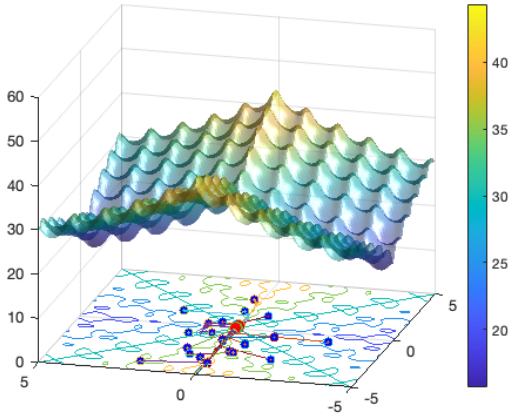


Figure 3.3 $F(x,y) = A(x) - A(y)$

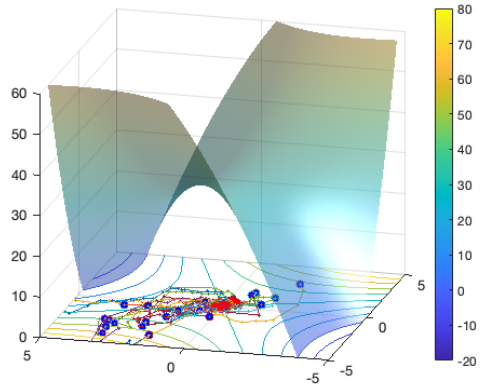


Figure 3.4 $F(x,y) = x^2 - y^2 - 2xy$

FIGURE 3. Illustration of the dynamics for multiscale CBO. The graphs are shifted by +30 units.

which a larger number of particles would be needed, and consequently a lower numerical complexity would be desirable.

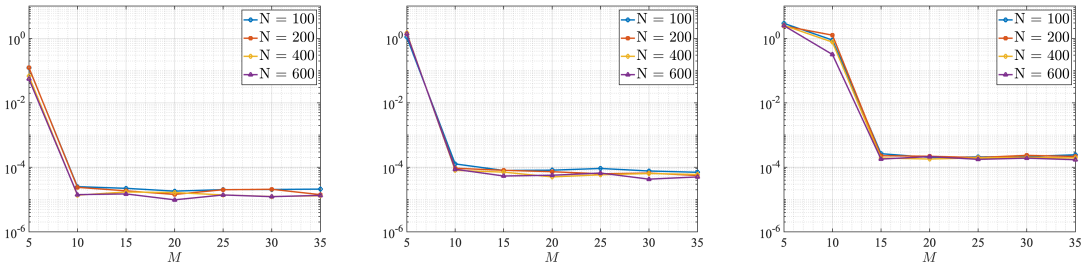
To further explore the performance of the algorithm `MS-CBO`, we now choose different M and N, while the other parameters are kept as in the previous test. The results we obtain in this setting are summarized in Figure 4.

Although optimal choices of M and N depend on the specific optimization problem, the numerical results in Figure 4 suggest that selecting $M \ll N$ is sufficient for solving min-max problems effectively, even in relatively high-dimension. These results will be further investigated in the parameter sensitivity analysis presented in the next section.

5.4. **Parameters sensitivity analysis.** The performance of `MS-CBO` depends on the choice of parameters. Therefore it is worth investigating how the parameters M, N, T_x, T_y, λ, and σ impact the accuracy of the algorithm. Figure 5 shows this sensitivity. First, we fix $N = 100$ and $T_y = 0.5$, and examine the influence of changing M and T_x on accuracy for all test functions in Section 5.1.

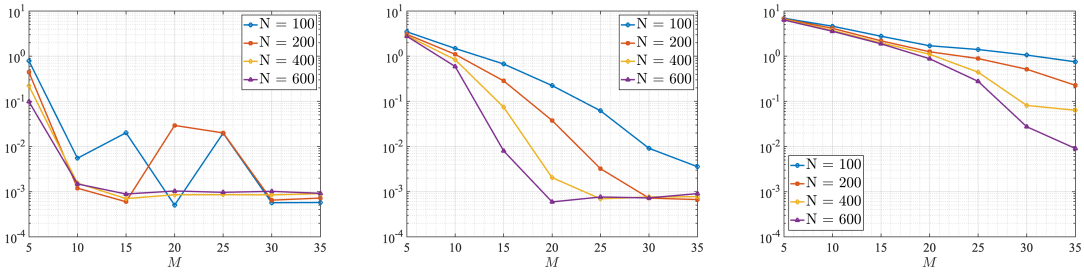
$d = 10$		MS-CBO		SP-CBO
		$\kappa = 1$	$\kappa = 1/\sqrt{M} - 0.01$	
Function **(a)** Ackley	success rate	100%	100%	99%
	$\mathbb{E}$[error]	7.689×10^{-5}	1.454×10^{-4}	8.714×10^{-3}
	running time (s)	2.991	2.939	0.011
Function **(b)** NS-Rastrigin	success rate	97%	77%	5%
	$\mathbb{E}$[error]	4.159×10^{-2}	0.245	2.597
	running time (s)	3.401	3.107	0.013
Function **(c)** Levy	success rate	100%	100%	99%
	$\mathbb{E}$[error]	9.007×10^{-5}	1.653×10^{-4}	2.664×10^{-2}
	running time (s)	4.736	4.734	0.015
Function **(d)** NS	success rate	100%	100%	100%
	$\mathbb{E}$[error]	1.964×10^{-3}	2.466×10^{-3}	1.474×10^{-3}
	running time (s)	2.019	2.007	0.008

TABLE 3. The numerical results of min-max optimization problems



$d = 5$ $d = 10$ $d = 20$

Figure 4.1 Function **(a)**



$d = 5$ $d = 10$ $d = 20$

Figure 4.2 Function **(b)**

FIGURE 4. Numerical results of the MS-CBO method with $\kappa = 1$ applied to min-max optimization problems. Results are shown for benchmark functions **(a)** (Figure 4.1) and **(b)** (Figure 4.2) with dimensions $d = 5, 10, 20$. Each subfigure illustrates the $\mathbb{E}$[error] achieved with different choices of M and N.

As shown in Figure 5.1, accuracy improves with ratio T_x/T_y increases, and stabilizes once this ratio is sufficiently large. Similarly, Figure 5.2 is consistent with the earlier observations from the min-max experiments (Figure 4), indicating that M need not be as large as N; in most cases, choosing $M \geq 20$ ensures satisfactory performance of the MS-CBO method. Note that for benchmark function **(iv)**,

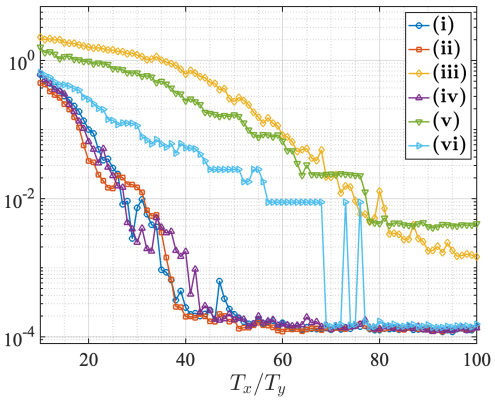


Figure 5.1 Varying parameter T_x/T_y

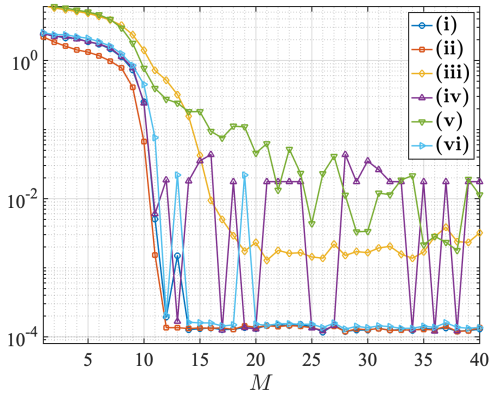


Figure 5.2 Varying parameter M

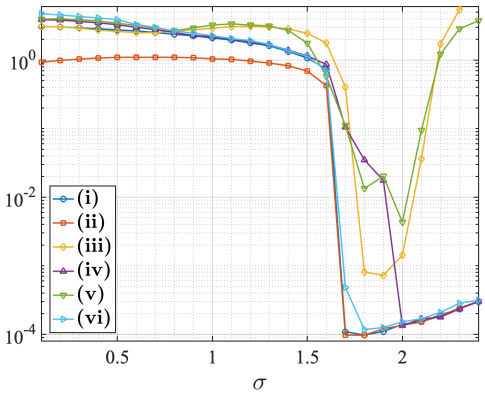


Figure 5.3 Varying parameter σ

FIGURE 5. $\mathbb{E}[\texttt{error}]$ for benchmark objectives **(i)-(vi)** (defined in Table 1) as a function of the parameter T_x/T_y (Figure 5.1), M (Figure 5.2) and σ (Figure 5.3). Only the indicated parameter is varied; all other parameters remain the same as (5.2).

the accuracy exhibits fluctuations as M increases. This occurs because, despite high success rates (98%–100%), a few unsuccessful runs significantly increase the averaged error. However, apart from these rare unsuccessful runs, the algorithm consistently solves the problem effectively.

Another important pair of parameters to choose is λ and σ. As mentioned in Remark 5.1 (1), in multi-level optimization problems, our empirical observations suggest that the optimal choice of σ differs from literature. Specifically, as can be seen in Figure 5.3, when fixing $\lambda = 1$, optimal values of σ lie in the interval $[1.8, 2.1]$. Values of σ below this range introduce insufficient randomness, while values above it cause excessive diffusion, both negatively impact the accuracy. All these results offer practical guidance for selecting parameters in applications of `MS-CBO` method.

6. Summary and conclusion

We have developed a multiscale version of the CBO algorithm, specifically designed for solving multi-level optimization problems. The main idea relies on a singularly perturbed system of SDEs: each scale corresponds to a level in the optimization problem. By using the celebrated averaging principle, this system is reduced to a single dynamics whose coefficients are averaged with respect to an invariant measure. The convergence of the original system to the averaged system has been proved in Theorem 3.1, and the presented algorithms have been tested on various optimization problems with satisfactory results.

Some practical modifications on the CBO model have been made in order to allow the averaging (obtained when $\varepsilon \to 0$) to be proved, in particular using a parameter κ to strengthen the drift term. In Theorem 3.1, this parameter is assumed to be small in order to guarantee the averaging. However, this is not a compulsory requirement in numerical implementation. Therefore, a sharper convergence analysis for broader choice of κ could be worth investigating. Another natural direction for further investigation is to explore the convergence guarantees of mean-field dynamics to the global solution of the optimization problem. We can study the mean-field limit of the averaged (effective) model (3.3) as $N \to \infty$, similar to the theoretical analysis of standard CBO methods. Yet, this is a challenging topic as difficulty arises from the invariant measure and its regularity with respect to the x-variable. On the other hand, the criteria of choosing optimal parameters in different dynamics remain to be understood, e.g., a more precise relationship is to be established between M and N, T_x and T_y in bi-level problems. Lastly, in both Algorithms 1 and 3, we used the approximation (FLA) to handle the averaging with respect to the invariant measure. But we believe the implementation of the two algorithms could be further improved if one improves the computation of the invariant measure or the averaging with its respect.

Acknowledgments. The authors wish to thank the two referees for their valuable comments which helped improve the paper.

Appendix A. Proof of Theorem 3.1

We start by defining some functional spaces that we shall use.

- If $\theta \in (0,1)$, then $C_b^\theta(\mathbb{R}^d)$ denotes the usual space of locally Hölder continuous functions with exponent θ, and that are moreover bounded.
- If $\theta = 1$, then $C_b^1(\mathbb{R}^d)$ denotes the space of bounded and Lipschitz continuous functions.
- If $\theta \in (1,2)$, then $C_b^\theta(\mathbb{R}^d)$ denotes the space of functions that are in $C_b^1(\mathbb{R}^d)$ and whose first-order derivatives are in $C_b^{\theta-1}(\mathbb{R}^d)$, noting that $\theta - 1 \in (0,1)$.
- If $\theta = 2$, then $C_b^2(\mathbb{R}^d)$ denotes the space of bounded functions whose first-order derivatives are Lipschitz continuous.

Now let $(x,y) \in \mathbb{R}^{d_1} \times \mathbb{R}^{d_2}$ for some $d_1, d_2 \in \mathbb{N}$, and let $f = f(x,y)$ be a function of (x,y). We denote by $C_b^{\theta,\delta} := C_b^\theta(\mathbb{R}^{d_1}; C_b^\delta(\mathbb{R}^{d_2}))$ the space functions f such that $x \mapsto f(x,y) \in C_b^\theta(\mathbb{R}^{d_1})$ locally uniformly in y, and $y \mapsto f(x,y) \in C_b^\delta(\mathbb{R}^{d_2})$ locally uniformly in x.

Proof of Theorem 3.1. This shall be a consequence of statement (i) in [63, Theorem 2.3] or, more precisely, of the particular case discussed in statement (i) of Remark 2.5 therein. (A similar result can also be found in [61, Theorem 2.5].)
Therefore, it suffices to check that our system of coupled SDEs (3.1) satisfies the assumptions in [63]. To this end, and for better readability, we first highlight the correspondence between our notation

and the one in [63] as follows:

- in [63] : $X^\varepsilon, Y^\varepsilon, \quad b, \sigma, \quad F, G, \quad c, H, \quad \alpha_\varepsilon.$
- in (3.1) : $\mathbb{Y}^\varepsilon, \mathbb{X}^\varepsilon, \quad \mathscr{B}, \mathscr{H}, \quad \mathscr{F}, \mathscr{G}, \quad 0,\ 0, \quad \sqrt{\varepsilon}.$

We can now state the assumptions needed, and prove that they are satisfied.

(i) $\mathscr{H}\mathscr{H}^*$ is bounded and non-degenerate in y, uniformly with respect to x. *(This corresponds to assumptions (A_σ) in [63].)*

(ii) $\mathscr{G}\mathscr{G}^*$ is bounded and non-degenerate in x, uniformly with respect to y. *(This corresponds to assumptions (A_G) in [63].)*

(iii) The drift of the fast process satisfies a recurrence assumption which ensures the existence of an invariant measure $\Upsilon_x(\mathrm{d}y)$, and that is

$$\lim_{|y|\to+\infty} \sup_x \langle y, \mathscr{B}(x,y)\rangle = -\infty.$$

(This corresponds to assumptions (A_b) in [63].)

(iv) The coefficients $\mathscr{F}, \mathscr{G}$ (of the slow variables $\mathbb{X}^\varepsilon$), and $\mathscr{B}, \mathscr{H}$ (of the fast variables $\mathbb{Y}^\varepsilon$) are in $C_b^{\theta,\delta}$, with $\theta \in (0,1)$ and $\delta \in (0,1)$. *(This is an assumption slightly stronger than the regularity requirement in [63].)*

Our construction of $\mathscr{H}$ and $\mathscr{G}$ in §2.3 guarantee the validity of assumptions (i) and (ii). To check assumption (iii), we proceed as follows. Given $\mathbf{x} = (x_i)_{1\le i\le N}$, where $x_i \in \mathbb{R}^n$, and setting $\mathbb{Y}^\varepsilon =: \mathbb{Y}$ where

$$\mathbb{Y} = (\mathbf{Y}_i)_{1\le i\le N} \in \mathbb{R}^{mMN}, \quad \text{where } \mathbf{Y}_i = (Y^j_{x_i})_{1\le j\le M} \in \mathbb{R}^{mM}, \quad \text{and } Y^j_{x_i} \in \mathbb{R}^m,$$

with $Y^j_{x_i} = (Y^j_{x_i}(k))_{1\le k\le m}$, we have

$$\mathscr{B}(\mathbf{x},\mathbb{Y}) = \big(B(x_i,\mathbf{Y}_i)\big)_{1\le i\le N}, \quad \text{where} \quad B(x_i,\mathbf{Y}_i) = \big(b_i^j(x_i,\mathbf{Y}_i)\big)_{1\le j\le M}$$

and

$$b_i^j(x_i,\mathbf{Y}_i) = -\lambda_2\,\psi_{R_2}\left(Y^j_{x_i} - \kappa\sum_{r=1}^M Y^r_{x_i}\frac{\varpi_\beta\left(x_i,Y^r_{x_i}\right)}{\sum_{\ell=1}^M \varpi_\beta\left(x_i,Y^\ell_{x_i}\right)}\right) \in \mathbb{R}^m$$

using the truncation function defined, for a vector $v \in \mathbb{R}^d$, by

$$\psi_R(v) = \big(\psi_R(v)_k\big)_{1\le k\le d} \quad \text{where } \psi_R(v)_k = \begin{cases} v_k\,, & \text{if } |v_k| \le R \\ \dfrac{v_k}{|v_k|}R\,, & \text{if } |v_k| > R. \end{cases}$$

Then we obtain

$$\langle\, \mathbb{Y},\, \mathscr{B}(x,\mathbb{Y})\,\rangle = \sum_{i=1}^N \langle\, \mathbf{Y}_i,\, B(x_i,\,\mathbf{Y}_i)\,\rangle = \sum_{i=1}^N\sum_{j=1}^M \langle\, Y^j_{x_i},\, b_i^j(x_i,\mathbf{Y}_i)\,\rangle \tag{A.1}$$

and

$$\langle\, Y^j_{x_i},\, b_i^j(x_i,\mathbf{Y}_i)\,\rangle = -\lambda_2\,\langle\, Y^j_{x_i},\, \psi_{R_2}\left(Y^j_{x_i} - \kappa\sum_{r=1}^M Y_i^r\frac{\varpi_\beta\left(x_i,Y^r_{x_i}\right)}{\sum_{\ell=1}^M \varpi_\beta\left(x_i,Y^\ell_{x_i}\right)}\right)\rangle.$$

Notice that, for any $w, v \in \mathbb{R}^d$, we have $\langle w, \psi_{R_2}(v)\rangle = \sum_{\tau=1}^d \langle w, \psi_{R_2}(v)\rangle_\tau$ and

$$\langle w, \psi_{R_2}(v)\rangle_\tau = \begin{cases} w_\tau v_\tau\,, & \text{if } |v_\tau| \le R_2, \\ w_\tau v_\tau\dfrac{R}{|v_\tau|}\,, & \text{if } |v_\tau| > R_2, \end{cases} \qquad \tau = 1,\dots,d.$$

We can then write $\langle w, \psi_{R_2}(v)\rangle_\tau = w_\tau v_\tau\, \xi(v_\tau)$ where $\xi(v_\tau) := 1 \wedge \frac{R_2}{|v_\tau|}$, and $\xi(0) = 1$. Note in particular that $0 < \xi(v_\tau) \leq 1$ for all $v_\tau \in \mathbb{R}$. Therefore we have, for $\tau = 1, \dots, m$

$$\begin{aligned}
&\langle\, Y^j_{x_i},\, b^j_i(x_i, \mathbf{Y}_i)\,\rangle_\tau \\
&\qquad = \Bigg(-\lambda_2\, Y^j_{x_i}(\tau)^2 + \lambda_2\,\kappa\, \sum_{r=1}^M Y^j_{x_i}(\tau) Y^r_{x_i}(\tau) \frac{\varpi_\beta\left(x_i, Y^r_{x_i}\right)}{\sum_{\ell=1}^M \varpi_\beta\left(x_i, Y^\ell_{x_i}\right)} \Bigg) \\
&\qquad\qquad \times \xi\Bigg(Y^j_{x_i}(\tau) - \kappa \sum_{r=1}^M Y^r_{x_i}(\tau) \frac{\varpi_\beta\left(x_i, Y^r_{x_i}\right)}{\sum_{\ell=1}^M \varpi_\beta\left(x_i, Y^\ell_{x_i}\right)} \Bigg).
\end{aligned}$$

Let us denote by $\underline{\xi}$ the following

$$\underline{\xi}(\mathbb{Y}) := \min_{\substack{1\leq\tau\leq m\\ 1\leq j\leq M\\ 1\leq i\leq N}} \left\{ \xi\Bigg(Y^j_{x_i}(\tau) - \kappa \sum_{r=1}^M Y^r_{x_i}(\tau) \frac{\varpi_\beta\left(x_i, Y^r_{x_i}\right)}{\sum_{\ell=1}^M \varpi_\beta\left(x_i, Y^\ell_{x_i}\right)} \Bigg) \right\} \in (0, 1].$$

We have

$$\langle\, Y^j_{x_i},\, b^j_i(x_i, \mathbf{Y}_i)\,\rangle_\tau \leq -\lambda_2\, Y^j_{x_i}(\tau)^2\, \underline{\xi}(\mathbb{Y}) + \lambda_2\,\kappa\, \Theta^j_i(\tau) \tag{A.2}$$

where

$$\begin{aligned}
\Theta^j_i(\tau) := &\Bigg(\sum_{k=1}^M Y^j_{x_i}(\tau) Y^k_{x_i}(\tau) \frac{\varpi_\beta\left(x_i, Y^k_{x_i}\right)}{\sum_{\ell=1}^M \varpi_\beta\left(x_i, Y^\ell_{x_i}\right)} \Bigg) \\
&\times\, \xi\Bigg(Y^j_{x_i}(\tau) - \kappa \sum_{k=1}^M Y^k_{x_i}(\tau) \frac{\varpi_\beta\left(x_i, Y^k_{x_i}\right)}{\sum_{\ell=1}^M \varpi_\beta\left(x_i, Y^\ell_{x_i}\right)} \Bigg).
\end{aligned}$$

Taking the absolute value, and using $|\xi(\cdot)| \leq 1$, one gets

$$\left|\Theta^j_i(\tau)\right| \leq \sum_{k=1}^M \left|Y^j_{x_i}(\tau)\right| \left|Y^k_{x_i}(\tau)\right| \frac{\varpi_\beta\left(x_i, Y^k_{x_i}\right)}{\sum_{\ell=1}^M \varpi_\beta\left(x_i, Y^\ell_{x_i}\right)},$$

and then

$$\sum_{\tau=1}^m \left|\Theta^j_i(\tau)\right| \leq \sum_{k=1}^M \sum_{\tau=1}^m \left|Y^j_{x_i}(\tau)\right| \left|Y^k_{x_i}(\tau)\right| \frac{\varpi_\beta\left(x_i, Y^k_{x_i}\right)}{\sum_{\ell=1}^M \varpi_\beta\left(x_i, Y^\ell_{x_i}\right)}.$$

Using Cauchy–Schwarz inequality yields

$$\begin{aligned}
\sum_{\tau=1}^m \left|\Theta^j_i(\tau)\right| &\leq \sum_{k=1}^M \left(\sum_{\tau=1}^m |Y^j_{x_i}(\tau)|^2 \right)^{\frac{1}{2}} \left(\sum_{\tau=1}^m |Y^k_{x_i}(\tau)|^2 \right)^{\frac{1}{2}} \frac{\varpi_\beta\left(x_i, Y^k_{x_i}\right)}{\sum_{\ell=1}^M \varpi_\beta\left(x_i, Y^\ell_{x_i}\right)} \\
&= \sum_{k=1}^M \|Y^j_{x_i}\|\, \|Y^k_{x_i}\| \frac{\varpi_\beta\left(x_i, Y^k_{x_i}\right)}{\sum_{\ell=1}^M \varpi_\beta\left(x_i, Y^\ell_{x_i}\right)} \\
&\leq \left(\sum_{k=1}^M \|Y^j_i\|^2 \frac{\varpi_\beta\left(x_i, Y^k_{x_i}\right)}{\sum_{\ell=1}^M \varpi_\beta\left(x_i, Y^\ell_{x_i}\right)} \right)^{\frac{1}{2}} \left(\sum_{k=1}^M \|Y^k_{x_i}\|^2 \frac{\varpi_\beta\left(x_i, Y^k_{x_i}\right)}{\sum_{\ell=1}^M \varpi_\beta\left(x_i, Y^\ell_{x_i}\right)} \right)^{\frac{1}{2}} \\
&= \|Y^j_{x_i}\| \left(\sum_{k=1}^M \|Y^k_{x_i}\|^2 \frac{\varpi_\beta\left(x_i, Y^k_{x_i}\right)}{\sum_{\ell=1}^M \varpi_\beta\left(x_i, Y^\ell_{x_i}\right)} \right)^{\frac{1}{2}} \\
&\leq \|Y^j_{x_i}\| \left(\sum_{k=1}^M \|Y^k_{x_i}\|^2 \right)^{\frac{1}{2}} = \|Y^j_{x_i}\|\, \|\mathbf{Y}_i\|.
\end{aligned}$$

The latter inequality together with (A.2) yield

$$\begin{aligned}\langle\, Y^j_{x_i},\, b^j_i(x_i,\mathbf{Y}_i)\,\rangle &= \sum_{\tau=1}^{m}\langle\, Y^j_{x_i},\, b^j_i(x_i,\mathbf{Y}_i)\,\rangle_\tau \\ &\leq -\lambda_2\,\underline{\xi}(\mathbb{Y})\,\|Y^j_{x_i}\|^2 + \lambda_2\,\kappa\,\|Y^j_{x_i}\|\,\|\mathbf{Y}_i\|,\end{aligned}$$

and with (A.1), one obtains (using again Cauchy-Schwarz inequality)

$$\begin{aligned}\langle\, \mathbb{Y},\, \mathscr{B}(x,\mathbb{Y})\,\rangle &= \sum_{i=1}^{N}\sum_{j=1}^{M}\langle\, Y^j_{x_i},\, b^j_i(x_i,\mathbf{Y}_i)\,\rangle \\ &\leq -\lambda_2\,\underline{\xi}(\mathbb{Y})\,\|\mathbb{Y}\|^2 + \lambda_2\,\kappa\sum_{i=1}^{N}\sum_{j=1}^{M}\|Y^j_{x_i}\|\,\|\mathbf{Y}_i\| \\ &\leq -\lambda_2\,\underline{\xi}(\mathbb{Y})\,\|\mathbb{Y}\|^2 + \lambda_2\,\kappa\sum_{i=1}^{N}\big(\|\mathbf{Y}_i\|\big)\,\big(\sqrt{M}\,\|\mathbf{Y}_i\|\big) \\ &= -\lambda_2\,\underline{\xi}(\mathbb{Y})\,\|\mathbb{Y}\|^2 + \lambda_2\,\kappa\,\sqrt{M}\,\|\mathbb{Y}\|^2 = -\lambda_2\big(\underline{\xi}(\mathbb{Y}) - \kappa\sqrt{M}\big)\|\mathbb{Y}\|^2.\end{aligned}$$

Therefore, in order to satisfy assumption (iii), it suffices to have

$$\kappa < \frac{\underline{\xi}(\mathbb{Y})}{\sqrt{M}}.$$

Let us note that $\underline{\xi}(\mathbb{Y})$ tends to 1 when the particles $Y^j_{x_i}$ tends to a consensus, and $\underline{\xi}(\mathbb{Y})$ tends to 0 only when a $Y^j_{x_i}$ gets (infinitely) far from the others, the latter being a behavior that is not likely to happen, provided we choose the radius R_2 of the truncation large enough compared to the closed ball containing the initial position of the particles. This will prevent $\xi(\mathbb{Y})$ from getting smaller, allowing κ to exist.

Finally, the coefficients in (3.1) are bounded and locally Hölder continuous[1] with any exponent in the open interval $(0,1)$, hence the validity of assumption (iv).

Thus we can apply the result in [63] with coefficients that are bounded in both x, y, and locally Hölder continuous with exponents $\theta, \delta \in (0,1)$, which concludes the proof. □

References

[1] Fatima Ezzahra Achamrah, Fouad Riane, and El-Houssaine Aghezzaf. Bi-level programming for modelling inventory sharing in decentralized supply chains. *Transportation Research Procedia*, 62:517–524, 2022.

[2] David H. Ackley. *A connectionist machine for genetic hillclimbing*. Kluwer Academic Publishers, USA, 1987.

[3] Houssam AlRachid, Mireille Bossy, Cristiano Ricci, and Lukasz Szpruch. New particle representations for ergodic McKean-Vlasov SDEs. *ESAIM: Proceedings and Surveys*, 65:68–83, 2019.

[4] Omid Amirtaheri, Mostafa Zandieh, Behrouz Dorri, and AR Motameni. A bi-level programming approach for production-distribution supply chain problem. *Computers & Industrial Engineering*, 110:527–537, 2017.

[5] G Anandalingam. Artificial intelligence based approaches for solving hierarchical optimization problems. *Impacts of recent computer advances on operations research*, pages 289–301, 1989.

[6] Hyeong-Ohk Bae, Seung-Yeal Ha, Myeongju Kang, Hyuncheul Lim, Chanho Min, and Jane Yoo. A constrained consensus based optimization algorithm and its application to finance. *Applied Mathematics and Computation*, 416:126726, 2022.

[7] Martino Bardi and Hicham Kouhkouh. Singular perturbations in stochastic optimal control with unbounded data. *ESAIM: Control, Optimisation and Calculus of Variations*, 29:52, 2023.

[8] Martino Bardi and Hicham Kouhkouh. Deep relaxation of controlled stochastic gradient descent via singular perturbations. *SIAM Journal on Control and Optimization*, 62(4):2229–2253, 2024.

[9] Jean-Francois M Barthelemy and Michael F Riley. Improved multilevel optimization approach for the design of complex engineering systems. *AIAA journal*, 26(3):353–360, 1988.

[10] Erhan Bayraktar, Ibrahim Ekren, and Hongyi Zhou. Uniform-in-time weak propagation of chaos for consensus-based optimization. *arXiv preprint arXiv:2502.00582*, 2025.

[1] In fact, they are locally Lipschitz continuous as can be seen from the results in [47].

[11] Mathieu Besançon, Miguel F Anjos, and Luce Brotcorne. Complexity of near-optimal robust versions of multilevel optimization problems. *Optimization Letters*, 15:2597–2610, 2021.
[12] George D Birkhoff. Proof of the ergodic theorem. *Proceedings of the National Academy of Sciences*, 17(12):656–660, 1931.
[13] Sabrina Bonandin and Michael Herty. Consensus-based algorithms for stochastic optimization problems. *arXiv preprint arXiv:2404.10372*, 2024.
[14] Sabrina Bonandin and Michael Herty. Kinetic variable-sample methods for stochastic optimization problems. *arXiv preprint arXiv:2502.17982*, 2025.
[15] Giacomo Borghi, Michael Herty, and Lorenzo Pareschi. An adaptive consensus based method for multi-objective optimization with uniform pareto front approximation. *Applied Mathematics & Optimization*, 88(2):58, 2023.
[16] Giacomo Borghi, Michael Herty, and Lorenzo Pareschi. Constrained consensus-based optimization. *SIAM Journal on Optimization*, 33(1):211–236, 2023.
[17] Giacomo Borghi, Hui Huang, and Jinniao Qiu. A particle consensus approach to solving nonconvex-nonconcave min-max problems. *arXiv preprint arXiv:2407.17373*, 2024.
[18] Zalán Borsos, Mojmir Mutny, and Andreas Krause. Coresets via bilevel optimization for continual learning and streaming. *Advances in neural information processing systems*, 33:14879–14890, 2020.
[19] Herminia I Calvete and Carmen Galé. A multiobjective bilevel program for production-distribution planning in a supply chain. In *Multiple Criteria Decision Making for Sustainable Energy and Transportation Systems: Proceedings of the 19th International Conference on Multiple Criteria Decision Making, Auckland, New Zealand, 7th-12th January 2008*, pages 155–165. Springer, 2010.
[20] Wilfred Candler, Jose Fortuny-Amat, and Bruce McCarl. The potential role of multilevel programming in agricultural economics. *American Journal of Agricultural Economics*, 63(3):521–531, 1981.
[21] José A Carrillo, Young-Pil Choi, Claudia Totzeck, and Oliver Tse. An analytical framework for consensus-based global optimization method. *Mathematical Models and Methods in Applied Sciences*, 28(06):1037–1066, 2018.
[22] Jean-Francois Chassagneux and Gilles Pagès. Computing the invariant distribution of McKean-Vlasov SDEs by ergodic simulation. *arXiv preprint arXiv:2406.13370*, 2024.
[23] Pratik Chaudhari, Anna Choromanska, Stefano Soatto, Yann LeCun, Carlo Baldassi, Christian Borgs, Jennifer Chayes, Levent Sagun, and Riccardo Zecchina. Entropy-sgd: Biasing gradient descent into wide valleys. *Journal of Statistical Mechanics: Theory and Experiment*, 2019(12):124018, 2019.
[24] Pratik Chaudhari, Adam Oberman, Stanley Osher, Stefano Soatto, and Guillaume Carlier. Deep relaxation: partial differential equations for optimizing deep neural networks. *Research in the Mathematical Sciences*, 5:1–30, 2018.
[25] Jian Chen, Pan Mu, Risheng Liu, Xin Fan, and Zhongxuan Luo. Flexible bilevel image layer modeling for robust deraining. In *2020 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2020.
[26] Caroline Crockett, Jeffrey A Fessler, et al. Bilevel methods for image reconstruction. *Foundations and Trends® in Signal Processing*, 15(2-3):121–289, 2022.
[27] Giuseppe Da Prato and Jerzy Zabczyk. *Ergodicity for infinite dimensional systems*, volume 229. Cambridge university press, 1996.
[28] M. Sajjad Edalatzadeh, Dante Kalise, Kirsten A. Morris, and Kevin Sturm. Optimal actuator design for the euler-bernoulli vibration model based on lqr performance and shape calculus. *IEEE Control Systems Letters*, 6:1334–1339, 2022.
[29] Massimo Fornasier, Hui Huang, Lorenzo Pareschi, and Philippe Sünnen. Anisotropic diffusion in consensus-based optimization on the sphere. *SIAM Journal on Optimization*, 32(3):1984–2012, 2022.
[30] Massimo Fornasier, Timo Klock, and Konstantin Riedl. Consensus-based optimization methods converge globally. *SIAM Journal on Optimization*, 34(3):2973–3004, 2024.
[31] Massimo Fornasier, Peter Richtárik, Konstantin Riedl, and Lukang Sun. Consensus-based optimisation with truncated noise. *European Journal of Applied Mathematics*, 36(2):292–315, 2025.
[32] Massimo Fornasier and Lukang Sun. A PDE framework of consensus-based optimization for objectives with multiple global minimizers. *Comm. in Partial Differential Equations*, pages 1–42, 2025.
[33] Jean-Pierre Fouque, George Papanicolaou, and K Ronnie Sircar. *Derivatives in financial markets with stochastic volatility*. Cambridge University Press, 2000.
[34] Jean-Pierre Fouque, George Papanicolaou, Ronnie Sircar, and Knut Solna. Multiscale stochastic volatility asymptotics. *Multiscale Modeling & Simulation*, 2(1):22–42, 2003.
[35] Jean-Pierre Fouque, George Papanicolaou, Ronnie Sircar, and Knut Solna. Singular perturbations in option pricing. *SIAM Journal on Applied Mathematics*, 63(5):1648–1665, 2003.
[36] Seung-Yeal Ha, Shi Jin, and Doheon Kim. Convergence of a first-order consensus-based global optimization algorithm. *Mathematical Models and Methods in Applied Sciences*, 30(12):2417–2444, 2020.
[37] Seung-Yeal Ha, Shi Jin, and Doheon Kim. Convergence and error estimates for time-discrete consensus-based optimization algorithms. *Numerische Mathematik*, 147:255–282, 2021.
[38] Seung-Yeal Ha, Myeongju Kang, Dohyun Kim, Jeongho Kim, and Insoon Yang. Stochastic consensus dynamics for nonconvex optimization on the Stiefel manifold: Mean-field limit and convergence. *Mathematical Models and Methods in Applied Sciences*, 32(03):533–617, 2022.

[39] Pierre Hansen, Brigitte Jaumard, and Gilles Savard. New branch-and-bound rules for linear bilevel programming. *SIAM Journal on scientific and Statistical Computing*, 13(5):1194–1217, 1992.
[40] Michael Herty and Hicham Kouhkouh. Relaxation and asymptotic expansion of controlled stiff differential equations. *International Journal of Control*, pages 1–15, 2024.
[41] Wei Hong, Shihu Li, and Wei Liu. Strong convergence rates in averaging principle for slow-fast McKean-Vlasov SPDEs. *Journal of Differential Equations*, 316:94–135, 2022.
[42] Wei Hong, Shihu Li, Wei Liu, and Xiaobin Sun. Central limit type theorem and large deviation principle for multi-scale McKean–Vlasov SDEs. *Probability Theory and Related Fields*, pages 1–69, 2023.
[43] Hui Huang and Hicham Kouhkouh. Self-interacting CBO: Existence, uniqueness, and long-time convergence. *Applied Mathematics Letters*, 161:109372, 2025.
[44] Hui Huang and Hicham Kouhkouh. Uniform-in-time mean-field limit estimate for the Consensus-Based Optimization. *ESAIM: Control, Optimisation and Calculus of Variations*, (To appear).
[45] Hui Huang, Hicham Kouhkouh, and Lukang Sun. Faithful global convergence for the rescaled Consensus–Based Optimization. *arXiv preprint arXiv:2503.08578*, 2025.
[46] Hui Huang and Jinniao Qiu. On the mean-field limit for the consensus-based optimization. *Mathematical Methods in the Applied Sciences*, 2022.
[47] Hui Huang, Jinniao Qiu, and Konstantin Riedl. Consensus-based optimization for saddle point problems. *SIAM Journal on Control and Optimization*, 62(2):1093–1121, 2024.
[48] Yuyang Huang, Michael Herty, Dante Kalise, and Nikolas Kantas. Fast and robust consensus-based optimization via optimal feedback control. *arXiv preprint arXiv:2411.03051*, 2024.
[49] Hideaki Iiduka. Iterative algorithm for solving triple-hierarchical constrained optimization problem. *Journal of Optimization Theory and Applications*, 148:580–592, 2011.
[50] Jia-Jia Jiang, Wen-Xue Wei, Wan-Lu Shao, Yu-Feng Liang, and Yuan-Yuan Qu. Research on large-scale bi-level particle swarm optimization algorithm. *IEEE Access*, 9:56364–56375, 2021.
[51] Dante Kalise, Karl Kunisch, and Kevin Sturm. Optimal actuator design based on shape calculus. *Mathematical Models and Methods in Applied Sciences*, 28(13):2667–2717, 2018.
[52] Dante Kalise, Akash Sharma, and Michael V Tretyakov. Consensus-based optimization via jump-diffusion stochastic differential equations. *Mathematical Models and Methods in Applied Sciences*, 33(02):289–339, 2023.
[53] Michael Keane and Karl Petersen. Easy and nearly simultaneous proofs of the ergodic theorem and maximal ergodic theorem. *Lecture Notes-Monograph Series*, pages 248–251, 2006.
[54] Yun Li, Fuke Wu, and Ji-feng Zhang. Near Optimality of Stochastic Control for Singularly Perturbed McKean–Vlasov Systems. *SIAM Journal on Control and Optimization*, 60(5):2859–2883, 2022.
[55] Katta G Murty and Santosh N Kabadi. Some NP-complete problems in quadratic and nonlinear programming. Technical report, 1985.
[56] Peter Ochs, René Ranftl, Thomas Brox, and Thomas Pock. Bilevel optimization with nonsmooth lower level problems. In *Scale Space and Variational Methods in Computer Vision: 5th International Conference, SSVM 2015, Lège-Cap Ferret, France, May 31-June 4, 2015, Proceedings 5*, pages 654–665. Springer, 2015.
[57] V Oduguwa and R Roy. Bi-level optimisation using genetic algorithm. In *Proceedings 2002 IEEE International Conference on Artificial Intelligence Systems (ICAIS 2002)*, pages 322–327. IEEE, 2002.
[58] Grigorios A Pavliotis and Andrew Stuart. *Multiscale methods: averaging and homogenization*, volume 53. Springer Science & Business Media, 2008.
[59] René Pinnau, Claudia Totzeck, Oliver Tse, and Stephan Martin. A consensus-based model for global optimization and its mean-field limit. *Mathematical Models and Methods in Applied Sciences*, 27(01):183–204, 2017.
[60] Leonard Andreevič Rastrigin. Systems of extremal control. *Nauka*, 1974.
[61] Michael Röckner, Xiaobin Sun, and Longjie Xie. Strong and weak convergence in the averaging principle for SDEs with Hölder coefficients. *arXiv preprint arXiv:1907.09256*, 2019.
[62] Michael Röckner, Xiaobin Sun, and Yingchao Xie. Strong convergence order for slow–fast McKean–Vlasov stochastic differential equations. In *Annales de l'Institut Henri Poincaré – Probabilités et Statistiques*, volume 57, pages 547–576, 2021.
[63] Michael Röckner and Longjie Xie. Diffusion approximation for fully coupled stochastic differential equations. *The Annals of Probability*, 49(3):1205 – 1236, 2021.
[64] Ryo Sato, Mirai Tanaka, and Akiko Takeda. A gradient method for multilevel optimization. *Advances in Neural Information Processing Systems*, 34:7522–7533, 2021.
[65] Allahkaram Shafiei, Vyacheslav Kungurtsev, and Jakub Marecek. Trilevel and multilevel optimization using monotone operator theory. *Mathematical Methods of Operations Research*, pages 1–38, 2024.
[66] Haoran Sun, Wenqiang Pu, Xiao Fu, Tsung-Hui Chang, and Mingyi Hong. Learning to continuously optimize wireless resource in a dynamic environment: A bilevel optimization perspective. *IEEE Transactions on Signal Processing*, 70:1900–1917, 2022.
[67] Surafel Luleseged Tilahun, Semu Mitiku Kassa, and Hong Choon Ong. A new algorithm for multilevel optimization problems using evolutionary strategy, inspired by natural adaptation. In *PRICAI 2012: Trends in Artificial Intelligence: 12th Pacific Rim International Conference on Artificial Intelligence, Kuching, Malaysia, September 3-7, 2012. Proceedings 12*, pages 577–588. Springer, 2012.

[68] Konstantinos I Tsianos, Sean Lawlor, and Michael G Rabbat. Consensus-based distributed optimization: Practical issues and applications in large-scale machine learning. In *2012 50th annual allerton conference on communication, control, and computing (allerton)*, pages 1543–1550. IEEE, 2012.
[69] Jinhuan Wang, Keyu Li, and Hui Huang. Mathematical analysis of the PDE model for the consensus-based optimization. *arXiv preprint arXiv:2504.10990*, 2025.
[70] Xu Wang, Zhaohong Bie, Fan Liu, Yu Kou, and Lizhou Jiang. Bi-level planning for integrated electricity and natural gas systems with wind power and natural gas storage. *International Journal of Electrical Power & Energy Systems*, 118:105738, 2020.
[71] E Weinan. *Principles of multiscale modeling*. Cambridge University Press, 2011.
[72] Gerald Whittaker, Rolf Färe, Shawna Grosskopf, Bradley Barnhart, Moriah Bostian, George Mueller-Warrant, and Stephen Griffith. Spatial targeting of agri-environmental policy using bilevel evolutionary optimization. *Omega*, 66:15–27, 2017.
[73] Chao Xue, Xiaoxing Wang, Junchi Yan, Yonggang Hu, Xiaokang Yang, and Kewei Sun. Rethinking bi-level optimization in neural architecture search: A gibbs sampling perspective. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 10551–10559, 2021.
[74] Peng Yang, Yingjie Lao, and Ping Li. Robust watermarking for deep neural networks via bi-level optimization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14841–14850, 2021.
[75] Arthur Yeh, Dennis Lin, Honghong Zhou, and Chandramouliswaran Venkataramani. A multivariate exponentially weighted moving average control chart for monitoring process variability. *Journal of Applied Statistics*, 30(5):507–536, 2003.
[76] Yihua Zhang, Prashant Khanduri, Ioannis Tsaknakis, Yuguang Yao, Mingyi Hong, and Sijia Liu. An introduction to bi-level optimization: Foundations and applications in signal processing and machine learning. *arXiv preprint arXiv:2308.00788*, 2023.
[77] Yihua Zhang, Yuguang Yao, Parikshit Ram, Pu Zhao, Tianlong Chen, Mingyi Hong, Yanzhi Wang, and Sijia Liu. Advancing model pruning via bi-level optimization. *Advances in Neural Information Processing Systems*, 35:18309–18326, 2022.
[78] Yihua Zhang, Guanhua Zhang, Prashant Khanduri, Mingyi Hong, Shiyu Chang, and Sijia Liu. Revisiting and advancing fast adversarial training through the lens of bi-level optimization. In *International Conference on Machine Learning*, pages 26693–26712. PMLR, 2022.
[79] W Zhao. Penalty function method for bilevel multiobjective programming. *Acta Automatica Sinica*, 24:331–337, 1998.
[80] Xiao Zhou, Renjie Pi, Weizhong Zhang, Yong Lin, Zonghao Chen, and Tong Zhang. Probabilistic bilevel coreset selection. In *International Conference on Machine Learning*, pages 27287–27302. PMLR, 2022.
[81] Simiao Zuo, Chen Liang, Haoming Jiang, Xiaodong Liu, Pengcheng He, Jianfeng Gao, Weizhu Chen, and Tuo Zhao. Adversarial Regularization as Stackelberg Game: An Unrolled Optimization Approach. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6562–6577, 2021.

Michael Herty
Institut für Geometrie und Praktische Mathematik, RWTH Aachen University, Templergraben 55, 52062 Aachen, Germany.
Extraordinary Professor, Department of Mathematics and Applied Mathematics, University of Pretoria, Private Bag X20, Hatfield 0028, South Africa

Email address: herty@igpm.rwth-aachen.de

Yuyang Huang
Department of Mathematics, Imperial College London, South Kensington Campus SW72AZ London, UK

Email address: yuyang.huang21@imperial.ac.uk

Dante Kalise
Department of Mathematics, Imperial College London, South Kensington Campus SW72AZ London, UK

Email address: d.kalise-balza@imperial.ac.uk

Hicham Kouhkouh
Department of Mathematics and Scientific Computing, NAWI, University of Graz, Heinrichstrasse 36, 8010, Graz, Austria

Email address: hicham.kouhkouh@uni-graz.at