CONSENSUS-BASED OPTIMIZATION ALGORITHM WITH REPULSION AND ITS APPLICATION TO ONLINE PORTFOLIO OPTIMIZATION

HYEONG-OHK BAE, SEUNG-YEAL HA, MYEONGJU KANG, CHANHO MIN, AND JANE YOO

Abstract. We propose a CBO algorithm incorporating repulsion between particles. The repulsive term is introduced to challenge the discontinuation of searching, after particles are frozen around an optimum at an instant. By introducing repulsion to the CBO algorithm, both attractive and repulsive forces of particles are controlled to efficiently search for optimal points in both static and dynamic environments. We provide an analytical framework of the proposed CBO model with repulsion (in short, Repulsive CBO or R-CBO) and demonstrate its emergent behavior analytically and numerically. We test R-CBO's performance in solving both convex and non-convex optimization problems. Our numerical simulation results show that the repulsive CBO algorithm is superior to the vanilla(original) CBO algorithm and its variants.

1. Introduction

Consensus Based Optimization (CBO) algorithm has been referred as a simple but advanced tool of approximating the optimum in complex optimization problems [8, 10]. The CBO algorithms are efficiently viable in obtaining a precise solution faster than other gradient-based methods [3], e.g., particle swarm optimization [7] by the nature of first-order searching [4, 8]. Particularly, the CBO has proved its efficiency in solving non-convex, non-regular objective function over high-dimensional manifolds [5]. Its beauty is also found in its variations from a predictor-corrector type of CBO [1], the CBO with adaptive momentum (Adam-CBO) [6], to the CBO with adaptive momentum and average drift [2].

In this paper, we propose a new CBO-based algorithm by considering repulsion of particles. The major motivation of developing a repulsion term is a sinking problem: particles of the vanilla CBO [8] are sunk near an instant optimum and easily lose their momentum to restart searching for a new optimum. The challenge is nontrivial to solve online optimization problems like developing intelligent robotic systems, a predator-prey problem, a probabilistic mutation model in genome evolution, and online portfolio selection. The essence of such real problems is on developing a tool to track a target or plan a path in complex, stochastic, and dynamic environments. When repulsive forces are well-managed, a cloud of particles sustain its shape to keep their momentum.

we introduce both drift and repulsion rates to control attractive and repulsive forces in continuous searching for optimal points. We provide an analytical framework of the proposed CBO model with repulsion (in short, Repulsive CBO or R-CBO) and demonstrate its emergent behavior analytically and numerically. We test R-CBO's performance in solving both convex and non-convex optimization problems. The simulation results demonstrate the superiority of R-CBO over vanilla CBO [4, 5] and Adam-CBO [6].

The results imply that a repulsive force controller is a key to challenge online optimization problems that are often found from many real problems including autonomous multi-agent control and online learning from flexible big data. Particularly, the repulsion term is significant in lowering tracking errors and reducing searching time in real online optimization problems.

The rest of this paper is organized as follows. In Section 2, we briefly provide preliminaries of the vanilla CBO, and we introduce R-CBO and its mathematical analysis. Finally, Section 3 is devoted

Key words and phrases. Consensus based optimization; online optimization; repulsion.

to numerical simulations to compare the vanilla CBO and the proposed R-CBO.

Notation. Throughout the paper, we use the following handy notation:

$$X = (\boldsymbol{x}^1, \cdots, \boldsymbol{x}^N) \in \mathbb{R}^{d\times N}, \quad \boldsymbol{x}^i = (x^{i,1}, \cdots, x^{i,d}) \in \mathbb{R}^d, \quad \forall\, i \in [N] := \{1, \cdots, N\}.$$

When X, $\boldsymbol{x}^i$'s, and $x^{i,k}$'s depend on time t, subscript will be used to denote this dependencies: X_t, $\boldsymbol{x}_t^i$, and $x_t^{i,k}$.

2. CBO with repulsion (repulsive CBO, R-CBO).

In this section, we present the CBO with repulsion. First, we begin with a brief discussion on the vanilla CBO.

Consider N particles with position $\boldsymbol{x}^k \in \mathbb{R}^d$ for $k \in [N]$. The d-dimensional position of the k-th sample path at time t is denoted by $\boldsymbol{x}_t^k \in \mathbb{R}^d$, and we set $\boldsymbol{x}^* \in \mathbb{R}^d$ to be the global optimum for an objective function L over $\mathbb{R}^d$:

$$\boldsymbol{x}^* \in \operatorname*{argmin}_{\boldsymbol{x}\in\mathbb{R}^d} L(\boldsymbol{x}).$$

Here, L may be neither convex nor differentiable. Then, the vanilla CBO [10] is given by the following system of stochastic differential equations:

$$\begin{cases} d\boldsymbol{x}_t^i = \lambda(M(X_t) - \boldsymbol{x}_t^i)dt + \sigma(M(X_t) - \boldsymbol{x}_t^i) \odot dW_t, \quad \forall\, t > 0, \\ \omega_t^i := \dfrac{e^{-\beta L(\boldsymbol{x}_t^i)}}{\sum_{j=1}^N e^{-\beta L(\boldsymbol{x}_t^j)}}, \quad \displaystyle\sum_{i=1}^N \omega_t^i = 1, \quad M(X_t) := \sum_{i=1}^N \omega_t^i \boldsymbol{x}_t^i, \quad X_t := (\boldsymbol{x}_t^1, \cdots, \boldsymbol{x}_t^N), \quad \forall\, i \in [N] \end{cases} \tag{2.1}$$

where $M(X_t)$ is the adaptive weighted average of $\boldsymbol{x}_t^k$'s, λ and σ represent the drift rate and noise intensity, respectively, and $\beta > 0$ is given hyperparameter. The coefficient process ω_t^i can be regarded as the weight of each particle $\boldsymbol{x}_t^i$. Here, $W_t := [W_t^1, \cdots, W_t^d]^\top$ and each component W_t^l is one-dimensional Brownian motion defined on the common probability space $(\Omega, \mathcal{F}, (\mathcal{F}_t)_{t\geq 0}, \mathbb{P})$. All W_t^l's are i.i.d. and their mean and covariance are given as follows:

$$\mathbb{E}[W_t^l] = 0 \quad \forall\, l \in [d] \quad \text{and} \quad \mathbb{E}[W_t^{l_1} W_t^{l_2}] = \delta_{l_1 l_2} t, \quad \forall\, l_1,\ l_2 \in [d].$$

In addition, for two matrices $A = (a_{ij})$ and $B = (b_{ij})$ of the same size, we denote by $A \odot B$ their Hadamard product, i.e. $A \odot B = (a_{ij} b_{ij})$. On the mathematical analysis of the CBO model, we refer to [4, 8] for the emergence of global consensus and [8] for the asymptotic convergence to a global optimum.

Next, we are ready to state the our problem setting and the CBO with a repulsion. Consider a global optimization problem for the time-dependent stochastic objective function $L : [0, \infty) \times \mathbb{R}^d \times \Omega \to \mathbb{R}$. Note that global minimizer $\boldsymbol{x}_t^*$ is time-dependent:

$$\boldsymbol{x}_t^*(\omega) \in \operatorname*{argmin}_{\boldsymbol{x}\in\mathbb{R}^d} L(t, \boldsymbol{x}, \omega), \quad \forall\, t \geq 0, \quad \text{a.s. } \omega \in \Omega.$$

From now on, if there is no confusion, ω can be omitted. Here, $L(t, \boldsymbol{x}, \omega)$ may be neither convex nor differentiable with respect to $\boldsymbol{x}$ for each t. Now, we propose the R-CBO model as follows:

$$\begin{cases} d\boldsymbol{x}_t^i = \lambda_0(M(X_t) - \boldsymbol{x}_t^i)dt - \lambda_1 N \omega_t^i(\bar{\boldsymbol{x}}_t - \boldsymbol{x}_t^i)dt + \sigma(M(X_t) - \boldsymbol{x}_t^i) \odot dW_t, \quad \forall\, t > 0, \\ \omega_t^i := \dfrac{e^{-\beta L(t, \boldsymbol{x}_t^i)}}{\sum_{j=1}^N e^{-\beta L(t, \boldsymbol{x}_t^j)}}, \quad M(X_t) := \displaystyle\sum_{i=1}^N \omega_t^i \boldsymbol{x}_t^i, \quad \bar{\boldsymbol{x}}_t := \frac{1}{N}\sum_{j=1}^N \boldsymbol{x}_t^j, \quad \forall\, i \in [N], \end{cases} \tag{2.2}$$

where, $\lambda_0, \lambda_1 > 0$ are the rates of drift and repulsion, respectively. In the absence of the repulsion term $-\lambda_1 N \omega_t^i(\bar{\boldsymbol{x}}_t - \boldsymbol{x}_t^i)dt$, the system (2.2) becomes the vanilla CBO (2.1).

Note that any point in $\{X \in \mathbb{R}^{d\times N} : \boldsymbol{x}^1 = \cdots = \boldsymbol{x}^N\}$ is an equilibrium of (2.2). If $\lambda_0 \gg \lambda_1$, particles might relax to a consensus state exponentially fast as can be seen in [4, 8]. Thus, it stops

chasing time-dependent global optimum $\boldsymbol{x}_t^*$. On the other hand, if $\lambda_0 \ll \lambda_1$, then the repulsive term is stronger than that of the relaxation (attractive) term, as a result, resulting relative distances between particles diverge asymptotically. Since we require that particles form a point cloud aroud $\boldsymbol{x}_t^*$, that is

$$\liminf_{t\to\infty} |\boldsymbol{x}_t^i(\omega) - \boldsymbol{x}_t^*(\omega)| < \varepsilon, \quad \forall\, i \in [N], \quad \text{a.s. } \omega \in \Omega, \tag{2.3}$$

for sufficiently small $\varepsilon > 0$, we need to consider the case of $\lambda_0 \approx \lambda_1$ in what follows. To illustrate the role of repulsion term in (2.2), we consider the following simple setting:

$$d = 1, \quad N = 2, \quad \sigma = 0, \quad \lambda_0 = \lambda_1 = \lambda > 0,$$

and we further assume that $x_0^{1,1}$ and $x_0^{2,1}$ are deterministic constants. In this setting, the model (2.2) becomes

$$\begin{cases} (x_1(t,\omega), x_2(t,\omega), \omega_1(t,\omega), \omega_2(t,\omega), x^*(t,\omega)) := (x_t^{1,1}(\omega), x_t^{2,1}(\omega), \omega_t^1(\omega), \omega_t^2(\omega), x_t^*(\omega)), \\ \dot{x}_1 = \lambda(\omega_1 x_1 + \omega_2 x_2 - x_1) - \lambda\omega_1(x_2 - x_1), \\ \dot{x}_2 = \lambda(\omega_1 x_1 + \omega_2 x_2 - x_2) - \lambda\omega_2(x_1 - x_2), \quad \forall\, t > 0. \end{cases} \tag{2.4}$$

Then, it is easy to see that $\omega_1 + \omega_2 = 1$ and

$$\begin{aligned} &\frac{d}{dt}(x_2 - x_1) = 0, \quad \forall\, t > 0, \quad \text{i.e.,} \\ &x_2(t,\omega) - x_1(t,\omega) = x_2(0) - x_1(0), \quad \forall\, t \geq 0, \quad \forall\, \omega \in \Omega. \end{aligned} \tag{2.5}$$

Next, we first state our sufficient framework $(\mathcal{F}1) - (\mathcal{F}3)$ which exhibits the finite-time entrance of optimum point x^* inside the interval with ending points x_1 and x_2 as follows.

- $(\mathcal{F}1)$: Optimal point path $x^*(t,\omega)$ is almost surely continuous in time t, and initial data satisfy

$$x_2(0) - x_1(0) = d_0, \quad \text{for some } d_0 > 0. \tag{2.6}$$

- $(\mathcal{F}2)$: For almost sure $\omega \in \Omega$ and for each $t \geq 0$, $L(t,x,\omega)$ is strictly convex in x-variable and there exist positive constants m and $d \in (0, d_0)$ such that

$$|L(t,x,\omega) - L(t,x \pm d_0, \omega)| \geq m d_0, \quad \forall\, x \in (-\infty, x^*(t,\omega) - d] \,\cup\, [x^*(t,\omega) + d, \infty). \tag{2.7}$$

- $(\mathcal{F}3)$: R is a positive constant such that

$$0 < R < \frac{\beta m d_0^2 \lambda}{2 + \beta m d_0}, \tag{2.8}$$

Theorem 2.1. *Suppose the framework $(\mathcal{F}1) - (\mathcal{F}3)$ holds, and let (x_1, x_2) be a global solution to (2.4) with deterministic initial data $(x_1(0), x_2(0))$. Then, the following assertions hold.*

(1) *Suppose that $\omega \in \Omega$ satisfies $(\mathcal{F}2)$,*

$$x^*(0,\omega) \notin \big(x_1(0),\ x_2(0)\big), \quad \textit{and} \quad \limsup_{t\to\infty} \frac{|x^*(t,\omega)|}{t} < R. \tag{2.9}$$

Then, there exists a finite time $t_e = t_e(\omega) > 0$ such that

$$x_1(t_e) < x^*(t_e,\omega) < x_2(t_e).$$

(2) *Suppose that $\omega \in \Omega$ satisfies $(\mathcal{F}2)$,*

$$x^*(0,\omega) \in \big(x_1(0),\ x_2(0)\big), \quad \textit{and} \quad \frac{|x^*(t,\omega) - x^*(s,\omega)|}{t-s} < R, \quad \textit{for all} \ \ t > s \geq 0.$$

Then, we have

$$x^*(t,\omega) \in \big(x_1(t,\omega),\ x_2(t,\omega)\big), \quad \forall\, t \geq 0. \tag{2.10}$$

(3) *Suppose that* $\omega \in \Omega$ *satisfies* $(\mathcal{F}2)$ *and*

$$\exists\ t_0 \in [0,\infty) \quad \textit{such that} \quad \frac{|x^*(t,\omega) - x^*(s,\omega)|}{t-s} < R, \quad \textit{for all}\ \ t > s \geq t_0.$$

Then, there exists a finite time $t_e = t_e(\omega) \geq t_0$ *such that*

$$x^*(t,\omega) \in \big(x_1(t,\omega),\ x_2(t,\omega)\big), \quad \forall\ t \geq t_e.$$

Proof. It follows from (2.5) and (2.6) that

$$x_2(t) = x_1(t) + d_0, \quad \forall\ t \geq 0, \quad \forall\ \omega \in \Omega. \tag{2.11}$$

From now on, if there is no confusion, ω can be omitted in the proof.

(1) Without loss of generality, we may assume that

$$x_1(0) < x_2(0) < x^*(0).$$

Since $L(t,\cdot)$ is strictly convex, we have

$$L\big(0, x_1(0)\big) > L\big(0, x_2(0)\big). \tag{2.12}$$

Now, we split the proof into two cases depending on whether (2.12) holds for all $t \geq 0$ or not.

• Case A: Suppose that there exists a finite time $t' = t'(\omega) > 0$ such that

$$L\big(t', x_1(t')\big) = L\big(t', x_2(t')\big),$$

then strict convexity of $L(t,\cdot)$ implies

$$x_1(t') < x^*(t') < x_2(t'),$$

which gives our desired result.

• Case B: Suppose that

$$L\big(t, x_1(t)\big) > L\big(t, x_2(t)\big), \quad \forall\ t \geq 0. \tag{2.13}$$

By the convexity of $L(t,\cdot)$, one has

$$x_1(t) < x_2(t) < x^*(t), \quad \forall\ t \geq 0.$$

For some $d \in (0, d_0)$ appearing in (2.7), we define a temporal set $\mathcal{T}$ and its supremum $\tau_* = \tau_*(\omega)$:

$$\mathcal{T} := \Big\{ t \in [0,\infty) :\ x_1(t) < x^*(t) - d \Big\} \quad \text{and} \quad \tau^* := \sup \mathcal{T}.$$

It follows from

$$x^*(0) > x_2(0) = d_0 + x_1(0) > d + x_1(0)$$

that $0 \in \mathcal{T}$, and the continuity of $x^* - x_1$ in t-variable implies that set $\mathcal{T}$ contains a nonempty open interval. Thus, we have

$$\tau^* \in (0, \infty].$$

Now, we consider the following two cases:

$$\text{Either } 0 < \tau^* < \infty \quad \text{or} \quad \tau^* = \infty.$$

◇ Case B.1: If $0 < \tau^* < \infty$, then we obtain the desired estimate with $t_e = \tau^*$:

$$x^*(\tau^*) = x_1(\tau^*) + d < x_1(\tau^*) + d_0 = x_2(\tau^*), \quad \text{i.e.,} \quad x^*(\tau^*) \in (x_1(\tau^*), x_2(\tau^*)).$$

◇ Case B.2: Suppose $\tau^* = \infty$. By (2.9), there exists a finite time $\tau_0 > 0$ such that

$$|x^*(t)| \leq Rt \quad \text{for all } t \geq \tau_0.$$

Then, we use (2.7), (2.13) and $e^x \geq 1+x$ for $x \geq 0$ to get

$$\begin{aligned}\omega_1(t) &= \frac{1}{1+e^{\beta(L(t,x_1(t))-L(t,x_2(t)))}} \leq \frac{1}{2+\beta(L(t,x_1(t))-L(t,x_2(t)))} \\ &= \frac{1}{2+\beta|L(t,x_1(t))-L(t,x_2(t))|} \leq \frac{1}{2+\beta m d_0}, \qquad \forall\, t \geq 0.\end{aligned}$$

This yields

$$\omega_2 - \omega_1 = 1 - 2\omega_1 \geq \frac{\beta m d_0}{2+\beta m d_0}, \quad \forall\, t \geq 0. \tag{2.14}$$

It follows from $(2.4)_1$, (2.5), (2.14), and $\omega_1+\omega_2 = 1$ that

$$\dot{x}_1 = d_0\lambda(\omega_2-\omega_1) \geq \frac{\beta m d_0^2 \lambda}{2+\beta m d_0}, \quad \forall\, t \geq 0. \tag{2.15}$$

Thus, for all $t \geq \tau_0$, we have

$$0 < d < x^*(t) - x_1(t) = x^*(t) - \int_0^t \dot{x}_1(s)ds - x_1(0) \leq \underbrace{\left(R - \frac{\beta m d_0^2\lambda}{2+\beta m d_0}\right)}_{<0 \text{ by (2.8)}} t - x_1(0).$$

We let $t \to \infty$ to derive a contradiction.

(2) Suppose the contrary of (2.10) holds, then without loss of generality, we assume that there exists a finite time $t'' > 0$ such that

$$x^*(t'') = x_2(t'') \quad \text{and} \quad x^*(t) \in (x_1(t), x_2(t)) \quad \text{for all } t < t''.$$

For $d \in (0, d_0)$, we choose sufficiently small $\delta \in (0, t'')$ such that

$$x^*(t) - x_1(t) > d, \quad \forall\, t \in [t''-\delta,\ t''].$$

This gives a contradiction to the following relation:

$$\begin{aligned}0 < d_0 - \big(x^*(t''-\delta) - x_1(t''-\delta)\big) &= x^*(t'') - x_1(t'') - \big(x^*(t''-\delta) - x_1(t''-\delta)\big) \\ = x^*(t'') - x^*(t''-\delta) - \int_{t''-\delta}^{t''} \dot{x}_1(s)ds &\leq \left(R - \frac{\beta m d_0^2\lambda}{2+\beta m d_0}\right)\delta < 0,\end{aligned}$$

where we used (2.11) and (2.15).

(3) Since (2.4) is autonomous, desired result directly comes from *(1)* and *(2)*. □

In Theorem 2.1 (1), (2.9) does not guarantee

$$\frac{|x^*(t,\omega) - x^*(s,\omega)|}{t-s} < R, \quad \text{for all } \ t > s \geq t_e.$$

Hence, $x^*(t',\omega) \notin \big(x_1(t'),\ x_2(t')\big)$ can occur at some $t' > t_e$. However, since (2.4) is autonomous, there exists $t'' = t''(\omega) > t'$ such that

$$x_1(t'') < x^*(t'',\omega) < x_2(t'').$$

Inductively, there exists a positive sequence $(t_n = t_n(\omega))_{n\geq 0}$ such that

$$t_n \nearrow \infty \quad \text{and} \quad x_1(t_n) < x^*(t_n,\omega) < x_2(t_n),$$

which implies that

$$\liminf_{t\to\infty} |x_i(t) - x^*(t,\omega)| < d_0, \quad i = 1,2.$$

We summarize aforementioned arguments in the following corollary on the relation between Theorem 2.1 and (2.3) which is the main purpose of our R-CBO model.

Corollary 2.2. *Suppose the framework* $(\mathcal{F}1)-(\mathcal{F}3)$ *holds, and let* (x_1, x_2) *be a global solution to* (2.4) *with deterministic initial data* $(x_1(0), x_2(0))$. *If we further assume that*

$$\limsup_{t\to\infty} \frac{|x^*(t,\omega)|}{t} < R, \quad \text{a.s. } \omega \in \Omega,$$

then we have

$$\liminf_{t\to\infty} |x_i(t) - x^*(t,\omega)| < d_0, \quad i = 1, 2, \quad \text{a.s. } \omega \in \Omega.$$

3. Numerical simulations

We present results from various numerical simulations in this section. We investigate the predicting power of our R-CBO in optimization problems of a high-dimensional nonconvex function, a time-dependent stochastic function, and a real online portfolio optimization problem. Prediction errors and computation time of R-CBO are compared with those of vanilla CBO [4] and CBO with adaptive momentum (Adam-CBO) [6]. Second test results show that R-CBO performs better than other two CBO-based algorithms in minimizing average loss of approximating an optimal point. The third test aims to evaluate the efficiency of R-CBO in searching for solutions of real online high-dimensional problems with constraints.

3.1. **High-dimensional Optimization.**

To evaluate the performance of R-CBO compared to vanilla and Adam CBO, we measure prediction error and computation errors of three CBO-based algorithms in approximating the Rastrigin function:

$$f(\boldsymbol{x}) = \frac{1}{d}\sum_{i=1}^{d}\Big[(x^i)^2 - 10\cos(2\pi x^i) + 10\Big].$$

Note that $f(\boldsymbol{x})$ is non-convex and its global minimum value is attained at $\boldsymbol{x} = 0$. In the experiment, we set the global minima not to be in the support of the distribution of initial data. That is, $\boldsymbol{x}_0^i$'s are independent and identically distributed random variables and 0 is not contained in the support of $\mathrm{law}(\boldsymbol{x}_0^i)$. We intend this setting to see if R-CBO can resolve the problem in which the vanilla CBO stops continuous search by particles sinked near local minima.

For numerical simulations, we set (hyper)parameters as

$$d = 50, \quad N = 150, \quad \Delta t = 0.3, \quad t \in [0, 600], \quad \beta = 200, \quad \lambda_0 = 1, \quad \sigma = 0.8.$$

We use the following values of hyper-parameters for the Adam-CBO, referring to the most frequently setting in the Adam-based approximation experiment like tests in [6]:

$$\beta_1 = 0.9 \quad \text{and} \quad \beta_2 = 0.99.$$

Finally, the initial data $\boldsymbol{x}_0^i$'s are randomly chosen from the uniform distribution $\mathcal{U}([2,4]^d)$ so that the global minimizer 0 of $f(\boldsymbol{x})$ is not included in $[2,4]^d$.

The average loss is defined by

$$\mathbb{E}\bigg[\frac{1}{n_{\text{iter}}}\sum_{n=1}^{n_{\text{iter}}} L\bigg(\frac{1}{N}\sum_{k=1}^{N}\boldsymbol{x}_{n\Delta t}^k\bigg)\bigg],$$

where n_{iter} is the number of iteration and the expectation is calculated using the average of 100 sample paths.

In Figure 1, we summarize average losses of Vanilla, Adam and Repulsive CBO algorithms. Given the vanilla CBO's average loss as a unity (black-solid line), we plot relative average losses of Adam-CBO and R-CBO to the unity. The relative losses are plotted over λ_1 in Figure 1 to examine the predicting power of R-CBO and Adam-CBO which show different errors by changes in λ_1.

The blue-dash-dotted curve shows that the average loss of R-CBO decreases with increasing λ_1 until $\lambda_1 = 4.0$. Furthermore, the R-CBO shows better performance than the Adam-CBO given $1.5 < \lambda_1 \le 4$. The results imply that the R-CBO outperforms the vanilla CBO for some bounded λ_1. Both Adam-CBO and R-CBO took 0.0019 seconds for each iterations.

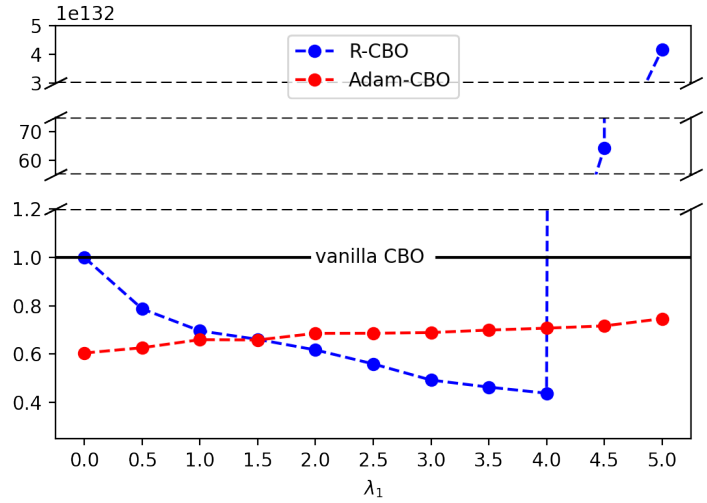


FIGURE 1. The average losses of R-CBO and Adam-CBO relative to the vanilla CBO for various λ_1 in nonconvex high-dimensional optimization

It is noteworthy to find from Figure 1 that λ_1 is a key parameter to manage the approximating errors of the R-CBO. Although the average loss of Adam-CBO maintains around 0.7 over λ_1, the R-CBO is capable of lowering its error by 60% at maximum with the appropriate choice of λ_1. As explained in (2.2), λ_1 controls the repulsive force of a particle. When the repulsive force is too strong by a large λ_1 like $\lambda_1 > 4$, relative distances between particles are diverged.

3.2. **Online Optimization.** When the problem is to keep tracking down optimal points over time-varying searching space, we expect R-CBO outperforms other CBO-based algorithms. It is also worth to investigate the role of repulsive force in searching optimal points over stochastically changing environment. In this section, we conduct numerical tests on the efficiency of R-CBO relative to other CBO-based methods in optimization in such an environment. Particularly, by comparing R-CBO's results with them of Adam-CBO, which is well-known for its efficiency in online optimization, we examine the potential use of R-CBO in real complex problems.

As online optimization tasks, we introduce two online optimization problems: for $\boldsymbol{x} = (x^1, \cdots, x^d)^\top \in \mathbb{R}^d$,

$$\begin{aligned} f(t,\boldsymbol{x}) &= \frac{1}{d}\sum_{i=1}^{d}\Big[(x^i - B_t^i)^2\Big], \\ g(t,\boldsymbol{x}) &= \frac{1}{d}\sum_{i=1}^{d}\Big[(x^i - B_t^i)^2 - 10\cos(2\pi(x^i - B_t^i)) + 10\Big]. \end{aligned}$$

Here, $B_t = (B_t^1, \cdots, B_t^d)^\top$ is a d-dimensional Brownian motion with $B_0 = \mathbf{0}$. Note that $f(t,\boldsymbol{x})$ is a time-dependent convex function in spatial variable $\boldsymbol{x}$, whereas $g(t,\boldsymbol{x})$ is a time-dependent non-convex function with respect to $\boldsymbol{x}$. It is obtained by shifting the Rastrigin function along B_t. Moreover, each function's minimum value is found at $\boldsymbol{x} = B_t$.

In numerical examples below, we use parameters

$$d = 10, \quad N = 30, \quad \Delta t = 0.1, \quad t \in [0, 10].$$

We choose hyperparameters as $\beta = 50$, $\lambda_0 = 1$, and $\sigma = 0.8$. Hyperparameters for Adam-CBO are set by:

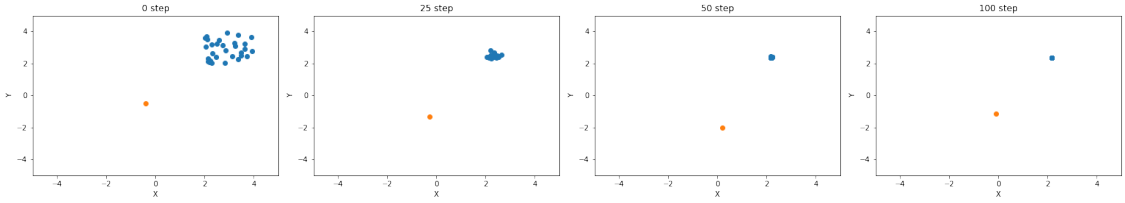
$$\beta_1 = 0.9 \quad \text{and} \quad \beta_2 = 0.99.$$

The initial data $\{\boldsymbol{x}_0^i\}$ are randomly chosen from the uniform distribution $\mathcal{U}([2,4]^d)$. By using same initial data with them of previous experiment, the global minimizers of $f(t,\boldsymbol{x})$ and $g(t,\boldsymbol{x})$ are not included in $[2,4]^d$ at $t=0$. We compute the average loss by

$$\frac{1}{n_{\text{iter}}}\sum_{n=1}^{n_{\text{iter}}} L\left(\frac{1}{N}\sum_{k=1}^{N}\boldsymbol{x}_{n\Delta t}^k\right),$$

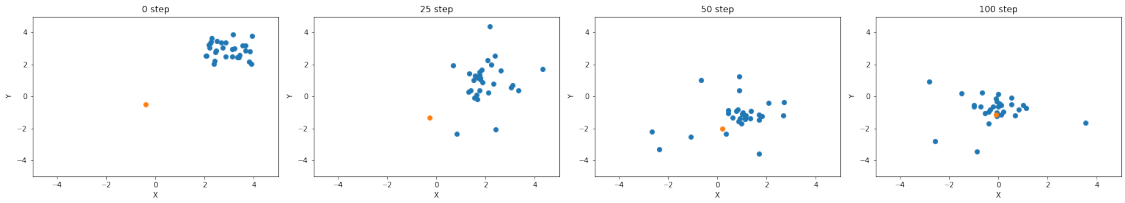
where n_{iter} is the number of iteration. Each experiment was conducted in 100 times (100 sample paths).

We describe positions of particles over different time-frames in Figure 2 and 3. The first and second coordinates of particles (depicted in blue) $\boldsymbol{x}_t^i$ and the global minimizer (depicted in orange) according to Brownian motion from $t=0$ to $t=10$. In Figure 2, particles that move according to the vanilla-CBO algorithm sink in one point $t=5$. The vanilla CBO fails to approximate the moving optimum over time. On the other hand, Figure 3 shows trajectories of particles approaching to the global minimizer. Notably, the R-CBO algorithm continuously generates different particle-clouds that keep including the moving global minimizer.



(A) $t=0$ (B) $t=2.5$ (C) $t=5$ (D) $t=10$

FIGURE 2. The optimal point (in orange) and vanilla CBO particles (in blue)



(A) $t=0$ (B) $t=2.5$ (C) $t=5$ (D) $t=10$

FIGURE 3. The optimal point (in orange) and R-CBO particles (in blue)

We summarize the results of $L=f(t,\boldsymbol{x})$ on the left and $L=g(t,\boldsymbol{x})$ on the right panel in Table 1. In each panel, we present the average loss of the R-CBO under "R-CBO" and that of Adam-CBO under "Adam-CBO", respectively. Also, note that we report errors from the vanilla CBO under the R-CBO with $\lambda_1=0$. The results show that our R-CBO method outperforms the Adam-CBO when the coefficient of controlling repulsive force, λ_1 is close to $\lambda_0=0.5$. Both R-CBO and Adam-CBO algorthims show better performance than the vanilla CBO.

3.3. **Online Portfolio Optimization on High-Dimensional Stochastic Space.** From our numerical tests of simple online optimization problems in Section 3.2, we find that well-managed repulsive force contributes to increasing searching efficiency of CBO by supporting candidate solutions (particles) distributed to catch moving an optimal point. In this section, we test the role of repulsive force to balance out particles' strong attraction in improving its ability to solve real optimization problems.

λ_1	$f(t,\boldsymbol{x})$		$g(t,\boldsymbol{x})$	
	R-CBO	Adam-CBO	R-CBO	Adam-CBO
0	1,040.68*	150.17	107.43*	29.04
0.1	494.71	133.67	99.16	25.34
0.2	91.55	73.91	83.18	23.14
0.3	40.61	65.48	54.61	21.28
0.4	32.12	68.78	35.33	21.06
0.5	33.09	78.19	22.29	20.91
0.6	34.40	86.99	19.39	21.09
0.7	37.72	100.52	19.20	22.48
0.8	40.83	124.63	18.69	26.60
0.9	44.70	175.05	18.69	28.14
1.0	49.08	196.40	19.23	31.64

TABLE 1. Average loss of CBO-based algorithms in online optimization
Note: Results are obtained from R-CBO and Adam-CBO algorithms with various λ_1 applied to approximate the optimal points of $f(t,\boldsymbol{x})$ and $g(t,\boldsymbol{x})$, respectively. In each experiment, we set $d = 10, \quad N = 30, \quad \Delta t = 0.1, \quad t \in [0,10]$. Hyperparameters are set by $\beta = 50$, $\lambda_0 = 1$, and $\sigma = 0.8$. For the test of Adam-CBO, we set $\beta_1 = 0.9 \quad$ and $\quad \beta_2 = 0.99$. Results marked by * are from R-CBO with $\lambda_1 = 0$ so that the same results are obtained from vanilla CBO.

One of popular online optimization problems in complex environment is finding the optimal portfolio weight [1, 2]. By referring to the theoretical foundation of asset pricing in a no-arbitrage market [9], we consider an investor's problem of choosing optimal weights on multiple (possibly infinite number of) assets with his/her total saving. The simple rule of the optimization problem is to choose the best combination of risky assets yielding maximal returns given the portfolio's total risk.

Since countless buyers and suppliers interact each other at any time with no entry barrier assumption, an asset market is stochastically and continuously changing. Based on simple investment returns expressed in percentage changes in values of assets over n periods such that $\frac{P_t - P_{t-n}}{P_{t-n}}$, the continuously compounding returns from an asset investment are given as $\log P_t - \log P_{t-n}$. Having observed expected returns of multiple assets, an investor chooses the portfolio that can generate maximal returns given its volatility over time [11]. Since then, the Sharpe ratio, the excess return relative to volatility, has been widely used as a measure to evaluate assets and portfolios.

We use the Sharpe ratio as an objective function in (2.2) and design numerical experiments to evaluate the performance of R-CBO. Our experiment was conducted in similar but more complex environment than that of [1, 2]. Specifically, we consider all stocks that have been consistently listed on the S&P 500 index. The optimal portfolio weight is found for each of all 484 stocks consistently listed on the S&P 500 index from January 1, 2017 to December 31, 2019[1]. We conduct additional tests on optimizing the portfolio weight on 50 random stocks from the S&P500 and 500 randomly picked stocks on S&P 500 index, respectively, to see the variance of prediction results.

In the experiment, particles are minimizers of the loss function, defined as the negative Sharpe ratio. Searching is designed online to find another minimizer of the new loss function given at $t+1$ after its search at t. Each search is conducted within a window of 60 trading days by following [2]. The online optimization rule is applied to the 60-day window over the sample period, in which a

[1]Although five hundred stocks are listed at each time frame, 484 stocks have been listed consistently over the sample period. Some stocks are delisted or newly-listed by regular examination of U.S. Securities and Exchange Commission, SEC.

window is marked at t of a starting date of the window and rolled over by one trading day until the last trading day (Dec. 31st, 2019) of the total sample period. Over the total sample period, we evaluate the performance of R-CBO relative to the benchmark solution of Sequential Least Squares Programming (SLSQP) algorithm[2]. We consider the SLSQP's solution as a correct answer, because it is based on the Jacobian after observing the entire period. However, the SLSQP's solution is not feasible for an investor without foreseeing stochastically time-varying returns.

The R-CBO test environment is set by $\lambda_0 = 1$ and $\lambda_1 = 12$. Additionally, we configure the number of particles N at 2,500 and 250 for S&P 500 data set and randomly sampled 50 data set, respectively. The parameters β and σ are fixed at 2,000 and 0.8, respectively. The initial data set, denoted as $\boldsymbol{x}_0^i$, is sampled from a uniform distribution $\mathcal{U}([0,1]^d)$ which then projected in to $\mathcal{S} := \{x | \sum x^i = 1, x^i \geq 0\}$. Following each iteration, a projection is performed onto the set $\mathcal{S}$. This step ensures that each particle accurately represents the weight of a stock in the portfolio.

Algorithm	S&P 500 Index		Random 50 stocks	
	Computing Time	Sharpe Ratio	Computing Time	Sharpe Ratio
SLSQP	38.73 sec.	13.95	0.25 sec.	7.63
R-CBO 250	0.92 sec.	3.65	0.23 sec.	5.66
R-CBO 2500	8.59 sec.	5.35	2.07 sec.	6.71

TABLE 2. Results of R-CB0 on the Portfolio Section Problem
Note: Results are obtained from SLSQP and R-CBO algorithms with different number of particles applied to approximate the optimal portfolio weight vector of maximizing the expected excess return relative to volatility (Sharpe ratio). We set $\lambda_0 = 1$, $\lambda_1 = 12$, $\beta = 2,000$, and $\sigma = 0.8$. See text for initial data setting. Computing time is in seconds.

Table 2 presents the computing time and Sharpe ratio achieved by SLSQP, R-CBO with 250 particles, and R-CBO with 2,500 particles. Both computing time and Sharpe ratio are given by the average of measures over windows. We conduct two experiments to evaluate the performance of R-CBO by setting the portfolio optimization problem using i) all stocks of the S&P500 index (on the left panel) and ii) randomly chosen 50 stocks from the S&P500 index. The optimal portfolio predicted by R-CBO (3.65) over all S&P 500 stocks achieves 26% of the SLSQP's solution (13.95). However, we find potential benefits of using R-CBO from its searching speed (2.23% of the SLSQP's computational speed) and its better improvement in the Sharpe ratio by using more particles (from 250 to 2500). Such tendencies are consistently found from tests of using random 50 stocks.

In our examination of numerical results, we find that there are certain types of portfolios are hardly projected by the R-CBO. Particularly, when the portfolio at $t+1$ experience a large jump from the portfolio at t by unexpected shocks or noise to a firm or an asset market, such a large jump is hardly predicted by the R-CBO based on realized returns data until t. Specifically, we show details about results from numerical tests in Figure 4. Figure 4 illustrates time series of the Sharpe ratio attained by SLSQP and R-CBO, respectively. While the SLSQP method generally exhibits higher Sharpe ratios than others, our R-CBO algorithm can trace out the trend of SLSQP. The R-CBO with 250 particles show the least Sharpe ratio, particularly when the ratio changes through a dramatic turning point.

For example, we find multiple corner solutions with many zero weights on assets are not easily found from on-time search. However, when we limit the searching space with a reasonable number of assets from 500 to 50 stocks as suggested in Table 2, we find a room to improve the R-CBO's searching capability. For example, 30-40 stocks are included in a basket of a diversified portfolio [12]. Also, although it is hard to deal with the corner solution problem for any optimizer, we find that a

[2]SLSQP uses the Han-Powell quasi-Newton method with a BFGS update of the B-matrix and an L1-test function in the step-length algorithm. The optimizer uses a slightly modified version of Lawson and Hanson's NNLS nonlinear least-squares solver, which is available in the SciPy optimization library.

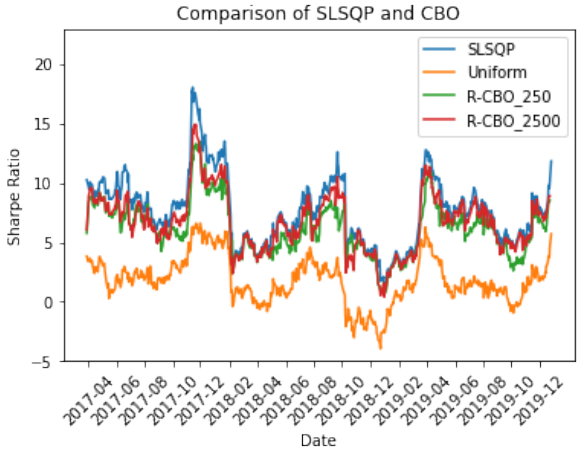


FIGURE 4. Sharpe Ratio's Time Series Plot by SLSQP and R-CBO
Note: Results are obtained from SLSQP and R-CBO algorithms with different number of particles applied to approximate the optimal portfolio weight vector of maximizing the expected excess return relative to volatility (Sharpe ratio). We set $\lambda_0 = 1$, $\lambda_1 = 12$, $\beta = 2,000$, and $\sigma = 0.8$. See text for initial data setting.

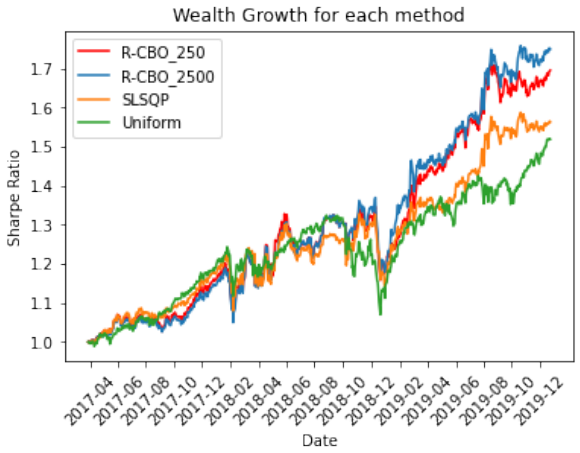


FIGURE 5. Time Series Wealth of R-CBO Algorithms on Online Portfolio Optimization
Note: Results are obtained from SLSQP and R-CBO algorithms with different number of particles applied to approximate the optimal portfolio weight vector of maximizing the expected excess return relative to volatility (Sharpe ratio). We set $\lambda_0 = 1$, $\lambda_1 = 12$, $\beta = 2,000$, and $\sigma = 0.8$. See text for initial data setting.

repulsive force controller is a key to resolve this issue as described by the R-CBO's ability to maintain its non-degenerate searching radius as shown in Figure 3.

Furthermore, we conduct another numerical test for predicting powers of the SLSQP and R-CBO. In this setting, we do not provide the full information of asset returns to educate the SLSQP to make the comparison fair. In Figure 5, we present the results by depicting the time series of wealth accumulated by the suggested portfolio optimization rule. In this test, we also test the uniform portfolio, which applies the equal weight to any asset in a portfolio.

Without seeing entire data, the R-CBO with either 250 or 2500 particles outperform the SLSQP and uniform portfolio. Table 3 summarizes the average terminal wealth for each method from 100

Optimization algorithm	Wealth
SLSQP	1.52
Uniform-style (Random particles)	1.56
R-CBO (using 250 particles)	1.69 (0.313)
R-CBO (using 2,500 particles)	1.74 (0.07)

TABLE 3. Final Wealth of SLSQP, Uniform Portfolio, and R-CBO Algorithms on Online Portfolio Optimization
Note: Results are obtained from SLSQP and R-CBO algorithms with different number of particles applied to approximate the optimal portfolio weight vector of maximizing the expected excess return relative to volatility (Sharpe ratio). We set $\lambda_0 = 1$, $\lambda_1 = 12$, $\beta = 2,000$, and $\sigma = 0.8$. See text for initial data setting.

independent experiments. Notably, our R-CBO method exhibits the highest terminal wealth and the standard deviation reduces with increasing the number of particles. Remark that the larger the number of particles, the terminal wealth increases with lower standard deviation, indicating the number of particles increase performance and robustness of our model. Our proposed method (R-CBO) exhibits promising outcomes in solving real problems in an efficient computational environment with appropriate choices of hyperparameters.

DECLARATIONS

Bae is funded by the Basic Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education and Technology (NRF-2021R1A2C109338). Ha is funded by NRF-2020R1A2C3A01003881. Min is funded by NRF-2021R1G1A1095140. Yoo is funded by NRF-2022S1A5C2A02090368. Bae, Min, and Yoo are also supported by Ajou University Research Fund. The authors have no competing interests to declare that are relevant to the content of this article.

REFERENCES

[1] Bae, H.-O., Ha, S.-Y., Kang, M., Lim, H., Min, C. and Yoo, J.: *A constrained consensus based optimization algorithm and its application to finance.* App. Math. Comp **416** Paper No. 126726 1-10 (2022)
[2] Bae, H.-O., Ha, S.-Y., Min, C., Yoo, J. and Yoon, J.: *Online optimization by consensus based optimization algorithm with average drift.* Submitted. (2023)
[3] Bertsekas, D.: Convex Analysis and Optimization. Athena Scientific. (2003)
[4] Carrillo, J. A., Jin, S., Li, L. and Zhu, Y.: *A consensus-based global optimization method for high dimensional machine learning problems.* ESAIM: COCV 27 S5 (2021)
[5] Carrillo, J., Choi, Y.-P., Totzeck, C. and Tse, O.: *An analytical framework for consensus-based global optimization method.* Mathematical Models and Methods in Applied Sciences **28** 1037-1066 (2018)
[6] Chen, J. Jin, S. and Lyu, L.: *A consensus-based global optimization method with adaptive momentum estimation.* Comm. Comp. Phys. **31** 1296-1316 (2022)
[7] Grassi, S. and Pareschi, L.: *From particle swarm optimization to consensus based optimization: stochastic modeling and meanfield limit.* Available at: https://arxiv.org/abs/2012.05613 (2020)
[8] Ha, S.-Y., Jin, S. and Kim, D.: *Convergence of a first-order consensus-based global optimization algorithm.* Math. Models. Meth. in App. Sci. **30** 2417-2444 (2020)
[9] Markowitz, H.: *Portfolio selection.* Journal of Finance **7** 77-91 (1952)
[10] Pinnau, R., Totzeck, C., and Tse, O.: *A consensus-based model for global optimization and its mean-field limit.* Math. Models Methods Appl. Sci., **27** 183-204 (2017)
[11] Sharpe, W. F.: *Mutual Fund Performance.* J. of Business. **39(S1)** 119–138 (1966)
[12] Statman, M.: *How Many Stocks Make a Diversified Portfolio?* The J. of Financial and Quantitative Analysis, **22(3)** 353–363 (1987)

(Bae) Department of Financial Engineering, Ajou University, Suwon, Republic of Korea
Email address: hobae@ajou.ac.kr

(Ha) Department of Mathematical Sciences, Seoul National University, Seoul, Republic of Korea
Email address: syha@snu.ac.kr

(Kang) Department of Finance and Big Data, Gacheon University, Seongnam, Republic of Korea
Email address: mathemjkang@gachon.ac.kr

(Min) Department of Financial Engineering, Ajou University, Suwon, Republic of Korea
Email address: chanhomin@ajou.ac.kr

(Yoo) Department of Financial Engineering, Ajou University, Suwon, Republic of Korea
Email address: janeyoo@ajou.ac.kr