axioms

MDPI

Article

RPCGB Method for Large-Scale Global Optimization Problems

Abderrahmane Ettahiri [1,*] and Abdelkrim El Mouatasim [2]

[1] Laboratory LABSI, Faculty of Sciences Agadir (FSA), Ibnou Zohr University, B.P. 8106, Agadir 80000, Morocco
[2] Department of Mathematics and Management, Faculty of Polydisciplinary Ouarzazate (FPO), Ibnou Zohr University, B.P. 284, Ouarzazate 45800, Morocco; a.elmouatasim@uiz.ac.ma
[*] Correspondence: ae.ettahiri@gmail.com

Abstract: In this paper, we propose a new approach for optimizing a large-scale non-convex differentiable function subject to linear equality constraints. The proposed method, RPCGB (random perturbation of the conditional gradient method with bisection algorithm), computes a search direction by the conditional gradient, and an optimal line search is found by a bisection algorithm, which results in a decrease of the cost function. The RPCGB method is designed to guarantee global convergence of the algorithm. An implementation and testing of the method are given, with numerical results of large-scale problems that demonstrate its efficiency.

Keywords: non-convex optimization; large-scale problem; conditional gradient method; bisection algorithm; random perturbation

MSC: 90C06; 90C15; 90C25; 90C26

check for updates

Citation: Ettahiri, A.; El Mouatasim, A. RPCGB Method for Large-Scale Global Optimization Problems. *Axioms* **2023**, *12*, 603. https://doi.org/10.3390/axioms12060603

Academic Editors: Ivan Mauricio Amaya-Contreras and José Carlos Ortiz-Bayliss

Received: 25 April 2023
Revised: 15 June 2023
Accepted: 16 June 2023
Published: 18 June 2023

1. Introduction

Non-convex optimization is a type of mathematical optimization problem in which the objective function to be optimized is not convex. Unlike convex optimization problems, non-convex problems can have multiple local optima, which can make it difficult to find the global optimum. Non-convex optimization has many applications in various fields, including finance (portfolio optimization, risk management, and option pricing) [1–3], computer vision (image segmentation, object recognition) [4,5], signal processing (compressed sensing, channel estimation, and equalization) [6–8], engineering (control systems, optimization of structures) [9,10], machine learning [11–14], and damage characterization [15] based on deep neural networks and the YUKI algorithm [16].

To solve non-convex optimization problems, two broad classes of techniques have been developed: deterministic and stochastic methods [17]. Deterministic methods include gradient-based methods, which rely on computing gradients of the objective function, and which can be sensitive to the choice of initialization and can converge to local optima. On the other hand, stochastic methods use randomness to explore the search space and can be less sensitive to initialization and more likely to find the global optimum.

Several reasons make stochastic methods more appropriate for non-convex optimization problems. Stochastic methods avoid getting stuck in local optima or saddle points, as they explore the search space more thoroughly. Additionally, complex non-convex optimization problems often have a large number of variables, making gradient-based methods computationally expensive. In contrast, stochastic methods can scale better to high-dimensional problems. Stochastic methods can also be more robust to noise and uncertainty in the problem formulation.

Scientific studies have shown the effectiveness of stochastic methods in solving non-convex optimization problems. One of the most widely studied deterministic methods that has been extended with random perturbations is gradient descent. For example, a study by Pogu and Souza de Cursi (1994) [18] compared the performance of deterministic gradient descent and stochastic gradient descent on a variety of non-convex optimization problems and found that stochastic gradient descent was more robust and could converge to better solutions. Another study by Mandt et al. (2016) [19] investigated the use of random perturbations in the context of Bayesian optimization and found that it could lead to better exploration of the search space and improved optimization performance. In addition to stochastic gradient descent, other deterministic methods have also been extended with random perturbations. For example, a study by Nesterov and Spokoiny (2017) [20] proposed a variant of the conjugate gradient method that adds random noise to the search direction at each iteration and showed that it could improve the convergence rate and solution quality compared to the standard conjugate gradient method. Another study by Songtao Lu et al. (2019) [21] proposed a variant of the projected gradient descent method that added random perturbations to the method, and they demonstrated its effectiveness in solving non-convex optimization problems.

We consider non-convex optimization problems with linear equality or inequality constraints of the form

$$\begin{cases} min & f(x) \\ s.t & Ax \le b \\ & v \le x \le \mu \end{cases}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable function, A is an $m \times n$ matrix with rank m, b is an m-vector, and the lower and upper bound vectors, v and μ, may contain some infinite components; and

$$\begin{cases} min & f(x) \\ s.t & Ax = b \end{cases} \tag{1}$$

where $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, and $f : \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable, non-convex objective function.

One possible numerical method to solve problem (1) is the conditional gradient with bisection (CGB) method. This method generates a sequence of feasible points $\{x^t\}_{t\ge 0}$, starting with an initial feasible point x^0. A new feasible point x^{t+1} is obtained from x^t for each $t > 0$, using an operator Q_t (details can be found in Section 3). The iterations can be expressed as follows:

$$x^{t+1} = Q_t(x^t), \quad \forall t \ge 0.$$

In this paper, we present a new approach for solving large-scale non-convex optimization problems by using a modified version of the conditional gradient algorithm that incorporates stochastic perturbations. The main contribution of this paper is to propose the RPCGB algorithm, which is an extension of a method previously presented in [22] that was designed for small- and medium-scale problems. The RPCGB algorithm was developed to deal with large-scale global optimization problems and aims to determine the global optimum.

This method involves replacing the sequence of vectors $\{x^t\}_{t\ge 0}$ with a sequence of random vectors $\{\mathbf{X}^t\}_{t\ge 0}$, and the iterations are modified as follows:

$$\mathbf{X}^{t+1} = Q_t(\mathbf{X}^t) + P_t, \qquad \forall t \ge 0,$$

where P_t is a random variable that is chosen appropriately, which is commonly known as the stochastic perturbation. It is important that the sequence $\{P_t\}_{t\ge 0}$ converges to zero at a rate slow enough to avoid the sequence $\{\mathbf{X}^t\}_{t\ge 0}$ convergence to local minima. For more details, refer to Section 4.

The paper is structured as follows: Section 3 revisits the principle of the conditional gradient with bisection method, while Section 4 provides details on the random perturbation of the CGB method. Notations are introduced in Section 2, and in Section 5, the results of numerical experiments for non-convex optimization tests with linear constraints are presented for large-scale problems.

2. Notations and Assumptions

We denote by $\mathbb{R}$ the set of the real numbers, and $E = \mathbb{R}^n$ is the n-dimensional real Euclidean space. x^T denotes the transpose of x. We denote by $\|x\| = \sqrt{x^T x} = (x_1^2 + \cdots + x_n^2)^{1/2}$ the euclidean norm of x, and let

$$M = \{x \in E \mid Ax = b, \ \ x \geq 0\}$$

and $\eta^* = \min_M f$, the lower bound of f on M. Let us introduce

$$M_\varphi = N_\varphi \cap M; \quad \text{where} \ \ N_\varphi = \{x \in E \mid f(x) \leq \varphi\}.$$

Supposing

$$\forall \varphi > \eta^* : meas(M_\varphi) > 0, \tag{2}$$

$$\forall \varphi > \eta^* \ : M_\varphi \text{ is not empty, closed, and bounded,} \tag{3}$$

$$f \text{ is continuously differentiable on } E, \tag{4}$$

where $meas(M_\varphi)$ is the measure of M_φ.

As the space E is of finite dimensions, condition (3) holds true if M is bounded or if f is coercive, i.e., $\lim_{\|x\| \to +\infty} f(x) = +\infty$. Assumption (3) is satisfied when M comprises a series of neighborhoods of an optimal point x^* that possesses a strictly positive measure, meaning x^* can be approximated by a sequence of points from the interior of M.

We see that the results of assumptions (3) and (4) are

$$M = \bigcup_{\varphi > \eta^*} N_\varphi, \quad i.e., \quad \forall x \in M : \exists \varphi > \eta^* \ \text{ such that } \ x \in M_\varphi.$$

From (3) and (4), one has:

$$\delta_1 = \sup\{\|\nabla f(x)\| : x \in M_\varphi\} < +\infty.$$

Consequently, one deduces

$$\delta_2 = \sup\{\|d\| : x \in M_\varphi\} < +\infty,$$

where d is the direction of conditional gradient method.

Thus,

$$\rho(\varphi, \varepsilon) = \sup\{\|y - (x + \alpha d)\| : (x, y) \in M_\varphi \times M_\varphi, \ 0 \leq \alpha \leq \varepsilon\} < +\infty, \tag{5}$$

where α and ε are positive real numbers.

3. Conditional Gradient Method

3.1. Conditional Gradient Algorithm

The conditional gradient method, also known as the Frank–Wolfe algorithm, is an iterative optimization algorithm used to find the minimum of a convex function over a convex set. It was introduced by Philip Wolfe and Marguerite Frank in 1956 [23] and is one of the oldest nonlinear constrained optimization techniques. It has recently gained renewed interest due to its projection-free iterations and low memory requirement. This algorithm enables the approximation of a function during each iteration by utilizing the first-order Taylor series expansion.

The algorithm starts with an initial point in the feasible set and iteratively moves towards a direction that minimizes the gradient of the objective function over the feasible set. At each iteration, the algorithm solves a linear optimization problem over the feasible set to find the direction that minimizes the gradient.

The conditional gradient algorithm has several advantages over other optimization methods, including its ability to handle large-scale problems and its ability to find sparse solutions. However, it may converge slowly and may not always find the global minimum. Moving forward, we focus on a problem related to nonlinear programming that involves constraints in the form of linear equalities or inequalities of the form

$$\begin{cases} \text{minimize} \quad f(x) \\ \\ \text{subject to} \quad x \in M \end{cases} \tag{6}$$

The search direction is $d_t := s_t - x^t$, with s_t being the optimal solution of a linear programming problem and

$$x^{t+1} = Q_t(x^k) = x^t + \alpha_k d_t. \tag{7}$$

We ascertain the optimal step by selecting the value of α_t that satisfies

$$f\left(x^t + \alpha_t d_t\right) = \min_{0 \leq \alpha \leq 1} \left\{ f\left(x^t + \alpha d_t\right) \right\}. \tag{8}$$

The conditional gradient algorithm can be summarized as follows (Algorithm 1):

Algorithm 1 Conditional gradient algorithm.

```
Choose an initial point $x^{(0)} \in M$ in the feasible set $M$.
for $t = 0, 1, 2, \ldots, T$ do
    Compute $\mathbf{s}_t := \mathbf{LMO}_M\ (\nabla f(x^{(t)})) := \arg\min_{s \in M} \nabla f(x^{(t)})^\top s$
                                                            (LMO: Linear minimization oracle)
    Let $\mathbf{d}_t := \mathbf{s}_t - x^{(t)}$                  (Conditional gradient direction)
    Compute $g_t := \left\langle -\nabla f(x^{(t)}), \mathbf{d}_t \right\rangle$     (Conditional gradient gap)
    if $g_t < \varepsilon$ then return $x^{(t)}$
    optimal line search step size
            $\alpha_k \in \arg\min_{\alpha \in [0,1]} f(x^{(k)} + \alpha \mathbf{d}_k)$
    Update $x^{(t+1)} := x^{(t)} + \alpha_k \mathbf{d}_t$
end for
return $x^{(T)}$
```

For non-convex objectives, the conditional gradient algorithm may not converge to a global minimum, but it can still converge to a stationary point under certain conditions. Simon Lacoste-Julien and his colleagues have shown that the Frank–Wolfe algorithm can converge to a stationary point for non-convex objectives, as shown in [24].

3.2. Bisection Algorithm

In this paper, we employ the bisection algorithm to tackle the unconstrained optimization problem with one variable (8). The method is described in [25]. We refer to the recursive bisection procedure as bis$(h, \theta_1, \theta_2, \epsilon)$, which takes as inputs the h calculation procedure, the $[\theta_1, \theta_2]$ interval, and the precision ϵ. The outputs of this procedure are an approximation of x_m for the minimizer x^* and h_m for the minimum value of the h function over the $[\theta_1, \theta_2]$ interval.

The recursive procedure iteration involves the application of the subsequent steps.

Step 0: If $\theta_2 - \theta_1 \geq \epsilon$, go to step 1, otherwise stop.

Step 1: Compute

$$\theta_3 = \frac{\theta_1 + \theta_2}{2}, \theta_1' = \frac{\theta_2 + \theta_3}{2}, \theta_2' = \frac{\theta_3 + \theta_2}{2}, h(\theta_3), h(\theta_1'), h(\theta_2').$$

Step 2: If $h(\theta_1') \leq h(\theta_3) \leq h(\theta_2')$, set $\theta_2 = \theta_2'$ If $h(\theta_1') \geq h(\theta_3) \geq h(\theta_2')$, set $\theta_1 = \theta_1'$.

$$\text{If } h(\theta_3) \leq \min\{h(\theta_1'), h(\theta_2')\}, \text{ set } \theta_1 = \theta_1', \theta_2 = \theta_2'.$$

Step 3: Execute $\text{bis}(h, \theta_1, \theta_2, \epsilon)$ with new inputs.

4. RPCGB Method

From [23], when it comes to objective functions that are non-convex, optimization algorithms based on gradients (CGB) cannot guarantee the discovery of the global minimum. Convex functions are the only ones for which CGB methods can find the global minimum. To deal with this issue, we suggest utilizing a suitable random perturbation method. Next, we will demonstrate how RPCGB can converge to a global minimum for non-convex optimization problems.

The sequence of real numbers $\{x^t\}_{t\geq 0}$ is replaced by a sequence of random variables $\{\mathbf{X}^t\}_{t\geq 0}$ involving a random perturbation P_t of the deterministic iteration (7). We have $\mathbf{X}^0 = x^0$;

$$\forall t \geq 0 \; \mathbf{X}^{t+1} = Q_t(\mathbf{X}^t) + P_t = \mathbf{X}^t + \alpha_k d^t + P_t = \mathbf{X}^t + \alpha_t (d^t + \frac{P_t}{\alpha_t}), \tag{9}$$

$$P_t \text{ is independent from } (\mathbf{X}^{t-1}, \ldots, \mathbf{X}^0), \quad \forall t \geq 1,$$

where $\alpha_t \neq 0$ satisfies Step 7 in the conditional gradient algorithm (Algorithm 1), and

$$\mathbf{X} \in M \Rightarrow Q_t(\mathbf{X}) + P_t \in M.$$

Equation (9) can be considered a perturbation of the upward direction d^t, which is substituted with a new direction $D_k = d^t + \frac{P_t}{\alpha_t}$. As a result, iterations (9) become:

$$\mathbf{X}^{t+1} = \mathbf{X}^t + \alpha_t D_t.$$

In the literature [18,26,27], general properties can be found to select a sequence suitable for perturbation $\{P_t\}_{t\geq 0}$. Typically, perturbations that satisfy these features are produced using sequences of Gaussian laws.

We define a random vector Z_t and use the symbols Φ_t and ϕ_t to represent its cumulative distribution function and probability density function, respectively.

The conditional probability density function of $\mathbf{X}^{t+1}$ is represented by f_{t+1}, and the conditional cumulative distribution function is designated as $F_{t+1}(y \mid \mathbf{X}^t = x)$.

$$F_{t+1}(y \mid \mathbf{X}^t = x) = P(\mathbf{X}^{t+1} < y \mid \mathbf{X}^t = x).$$

We define a sequence of n-dimensional random vectors $\{Z_t\}_{t\geq 0} \in M$. Additionally, we also take into account $\{\xi_t\}_{t\geq 0}$, a decreasing sequence of positive real numbers that steadily approaches 0, where ξ_0 is less than or equal to 1.

Let $P_t = \xi_t Z_t$

$$F_{t+1}(y \mid \mathbf{X}^t = x) = P(\mathbf{X}^{t+1} < y \mid \mathbf{X}^t = x).$$

It follows that

$$F_{t+1}(y \mid \mathbf{X}^t = x) = P\left(Z_t < \frac{y - Q_t(x)}{\xi_t}\right) = \Phi_t\left(\frac{y - Q_t(x)}{\xi_t}\right).$$

Therefore, we have

$$f_{t+1}(y \mid \mathbf{X}^t = x) = \frac{1}{\xi_t^n}\phi_t\left(\frac{y - Q_t(x)}{\xi_t}\right), \quad y \in M. \tag{10}$$

Relation (5) shows that

$$\|y - Q_t(x)\| \leq \rho(\varphi, \varepsilon) \text{ for } (x,y) \in M_\varphi \times M_\varphi.$$

We suppose that $t \mapsto h_t(t) > 0$ is a decreasing function defined on $\mathbb{R}^+$ such that

$$y \in M_\varphi \Rightarrow \phi_t\left(\frac{y - Q_t(x)}{\xi_t}\right) \geq h_t(\frac{\rho(\varphi, \varepsilon)}{\xi_t}). \tag{11}$$

For simplicity, let

$$Z_t = \mathbf{1}_M(Z_t) Z_t, \tag{12}$$

and $Z \sim \mathbf{N}(0,1)$, where Z is a random variable.

The procedure generates a sequence $V_t = f(\mathbf{X}^t)$. By construction, this sequence is increasing and upper-bounded by η^*.

$$\forall t \geq 0: \ \eta^* \geq V_{t+1} \geq V_t. \tag{13}$$

Thus, there exists $V \leq \eta^*$ such that

$$V_t \to V \ \ for \ t \to +\infty.$$

Lemma 1. *Let $P_t = \xi_t Z_t$ and $\gamma = f(x^0)$ if Z_t is given by (12). Then, there exists $\ell > 0$ such that*

$$P(V_{t+1} > \omega | V_t \leq \omega) \geq \frac{meas(\mathrm{M}_\gamma - \mathrm{M}_\omega)}{\xi_t^n} h_t\left(\frac{\rho(\gamma, \varepsilon)}{\xi_t}\right) > 0 \quad \forall \omega \in (\eta^*, \eta^* + \ell],$$

where $n = \dim(E)$.

Proof. Let $M_\omega = \{x \in M \mid f(x) < \omega\}$, for $\omega \in (\eta^*, \eta^* + \ell]$.

Since $M_\varphi \subset \hat{M}_\omega$, $\eta^* < \varphi < \omega$, it can be deduced from (2) that $\hat{M}_\omega$ is non-empty and has a strictly positive measure.

If $meas(M - \hat{M}_\omega) = 0$ for any $\omega \in (\eta^*, \eta^* + \ell]$, the result is immediate, since we have $f(x) = \eta^*$ on M.

Let us assume that there exists $\varepsilon > 0$ such that $meas(M - \hat{M}_\omega) > 0$. For $\omega \in (\eta^*, \eta^* + \varepsilon]$, we have $\hat{M}_\omega \subset \hat{M}_\varepsilon$ and $meas(M - \hat{M}_\omega) > 0$.

$P(\mathbf{X}^t \notin \hat{M}_\omega) = P(\mathbf{X}^t \in S - \hat{M}_\omega) = \int_{M - \hat{M}_\omega} P(\mathbf{X}^t \in dx) > 0$ for any $\omega \in (\eta^*, \eta^* + \varepsilon]$ and, since the sequence $\{\mathbf{V}_i\}_{i \geq 0}$ is increasing, we also have

$$\left\{\mathbf{X}^i\right\}_{i \geq 0} \subset M_\gamma. \tag{14}$$

Thus

$$P(\mathbf{X}^t \notin \hat{M}_\omega) = P(\mathbf{X}^t \in N - \hat{M}_\omega) = \int_{M_\gamma - \hat{M}_\omega} P(\mathbf{X}^t \in dx) > 0 \ \text{ for any } \omega \in (\eta^*, \alpha^* + \varepsilon].$$

Letting $\omega \in (\eta^*, \alpha^* + \varepsilon]$, we have from (13)

$$P(V_{t+1} > \omega \mid V_t \leq \omega) = P(\mathbf{X}^{t+1} \in \hat{M}_\omega \mid \mathbf{X}^i \notin \hat{M}_\omega, i = 0, \ldots, t).$$

However, the Markov chain produces

$$P(\mathbf{X}^{t+1} \in \hat{M}_\omega \mid \mathbf{X}^i \notin \hat{M}_\omega, i = 0, \ldots, t) = P(\mathbf{X}^{t+1} \in \hat{M}_\omega \mid \mathbf{X}^t \notin \hat{M}_\omega).$$

By the conditional probability rule,

$$P(\mathbf{X}^{t+1} \in \hat{M}_\omega \mid \mathbf{X}^t \notin \hat{M}_\omega) = \frac{P(\mathbf{X}^{t+1} \in \hat{M}_\omega, \mathbf{X}^t \notin \hat{M}_\omega)}{P(\mathbf{X}^t \notin \hat{M}_\omega)}.$$

Moreover,

$$P(\mathbf{X}^{t+1} \in \hat{M}_\omega \mid \mathbf{X}^t \notin \hat{M}_\omega) = \int_{M-\hat{M}_\omega} P(\mathbf{X}^t \in dx) \int_{\hat{M}_\omega} f_{t+1}(y \mid \mathbf{X}^t = x) dy.$$

From (14), we have

$$P(\mathbf{X}^{t+1} \in \hat{M}_\omega \mid \mathbf{X}^t \notin \hat{M}_\omega) = \int_{M_\gamma-\hat{M}_\omega} P(\mathbf{X}^t \in dx) \int_{\hat{M}_\omega} f_{t+1}(y \mid \mathbf{X}^t = x) dy,$$

and

$$P(\mathbf{X}^{t+1} \in \hat{M}_\omega \mid \mathbf{X}^t \notin \hat{M}_\omega) \geq \inf_{x \in M_\gamma-\hat{M}_\omega} \left\{ \int_{\hat{M}_\omega} f_{t+1}(y \mid \mathbf{X}^t = x) dy \right\} \int_{M_\gamma-\hat{M}_\omega} P(\mathbf{X}^t \in dx).$$

Thus

$$P(\mathbf{X}^{t+1} \in \hat{M}_\omega \mid \mathbf{X}^t \notin \hat{M}_\omega) \geq \inf_{x \in M_\gamma-\hat{M}_\omega} \left\{ \int_{\hat{M}_\omega} f_{t+1}(y \mid \mathbf{X}^t = x) dy \right\}.$$

Taking (10) into account, we have

$$P(\mathbf{X}^{t+1} \in \hat{M}_\omega \mid \mathbf{X}^t \notin \hat{M}_\omega) \geq \frac{1}{\xi_t^n} \inf_{x \in M_\gamma-\hat{M}_\omega} \left\{ \int_{\hat{M}_\omega} \phi_t \left(\frac{y - Q_t(x)}{\xi_t} \right) dy \right\}.$$

Relation (5) shows that

$$\|y - Q_t(x)\| \leq \rho(\gamma, \varepsilon),$$

and (11) yields

$$\phi_t \left(\frac{y - Q_t(x)}{\xi_t} \right) \geq h_t \left(\frac{\rho(\gamma, \varepsilon)}{\xi_t} \right).$$

Hence,

$$P(\mathbf{X}^{t+1} \in \hat{M}_\omega | \mathbf{X}^t \notin \hat{M}_\omega) \geq \frac{1}{\xi_t^n} \inf_{x \in M_\gamma-\hat{M}_\omega} \int_{\hat{M}_\omega} h_t \left(\frac{\rho(\gamma, \varepsilon)}{\xi_t} \right) dy.$$

$$P(\mathbf{X}^{t+1} \in \hat{M}_\omega | \mathbf{X}^t \notin \hat{M}_\omega) \geq \frac{meas(M_\gamma - M_\omega)}{\xi_t^n} h_t \left(\frac{\rho(\gamma, \varepsilon)}{\xi_t} \right).$$

□

The following result, which follows from Borel–Catelli's lemma (as described in [18], for example), is a consequence of the global convergence:

Lemma 2. *Let $\{V_t\}_{t\geq 0}$ be a increasing sequence, upper-bounded by η^*. Then, there exists V such that $V_t \to V$ for $t \to +\infty$. Assume that there exists $\ell > 0$ such that, for any $\omega \in (\alpha^*, \alpha^* + \ell]$, there is a sequence of strictly positive real numbers $\{\mathbf{c}_t(\omega)\}_{t\geq 0}$, such that*

$$\forall t \geq 0 \ : \ P(V_{t+1} > \omega \mid V_t \leq \omega) \geq \mathbf{c}_t(\omega) > 0 \ \textit{and} \ \sum_{t=0}^{+\infty} \mathbf{c}_t(\omega) = +\infty.$$

Then $V{=}\eta^*$ almost surely.

Proof. For instance, see [18,28]. □

Theorem 1. *Assuming x^0 belongs to M, and letting $\gamma = f(x^0)$, let the sequence ξ_t be non-increasing, and*

$$\sum_{t=0}^{+\infty} h_t\left(\frac{\rho(\gamma,\varepsilon)}{\xi_t}\right) = +\infty. \tag{15}$$

Then $V{=}\eta^*$ almost surely.

Proof. Let

$$\mathbf{c}_t(\omega) = \frac{meas(M_\gamma - M_\omega)}{\xi_t^n} h_t\left(\frac{\rho(\gamma,\varepsilon)}{\xi_t}\right) > 0.$$

Since the sequence $\{\xi_t\}_{t\geq 0}$ is non increasing,

$$\mathbf{c}_t(\omega) \geq \frac{meas(M_\gamma - M_\omega)}{\xi_t^n} h_t\left(\frac{\rho(\gamma,\varepsilon)}{\xi_t}\right) > 0.$$

Thus, Equation (15) shows that

$$\sum_{t=0}^{+\infty} \mathbf{c}_t(\omega) \geq \frac{meas(M_\gamma - M_\omega)}{\xi_t^n} \sum_{t=0}^{+\infty} h_t\left(\frac{\rho(\gamma,\varepsilon)}{\xi_t}\right) = +\infty.$$

We can conclude that $V{=}\eta^*$ almost surely by applying Lemmas 1 and 2. □

Theorem 2. *Let Z_t be defined by (12) and ξ_t by*

$$\xi_t = \sqrt{\frac{\hat{b}}{\log(t+\hat{a})}}, \tag{16}$$

where $\hat{b} > 0$, $\hat{a} > 0$, and t is the iteration number. If $x^0 \in$ M, then for $\hat{b}$ large enough, $V{=}\eta^*$ almost surely.

Proof. We have

$$\phi_t(Z) = \frac{1}{(\sqrt{2\pi})^n} \exp(-\frac{1}{2}\|Z\|^2) = h_t(\|Z\|) > 0,$$

so

$$h_t\left(\frac{\rho(\gamma,\varepsilon)}{\xi_t}\right) = \frac{1}{(\sqrt{2\pi})^n (t+\hat{a})^{\rho(\gamma,\varepsilon)^2/2\hat{b}}}.$$

For $\hat{b}$, such that

$$0 < \frac{\rho(\gamma,\varepsilon)^2}{2\hat{b}} < 1,$$

we have

$$\sum_{t=0}^{+\infty} h_t \left(\frac{\rho(\gamma, \varepsilon)}{\xi_t} \right) = +\infty;$$

furthermore, as per the previous Theorem 2, it can be deduced that V is almost surely equal to η^*. □

5. Numerical Results

In this section, we present numerical results of six examples implemented using the CGB method and the perturbed RPCGB method. Our aim is to compare the performance of these two algorithms.

We begin by applying the algorithm to the initial value, which is $\mathbf{X}^0 = x^0 \in M$. At each step $t \geq 0$, $\mathbf{X}^t$ is known, and we calculate $\mathbf{X}^{t+1}$.

k_{sto} denotes the number of perturbations. When $k_{\text{sto}} = 0$, the method used is the conditional gradient with bisection, without any perturbations (unperturbed conditional gradient with bisection method).

The Gaussian variates used in our experiments are generated using regular generator calls. Specifically, we use

$$\xi_t = \sqrt{\frac{\hat{b}}{\log(t+2)}}, \quad \text{where } \hat{b} > 0.$$

The definitions of the methods listed in the tables are as follows:

(i) "CGB", the method of conditional gradient and bisection;
(ii) "RPCGB", the method of random perturbation of conditional gradient and bisection.

The proposed RPCGB algorithm is implemented using the MATLAB programming language. We evaluate the performance of the RPCGB method and compare it with the CGB method for high-dimensional problems. We test the efficacy of these algorithms on several problems [29–32] with linear constraints, using predetermined feasible starting points x^0. The results are presented in Tables 1–6 and Figures 1–6, where n denotes the dimension of the problem under consideration and n_c represents the number of constraints. The reported test results include the optimal value f^*_{RPCGB} and the number of iterations k_{iter}.

The optimal line search process of CGB and RPCGB is found using the bisection method with $\epsilon = 10^{-4}$. We terminate the iterative process when either the best solution (global solution) is found or the maximum number of iterations has been reached.

All algorithms were run on a TOSHIBA Intel(R) Core(TM) CPU running at 2.40 GHz with 6 GB of RAM, a Core i7 processor, and the 64-bit Windows 7 Professional operating system. The "CPU" column in the table displays the mean CPU time for one run in seconds.

Problem 1. *The Neumaier 3 Problem (NF3) is a mathematical optimization problem introduced by Arnold Neumaier in 2003 (see [29]). The problem is defined as follows:*

$$\begin{cases} \textit{minimize:} & \sum\limits_{j=1}^{n} (x_j - 1)^2 - \sum\limits_{j=2}^{n} x_j x_{j-1} \\ \textit{subject to:} & -n^2 \leq x_j \leq n^2, \qquad j = 1, \ldots, n \end{cases}$$

Table 1. The results obtained from the CGB and RPCGB algorithms.

Problem 1		**Algorithm**						
		CGB			**RPCGB**			
n	n_c	k_{iter}	CPU	f^*_{CGB}	k_{iter}	CPU	k_{sto}	f^*_{RPCGB}
500	1000	9	0.11	-1.06×10^6	9	0.68	2	-1.61×10^7
900	1800	9	0.13	-3.41×10^6	4	0.25	2	-6.32×10^7
2000	4000	10	3.27	-1.08×10^7	6	5.31	5	-3.53×10^8
4000	8000	12	19.33	-5.53×10^7	7	27.27	10	-1.48×10^9
6000	12,000	19	35.70	-1.34×10^7	9	46.54	10	-3.35×10^9
9000	18,000	27	87.92	-1.61×10^7	13	91.16	10	-7.62×10^9

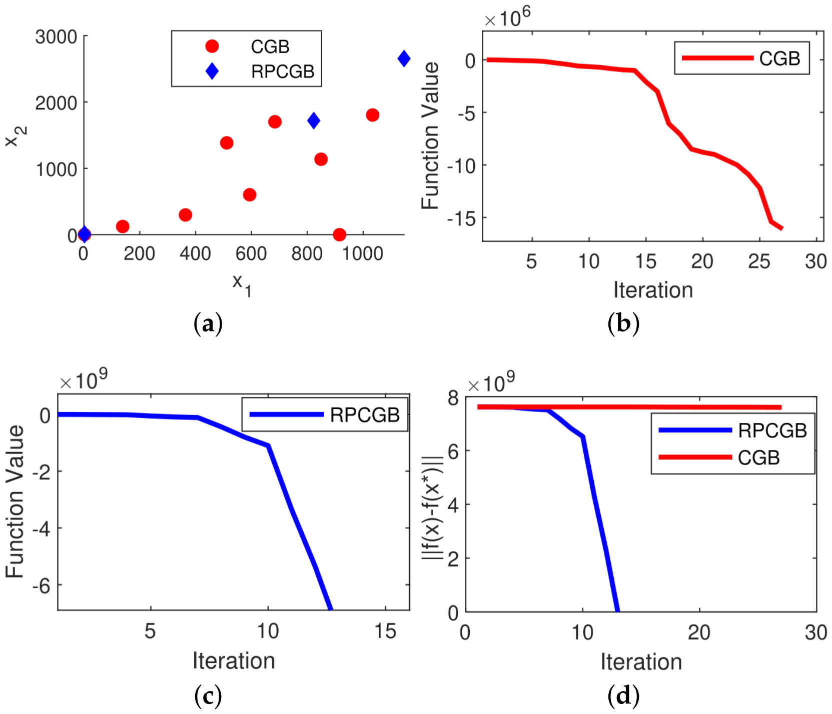


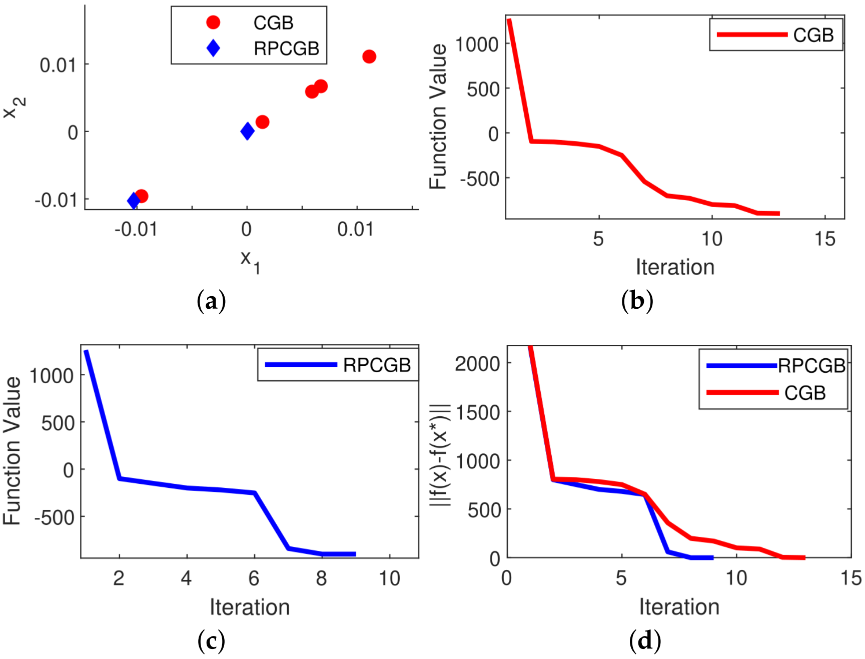
Figure 1. (**a**) Scatter plot of solution distribution for the CGB and RPCGB algorithms (n = 900). (**b**) Objective function values over iterations for the CGB method (n = 9000). (**c**) Objective function values over iterations for the RPCGB method (n = 9000). (**d**) Convergence performance for the CGB and RPCGB methods with n = 9000 for Problem 1.

Problem 2. *The Cosine Mixture Problem (CM) is an optimization problem introduced by Breiman and Cutler in 1993 (see [29]). The problem is defined as follows:*

$$\begin{cases} \textit{minimize:} \quad \sum\limits_{j=1}^{n} x_j^2 - 0.1 \sum\limits_{j=1}^{n} \cos(5\pi x_j) \\ \\ \textit{subject to:} \quad -1 \leq x_j \leq 1, \qquad j = 1, \ldots, n \end{cases}$$

Table 2. The results obtained from the CGB and RPCGB algorithms.

Problem 2		Algorithm						
		CGB			RPCGB			
n	n_c	k_{iter}	CPU	f^*_{CGB}	k_{iter}	CPU	k_{sto}	f^*_{RPCGB}
500	1000	4	0.07	−50	2	0.02	1	−50
900	1800	5	0.09	−90	2	0.05	1	−90
2000	4000	7	0.11	−199.99	3	0.09	1	−200
4000	8000	9	0.19	−400	4	0.12	1	−400
6000	12,000	10	0.37	−599.99	7	0.15	1	−600
9000	18,000	13	0.42	−900	9	0.21	1	−900



(**a**)

(**b**)

(**c**)

(**d**)

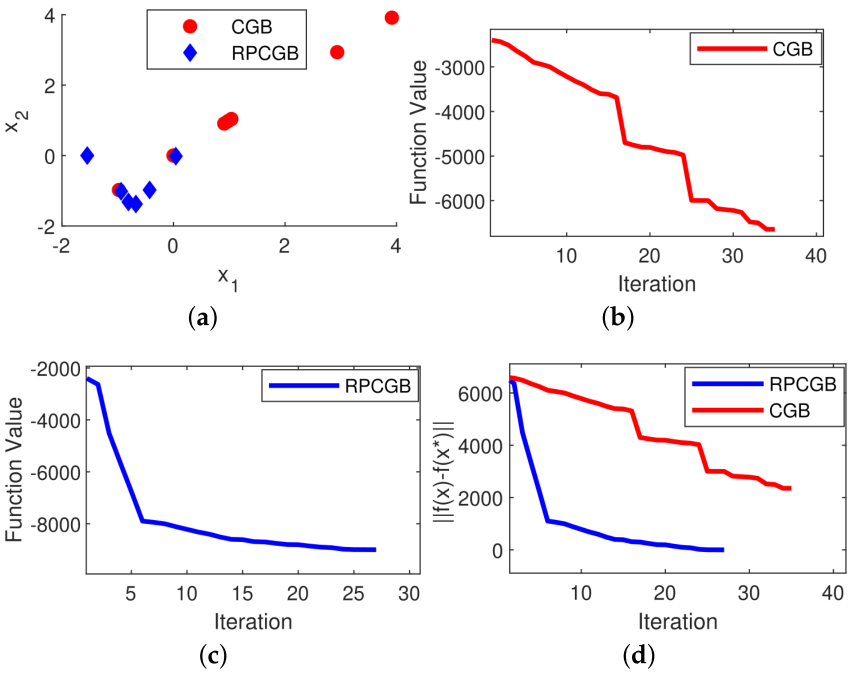
Figure 2. (**a**) Scatter plot of solution distribution for the CGB and RPCGB algorithms (n = 900). (**b**) Objective function values over iterations for the CGB method (n = 9000). (**c**) Objective function values over iterations for the RPCGB method (n = 9000). (**d**) Convergence performance for the CGB and RPCGB methods with n = 9000 for Problem 2.

Problem 3. *The Inverted Cosine Wave Function or the Cosine Mixture with Exponential Decay Problem. This is a commonly used benchmark problem in global optimization and was introduced by Price et al. in 2006 (see [30]). The problem is defined as follows:*

$$\begin{cases} \textit{minimize:} & -\sum_{j=1}^{n} \exp(\dfrac{-x_j^2 - x_{j+1}^2 - 0.5x_j x_{j+1}}{8}) \cos(4\sqrt{x_j^2 + x_{j+1}^2 + 0.5x_j x_{j+1}}) \\ \textit{subject to:} & -5 \le x_j \le 5, \qquad j = 1, \dots, n \end{cases}$$

Table 3. The results obtained from the CGB and RPCGB algorithms.

Problem 3		**Algorithm**						
		CGB			**RPCGB**			
n	n_c	k_{iter}	CPU	f^*_{CGB}	k_{iter}	CPU	k_{sto}	f^*_{RPCGB}
500	1000	5	0.05	−498.99	3	0.04	1	−499
900	1800	7	0.07	−898.99	6	0.07	1	−898.76
2000	4000	11	0.12	−1475.44	11	0.19	1	−1998.99
4000	8000	18	0.31	−2951.62	12	0.47	1	−3999
6000	12,000	24	0.79	−4427.81	19	0.74	1	−5998.87
9000	18,000	35	1.03	−6642.08	27	0.96	1	−8998.25



(**a**)

(**b**)

(**c**)

(**d**)

Figure 3. (**a**) Scatter plot of solution distribution for the CGB and RPCGB algorithms (n = 900). (**b**) Objective function values over iterations for the CGB method (n = 9000). (**c**) Objective function values over iterations for the RPCGB method (n = 9000). (**d**) Convergence performance for the CGB and RPCGB methods with n = 9000 for Problem 3.

Problem 4. *The Epistatic Michalewicz Problem (EM) is a type of optimization problem commonly used as a benchmark in evolutionary computation and optimization. It was introduced by Michalewicz in 1996 (see [29]). The problem is defined as follows:*

$$\begin{cases} \textit{minimize:} \quad -\sum_{j=1}^{n} \sin(y_j)(\sin(\frac{jy_j^2}{\pi}))^{20} \\ \textit{subject to:} \quad 0 \le x_j \ \le \pi, \quad j = 1, \ldots, n \\ y_j = \begin{cases} x_j \cos(\frac{\pi}{6}) - x_{j+1} \sin(\frac{\pi}{6}), & j = 1, 3, 5, \ldots, \le n \\ x_j \sin(\frac{\pi}{6}) + x_{j+1} \cos(\frac{\pi}{6}), & j = 2, 4, 6, \ldots, \le n \\ x_j, & j = n \end{cases} \end{cases}$$

Table 4. The results obtained from the CGB and RPCGB algorithms.

Problem 4		**Algorithm**						
		CGB			**RPCGB**			
n	n_c	k_{iter}	CPU	f^*_{CGB}	k_{iter}	CPU	k_{sto}	f^*_{RPCGB}
500	1000	23	11.41	−131.81	45	19.53	25	−176.72
900	1800	29	17.06	−214.79	57	21.34	30	−293.51
2000	4000	34	42.66	−417.95	69	71.26	30	−536.38
4000	8000	56	67.18	−768.22	75	96.02	50	-1.06×10^3
6000	12,000	73	79.63	−846.01	94	110.63	70	-1.11×10^3
9000	18,000	89	99.25	−919.85	124	136.71	90	-1.35×10^3

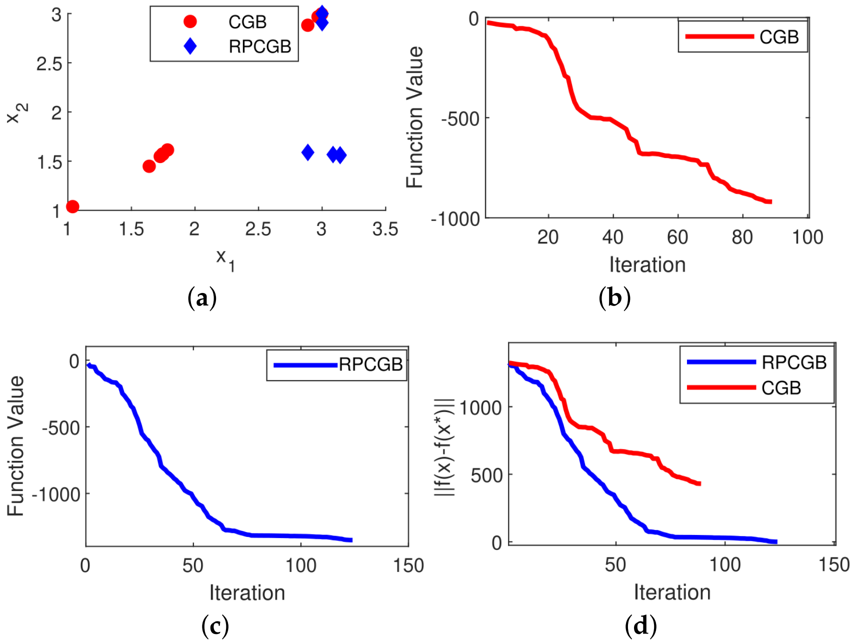


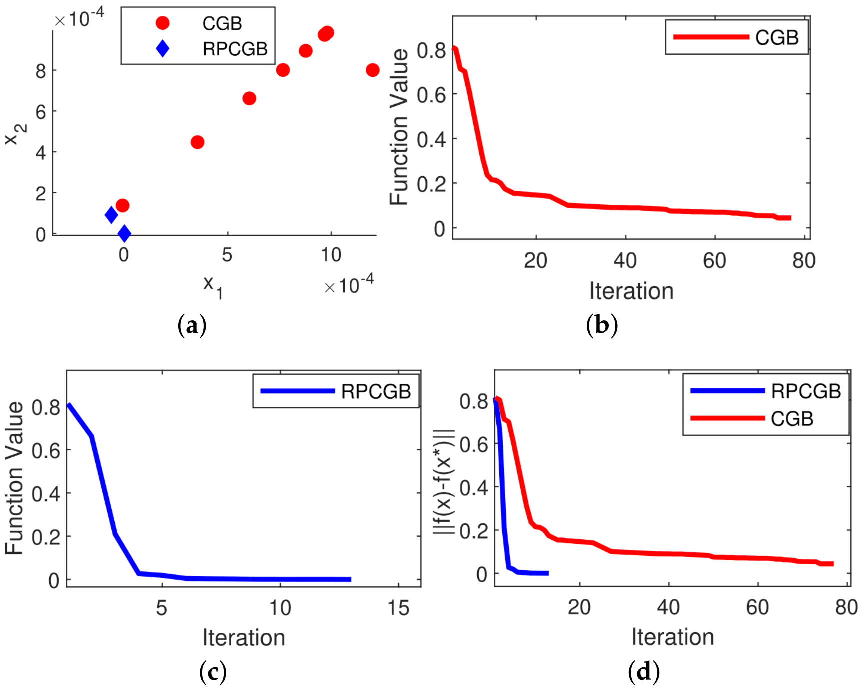
Figure 4. (**a**) Scatter plot of solution distribution for the CGB and RPCGB algorithms (n = 2). (**b**) Objective function values over iterations for the CGB method (n = 9000). (**c**) Objective function values over iterations for the RPCGB method (n = 9000). (**d**) Convergence performance for the CGB and RPCGB methods with n = 9000 for Problem 4.

Problem 5. *The problem is a mathematical optimization problem used in global optimization which comes from [32] and is defined as follows:*

$$\begin{cases} \textit{minimize:} & \sum\limits_{j=1}^{n} (x_j^2 - 10\cos(2\pi x_j) + 10) \\ \textit{subject to:} & \sum\limits_{j=1}^{n} x_j = 0, \\ & -5.12 \le x_j \le 5.12, \quad j = 1, \ldots, n \end{cases}$$

Table 5. The results obtained from the CGB and RPCGB algorithms.

Problem 5		**Algorithm**						
		CGB			**RPCGB**			
n	n_c	k_{iter}	**CPU**	f^*_{CGB}	k_{iter}	CPU	k_{sto}	f^*_{RPCGB}
500	500	19	8.04	4.71×10^{-4}	14	9.23	70	1.38×10^{-11}
900	900	29	14.13	8.51×10^{-4}	14	14.25	70	3.05×10^{-10}
2000	2000	42	33.09	0.0097	26	45.31	150	4.96×10^{-10}
4000	4000	30	53.63	0.0194	17	97.27	200	7.93×10^{-7}
6000	6000	59	71.47	0.0291	19	122.54	300	7.93×10^{-9}
9000	9000	77	92.55	0.0436	13	153.16	800	7.93×10^{-5}



(**a**) (**b**) (**c**) (**d**)

Figure 5. (**a**) Scatter plot of solution distribution for the CGB and RPCGB algorithms (n = 2). (**b**) Objective function values over iterations for the CGB method (n = 9000). (**c**) Objective function values over iterations for the RPCGB method (n = 9000). (**d**) Convergence performance for the CGB and RPCGB methods with n = 9000 for Problem 5.

Problem 6. *Rastrigin's function is a non-convex, multi-modal function commonly used as a benchmark problem in optimization. It was introduced by Rastrigin in 1974 (see [31]) and is defined as:*

$$\begin{cases} \textit{minimize:} & \sum_{j=1}^{n} \cos(2\pi x_j \sin(\frac{\pi}{20})) \\ \\ \textit{subject to:} & x_j - x_{j+1} = 0.4 \quad j = 1, \ldots, n-1 \end{cases}$$

Table 6. The results obtained from the CGB and RPCGB algorithms.

Problem 6		Algorithm						
		CGB			**RPCGB**			
n	n_c	k_{iter}	**CPU**	f^*_{CGB}	k_{iter}	**CPU**	k_{sto}	f^*_{RPCGB}
500	499	2	0.09	−1.8856	5	0.28	10	−4.0147
900	899	19	1.65	0.7033	7	0.38	10	−4.4432
1000	999	13	0.92	0.1626	12	1.08	10	−4.6572
2000	1999	21	1.70	2.0535	8	17.24	40	−3.1399
3000	2999	23	2.96	0.9158	9	27.54	60	−3.2145
4000	3999	12	1.47	2.0097	11	48.56	100	−4.6168

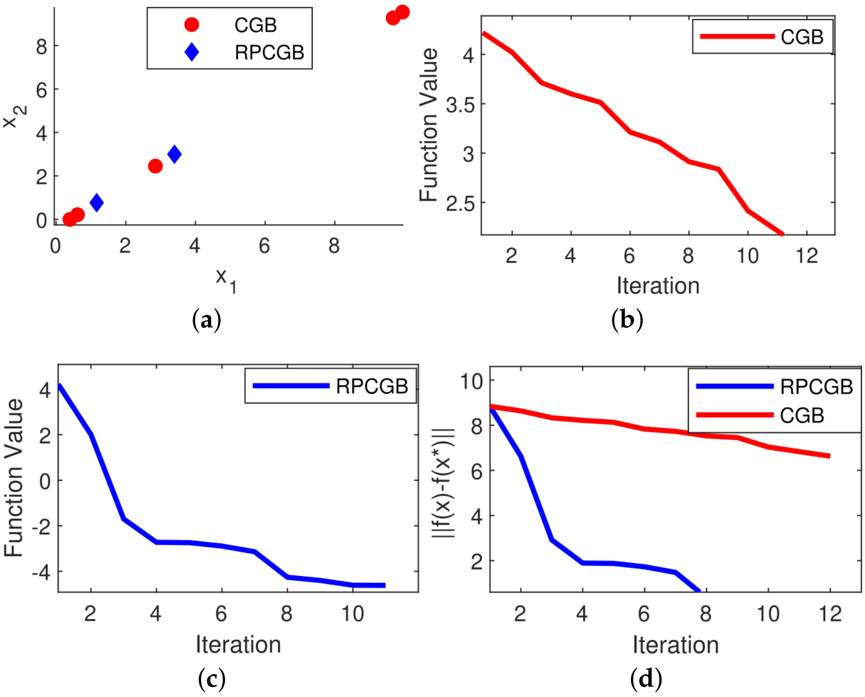


Figure 6. (**a**) Scatter plot of solution distribution for the CGB and RPCGB algorithms (n = 2). (**b**) Objective function values over iterations for the CGB method (n = 4000). (**c**) Objective function values over iterations for the RPCGB method (n = 4000). (**d**) Convergence performance for the CGB and RPCGB methods with n = 4000 for Problem 6.

To gain a deeper understanding of the effect of the modifications on the proposed algorithm, we utilized a scatter plot that illustrates the distribution of the algorithm's solutions in two dimensions for both the CGB and RPCGB algorithms. The goal was to generate scatter plots to depict the distribution of solutions in a 2D space for all problems when using two variables (n = 2). However, we found that for Problems 1 to 3, we were able to obtain the optimal solution value using only one iteration, which made the creation of a scatter plot unnecessary in this case. Thus, we generated a scatter plot of the solution distribution for the case of n = 900. This allowed us to effectively illustrate the distribution of solutions using scatter plots. In each algorithm, the scatter plot was generated at the first iteration and continued up to the limit of the required number of iterations to reach the solution. After analyzing Figures 1–6, we concluded that the modified algorithm exhibited a more tightly clustered solution distribution in the scatter plot compared to the original algorithm.

We also present in Figures 1–6 the results of plotting the objective function values in each iteration and the convergence performance for the CGB and RPCGB methods with 9000 variables. The plots (**d**) show that the proposed algorithm performs better than the

CGB algorithm, with the majority of cases showing that the suggested algorithm achieves convergence in fewer iterations than the CGB algorithm. However, there is an exception observed in Problem 4, as presented in Figure 4, where the CGB algorithm stops early. This demonstrates that the convergence behavior of optimization algorithms can vary based on the problem being solved. It is worth noting that both algorithms terminated their execution before the 30th iteration, which is because a stopping criterion of approximately $\epsilon = 10^{-4}$ was met. The algorithms cease their iterations upon reaching the optimal solution (the local or global solution) or upon reaching the maximum number of iterations. We observe that the random perturbation has a significant effect on the convergence. This suggests that the changes made to the algorithm led to an improvement in its performance.

The results presented in Tables 1–6 demonstrate that the CGB algorithm is capable of obtaining global solutions in certain instances regardless of the number of dimensions, such as Problem 2 (see Table 2). However, in some cases, the CGB algorithm fails to obtain global solutions as the number of dimensions increases, as seen in Problem 3 (see Table 3). In contrast, our RPCGB algorithm can obtain a global solution for all cases, and the computational results indicate that it performs effectively for these high-dimensional problems. These results also indicate that, for larger-dimensions problems, the CGB method necessitates a greater number of iterations to finalize the optimization process, whereas the RPCGB method does not, as evidenced by Problem 5. This difference can be explained by the k_{sto} parameter, which denotes the number of perturbations. If the number of perturbations is raised, then the number of iterations needed to achieve the optimal solution is reduced.

When analyzing the obtained results, it is evident that the perturbed conditional gradient method with bisection algorithm (RPCGB) performs well compared to the conditional gradient algorithm (CGB).

6. Conclusions

In this work, we generalized the RPCGB method to solve large-scale non-convex optimization problems. The algorithms mentioned in this paper, specifically the conditional gradient algorithm, are commonly used optimization techniques for solving convex optimization problems. However, in the case of non-convex optimization problems, these algorithms may converge to a local minimum instead of the global minimum. To overcome this problem, the proposed approach introduces a random perturbation to the optimization problem. Specifically, at each iteration of the algorithm, a random perturbation is added to the Q_t operator, which allows the algorithm to escape from local minima and explore the search space more effectively. The bisection algorithm is used to find the optimal step size along the search direction. It involves solving a one-dimensional optimization problem to find the step size that minimizes the objective function. By combining these two algorithms with the random perturbation approach, the proposed method is able to efficiently explore the search space for large-scale non-convex optimization problems under linear constraints and reach a global minimum. The tuning of the parameters k_{sto} and $\hat{b}$ is related to the main difficulty in applying random perturbation in practice.

The RPCGB algorithm has the ability to solve various problems, such as control systems, as well as optimization problems in machine learning, robotics, and image reconstruction. There are problems that contain a part that is not smooth. Therefore, in the future, we plan to use the random perturbation of the conditional subgradient method with bisection algorithm to solve non-convex, non-smooth (non-differentiable) programming under linear constraints. Additionally, we intend to use the perturbed conditional gradient method to address non-convex optimization problems in support vector machines (SVM).

Author Contributions: Conceptualization, methodology, software, validation, formal analysis, investigation, writing—original draft preparation, writing—review and editing: A.E. and A.E.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to thank the referees for their fruitful suggestions, which helped us improve the quality of the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Frausto Solis, J.; Purata Aldaz, J.L.; González del Angel, M.; González Barbosa, J.; Castilla Valdez, G. SAIPO-TAIPO and Genetic Algorithms for Investment Portfolios. *Axioms* **2022**, *42*, 11. [CrossRef]
2. Kuang, X.; Lamadrid, A.J.; Zuluaga, L.F. Pricing in non-convex markets with quadratic deliverability costs. *Energy Econ.* **2019**, *80*, 123–131. [CrossRef]
3. Pang, L.P.; Chen, S.; Wang, J.H. Risk management in portfolio applications of non-convex stochastic programming. *Appl. Math. Comput.* **2015**, *258*, 565–575. [CrossRef]
4. Chan, R.; Lanza, A.; Morigi, S.; Sgallari, F. Convex non-convex image segmentation. *Numer. Math.* **2018**, *138*, 635–680. [CrossRef]
5. Oh, S.; Woo, H.; Yun, S.; Kang, M. Non-convex hybrid total variation for image denoising. *J. Vis. Commun. Image Represent.* **2013**, *24*, 332–344. [CrossRef]
6. Chu, H.; Zheng, L.; Wang, X. Semi-blind millimeter-wave channel estimation using atomic norm minimization. *IEEE Commun.* **2018**, *22*, 2535–2538. [CrossRef]
7. Di Martino, F.; Sessa, S. A Multilevel Fuzzy Transform Method for High Resolution Image Compression. *Axioms* **2022**, *11*, 551. [CrossRef]
8. Wen, S.; Liu, G.; Chen, Q.; Qu, H.; Wang, Y.; Zhou, P. Optimization of precoded FTN signaling with MMSE-based turbo equalization. In Proceedings of the IEEE International Conference on Communications (ICC), Shanghai, China, 20–24 May 2019; pp. 1–6.
9. Kaveh, A.; Hamedani, K.B. Improved arithmetic optimization algorithm and its application to discrete structural optimization. *Structures* **2022**, *35*, 748–764. [CrossRef]
10. Zeng, G.Q.; Xie, X.Q.; Chen, M.R.; Weng, J. Adaptive population extremal optimization-based PID neural network for multivariable nonlinear control systems. *Swarm Evolut. Comput.* **2019**, *44*, 320–334. [CrossRef]
11. El Mouatasim, A. Fast gradient descent algorithm for image classification with neural networks. *Signal Image Video Process.* **2020**, *14*, 1565–1572. [CrossRef]
12. Nanuclef, R.; Frandi, E.; Sartori, C.; Allende, H. A novel Frank-Wolfe algorithm. Analysis and applications to large-scale SVM training. *Inf. Sci.* **2014**, *285*, 66–99. [CrossRef]
13. Zheng, M.; Wang, F.; Hu, X.; Miao, Y.; Cao, H.; Tang, M. A Method for Analyzing the Performance Impact of Imbalanced Binary Data on Machine Learning Models. *Axioms* **2022**, *11*, 607. [CrossRef]
14. Berrada, L.; Zisserman, A.; Kumar, M.P. Deep Frank-Wolfe for neural network optimization. In Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019.
15. Amoura, N.; Benaissa, B.; Al Ali, M.; Khatir, S. Deep Neural Network and YUKI Algorithm for Inner Damage Characterization Based on Elastic Boundary Displacement; Capozucca. *Lect. Notes Civ. Eng.* **2023**, *317*, 220–233.
16. Benaissa, B.; Hocine, N.A.; Khatir, S.; Riahi, M.K.; Mirjalili, S. YUKI Algorithm and POD-RBF for Elastostatic and Dynamic Crack Identification. *J. Comput. Sci.* **2021**, *55*, 101451. [CrossRef]
17. Moxnes, E. *An Introduction to Deterministic and Stochastic Optimization, Analytical methods for Dynamic Modelers*; MIT Press: Cambridge, MA, USA, 2015.
18. Pogu, M.; Souza de Cursi, J.E. Global optimization by random perturbation of the gradient method with a fixed parameter. *J. Glob. Optim.* **1994**, *5*, 159–180. [CrossRef]
19. Mandt, S.; Hoffman, M.; Blei, D. A variational analysis of stochastic gradient algorithms. In Proceedings of the 33rd International Conference on Machine Learning, New York, NY, USA, 19–24 June 2016; pp. 354–363.
20. Nesterov, Y.; Spokoiny, V. Random gradient-free minimization of convex functions. *Found. Comput. Math.* **2017**, *17*, 527–566. [CrossRef]
21. Lu, S; Zhao, Z; Huang, K; Hong, M. Perturbed projected gradient descent converges to approximate second-order points for bound constrained nonconvex problems. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, 12–17 May 2019; pp. 5356–5360.
22. El Mouatasim, A.; Ettahiri, A. Conditional gradient and bisection algorithms for non-convex optimization problem with random perturbation. *Appl. Math. E-Notes* **2022**, *22*, 142–159.
23. Frank, M.; Wolfe, P. An Algorithm for Quadratic Programming. *Naval Res. Logist. Q.* **1956**, *3*, 95–110. [CrossRef]
24. Khamaru, K.; Wainwright, M.J. Convergence guarantees for a class of non-convex and non-smooth optimization problems. *J. Mach. Learn. Res.* **2019**, 20, 1–52.
25. Baushev, A.N.; Morozova, E.Y. A multidimensional bisection method for minimizing function over simplex. *Lect. Notes Eng. Comput. Sci.* **2007**, *2*, 801–803.
26. El Mouatasim, A.; Ellaia, R.; Souza de Cursi, J.E. Random perturbation of projected variable metric method for linear constraints nonconvex nonsmooth optimization. *Int. J. Appl. Math. Comput. Sci.* **2011**, *21*, 317–329.

27. Bouhadi, M.; Ellaia, R.; Souza de Cursi, J.E. Random perturbations of the projected gradient for linearly constrained problems. *Nonconvex Optim. Appl.* **2001**, 487–499.
28. L'Ecuyer, P.; Touzin, R. On the Deng-Lin random number generators and related methods. *Stat. Comput.* **2003**, *14*, 5–9. [CrossRef]
29. Ali, M.M.; Khompatraporn, C.; Zabinsky, Z.B. A numerical evaluation of several stochastic algorithms on selected continuous global optimization test problems. *J. Glob. Optim.* **2005**, *31*, 635–672. [CrossRef]
30. Aslimani, N.; Ellaia, R. A new chaos optimization algorithm based on symmetrization and levelling approaches for global optimization. *Numer. Algorithms* **2018**, *79*, 1021–1047. [CrossRef]
31. Che, H.; Li, C.; He, X.; Huang, T. An intelligent method of swarm neural networks forequalities constrained nonconvex optimization. *Neurocomputing* **2015**, *167*, 569–577. [CrossRef]
32. Li, C.; Li, D. An extension of the Fletcher Reeves method to linear equality constrained optimization problem. *Appl. Math. Comput.* **2003**, *219*, 10909–10914. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.