Conditional Gradient And Bisection Algorithms For Non-convex Optimization Problem With Random Perturbation

Abdelkrim El Mouatasim†, Abderrahmane Ettahiri‡

Received 23 November 2020

Abstract

In this paper, we propose an implementation of stochastic perturbation of conditional gradient and bisection (SPCGB) method (a.k.a. Frank-Wolfe method) for solving non-convex differentiable programming under linear constraints. The goal is to attempt to avoid getting stuck in local optimum solutions. Theoretical results guarantee the convergence of the proposed method towards a global minimizer. To demonstrate the effectiveness of our method, some numerical results of small and medium scale problems are given.

1 Introduction

Convex optimization has played an important role in recent years with the advent of the computer to study a given phenomenon, or to study a range of phenomena. A main challenge today is on non-convex problems in these phenomena. There exist several application areas for non-convex optimization with linear constraints (NCOLC) problems like combinatorial optimization (water distribution [9], co-localization image and video), optimal control [10], integer programming of call center [2], machine learning [20, 21], and or learning neural networks based on parsimonious coding and conditional gradient algorithm [4]. This algorithm also known as the Frank-Wolfe, was originally proposed by Marguerite Frank and Philip Wolfe in 1956 [13], is one of the oldest methods for nonlinear constrained optimization and has seen an impressive revival in recent years due to its low memory requirement and projection-free iterations. It makes it possible to approximate to each iteration a function by its development in first-order Taylor series.

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

\[
\begin{align*}
\text{minimize} & \quad f(x) \\
\text{subject to} & \quad Ax \leq b \\
& \quad \ell \leq x \leq \eta
\end{align*}
\]

(1)

where \( f : \mathbb{R}^n \to \mathbb{R} \) is a twice 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, \( \ell \) and \( \eta \), may contain some infinite components; and

\[
\begin{align*}
\text{minimize} & \quad f(x) \\
\text{subject to} & \quad Ax = b \\
& \quad 0 \leq x
\end{align*}
\]

(2)

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

In convex situations, the global optimization problem can be tackled by a set of classical methods, such as, for example, those based on the gradient, which have shown their effectiveness in this field. When...
the situation is not convex, this problem cannot be solved using the classic deterministic methods like the conditional gradient. The stochastic algorithms like the genetic algorithm and the simulated annealing algorithm are also inefficient for solving this type of problems. For this reason, in order to solve this kind of problems, we try to stochastically perturb the deterministic classic method.

The problem (2) can be numerically approached by using conditional gradient with bisection (CGB) method, which generates a sequence \( \{x^k\}_{k \geq 0} \), where \( x^0 \) is an initial feasible point and, for each \( k > 0 \), a new feasible point \( x^{k+1} \) is generated from \( x^k \) by using an operator \( Q_k \) (see Section 3). Thus, the iterations are given by:

\[
\forall k \geq 0 : x^{k+1} = Q_k(x^k).
\]

We introduce in this paper a different approach, inspired from the method of stochastic perturbations introduced in [23] for unconstrained minimization of continuously differentiable functions and adapted to linearly constrained problems in [8].

In such a method, the sequence \( \{x^k\}_{k \geq 0} \) is replaced by a random vectors sequence \( \{X^k\}_{k \geq 0} \) and the iterations are modified as follows:

\[
\forall k \geq 0 : X^{k+1} = Q_k(X^k) + P_k,
\]

where \( P_k \) is a suitable random variable, usually referred as the stochastic perturbation. The sequence \( \{P_k\}_{k \geq 0} \) must converge to zero slowly enough in order to prevent convergence of the sequence \( \{X^k\}_{k \geq 0} \) to a local minimum (see Section 4).

The rest of the article is organized as follows. In section 2, we introduce some notations and give some precise assumptions that will be useful for the rest of the article. The principle of the conditional gradient with bisection method is recalled in section 3. Then, in section 4, we present the stochastic perturbation of CGB method. Finally, in section 5, we provide some numerical experiments of linear constraints non-convex optimization test of small and medium scale problems.

### 2 Notations and Assumptions

We use the following notations:

- \( E = \mathbb{R}^n \), the n-dimensional positive real Euclidean space.
- \( x = (x_1, \ldots, x_n)^t \in E \).
- \( \|x\| = \sqrt{x^t x} = (x_1^2 + \cdots + x_n^2)^{1/2} \) the Euclidean norm of \( x \).
- \( x^t \) denotes the transpose of \( x \).

Let

\[
S = \{ x \in E \mid Ax = b, \ x \geq 0 \}.
\]

The objective function is \( f : E \to \mathbb{R} \), its lower bound on \( S \) is denoted by \( \alpha^* \) i.e. \( \alpha^* = \min S f \). Let us introduce

\[
S_\lambda = C_\lambda \cap S; \quad \text{where} \quad C_\lambda = \{ x \in E \mid f(x) \leq \lambda \}.
\]

We assume that

\[
f \text{ is twice continuously differentiable on } E, \quad (3)
\]

\[
\forall \lambda > \alpha^* : S_\lambda \text{ is not empty, closed and bounded}, \quad (4)
\]

\[
\forall \lambda > \alpha^* : \text{meas}(S_\lambda) > 0, \quad (5)
\]

where \( \text{meas}(S_\lambda) \) is the measure of \( S_\lambda \).

Since \( E \) is a finite dimensional space, the assumption (4) is verified when \( S \) is bounded or \( f \) is coercive, i.e., \( \lim_{\|x\| \to +\infty} f(x) = +\infty \). Assumption (4) is verified when \( S \) contains a sequence of neighborhoods of a point
of optimum $x^*$ having strictly positive measure, i.e., when $x^*$ can be approximated by a sequence of points of the interior of $S$.

We observe that the assumptions (3) and (4) yield that

$$ S = \bigcup_{\lambda > \alpha^*} S_\lambda, \quad i.e., \quad \forall x \in S : \exists \lambda > \alpha^* \text{ such that } x \in S_\lambda. $$

From (3)–(4), one has

$$ \gamma_1 = \sup \{ \| \nabla f(x) \| : x \in S_\lambda \} < +\infty. $$

Consequently, one deduces

$$ \gamma_2 = \sup \{ \| d \| : x \in S_\lambda \} < +\infty, $$

where $d$ is the direction of conditional gradient method. Thus

$$ \beta(\lambda, \varepsilon) = \sup \{ \| y - (x + \eta d) \| : (x, y) \in S_\lambda \times S_\lambda, \ 0 \leq \eta \leq \varepsilon \} < +\infty, \quad (6) $$

where $\varepsilon, \eta$ are positive real numbers.

## 3 Conditional Gradient Method and Bisection Algorithms

In this section, we recall conditional gradient method for convex optimization, see Frank and Wolfe [13], as well as Demyanov and Rubinov [7], cited here for minimization problems. From now on, we consider a nonlinear programming problem with linear equality or inequality constraints of the form

$$ \begin{aligned} \min \ f(x) \\ s.t \ Ax = (or \leq) b \end{aligned} \quad (7) $$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is non-convex twice continuously differentiable function, $A$ is $m \times n$ matrix with $m \leq n$ and $b$ is a vector in $\mathbb{R}^m$.

In the conditional gradient algorithm one determines $d_k$ through the solution of the approximation of the problem (7) that is obtained by replacing the function $f$ with its first-order Taylor expansion around $x^k$:

$$ f(x) \sim f(x^k) + \nabla f(x^k)(x - x^k). $$

By eliminating the constants, this amounts to minimizing the linear function:

$$ \begin{aligned} \minimize \ \nabla f(x^k)^\top s \\ \text{subject to} \quad As = (or \leq) b \end{aligned} \quad (8) $$

This is an linear programming problem, and it gives an extreme point $s_k$ as an optimal solution. The search direction is $d_k := s_k - x^k$ and then updates

$$ x^{k+1} = Q_k(x^k) = x^k + \eta kd_k. \quad (9) $$

We determine the optimal step as the value $\eta_k$ such that

$$ f(x^k + \eta_k d_k) = \min_{0 \leq \eta \leq 1} \{ f(x^k + \eta d_k) \}. \quad (10) $$
3.1 Bisection Algorithm

We use the bisection algorithm in this paper to solve the unconstrained optimization problem with one variable (10). For example, see [3]. Let us denote the recursive bisection procedure by bis(h, a, b, ϵ). The inputs for this procedure are the h calculation procedure, the [a, b] segment, and the accuracy ϵ. The outputs are the approximation of x_m for the minimizer x*_m and h_m for the value of the h function minimum over the [a, b] segment.

The following steps are used in the iteration of the recursive procedure.

Step 0: If b – a ≥ ϵ, go to step 1, otherwise stop.

Step 1: Compute
\[ d = \frac{a + b}{2}, \quad a' = \frac{a + d}{2}, \quad b' = \frac{d + b}{2}, \quad h(d), \quad h(a'), \quad h(b'). \]

Step 2: If h(a') ≤ h(d) ≤ h(b'), set b = b'. If h(a') ≥ h(d) ≥ h(b'), set a = a'.

Step 3: If h(d) ≤ min{h(a'), h(b')}, set a = a', b = b'.

Step 4: Execute bis(h, a, b, ϵ) with new inputs.

3.2 Algorithm of Conditional Gradient

The conditional gradient algorithm is an iterative first-order optimization algorithm for constrained non-convex optimization, that given an initial guess x^0 constructs a sequence of estimates x^1, x^2, . . . that converges towards a solution of the optimization problem. The algorithm is defined as follows (Algorithm 1):

Algorithm 1 Standard conditional gradient algorithm
1: Choose a feasible point x(0) ∈ S
2: for k = 0 . . . K do
3: Compute s_k := LMO_S (∇ f(x(k))) := arg min_{s ∈ S} ∇ f(x(k))^T s (the linear minimization oracle)
4: Let d_k := s_k – x(k)
5: Compute g_k := ⟨−∇ f(x(k)), d_k⟩ (the CG direction)
6: if g_k < ε then return x(k)
7: Step size by optimal line search
   \[ η_k ∈ \arg \min_{η ∈ [0,1]} f(x(t) + η d_k) \]
8: Update \[ x(k+1) := x(k) + η_k d_k \]
9: end for
10: return x(K)

3.3 Convergence of Conditional Gradient for Non-convex Objectives

Let us present a convergence rate result which is valid for objectives with L-Lipschitz gradient but not necessarily convex. This was first proven by Simon Lacoste-Julien (see for instance [17]):

Theorem 1 (Convergence of CG on non-convex objectives) If f is differentiable with L-Lipschitz gradient and the domain D is a convex and compact set., then we have the following \( O(1/\sqrt{T}) \) bound on the best conditional gradient gap:

\[ \min_{0 \leq i \leq T} g_i \leq \frac{2h_0, L \text{diam}(D)^2}{\sqrt{T} + 1} \quad \text{for } t \geq 0, \]

where \( h_0 := f(x_0) - \min_{x \in D} f(x) \) is the initial global suboptimality.

Proof. See, [17].
4 Stochastic Perturbation of Conditional Gradient with Bisection (SPCGB) Method

From [13], it is well-known that if $f$ is not convex, the global minimum can not be found using a CGB algorithm. To overcome this difficulty, we propose an appropriate random perturbation. In the next, we will establish the convergence of SPCGB to a global minimum for non-convex optimization problems.

The sequence of real numbers $\{x^k\}_{k \geq 0}$ is replaced by a sequence of random variables $\{X^k\}_{k \geq 0}$ involving a random perturbation $P_k$ of the deterministic iteration (9); then we have $X^0 = x^0$;

$$\forall k \geq 0 \ X^{k+1} = Q_k(X^k) + P_k = X^k + \eta_k d^k + P_k = X^k + \eta_k (d^k + \frac{P_k}{\eta_k}), \tag{11}$$

where $\eta_k \neq 0$ satisfied the Step 8 in CG algorithm, and

$$\forall k \geq 1 \ P_k \text{ is independent from } (X^{k-1}, \ldots, X^0)$$

and

$$X \in S \Rightarrow Q_k(X) + P_k \in S.$$  
Equation (11) can be viewed as perturbation of the ascent direction $d^k$, which is replaced by a new direction $D_k = d^k + \frac{P_k}{\eta_k}$ and the iterations (11) become

$$X^{k+1} = X^k + \eta_k D_k.$$  

General properties defining convenient sequences of perturbation $\{P_k\}_{k \geq 0}$ can be found in the literature [8, 23]: usually, sequence of Gaussian laws may be used in order to produce elements satisfying these properties.

We introduce a random vector $Z_k$, we denote by $\Phi_k$ and $\phi_k$ the cumulative distribution function and the probability density of $Z_k$, respectively.

We denote by $F_{k+1}(y \mid X^k = x)$ the conditional cumulative distribution function

$$F_{k+1}(y \mid X^k = x) = P(X^{k+1} < y \mid X^k = x),$$

and the condition probability density of $X^{k+1}$ is denoted by $f_{k+1}$.

Let us introduce a sequence of n-dimensional random vectors $\{Z_k\}_{k \geq 0} \in S$. We consider also $\{\xi_k\}_{k \geq 0}$, a suitable decreasing sequence of strictly positive real numbers converging to 0 and such that $\xi_0 \leq 1$.

The optimal choice for $\eta_k$ is determined by Step 8. Let $P_k = \xi_k Z_k$ and

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

It follow that

$$F_{k+1}(y \mid X^k = x) = P \left( Z_k < \frac{y - Q_k(x)}{\xi_k} \right) = \Phi_k \left( \frac{y - Q_k(x)}{\xi_k} \right).$$

So, we have

$$f_{k+1}(y \mid X^k = x) = \frac{1}{\xi_k^2} \phi_k \left( \frac{y - Q_k(x)}{\xi_k} \right), \quad y \in S. \tag{12}$$

The relation (6) shows that

$$\|y - Q_k(x)\| \leq \beta(\lambda, \varepsilon) \quad \text{for } (x, y) \in S_\lambda \times S_\lambda.$$  

We assume that there exists a decreasing function $t \mapsto h_k(t) > 0$ on $\mathbb{R}^+$ such that

$$y \in S_\lambda \Rightarrow \phi_k \left( \frac{y - Q_k(x)}{\xi_k} \right) \geq h_k \left( \frac{\beta(\lambda, \varepsilon)}{\xi_k} \right). \tag{13}$$
For simplicity, let
\[ Z_k = 1_C(Z_k)Z_k, \] (14)
where \( Z \) is a random variable, for simplicity let \( Z \sim N(0,1) \).

The procedure generates a sequence \( U_k = f(X^k) \). By construction this sequence is increasing and upper bounded by \( \alpha^* \).

\[ \forall k \geq 0 : \ \alpha^* \geq U_{k+1} \geq U_k. \] (15)

Thus, there exists \( U \leq \alpha^* \) such that
\[ U_k \to U \ \text{for} \ k \to +\infty. \]

**Lemma 1** Let \( P_k = \xi_k Z_k \) and \( \gamma = f(x^0) \) if \( Z_k \) is given by (14). Then there exists \( v > 0 \) such that
\[ P(U_{k+1} > \theta | U_k \leq \theta) \geq \frac{\text{meas}(S_{\gamma} - S_{\theta})}{\xi^n_k} h_k \left( \frac{\beta(\gamma, \epsilon)}{\xi_k} \right) > 0 \ \forall \theta \in (\alpha^*, \alpha^* + v], \]
where \( n = \dim(E) \).

**Proof.** Let \( S_{\theta} = \{ x \in S \mid f(x) < \theta \} \), for \( \theta \in (\alpha^*, \alpha^* + v] \). Since \( S_{\lambda} \subset \hat{S}_{\theta} \), \( \alpha^* < \lambda < \theta \), it follows from (5) that \( S_{\theta} \) is not empty and has a strictly positive measure. If \( \text{meas}(S - \hat{S}_{\theta}) = 0 \) for any \( \theta \in (\alpha^*, \alpha^* + v] \), the result is immediate, since we have \( f(x) = \alpha^* \) on \( S \).

Let us assume that there exists \( \epsilon > 0 \) such that \( \text{meas}(S - \hat{S}_{\theta}) > 0 \). For \( \theta \in (\alpha^*, \alpha^* + \epsilon] \), we have \( \hat{S}_{\theta} \subset \hat{S}_{\epsilon} \) and \( \text{meas}(S - \hat{S}_{\theta}) > 0 \).

\[ P(X^k \notin \hat{S}_{\theta}) = \int_{S - \hat{S}_{\theta}} P(X^k \in dx) > 0 \]
for any \( \theta \in (\alpha^*, \alpha^* + \epsilon] \), since the sequence \( \{U_i\}_{i \geq 0} \) is increasing, we have also
\[ \{X_i\}_{i \geq 0} \subset S_{\gamma}. \] (16)

Thus
\[ P(X^k \notin \hat{S}_{\theta}) = \int_{S - \hat{S}_{\theta}} P(X^k \in dx) > 0 \]
for any \( \theta \in (\alpha^*, \alpha^* + \epsilon] \).

Let \( \theta \in (\alpha^*, \alpha^* + \epsilon] \), we have from (15)
\[ P(U_{k+1} > \theta \mid U_k \leq \theta) = P(X^{k+1} \in \hat{S}_{\theta} \mid X^i \notin \hat{S}_{\theta}, i = 0, \ldots, k). \]

But Markov chain yield that
\[ P(X^{k+1} \in \hat{S}_{\theta} \mid X^i \notin \hat{S}_{\theta}, i = 0, \ldots, k) = P(X^{k+1} \in \hat{S}_{\theta} \mid X^k \notin \hat{S}_{\theta}). \]

By the conditional probability rule
\[ P(X^{k+1} \in \hat{S}_{\theta} \mid X^k \notin \hat{S}_{\theta}) = \frac{P(X^{k+1} \in \hat{S}_{\theta}, X^k \notin \hat{S}_{\theta})}{P(X^k \notin \hat{S}_{\theta})}. \]

Moreover
\[ P(X^{k+1} \in \hat{S}_{\theta} \mid X^k \notin \hat{S}_{\theta}) = \int_{S - \hat{S}_{\theta}} P(X^k \in dx) \int_{\hat{S}_{\theta}} f_{k+1}(y \mid X^k = x) dy. \]

From (16) we have
\[ P(X^{k+1} \in \hat{S}_{\theta} \mid X^k \notin \hat{S}_{\theta}) = \int_{S - \hat{S}_{\theta}} P(X^k \in dx) \int_{\hat{S}_{\theta}} f_{k+1}(y \mid X^k = x) dy. \]
and
\[ P(X^{k+1} \in \hat{S}_\theta \mid X^k \notin \hat{S}_\theta) \geq \inf_{x \in S_\gamma - \hat{S}_\theta} \left\{ \int_{\hat{S}_\theta} f_{k+1}(y \mid X^k = x) dy \right\} \int_{S_\gamma - \hat{S}_\theta} P(X^k \in dx). \]

Thus
\[ P(X^{k+1} \in \hat{S}_\theta \mid X^k \notin \hat{S}_\theta) \geq \inf_{x \in S_\gamma - \hat{S}_\theta} \left\{ \int_{\hat{S}_\theta} f_{k+1}(y \mid X^k = x) dy \right\}. \]

Taking (12) into account, we have
\[ P(X^{k+1} \in \hat{S}_\theta \mid X^k \notin \hat{S}_\theta) \geq \frac{1}{\xi^k} \inf_{x \in S_\gamma - \hat{S}_\theta} \left\{ \int_{\hat{S}_\theta} \phi_k \left( \frac{y - Q_k(x)}{\xi_k} \right) dy \right\}. \]

The relation (6) shows that
\[ \|y - Q_k(x)\| \leq \beta(\gamma, \varepsilon). \]

and (13) yields that
\[ \phi_k \left( \frac{y - Q_k(x)}{\xi_k} \right) \geq h_k \left( \frac{\beta(\gamma, \varepsilon)}{\xi_k} \right). \]

Hence
\[ P(X^{k+1} \in \hat{S}_\theta \mid X^k \notin \hat{S}_\theta) \geq \frac{1}{\xi^k} \inf_{x \in S_\gamma - \hat{S}_\theta} \int_{\hat{S}_\theta} h_k \left( \frac{\beta(\gamma, \varepsilon)}{\xi_k} \right) dy, \]
\[ P(X^{k+1} \in \hat{S}_\theta \mid X^k \notin \hat{S}_\theta) \geq \frac{\text{meas}(S_\gamma - \hat{S}_\theta)}{\xi^k} h_k \left( \frac{\beta(\gamma, \varepsilon)}{\xi_k} \right). \]

### 4.1 Global Convergence

The global convergence is a consequence of the following result, which is a consequence of the Borel-Catelli’s lemma (for instance, see [23]):

**Lemma 2** Let \( \{U_k\}_{k \geq 0} \) be a increasing sequence, upper bounded by \( \alpha^* \). Then, there exists \( U \) such that \( U_k \to U \) for \( k \to +\infty \). Assume that there exists \( v > 0 \) such that for any \( \theta \in (\alpha^*, \alpha^* + v] \), there is a sequence of strictly positive real numbers \( \{c_k(\theta)\}_{k \geq 0} \) such that

\[ \forall k \geq 0: P(U_{k+1} > \theta \mid U_k \leq \theta) \geq c_k(\theta) > 0 \quad \text{and} \quad \sum_{k=0}^{+\infty} c_k(\theta) = +\infty. \]

Then \( U = \alpha^* \) almost surely.

**Proof.** For instance, see [18, 23].

**Theorem 2** Let \( \gamma = f(x^0) \). Assume that \( x^0 \in S \), the sequence \( \xi_k \) is non increasing and

\[ \sum_{k=0}^{+\infty} h_k \left( \frac{\beta(\gamma, \varepsilon)}{\xi_k} \right) = +\infty. \] (17)

Then \( U = \alpha^* \) almost surely.

**Proof.** Let
\[ c_k(\theta) = \frac{\text{meas}(S_\gamma - S_\theta)}{\xi^k} h_k \left( \frac{\beta(\gamma, \varepsilon)}{\xi_k} \right) > 0. \]
Since the sequence \( \{\xi_k\}_{k \geq 0} \) is non increasing,
\[
c_k(\theta) \geq \frac{\text{meas}(S_\gamma - S_\theta)}{\xi_k^n} h_k \left( \frac{\beta(\gamma, \varepsilon)}{\xi_k} \right) > 0.
\]

Thus, Eq. (17) shows that
\[
\sum_{k=0}^{+\infty} c_k(\theta) \geq \frac{\text{meas}(S_\gamma - S_\theta)}{\xi_k^n} \sum_{k=0}^{+\infty} h_k \left( \frac{\beta(\gamma, \varepsilon)}{\xi_k} \right) = +\infty.
\]

Using Lemmas 1 and 2 we have \( U = \alpha^* \) almost surely.

**Theorem 3** Let \( Z_k \) define by (14), and let
\[
\xi_k = \sqrt{\frac{\hat{a}}{\log(k + \hat{d})}},
\]
where \( \hat{a} > 0, \hat{d} > 0 \) and \( k \) is the iteration number. If \( x^0 \in S \) then, for \( \hat{a} \) large enough, \( U = \alpha^* \) almost surely.

**Proof.** We have
\[
\phi_k(Z) = \frac{1}{(\sqrt{2\pi})^n} \exp(-\frac{1}{2} \|Z\|^2) = h_k(\|Z\|) > 0,
\]
so,
\[
h_k \left( \frac{\beta(\gamma, \varepsilon)}{\xi_k} \right) = \frac{1}{(\sqrt{2\pi})^n(k + \hat{d})^{\beta(\gamma, \varepsilon)/2\hat{a}}}.
\]

For \( \hat{a} \) such that
\[
0 < \frac{\beta(\gamma, \varepsilon)^2}{2\hat{a}} < 1,
\]
we have
\[
\sum_{k=0}^{+\infty} h_k \left( \frac{\beta(\gamma, \varepsilon)}{\xi_k} \right) = +\infty,
\]
and, from the preceding Theorem 3, we have \( U = \alpha^* \) almost surely.

### 4.2 Practical Implementation of Algorithm SPCGB

The above results suggest the following numerical algorithm:

1. An initial guess \( X^0 \in S \) is given.
2. At the iteration number \( k \geq 0 \), \( X^k \) is known and \( X^{k+1} \) is determined by performing the following three substeps:
   1. We determine the direction \( d_k \) and the step \( \eta_k \) using ascent method (9). This generates the first trial point:
      \[
      T^0_{k+1} = Q_k(X^k).
      \]
   2. We determine a sample \( \left( P^1_k, \ldots, P^{k_{sto}}_k \right) \) of \( k_{sto} \) new trial points:
      \[
      T^i_{k+1} = T^0_{k+1} + P^i_k, \quad i = 1, \ldots, k_{sto}.
      \]
We determine $X^{k+1}$ by selecting it from the set of available points:

$$A_k = \left\{ X^k, T^0_{k+1}, \ldots, T_{k+1}^{k_{sto}} \right\}.$$ 

As was shown in Theorem 3, substep (2.2) may use $P_k = \xi_k Z_k$, where $Z = (Z_1, \ldots, Z_{k+1})$ is a sample of $N(0,1)$ and $\xi_k$ is given by the equation (18). The computation of $X^{k+1}$ is performed by

$$X^{k+1} = \arg \min_{X \in A_k} f(X).$$

5 Numerical Experiments

In this section, we describe practical implementation of stochastic perturbation and we present the results of some numerical experiments which illustrate the numerical behavior of the method.

In order to apply the method, we start with the initial value $X^0 = x^0 \in S$. At step $k \geq 0$, $X^k$ is known and $X^{k+1}$ is determined. We generate $k_{sto}$ the number of perturbation, the case $k_{sto} = 0$ corresponds to the unperturbed conditional gradient with bisection method. In our experiments, the Gaussian variates are obtained from calls to standard generators. We use

$$\xi_k = \sqrt{\frac{\hat{a}}{\log(k+2)}}, \quad \text{where } \hat{a} > 0.$$ 

The methods in the tables have the following meanings:

(i) “SQP” stands for sequential quadratic programming [5].

(ii) “IP” stands for interior-point algorithm [16].

(iii) “CGB” stands for the method of conditional gradient and bisection.

(iv) “SPCGB” stands for the method of stochastic perturbation of conditional gradient and bisection.

The code of the proposed algorithm SPCGB is written by using Matlab programming language. We test SPCGB method and compare it with interior-point algorithm [16] and Sequential quadratic programming [5], using Matlab fmincon function on low and large dimensional problems. This algorithms has been tested on some problems from [1, 11, 12, 22, 24, 26, 27], where linear constraints are present with given initial feasible points $x^0$. The results are listed in Table 1 and Table 2, where $n$ stands for the dimension of tested problem and $n_c$ stands for the number of constraints. We will report the following results: the optimal value $f^*$ and the number of iteration $k_{iter}$.

The optimal line search process of CGB and SPCGB find by using bisection method, we set $\epsilon = 10^{-4}$. We stop the iteration if we find a best solution (global solution) or maximum iteration is satisfied. All algorithms were carried out in a TOSHIBA Intel(R) Core(TM) processor 2.40 GHz and 6G RAM, Core i7, running under windows 7 professional 64 bit operating system. The row cpu gives the mean CPU time in seconds for one run. We give in each small and medium scale problem the initial value $x^0$, the optimal solution $x^*$ of problem (1) the number of stochastic perturbation $k_{sto}$ and minimum value $f^*_{SPCGB}$.

Problem 1 ([1])

$$\begin{align*}
\text{minimize:} & \quad x_1^2 + 2x_2^2 - 0.3 \cos(3\pi x_1) \cos(4\pi x_2) + 0.3, \\
\text{subject to:} & \quad -50 \leq x_i \leq 50, \quad i = 1, 2,
\end{align*}$$

$k_{sto} = 10$ is used and the initial point $x^0 = (20, 10)^T$. This optimal solution $x^* = (-0.00049, 0.00016)^T$ and $f^*_{SPCGB} = 4.2077e-06$ is given by the Matlab code of our approach.
Problem 2 ([1])
\[
\begin{align*}
\text{minimize:} & \quad 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4, \\
\text{subject to:} & \quad -5 \leq x_i \leq 5, \quad i = 1, 2.
\end{align*}
\]

\(k_{sto} = 5\) is used and the initial point \(x^0 = (1, 1)^T\). This optimal solution \(x^* = (-0.09021, 0.71247)^T\) and \(f_{SPCGB}^* = -1.0316\) is given by the Matlab code of our approach.

Problem 3 ([1])
\[
\begin{align*}
\text{minimize:} & \quad 10^5x_1^2 + x_2^2 - (x_1^2 + x_2^2)^2 + 10^{-5}(x_1^2 + x_2^2)^4, \\
\text{subject to:} & \quad -20 \leq x_i \leq 20, \quad i = 1, 2.
\end{align*}
\]

\(k_{sto} = 2\) is used and the initial point \(x^0 = (5, 5)^T\). This optimal solution \(x^* = (-0.00015, 14.9119)^T\) and \(f_{SPCGB}^* = -24774.56\) is given by the Matlab code of our approach.

Problem 4 ([14])
\[
\begin{align*}
\text{minimize:} & \quad (x_1 - 1)^2(x_1 - 2)^2 + (x_2 - 1)^2(x_2 - 2)^2 + x_1 + 3x_2 + x_3 - 3, \\
\text{subject to:} & \quad -1 \leq x_i \leq 3, \quad i = 1, 2, \\
& \quad -1 \leq x_3 \leq 1.
\end{align*}
\]

\(k_{sto} = 5\) is used and the initial point \(x^0 = (0, 0, 0)^T\). This optimal solution \(x^* = (0.73927, 0.499992, -0.9989)^T\) and \(f_{SPCGB}^* = -1.0891\) is given by the Matlab code of our approach.

Problem 5 ([14])
\[
\begin{align*}
\text{minimize:} & \quad x_1^2(3x_1^2 - 4x_1 - 12) + 3x_2^2(3x_2^2 - 8x_2 - 18), \\
\text{subject to:} & \quad -2 \leq x_i \leq 3, \quad i = 1, 2.
\end{align*}
\]

\(k_{sto} = 100\) is used and the initial point \(x^0 = (0, 0)^T\). This optimal solution \(x^* = (1.9984, 2.9669)^T\) and \(f_{SPCGB}^* = -436.7659\) is given by the Matlab code of our approach.

Problem 6 ([1])
\[
\begin{align*}
\text{minimize:} & \quad -\cos(x_1)\cos(x_2)\exp(-(x_1 - \pi)^2 - (x_2 - \pi)^2), \\
\text{subject to:} & \quad -10 \leq x_i \leq 10, \quad i = 1, 2.
\end{align*}
\]

\(k_{sto} = 2\) is used and the initial point \(x^0 = (2, 1)^T\). This optimal solution \(x^* = (3.1282, 3.1551)^T\) and \(f_{SPCGB}^* = -0.9995\) is given by the Matlab code of our approach.

Problem 7 ([25])
\[
\begin{align*}
\text{minimize:} & \quad x_1^2(3x_1^2 - 4x_1 - 12) + 2x_2^2(3x_2^2 - 4x_2 - 12) + 3x_3^2(3x_3^2 - 4x_3 - 12), \\
\text{subject to:} & \quad 2x_1 + 2x_3 \leq 7, \\
& \quad -2 \leq x_i \leq 3, \quad i = 1, 2.
\end{align*}
\]

\(k_{sto} = 100\) is used and the initial point \(x^0 = (0, 0, 0)^T\). This optimal solution \(x^* = (1.5458, 2.004, 1.8875)^T\) and \(f_{SPCGB}^* = -185.0364\) is given by the Matlab code of our approach.

Problem 8 ([25])
\[
\begin{align*}
\text{minimize:} & \quad \sin(x_1) - \cos(x_2) - \frac{(x_1 + x_2)}{2}, \\
\text{subject to:} & \quad 0 \leq x_1 \leq 15, \\
& \quad 0 \leq x_2 \leq 20.
\end{align*}
\]

\(k_{sto} = 5\) is used and the initial point \(x^0 = (0, 0)^T\). This optimal solution \(x^* = (14.9996, 19.3742)^T\) and \(f_{SPCGB}^* = -17.4018\) is given by the Matlab code of our approach.
Problem 9 ([1])

\[
\begin{align*}
\text{minimize:} & \quad (\exp(x_1) - x_2)^2 + (x_2 - x_3)^6 + (\tan(x_3 - x_4))^4 + x_1^8, \\
\text{subject to:} & \quad -1 \leq x_i \leq 1, \quad i = 1, \ldots, 4,
\end{align*}
\]

\( k_{sto} = 40 \) is used and the initial point \( x^0 = (0.5, 0.9, 0.9, 0.9)^T \). This optimal solution

\[ x^* = (-0.15228, 0.84332, 0.81445, 0.81062)^T \]

and \( f_{SPCGB} = 4.0389 \times 10^{-7} \) is given by the Matlab code of our approach.

Problem 10 ([1])

\[
\begin{align*}
\text{minimize:} & \quad 100(x_2 - x_1^2)^2 + (1 - x_1)^2 + 90(x_4 - x_3^2)^2 + (1 - x_3)^2 + 10.1, \\
& \quad ((x_2 - 1)^2 + (x_4 - 1)^2) + 19.8(x_2 - 1)(x_4 - 1), \\
\text{subject to:} & \quad -10 \leq x_i \leq 10, \quad i = 1, \ldots, 4,
\end{align*}
\]

\( k_{sto} = 50 \) is used and the initial point \( x^0 = (0.9, 1, 1.2, 1.2)^T \). This optimal solution

\[ x^* = (0.92196, 0.8492, 1.0722, 1.151)^T \]

and \( f_{SPCGB} = 0.0206 \) is given by the Matlab code of our approach.

Problem 11 ([22])

\[
\begin{align*}
\text{minimize:} & \quad -x_1 + x_1 x_2 - x_2, \\
\text{subject to:} & \quad -6x_1 + 8x_2 \leq 3, \\
& \quad 3x_1 - x_2 \leq 3, \\
& \quad 0 \leq x_1, x_2 \leq 5,
\end{align*}
\]

\( k_{sto} = 15 \) is used and the initial point \( x^0 = (0, 0)^T \). This optimal solution \( x^* = (1.1667, 0.50005)^T \) and \( f_{SPCGB} = -1.0833 \) is given by the Matlab code of our approach.

Problem 12 ([24])

\[
\begin{align*}
\text{minimize:} & \quad -2x_1 - 6x_2 + x_1^3 + 8x_2^2, \\
\text{subject to:} & \quad x_1 + 6x_2 \leq 6, \\
& \quad 5x_1 + 4x_2 \leq 10, \\
& \quad 0 \leq x_1 \leq 2, \\
& \quad 0 \leq x_2 \leq 1,
\end{align*}
\]

\( k_{sto} = 2 \) is used and the initial point \( x^0 = (0, 1)^T \). This optimal solution \( x^* = (0.81618, 0.37523)^T \) and \( f_{SPCGB} = -2.2137 \) is given by the Matlab code of our approach.

Problem 13 ([27])

\[
\begin{align*}
\text{minimize:} & \quad x_1^2 - 10x_1 x_2 + 7x_1 + 7x_2 - 9, \\
\text{subject to:} & \quad -2x_1 + 3x_2 \leq 6, \\
& \quad 4x_1 - 5x_2 \leq 8, \\
& \quad 5x_1 + 3x_2 \leq 15, \\
& \quad -4x_1 - 3x_2 \leq -12, \\
& \quad x_1, x_2 \geq 0,
\end{align*}
\]

\( k_{sto} = 10 \) is used and the initial point \( x^0 = (1, 3)^T \). This optimal solution \( x^* = (1.547, 2.4188)^T \) and \( f_{SPCGB} = -16.27 \) is given by the Matlab code of our approach.
Problem 14 ([26])

\[
\begin{align*}
\text{minimize:} & \quad 2x_1 - 2x_1^2 + 2x_1x_2 + 3x_2 - 2x_2^2 \\
\text{subject to:} & \quad -x_1 + x_2 \leq 1, \\
& \quad x_1 - x_2 \leq 1, \\
& \quad -x_1 + 2x_2 \leq 3, \\
& \quad 2x_1 - x_2 \leq 3, \\
& \quad x_1, x_2 \geq 0.
\end{align*}
\]

\(k_{sto} = 30\) is used and the initial point \(x^0 = (0.5, 0.5)^T\). This optimal solution \(x^* = (3, 3)^T\) and \(f^*_\text{SPCGB} = -3\) is given by the Matlab code of our approach.

Problem 15 ([14])

\[
\begin{align*}
\text{minimize:} & \quad (x_1 - 1)^2 + (x_2 - x_3)^2 + (x_4 - x_5) \\
\text{subject to:} & \quad x_1 + x_2 + x_3 + x_4 + x_5 = 5, \\
& \quad x_3 - 2(x_4 + x_5) = -3,
\end{align*}
\]

\(k_{sto} = 30\) is used and the initial point \(x^0 = (2/3, 0, 3/2, 0)^T\). This optimal solution \(x^* = (1, 0.73399, 0.73398, 1.266, 1.266)^T\) and \(f^*_\text{SPCGB} = 2.0694e-10\) is given by the Matlab code of our approach.

Problem 16 ([14])

\[
\begin{align*}
\text{minimize:} & \quad -32.174(255 \ln((x_1 + x_2 + x_3 + 0.03)/(0.09x_1 + x_2 + x_3 + 0.03)) \\
& \quad + 280 \ln((x_2 + x_3 + 0.03)/(0.07x_2 + x_3 + 0.03)) \\
& \quad + 290 \ln((x_3 + 0.03)/(0.13x_3 + 0.03))), \\
\text{subject to:} & \quad x_1 + x_2 + x_3 = 1, \\
& \quad 0 \leq x_i \leq 1, \quad i = 1, 2, 3,
\end{align*}
\]

\(k_{sto} = 20\) is used and the initial point \(x^0 = (1, 0, 0)^T\). This optimal solution \(x^* = (0.61781, 0.3282, 0.053988)^T\) and \(f^*_\text{SPCGB} = -26250.46\) is given by the Matlab code of our approach.

Problem 17 ([14])

\[
\begin{align*}
\text{minimize:} & \quad - \sum_{i=1}^{235} \ln \left( (a_i(x) + b_i(x) + c_i(x))/\sqrt{2\pi} \right), \\
\text{subject to:} & \quad 1 - x_1 - x_2 \geq 0, \\
& \quad 0.001 \leq x_i \leq 0.499, \quad i = 1, 2, \\
& \quad 100 \leq x_3 \leq 180, \\
& \quad 130 \leq x_4 \leq 210, \\
& \quad 170 \leq x_5 \leq 240, \\
& \quad 5 \leq x_i \leq 25, \quad i = 6, \ldots, 8,
\end{align*}
\]

where

\[
\begin{align*}
a_i(x) &= \frac{x_1}{x_6} \exp\left(-y_i - x_3^2/(2x_6^2)\right), \\
b_i(x) &= \frac{x_2}{x_7} \exp\left(-y_i - x_4^2/(2x_7^2)\right), \\
c_i(x) &= \frac{1 - x_2 - x_1}{x_8} \exp\left(-y_i - x_5^2/(2x_8^2)\right).
\end{align*}
\]
Problem 18 ([15])

\[
\begin{align*}
\text{minimize: } & - (x_1 + 0.5x_2 + 0.667x_3 + 0.75x_4 + 0.8x_5)^{1.5}, \\
\text{subject to: } & Ax \leq b, \\
& x \geq 0,
\end{align*}
\]

where:

\[
A = \begin{pmatrix}
0.795137 & 0.225733 & 0.371307 & 0.225064 & 0.878756 \\
-0.905037 & -0.638848 & -0.134430 & -0.921211 & 0.150370 \\
0.905037 & 0.248231 & 0.278197 & 0.376265 & -0.597468 \\
0.762043 & -0.304755 & -0.012345 & -0.394012 & -0.792129 \\
0.564347 & 0.746523 & -0.822105 & -0.892331 & -0.922916 \\
-0.954276 & -0.196016 & 0.242000 & 0.797813 & -0.147119 \\
0.747682 & 0.912055 & -0.529338 & 0.243496 & 0.279402 \\
-0.109599 & 0.727219 & -0.741781 & -0.058455 & 0.749470 \\
0.209106 & -0.074202 & -0.022484 & -0.144214 & -0.735169
\end{pmatrix}
\]

and

\[
b = \begin{pmatrix}
4.242372 \\
-1.785220 \\
3.213560 \\
1.205676 \\
-0.891062 \\
-0.066698 \\
2.286079 \\
0.521564 \\
-0.730516
\end{pmatrix}
\]

\(k_{sto} = 50\) is used and the initial point \(x^0 = (0.1, 0.2, 1.80, 160, 210, 11.21, 3.21, 5.8)^T\). This optimal solution

\(x^* = (0.5009916, 0.5009925, 137.247, 187.1867, 174.5884, 16.48846, 24.89633, 10.55855)^T\)

and \(f_{SPCGB}^* = 1149.78\) is given by the Matlab code of our approach.

\(k_{sto} = 1\) is used and the initial point \(x^0 = (2.9, 0, 0.8, 0.2, 1.7)^T\). This optimal solution

\(x^* = (0.40964, 5.6011, 6.1354, 7.7007e - 12, 0.4258)^T\)

and \(f_{SPCGB}^* = -21.1304\) is given by the Matlab code of our approach.
Problem 19 ([19])

\[
\begin{aligned}
\text{minimize: } & \frac{\pi}{n} \left( k_1 \sin^2(\pi y_1) + \sum_{i=1}^{n-1} (y_i - k_2)^2 (1 + k_1 \sin^2(\pi y_{i+1})) \right) + (y_n - k_2)^2, \\
\text{subject to: } & 3x_1 + x_2 + 2x_5 + x_7 - x_9 + 6x_{10} \leq 120, \\
& 2x_1 + 4x_2 + 7x_4 + 3x_5 + x_8 \leq 57, \\
& x_5 + 2x_8 - x_{10} \leq 10, \\
& x_3 + x_8 + 2x_{10} \leq 42, \\
& x_4 + x_9 + x_{10} \leq 23, \\
& 0 \leq x_i \leq 6 \quad i = 1, 2, 5, \quad 0 \leq x_i \leq 8 \quad i = 3, 4, 8, 9, 10, \quad 0 \leq x_i \leq 10 \quad i = 6, 7,
\end{aligned}
\]

where \( y_i = 1 + 0.25(x_i - 1), i = 1, 2, \ldots, 10 \). \( k_{sto} = 5 \) is used and the initial point \( x^0 = (1, 1, 1, 1, 1, 1, 1, 1, 0.5, 0.5)^T \). This optimal solution

\[
x^* = (1.0001, 0.98776, 0.88898, 0.98776, 0.98776, 0.98776, 0.98776, 1.0865)^T
\]

and \( f^*_SPCGB = 4.1245e - 04 \) is given by the Matlab code of our approach.

Problem 20 ([6])

\[
\begin{aligned}
\text{minimize: } & x_1 - x_2 - x_3 - x_1x_3 + x_1x_4 + x_2x_3 - 2x_4, \\
\text{subject to: } & x_1 + 2x_2 \leq 8, \\
& 4x_1 + x_2 \leq 12, \\
& 3x_1 + 4x_2 \leq 12, \\
& 2x_3 + x_4 \leq 8, \\
& x_3 + 2x_4 \leq 8, \\
& x_3 + x_4 \leq 5, \\
& 0 \leq x_i, \quad i = 1, \ldots, 10,
\end{aligned}
\]

\( k_{sto} = 5 \) is used and the initial point \( x^0 = (0, 0, 0, 0)^T \). This optimal solution \( x^* = (3, -7.2287e - 12, 4, -3.3552e - 09)^T \) and \( f^*_SPCGB = -13 \) is given by the Matlab code of our approach.

Problem 21 ([6])

\[
\begin{aligned}
\text{minimize: } & -\sum_{i=1}^{10} (x_i^2 + 0.5x_i), \\
\text{subject to: } & 2x_1 - x_6 + x_7 \leq 3, \\
& x_3 - x_5 + x_7 \leq 1.5, \\
& 3x_4 - 2x_9 + x_{10} \leq 2.2, \\
& x_5 + 2x_6 - x_9 \leq 2.7, \\
& x_2 + x_9 - x_{10} \leq 2.3, \\
& x_3 + 2x_8 - x_{10} \leq 3, \\
& 0 \leq x_i \leq 1, \quad i = 1, 2, \ldots, 10,
\end{aligned}
\]

\( k_{sto} = 1 \) is used and the initial point \( x^0 = (0, 0, 0, 0, 0, 0, 0, 0, 0, 0)^T \). This optimal solution \( x^* = (1, 1, 1, 1, 1, 1, 1, 1, 1, 1)^T \) and \( f^*_SPCGB = -15 \) is given by the Matlab code of our approach.
Problem 22 ([12])

\[
\begin{align*}
\text{minimize:} & \quad \sum_{i=1}^{m} \sum_{j=1}^{n} (c_{ij}x_{ij} + d_{ij}x_{ij}^2), \\
\text{subject to:} & \quad \sum_{i=1}^{m} x_{ij} = b_j, \quad j = 1, \ldots, n, \\
& \quad \sum_{j=1}^{n} x_{ij} = a_i, \quad i = 1, \ldots, m, \\
& \quad 0 \leq x_{ij},
\end{align*}
\]

where

\[d_{ij} \leq 0, \quad \sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j.\]

This problem features \(n + m\) equality constraints and \(nm\) variables. There is exactly one redundant equality constraint.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>#</td>
<td>n</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>9</td>
<td>4</td>
</tr>
<tr>
<td>10</td>
<td>4</td>
</tr>
<tr>
<td>11</td>
<td>2</td>
</tr>
<tr>
<td>12</td>
<td>2</td>
</tr>
<tr>
<td>13</td>
<td>2</td>
</tr>
<tr>
<td>14</td>
<td>2</td>
</tr>
<tr>
<td>15</td>
<td>5</td>
</tr>
<tr>
<td>16</td>
<td>3</td>
</tr>
<tr>
<td>17</td>
<td>8</td>
</tr>
<tr>
<td>19</td>
<td>10</td>
</tr>
<tr>
<td>20</td>
<td>4</td>
</tr>
<tr>
<td>22</td>
<td>24</td>
</tr>
</tbody>
</table>

Table 1: Comparing optimal values of SQP, IP, CGB and SPCGB algorithms.

\[n = 4, \quad m = 6, \]
\[a = (8, 24, 20, 24, 16, 12)^T, \]
\[b = (29, 41, 13, 21)^T, \]
\[c = \begin{pmatrix} 300 & 740 & 300 & 430 & 210 & 360 \\ 270 & 600 & 490 & 250 & 830 & 290 \\ 460 & 540 & 380 & 390 & 470 & 400 \\ 800 & 380 & 760 & 600 & 680 & 310 \end{pmatrix} \]
\[\text{and} \quad d = \begin{pmatrix} -7 & -12 & -13 & -7 \\ -4 & -9 & -12 & -9 \\ -6 & -14 & -8 & -16 \\ -8 & -7 & -8 & -4 \end{pmatrix}. \]
We remark that the rank of the matrix of constraints is less than the number of there rows in this problem, so we need to add the intelligent variables. $k_{sto} = 5$ is used and the initial point $x^0 = (2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)^T$.

This optimal solution

$$x^* = (5.9998, 2.0002, 0, 0, 0, 2.9998, 0, 21, 20, 0, 0, 0, 0, 0, 24, 0, 0, 3.0002, 0, 0, 12.9998, 0, 0, 12, 0, 0)^T$$

and $f_{SPCGB}^{*} = 15639$ is given by the Matlab code of our approach. From Table 1 below, we see that our algorithm SPCGB can find a global solution, and the computation results illustrate that our algorithm SPCGB executes well for those problems. In contrast to the numerical results of CGB algorithm, IP and SQP algorithms failed to find a global solution of problems 14, 20, 22 and we show that

$$f_{SPCGB}^{*} < f_{CGB}^{*}$$

except in problems 2, 18, 20 and 21 we have $f_{SPCGB}^{*} = f_{CGB}^{*}$.

In Table 2 below, we remark that our algorithm SPCGB (and also CGB) can find its solutions with a large number of iterations for problems 3 and 4 and the execution takes more time. This is mainly due to the fact that one of the disadvantages of conditional gradient method is not very fast.

### 5 Conclusion

In this work, we have proposed an implementation of stochastic perturbation of conditional gradient and bisection (SPCGB) method for optimizing a non-convex differentiable function subject to linear constraints.
In particular, at each iteration, we compute a search direction by conditional gradient, and optimal line search by bisection algorithm along this direction yields a decrease in the objective value. In the lack of convexity assumptions, convergence to a global minimum cannot be ensured. We also introduced a stochastic modification of the method involving the incorporation of a random perturbation $P_k$, which may be interpreted as a perturbation of the direction. This approach leads to a stochastic method where the deterministic sequence generated by the conditional gradient is replaced by a sequence of random variables. A mathematical result concerning convergence to a global minimum was established for a convenient class of random perturbations. We established that perturbations such that $P_k = \xi_k Z$ belong to this class if $Z$ is a gaussian random vector ($N(0, 1)$ variate), and $\{\xi_k\}_{k \geq 0}$ is a decreasing sequence of strictly positive real numbers converging to zero and such that $\xi_0 \leq 1$. This provides a simple method for generation of convenient perturbations.

We proposed an algorithm for the implementation of the method and presented the results of some numerical experiments. The implementation and test of SPCGB algorithm proposed show that this approach is effective to calculate for non-convex optimization problems with linear constraints. The main difficulty in the practical use of the stochastic perturbation is connected to the tuning of the parameters $\hat{a}$ and $k_{sto}$.

The SPCGB algorithm can solve many problems such as the optimal control problems, optimization for machine learning and image regularization via penalty method. Also, we can generalized SPCGB implementation for solving large-scale optimization problems and non-smooth optimization problems.

**Acknowledgment.** The Authors would like to thank the referees for their fruitful suggestions, which helped us to improve the quality of the paper.

**References**


