Implementation of reduced gradient with bisection algorithms for non-convex optimization problem via stochastic perturbation

Article *in* Numerical Algorithms · May 2018

DOI: 10.1007/s11075-017-0366-1

1 author:

Abdelkrim El Mouatasim
University Ibn Zohr - FPO

Noname manuscript No.
(will be inserted by the editor)

Implementation of reduced gradient with bisection algorithms for non-convex optimization problem via stochastic perturbation

Abdelkrim El Mouatasim

Received: date / Accepted: date

Abstract In this paper, we proposed an implementation of stochastic perturbation of reduced gradient and bisection (SPRGB) method for optimizing a non-convex differentiable function subject to linear equality constraints and non-negativity bounds on the variables. In particular, at each iteration, we compute a search direction by reduced gradient, and optimal line search by bisection algorithm along this direction yields a decrease in the objective value. SPRGB method is desired to establish the global convergence of the algorithm. An implementation and tests of SPRGB algorithm are given, and some numerical results of large scale problems are presented, which show the efficient of this approach.

Keywords Large scale problem · Linear constraints · Non-convex optimization · Reduced gradient algorithm · Bisection algorithm · Stochastic perturbation

1 Introduction

Optimization problems with equality constraints contain the implicit difficulty arising from the need to solve a system of equations while optimizing a cost function [16].

The general non-convex optimization with linear constraints (NCOLC) problem is to minimize a non-convex function subject to bundle and linear constraints, see for instance [15]. There exist several application areas for

A. El Mouatasim
Université Ibn Zohr,
Faculté Polydisciplinaire de Ouarzazate (FPO),
Département de Mathématiques, Informatique et Gestion,
Ouarzazate 45800, Morocco.
Tel.: +212671311143
E-mail: a.elmouatasim@uiz.ac.ma

NCOLC problems like, combinatorial optimization (water distribution [12] and co-localization image and video), optimal control [13], integer programming of call center [2], machine learning [27,28] and neural networks [9].

Two equivalent formulations of this problem are useful for describing algorithms. They are

$$\begin{cases} \text{minimize} & f(y) \\ \text{subject to} & \bar{A}y \leq b \\ & \ell \leq y \leq \mathbf{u} \end{cases} \tag{1}$$

where $f : I\!R^{\bar{n}} \longrightarrow I\!R$ is twice continuously differentiable non-convex function, $\bar{A}$ is $m \times \bar{n}$ matrix, b is an m-vector, and the lower- and upper-bound vectors, ℓ and $\mathbf{u}$, may contain some infinite components;
and

$$\begin{cases} \text{minimize} & f(x) \\ \text{subject to} & Ax = b \\ & 0 \leq x \end{cases} \tag{2}$$

where A is $m \times n$ matrix.
To get equality constraints for problem (1) one might think about introducing slack variables and incorporating them into the linear equality constraints (2).

Many algorithms have proposed for solving NCOLC problems, such as the reduced gradient method see, for instance [5,4,10,21], Wolfe reduced gradient method [32] and Frank-Wolfe method see, for instance [27], proceed by solving a sequence of subproblems in which the number of variables has been implicitly reduced. These reduced problems are obtained by using the linear constraints to express certain variables, designated as 'basic', in terms of other variables.

We are mainly interested in the situation where, on one hand f is non-convex and, on the other hand, the constraints are in general not equality (1), my be with redundant constraints or inconsistent constraints (the rank of matrix A less than or equal m).

The problem (2) can be numerically approached by using reduced gradient with bisection (RGB) 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:

$$\forall k \geq 0 : x^{k+1} = Q_k(x^k). \tag{3}$$

In order to prevent from convergence to local minimum, various modifications of these basic methods have been introduced in the literature. For instance, the sequential convex programming in [20]; line search algorithm [16], Frank-Wilfe algorithm [23]. Moreover, stochastic or evolutionary search can be introduced, but these methods are usually considered as expensive and not accurate mince, the flexibility introduced by randomization implies often a large number of evaluations of the objective function and because, the pure random search generates many points which do not improve the value of the objective function. This remark has led some authors to introduce a controlled random search see, for instance [7,8,11,29,33].

Since we have the equality linear constraints, then we can't add a random perturbation to the vector $\{x^k\}$ directly. However the reduced gradient methods transform the optimization problem 2 to the optimization problem 10 with non-basic variables $\{x_N^k\}$ that we can perturbed.

In this paper, the sequence $\{x_N^k\}_{k\geq 0}$ is replaced by a random vectors sequence $\{\mathbf{X}_N^k\}_{k\geq 0}$ and the iterations are modified as follows:

$$\forall k \geq 0 : \mathbf{X}_N^{k+1} = Q_k(\mathbf{X}_N^k) + \mathcal{P}_N^k \tag{4}$$

where $\mathcal{P}_N^k$ is a suitable random variable, called the stochastic perturbation of non-basic variables. The sequence $\{\mathcal{P}_N^k\}_{k\geq 0}$ goes to zero slowly enough in order to prevent convergence to a local minimum. There are many choices for the random variables $\mathcal{P}_N^k$. In this paper we consider in particular that the are governed by a Gaussian law.

The main techniques that have been proposed for solving problem (2) in this work is SPRGB algorithm, that is extension of the work [13] for the cases in which the matrix A has redundant constraints or inconsistent constraints, by using two phase method, bisection algorithm and projected a random perturbation of non-basic variables.

In this paper we used a bisection algorithm without call a gradient of objective function for solving unconstraints optimization with one variables that find optimal step (linear search) η_k of reduced gradient algorithm, since we know a maximum step η_{max} in each iteration.

The organization of paper is as follows: Principe of the reduced gradient with bisection method is recalled in section 3, the stochastic perturbation of RGB method are given in section 4, the notations are introduced in section 2. The results of some numerical experiments of linear constraints non-convex optimization test of medium and large scale problems are given in section 5.

2 Notations and Assumptions

We denote by $I\!R$ the set of the real numbers and $I\!R^+$ the set of positive real numbers $[0, +\infty)$, $E = I\!R^n$, the n-dimensional real Euclidean space. For $x = (x_1, x_2, ..., x_n)^t \in E$, x^T denotes the transpose of x. We denote by $\|x\| = \sqrt{x^T x} = \sqrt{(x_1^2 + ... + x_n^2)}$ the Euclidean norm of x.

Let $C = \{x \in E \mid \ Ax = b, \ \ x \geq 0\}$, the set of feasible points of problem (2). The objective function is $f : E \longrightarrow I\!R$ and its lower bound on C is denoted

by l^* i.e. $l^* = \min_C f$.
Let us introduce

$$C_\alpha = S_\alpha \cap C \text{ where } S_\alpha = \{x \in E \mid \ f(x) \le \alpha\}.$$

We assume that

$$f \quad \text{is twice continuously differentiable on} \quad E, \tag{5}$$

$$\forall \alpha > l^* : \quad C_\alpha \text{ is not empty, closed and bounded,} \tag{6}$$

$$\forall \alpha > l^\star : \quad \text{meas}(C_\alpha) > 0, \tag{7}$$

where meas(C_α) is the measure of C_α.
Since E is a finite dimensional space, the assumption (6) is verified when C is bounded or f is coercive, i.e., $\lim\limits_{\|x\| \to +\infty} f(x) = +\infty$.
Assumption (6) is verified when C contains a sequence of neighborhoods of a point of optimum $x^\star$ having strictly positive measure, i.e., when $x^\star$ can be approximated by a sequence of points of the interior of C.
We observe that the assumptions (5)-(6) yield that

$$C = \bigcup_{\alpha > l^\star} C_\alpha \quad \text{i.e.,} \ \ \forall x \in C, \ \ \exists \alpha > l^* \quad \text{such that} \ \ x \in C_\alpha.$$

From (5)-(6), one has

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

Consequently, one deduces

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

where d is the direction of reduced gradient method.
Thus,

$$\beta(\alpha, \varepsilon) = \sup\{\|y - (x + \eta d)\| : (x, y) \in C_\alpha \times C_\alpha, 0 \le \eta \le \varepsilon\} < +\infty \tag{8}$$

where ε, η are positive real numbers.

3 Reduced Gradient Method

3.1 Reduced problem

From now on, we consider a non-convex programming problem with linear equality constraints (2). Let a feasible solution $x \in C$, we assume that a basis $m \times m$ invertible submatrix B of A and exists, where the components of $x_B > 0$, which correspond to the columns of B, and a $m \times (n - m)$ submatrix N such $A = [B, N]$.

We note I_B and I_N the set of index basic variables and the set of index non-basic variables.
The reduced gradient method begins with a basis B and a feasible solution $x = [x_B, x_N]$, $x_B > 0$ are called the basic variables and those of $x_N \geq 0$ the non-basic variables.
Furthermore the gradient of f may be conformally partitioned as

$$\nabla f(x)^T = \left[\nabla_B f(x)^T, \nabla_N f(x)^T\right] \tag{9}$$

where $\nabla_B f$ and $\nabla_N f$ are the basic gradient and non-basic gradient of f respectively.

The reduced gradient method begins with a basis B and a feasible solution $x^k = (x_B^k, x_N^k)$ such that $x_B^k > 0$. The solution x is not necessarily a basic solution, i.e. x_N do not has to be identically zero. Such a solution can be obtained e.g. by the usual first phase procedure of linear optimization. Using the basis B form $Bx_B + Nx_N = b$, we have

$$x_B = B^{-1}b - B^{-1}Nx_N,$$

hence the basic variables x_B can be eliminated from the problem (2)

$$\begin{cases} \text{minimize :} \quad F(x_N) \\ \text{subject to} : 0 \leq x_N, \end{cases} \tag{10}$$

where $F(x_N) = f(B^{-1}b - B^{-1}Nx_N, x_N)$. Using the equation (9), the gradient of F which is the so-called *reduced gradient*, can be expressed as

$$r = \nabla F(x)^T = -\left(\nabla_B f(x)^T B^{-1} N\right) + \left(\nabla_N f(x)^T\right).$$

Now let us assume that the basis is non-degenerate, i.e. only the non-negativity constraints $x_N \geq 0$ might be active at the current iterate x^k.
Let the search direction be a vector $d = (d_B, d_N)$ [13], where

$$(d_N)_j = \begin{cases} -r_j \text{ if } r_j < 0 \quad \text{or } (x_N)_j > 0, \\ 0 \quad \text{otherwise} \end{cases}$$

and

$$d_B^k = -B^{-1}Nd_N^k.$$

Determine the largest value $\eta_{\max}$ that can take the step for that the new solution continued feasibility for the problem (2) [13],

$$\eta_{\max} = \min_{j=1,\ldots,n} \left\{\frac{x_j^k}{-d_j^k} : d_j^k < 0\right\}.$$

We determine the optimal step as the value η_k such that

$$f(x^k + \eta_k d^k) = \min_{0 \leq \eta \leq \eta_{\max}} \{f(x^k + \eta d^k)\}. \tag{11}$$

3.2 Algorithm of reduced gradient and bisection

3.2.1 Pivoting

Once the optimal line search has been performed, we get a next iteration point $x^{k+1} = x^k + \eta_k d^k$. If all the coordinates $x_B^{k+1} > 0$ we keep the basis, else a pivot is made to eliminate the zero variable from the basis and replace it by a positive but currently non-basic coordinate. It will always be possible because we imposed the following non-degeneracy assumption.
Assumption 1: At any feasible point x for (2), there exists a basis B such that $x_B > 0$.

For more details on the basis change, see for instance [6]. The elimination of zero basic variables occurs as follows. We replace the zero basic variables by the non-basic variables as large as possible.

3.2.2 Two phase method

We used two phase method or big M method for determine unknown feasible initial points or if

$$\mathrm{rank}(A) < m,$$

by adding intelligence variables to the equality constraints optimization problem (2):

$$\begin{cases} \text{minimize} & f(x) + M \sum_{i=1}^{m} x_{n+i} \\ \text{subject to} & Ax + Id\tilde{x} = b \\ & 0 \leq [x, \tilde{x}], \end{cases}$$

where M is positive constant and Id is identical matrix.

3.2.3 Redundant constraints and inconsistent constraints

A constraints is called redundant in a path condition if the set of solutions to stay unchanged when the constraint is removed from the constraints system.

Two obvious deficiencies of any reduced gradient method defined above are that the constraints may be inconsistent if the matrix A does not have full row rank,

$$\mathrm{rank}(A) < m,$$

or the constraints may be redundant see for example problem 9.
Several modifications to the problem structure are designed to correct these difficulties, by using two phase method, big M method or eliminate the redundant constraints in the problem (2) by simply logical comparison see for instance [30].

3.2.4 Bisection algorithm

In this paper we used the bisection algorithm for solving the unconstrained optimization problem with one variable (11) see for instance [3].
Let bis(g,a,b,ϵ) denote the recursive bisection procedure. The inputs for this procedure are: the procedure for calculation values of g, the segment $[a, b]$ and the accuracy ϵ. The outputs are the estimation x_m for the minimizer x^* and g_m for the value of the minimum of the function g over the segment [a,b].
The iteration of the recursive procedure includes the following steps.

Step 0 If $b - a \geq \epsilon$, go to step 1, otherwise stop.
Step 1 Compute

$$c = \frac{a+b}{2}, a' = \frac{a+c}{2}, b' = \frac{c+b}{2}, g(c), g(a'), g(b')$$

Step 2
If $g(a') \leq g(c) \leq g(b')$, set $b = b'$.
If $g(a') \geq g(c) \geq g(b')$, set $a = a'$.
If $g(c) \leq \min\{g(a'), g(b')\}$, set $a = a', b = b'$
Step 3 Execute bis(g,a,b,ϵ) with new inputs.

3.2.5 Algorithm of RGB

We are now ready to present the details of our RGB algorithm.

Step 0 (Initialization).
Choose a feasible point $\mathbf{x}^0 \in C$, and I_B^0, I_N^0 such that B^0 is non-singular.
Set the iteration counter $k = 0$.
Step 1 (Independent variables choice).
If $k \neq 0$ choose the sets I_B^k, I_N^k;
Step 2 (Search direction computation).
1. Let $(d_N)_j = \begin{cases} -r_j \text{ if } r_j < 0 \quad \text{or } (x_N)_j > 0, \\ 0 \quad \text{otherwise} \end{cases}$
2. If d_N is zero, stop; the current point is a solution. Otherwise, fined $d_B = -B^{-1} N d_N$.

Step 3 (Line search computation).
Solve the following line search problem

$$\min_{0 \leq \eta \leq \eta_{\max}} f(x^k + \eta d^k),$$

where

$$\eta_{\max} = \begin{cases} \min\limits_{1 \leq j \leq n} \{ \frac{x_j^k}{-d_j^k} : d_j^k < 0\} & \text{if } d^k \ngeq 0, \\ \infty & \text{if } d^k \geq 0, \end{cases}$$

let η_k be an optimal solution.
Step 4 (Next point computation). Put

$$x^{k+1} = x^k + \eta_k d^k; \tag{12}$$

Step 5 If $\eta_k < \eta_{\max}$, return to Step 2. Otherwise, declare the vanishing variable in the dependent set independent and declare a strictly positive variable in the independent set dependent. Update B and N

Step 6 (Basic variables choice). Choose I_B^{k+1}, I_N^{k+1}. Let $k = k + 1$ and go to Step 2.

3.3 Convergence analysis

Stopping criteria:
The RGB method stops when the direction vector with respect to the non-basic variables $d_N^k = 0$. The justification of this stopping criterion is presented in the next result

Theorem 1 *If $d_N^k = 0$, then the KKT optimality conditions in problem (2) are satisfied at the point x^k.* ■

Proof Recall that $d_N^k = [d_{m+1}^k, \dots, d_n^k]^T$ is defined as follows: for $j = m + 1, \dots, n$

$$d_j^k = \begin{cases} -r_j \text{ if } r_j < 0 \quad \text{or } x_j^k > 0, \\ 0 \quad \text{otherwise} \end{cases}$$

Therefore, if $d_j^k = 0$, and we have

$$r_j = 0 \text{ and } x_j^k > 0$$

or

$$r_j \geq 0 \text{ and } x_j^k = 0$$

It follows that $\nabla_N F(x_N^k) \geq 0$ and $\nabla_N F(x_N^k)^T x_N^k = 0$.

Replace $\nabla_N F(x_N^k)$ by its expression in terms of $\nabla f(x^k)$:

$$\nabla_N F(x_N^k) = \nabla_N f(x^k) - (B^{-1}N)^T \nabla_B f(x^k) \geq 0$$
$$(\nabla_N f(x^k) - (B^{-1}N)^T \nabla_B f(x^k))^T x_N^k = 0.$$

So, we have the following equations:

$$\nabla_B f(x^k) - I\nabla_B f(x^k) \geq 0 \tag{13}$$
$$\nabla_N f(x^k) - (B^{-1}N)^T \nabla_B f(x^k) \geq 0 \tag{14}$$
$$(\nabla_N f(x^k) - (B^{-1}N)^T \nabla_B f(x^k))^T x_N^k = 0 \tag{15}$$
$$(\nabla_B f(x^k) - I\nabla_B f(x^k))^T x_B^k = 0. \tag{16}$$

We written equations (13-16) as:

$$\begin{bmatrix} \nabla_B f(x^k) \\ \nabla_N f(x^k) \end{bmatrix} - \begin{bmatrix} I \\ (B^{-1}N)^T \end{bmatrix} \nabla_B f(x^k) \geq 0$$
$$(\nabla_B f(x^k) - I\nabla_B f(x^k))^T x_B^k = 0$$
$$(\nabla_N f(x^k) - (B^{-1}N)^T \nabla_B f(x^k))^T x_N^k = 0$$

Adding expressions

$$\begin{aligned} Ix_B^k &= B^{-1}b - B^{-1}Nx_N^k \\ x_B^k &\geq 0 \;\; x_N^k \geq 0 \end{aligned}$$

we find the KKT optimality conditions for the problem

$$\begin{array}{ll} \min & f(x_B, x_N) \\ \text{subject to :} & Ix_B + B^{-1}Nx_N = B^{-1}b \\ & x_B \geq 0 \;\; x_N \geq 0 \end{array}$$

(Equivalent to the problem (2)) where the vector multipliers is equal to $-\nabla_B f(x^k)$.

$$\begin{aligned} \begin{bmatrix} \nabla_B f(x^k) \\ \nabla_N f(x^k) \end{bmatrix} - \begin{bmatrix} I \\ (B^{-1}N)^T \end{bmatrix} \nabla_B f(x^k) &\geq 0 \\ (\nabla_B f(x^k) - I\nabla_B f(x^k))^T x_B^k &= 0 \\ (\nabla_N f(x^k) - (B^{-1}N)^T \nabla_B f(x^k))^T x_N^k &= 0 \\ Ix_B^k &= B^{-1}b - B^{-1}Nx_N^k \\ x_B^k &\geq 0 \;\; x_N^k \geq 0 \end{aligned}$$

Transform the conditions as follows

$$\begin{aligned} \begin{bmatrix} \nabla_B f(x^k) \\ \nabla_N f(x^k) \end{bmatrix} - \begin{bmatrix} B^T B^{-1T} \\ N^T B^{-1T} \end{bmatrix} \nabla_B f(x^k) &\geq 0 \\ (\nabla_B f(x^k) - B^T B^{-1T} \nabla_B f(x^k))^T x_B^k &= 0 \\ (\nabla_N f(x^k) - N^T B^{-1T} \nabla_B f(x^k))^T x_N^k &= 0 \\ B(Ix_B^k + B^{-1}Nx_N^k) &= BB^{-1}b \\ x_B^k &\geq 0 \;\; x_N^k \geq 0 \end{aligned}$$

Collect terms B^{-1T} and $\nabla_B f(x^k)$

$$\begin{aligned} \begin{bmatrix} \nabla_B f(x^k) \\ \nabla_N f(x^k) \end{bmatrix} - \begin{bmatrix} B^T \\ N^T \end{bmatrix} B^{-1T} \nabla_B f(x^k) &\geq 0 \\ (\nabla_B f(x^k) - B^T B^{-1T} \nabla_B f(x^k))^T x_B^k &= 0 \\ (\nabla_N f(x^k) - N^T B^{-1T} \nabla_B f(x^k))^T x_N^k &= 0 \\ Bx_B^k + Nx_N^k &= b \\ x_B^k &\geq 0 \;\; x_N^k \geq 0 \end{aligned}$$

These equations represent the KKT optimality conditions for the problem (2) at the point x^k where the vector multiplier equals $-B^{-1T}\nabla_B f(x^k)$. ■

4 Stochastic Perturbation of the Reduced Gradient with bisection (SPRGB) Method

The main difficulty remains the lack of convexity: if f is not convex, the Kuhn-Tucker points may not correspond to a global minimum. In the next, we shall improve this point by using an appropriate random perturbation.

The sequence of real numbers (non-basic variables) $\{x_N^k\}_{k\geq 0}$ that is a solution of subproblems (10) is replaced by a sequence of non-basic random variables $\{\mathbf{X}_N^k\}_{k\geq 0}$ involving a non-basic random perturbation $\mathcal{P}_N^k$ of the deterministic iteration

$$x_N^{k+1} = x_N^k + \eta_k d_N^k,$$

then we have $\mathbf{X}_N^0 = x_N^0$.

1) the non-basic random perturbation $\mathcal{P}_N^k$ for non-basic variables,

$$\forall k \geq 0 \qquad \mathbf{X}_N^{k+1} = \mathbf{X}_N^k + \eta_k d_N^k + \mathcal{P}_N^k. \tag{17}$$

2) the basic random perturbation $\mathcal{P}_B^k$for basic variables by projection, we have

$$\mathbf{X}_B^{k+1} = B^{-1}b - B^{-1}\mathbf{N}X_N^{k+1} \tag{18}$$

the equations (17) and (18) give

$$\mathbf{X}_B^{k+1} = \mathbf{X}_B^{k+1} + \eta_k d_B^k - B^{-1}N\mathcal{P}_N^k$$

then

$$\mathcal{P}_B^k = -B^{-1}N\mathcal{P}_N^k.$$

Let $\mathcal{P}^k = (\mathcal{P}_B^k, \mathcal{P}_N^k)$. Then

$$\forall k \geq 0 \qquad \mathbf{X}^{k+1} = Q_k(\mathbf{X}^k) + \mathcal{P}^k = \mathbf{X}^k + \eta_k d^k + \mathcal{P}^k. \tag{19}$$

where

$$\forall k \geq 1 : \mathcal{P}^k \text{ is independent from } (\mathbf{X}^{k-1}, ..., \mathbf{X}^0) \tag{20}$$

$$\mathbf{X} \in C \Rightarrow Q_k(\mathbf{X}) + \mathcal{P}^k \in C \tag{21}$$

Equation (19) can be viewed as perturbation of the descent direction $\mathbf{d}^k$, which is replaced by a new direction $\mathbf{D}_k = d^k + \frac{\mathcal{P}^k}{\eta_k}$ and the iterations (19) become

$$\mathbf{X}^{k+1} = \mathbf{X}^k + \eta_k \mathbf{D}_k.$$

General properties defining convenient sequences of perturbation $\{\mathcal{P}_N^k\}_{k\geq O}$ can be found in the literature: usually, sequence of Gaussian laws may be used in order to produce elements satisfying these properties.
We introduce a sequence of $n-m$ dimensional random vectors $\{\mathbf{Z}_k\}_{k\geq 0}$, we denote by ϕ_k the probability density of $\mathbf{Z}_k$. 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 η_k is

$$\eta_k \in \operatorname{argmin}\{f(\mathbf{X}^k + \eta\mathbf{D}_k) : 0 \leq \eta \leq \eta_{\max}\} \tag{22}$$

where

$$\eta_{max} = \begin{cases} \min\limits_{1\leq j\leq n}\left\{\dfrac{-\mathbf{X}_j^k}{\mathbf{D}_j^k} : \quad \mathbf{D}_j^k < 0\right\} & \text{if } \mathbf{D}^k \not\geq 0 \\ \infty & \text{if } \ \mathbf{D}^k \geq 0. \end{cases}$$

We call the function bis() of bisection algorithm for determine η_k, the solution of problem (22) .

We assume that there exists a decreasing function $\mathbf{t} \mapsto g_k(\mathbf{t})$, $g_k(\mathbf{t}) > 0$ on $I\!R^+$ such that

$$\mathbf{y} \in C_\alpha \Rightarrow \phi_k\Big(\frac{\mathbf{y} - Q_k(\mathbf{x})}{\xi_k}\Big) \geq g_k\Big(\frac{\beta(\alpha,\varepsilon)}{\xi_k}\Big) \tag{23}$$

The procedure generates a sequence $U_k = f(X^k)$. By construction this sequence is decreasing and lower bounded by l^*:

$$\forall k \geq 0 : l^* \leq U_{k+1} \leq U_k \tag{24}$$

Thus, there exists $U \geq l^*$ such that $U_k \to U \ for \ k \to +\infty$.

Lemma 1 *Let $\mathcal{P}^k = \xi_k\mathbf{Z}_k$ and $\gamma = f(\mathbf{x}^0)$. Then there exists $\nu > 0$ such that*

$$P(U_{k+1} < \theta | U_k \geq \theta) \geq \frac{\operatorname{meas}(C_\gamma - C_\theta)}{\xi_k^n} g_k\Big(\frac{\beta(\gamma,\varepsilon)}{\xi_k}\Big) > 0 \quad \forall\theta \in (l^*, l^* + \nu].\blacksquare$$

Proof See, for instance, [13]. ■

The global convergence is a consequence of the following result, which yields from the Borel-Catelli's lemma (for instance, see, [13]):

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

$$\forall k \geq 0, P(U_{k+1} < \theta | U_k \geq \theta) \geq c_k(\theta) > 0 \quad and \quad \sum_{k=0}^{+\infty} c_k(\theta) = +\infty \tag{25}$$

Then $U = l^*$ almost surely. ■

Proof See, for instance, [13,22]. ■

Theorem 2
Let $\gamma = f(x^0)$ and η_k satisfy (22). Assume that $x^0 \in C$, the sequence ξ_k is decreasing and

$$\sum_{k=0}^{+\infty} g_k\Big(\frac{\beta(\gamma,\varepsilon)}{\xi_k}\Big) = +\infty \ . \tag{26}$$

Then $U = l^*$ almost surely. ■

Proof Let

$$c_k(\theta) = \frac{\text{meas}\,(C_\gamma - C_\theta)}{\xi_k^n} g_k(\frac{\beta(\gamma,\varepsilon)}{\xi_k}) > 0. \tag{27}$$

Since the sequence $\{\ \xi_k\ \}_{k\ \geq\ 0}$ is decreasing, one has $c_k(\theta) \geq \frac{\text{meas}\,(C_\gamma - C_\theta)}{\xi_0^n} g_k(\frac{\beta(\gamma,\varepsilon)}{\xi_k}) >$

0.
Thus, Eq. (26) shows that

$$\sum_{k=0}^{+\infty} c_k(\theta) \geq \frac{\textbf{meas}\,(\ C_\gamma - C_\theta\ )}{\xi_0^n} \sum_{k=0}^{+\infty} g_k\left(\frac{\beta(\gamma,\varepsilon)}{\xi_k}\right) = +\infty.$$

Using Lemma 1 and Lemma 2 we have $U = l^*$ almost surely. ■

Theorem 3 *Let $\mathbf{Z}_k$ random variable following $N\,(\ \mathbf{0},\ \sigma\mathbf{Id})$ where $(\ \sigma > 0)$ and let*

$$\xi_k = \sqrt{\frac{\mathbf{a}}{\log{(k+d)}}} \tag{28}$$

where $\mathbf{a} > 0$, $d > 0$ and k is the iteration number.
If $x^0 \in C$ then, for $\mathbf{a}$ large enough, $U = l^*$ almost surely. ■

Proof We have

$$\phi_k\ (\ \mathbf{Z}_k\ ) = \frac{1}{\left(\sigma\sqrt{\ 2\pi\ }\right)^n} \exp\left(\ -\frac{1}{2}\left\|\ \frac{\mathbf{Z}_k}{\sigma}\ \right\|^2\right) = g_k\ (\|\ \mathbf{Z}_k\ \|) > 0\ .$$

So,

$$g_k\left(\frac{\beta(\gamma,\varepsilon)}{\xi_k}\right) = \frac{1}{\left(\sigma\sqrt{\ 2\pi\ }\right)^n (k+d)^{\beta(\gamma,\varepsilon)^2/(2\sigma^2 a)}}$$

For $\mathbf{a}$ such that

$$0 < \frac{\beta(\gamma,\varepsilon)^2}{2\sigma\mathbf{a}} < 1,$$

we have

$$\sum_{k=0}^{\infty} g_k\left(\frac{\beta(\gamma,\varepsilon)}{\xi_k}\right) = +\infty,$$

and, from the preceding theorem 2 , we have $U = l^*$ almost surely. ■

5 Numerical experiments

This section reports some numerical experiments.
The methods in the tables have the following meanings:

(i) "IP" stands for interior-point algorithm [16].
(ii) "FW" stands for the Frank-Wolfe method [27].
(iii) "RGB" stands for the method of reduced gradient and bisection.
(iv) "SPRGB" stands for the method of stochastic perturbation of reduced gradient and bisection.

The code of proposed algorithm SPRGB, RGB, FW is written by using Octave/Matlab programming language. We test SPRGB method and compare it with interior-point algorithm [16], using Matlab's **fmincon** function on low and large dimensional problems.
The optimal line search process of RGB, FW and SPRGB fined by using bisection method, we set $\epsilon = 10^{-4}$.

We stop the iteration if we fined a best solution (global solution) or maximum iteration is satisfied.
All algorithms are run on a workstation HP Intel(R) Celeron(R) M processor 1.30 GHz., 224 Mo RAM, Core i3.
The row cpu gives the mean CPU time in seconds for one run.

5.1 Approximation gradient and starting point

For given $h \in I\!R^{+} - \{0\}$ and $x = (x_1, \ldots, x_n) \in I\!R^n$. The gradient of f at point x is

$$\nabla f(x) = (\frac{\partial f(x)}{\partial x_1}, \ldots, \frac{\partial f(x)}{\partial x_n})^T.$$

We approximate the gradient by centered finite difference.

$$\frac{\partial f(x)}{\partial x_i} = \frac{-f(x_i + 2h) + 8f(x_i + h) - 8f(x_i - h) + f(x_i - 2h)}{12h}.$$

We instead found a starting point by solving the following program using MATLAB's **linprog** function:

$$\begin{cases} \text{minimize} & 0 \\ \text{subject to} & Ax = b \\ & 0 \leq x \end{cases} \tag{29}$$

5.2 Numerical comparison

This algorithms has been tested on some problems, where equality constraints are present. We test the performance of IP, FW, RGB and SPRGB methods on the following test problems with given initial feasible points x^0. The

comparing results are listed in Table 2, n_1 and n_2 stands for the dimension of tested problem (1) and problem (2) respectively; and n_c stands for the number of equality constraints of problem (2), f^*_{IP}, f^*_{FW}, f^*_{RGB} and f^*_{SPRGB} the optimal value of IP, FW, RGB and SPRGB respectively.
We give in each problem the optimal solution x^* of problem (1), the number of stochastic perturbation k_{sto}, the optimality condition $\|d_N\|$ and best known minimum value f_{best}.
We will report the following results of SPRGB algorithm in Table 3: the CPU time, the iteration number k_{iter}, the number of objective function evaluation f_{eval}.

5.2.1 Medium-scale problems

Problem 1 (HS48 [17])

$$\begin{cases} \text{minimize}: & (x_1-1)^2+(x_2-x_3)^2+(x_4-x_5)^2 \\ \text{subject to}: & x_1+x_2+x_3+x_4+x_5=5 \\ & x_3-2(x_4+x_5)=-3 \end{cases}$$

We use $k_{sto}=1$ and initial point $x^0=(2,3/2,0,3/2,0)^T$. The Matlab code of our approach furnish this optimal solution $x^*=(1,\ldots,1)^T$ with optimality condition $\|d_N\|=1.4469e-04$.

Problem 2 (HS62 [17])

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

We use $k_{sto}=10$ and initial point $x^0=(0.7,0.2,0.1)^T$. The Matlab code of our approach furnish this optimal solution $x^*=(0.5929,0.3549,0.0522)^T$ with optimality condition $\|d_N\|=1.0126$ and $f^*=f_{best}$.

Problem 3 (Maximum-likelihood estimation HS105 [17])

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

where:

$$\begin{aligned} a_i(x) &= \tfrac{x_1}{x_6}\exp(-(y_i-x_3)^2/(2x_6^2)) \\ b_i(x) &= \tfrac{x_2}{x_7}\exp(-(y_i-x_4)^2/(2x_7^2)) \qquad i=1,\ldots,235 \\ c_i(x) &= \tfrac{1-x_2-x_1}{x_8}\exp(-(y_i-x_5)^2/(2x_8^2)) \end{aligned}$$

i	y_i	i	y_i	i	y_i
1	95	102-118	150	199-201	200
2	105	119-122	155	202-204	205
3-6	110	123-142	160	205-212	210
7-10	115	143-150	165	213	215
11-25	120	151-167	170	214-219	220
26-40	125	168-175	175	220-224	230
41-55	130	176-181	180	225	235
56-68	135	182-187	185	226-232	240
69-89	140	188-194	190	233	245
90-101	145	195-198	195	234-235	250

Table 1 Data of y.

and y_i is given in table 1.

We use $k_{sto} = 120$ and initial point $x^0 = (0.1, 0.2, 180, 160, 21011.213.215.8)^T$.
The Matlab code of our approach furnish this optimal solution
$x^* = (0.3657, 0.4417, 165.2680132.2312, 216.2224, 14.5885, 12.6037, 21.2516)^T$
with optimality condition $\|d_N\| = 1.0540$.

Problem 4 (HS112 [17])

$$\begin{cases} \text{minimize : } & \sum\limits_{j=1}^{10} x_j(c_j + \ln \frac{x_j}{x_1+\cdots+x_{10}}) \\ \text{subject to : } & x_1 + 2x_2 + 2x_3 + x_6 + x_{10} = 2 \\ & x_4 + 2x_5 + x_6 + x_7 = 1 \\ & x_3 + x_7 + x_8 + 2x_9 + x_{10} = 0 \\ & 1.e - 6 \leq x_i, i = 1, \dots, 10 \end{cases}$$

where $c = (-6.089, -17.164, -43.054, -5.914, -24.721, -14.986, -24.100, -10.708, -26.662, -22.179)^T$.

We use $k_{sto} = 100$ and initial point $x^0 = (1.4, 0.1, 0.1, 0.6, 0.1, 0.1, 0.1, 0.5, 0.1, 0.1)^T$.
The Matlab code of our approach furnish this optimal solution
$x^* = (0.0644, 0.1777, 0.7320, 0.0020, 0.4833, 0.0008, 0.0306, 0.0288, 0.0466, 0.115)^T$
with optimality condition $\|d_N\| = 0.8598$.

Problem 5 ([18])

$$\begin{cases} \text{minimize:} & -(x_1 + 0.5x_2 + 0.667x_3 + 0.75x_4 + 0.8x_5)^{1.5} \\ \text{subject to:} & Ax \leq b \\ & 0 \leq x \end{cases}$$

where

$$A=\begin{pmatrix} 0.795137 & 0.225733 & 0.371307 & 0.225064 & 0.878756\\ -0.905037 & -0.638848 & -0.134430 & -0.921211 & 0.150370\\ 0.905037 & 0.248231 & 0.278197 & 0.376265 & -0.597468\\ 0.762043 & -0.304755 & -0.012345 & -0.394012 & -0.792129\\ 0.564347 & 0.746523 & -0.822105 & -0.892331 & -0.922916\\ -0.954276 & -0.196016 & 0.242000 & 0.797813 & -0.147119\\ 0.747682 & 0.912055 & -0.529338 & 0.243496 & 0.279402\\ -0.109599 & 0.727219 & -0.741781 & -0.058455 & 0.749470\\ 0.209106 & -0.074202 & -0.022484 & -0.144214 & -0.735169 \end{pmatrix} \text{ and } \quad b=\begin{pmatrix} 4.242372\\ -1.785220\\ 3.213560\\ 1.205676\\ -0.891062\\ -0.066698\\ 2.286079\\ 0.521564\\ -0.730516 \end{pmatrix} \tag{30}$$

We use $k_{sto} = 5000$ and initial point $x^0 = (2.928723, 0, 0.857287, 0.249422, 1.751548)^T$. The Matlab code of our approach furnish this optimal solution $x^* = (0.409645, 5.6011101, 6.135397, 0.000002, 0.425802)^T$ with optimality condition $\|d_N\| = 0$.

Problem 6 ([24])

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

where $y_i = 1 + 0.25(x_i - 1), \quad i = 1, 2, ..., 10.$
The constants k_1 and k_2 are respectively 10 and 1.
We use $k_{sto} = 300$ and initial point $x^0 = (0,\dots,0)^T$. The Matlab code of our approach furnish this optimal solution $x^* = (1,\dots,1)^T$ with optimality condition $\|d_N\| = 7.7541e-05$.

Problem 7 ([7])

$$\begin{cases} \text{minimize: } x_1 - x_2 - x_3 - x_1x_3 + x_1x_4 + x_2x_3 - x_2x_4\\ \text{subject to: } x_1 + 4x_2 \le 8\\ \qquad 4x_1 + x_2 \le 12\\ \qquad 3x_1 + 4x_2 \le 12\\ \qquad 2x_3 + x_4 \le 8\\ \qquad x_3 + 2x_4 \le 8\\ \qquad x_3 + x_4 \le 5\\ \qquad 0 \le x_i \end{cases}$$

We use $k_{sto} = 500$ and initial point $x^0 = (0, \dots, 0)^T$. The Matlab code of our approach furnish this optimal solution $x^* = (3, 0, 4, 0)^T$ with optimality condition $\|d_N\| = 4.9957$.

Problem 8 ([7])

$$\begin{cases} \text{minimize:} & -\sum\limits_{i=1}^{10}(x_i^2+0.5x_i) \\ \text{subject to:} & 2x_1 - x_6 + x_7 \le 3 \\ & x_3 - x_5 + x_7 \le 1.5 \\ & 3x_4 - 2x_9 + x_{10} \le 2.2 \\ & x_2 + x_9 - x_{10} \le 2.3 \\ & x_5 + 2x_6 - x_9 \le 2.7 \\ & x_3 + 2x_8 - x_{10} \le 3 \\ & -1 \le x_i \le 1, \quad i = 1, 2, \dots, 10. \end{cases}$$

We use $k_{sto} = 1$ and initial point $x^0 = (0, \dots, 0)^T$. The Matlab code of our approach furnish this optimal solution $x^* = (2, \dots, 2)^T$ with optimality condition $\|d_N\| = 0$.

Problem 9 ([14])

$$\begin{cases} \text{minimize:} & \sum\limits_{i=1}^{m}\sum\limits_{j=1}^{n}(c_{ij}x_{ij}+d_{ij}x_{ij}^2) \\ \text{subject to:} & \sum\limits_{i=1}^{m} x_{ij} = b_j, \qquad j = 1, \dots, n \\ & \sum\limits_{j=1}^{n} x_{ij} = a_i, \qquad i = 1, \dots, m \\ & 0 \le x_{ij} \end{cases}$$

where

$$d_{ij} \le 0, \quad \sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j.$$

This problem features $n + m$ equality constraints and nm variables. There is exactly one redundant equality constraint. When any one of the constraints is dropped the remaining form a linearly independent system of constraints. Hence, a basic vector for the transportation consists of $n+m-1$ basic variables.

$$n = 4, \quad m = 6$$

$$a = (8, 24, 20, 24, 16, 12)^T$$

$$b = (29, 41, 13, 21)^T$$

$$c = \begin{pmatrix} 300 & 270 & 460 & 800 \\ 740 & 600 & 540 & 380 \\ 300 & 490 & 380 & 760 \\ 430 & 250 & 390 & 600 \\ 210 & 830 & 470 & 680 \\ 360 & 290 & 400 & 310 \end{pmatrix} \text{ and } \qquad d = \begin{pmatrix} -7 & -4 & -6 & -8 \\ -12 & -9 & -14 & -7 \\ -13 & -12 & -8 & -4 \\ -7 & -9 & -16 & -8 \\ -4 & -10 & -21 & -13 \\ -17 & -9 & -8 & -4 \end{pmatrix}.$$

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.

We use $k_{sto} = 100$ and initial point $x^0 = (2, \dots, 2)^T$. The Matlab code of our approach furnish this global optimal solution
$x^* = (6, 2, 0, 0, 0, 3, 0, 21, 20, 0, 0, 0, 0, 24, 0, 0, 3, 0, 13, 0, 0, 12, 0, 0)^T$ with optimality condition $\|d_N\| = 0$ and $f^* = 15639$ is global minimum value.

Problem 10 ([26])
Here we want to minimize the side lobes in the radiation pattern from a 15-element linear antenna array subject to constraints on the element positions. Mathematically the problem given as follows:

$$\begin{cases} \text{minimize:} & \max_{1\le i\le 163} |f_i(x)| \\ \text{subject to:} & x_1 - s \ge 0 \\ & -x_1 + x_2 - s \ge 0 \\ & -x_2 + x_3 - s \ge 0 \\ & -x_3 + x_4 - s \ge 0 \\ & -x_4 + x_5 - s \ge 0 \\ & -x_5 + x_6 - s \ge 0 \\ & -x_6 + x_7 - s \ge 0 \end{cases} \tag{31}$$

where

$$f_i(x) = \frac{1}{15} + \frac{2}{15} \sum_{j=1}^{7} \cos(2\pi x_j \sin\theta_i),$$

$$\theta_i = \frac{\pi}{180}(8.5 + i0.5), \quad i = 1, \dots, 163,$$

$$x_7 = 3.5,$$

and $s = 0.375$ is the minimum allowable spacing between two neighboring elements.
We use $k_{sto} = 100$ and initial point $x^0 = (0.5, 1, 1.5, 2, 2.5, 3, 3.5)^T$. The Matlab code of our approach furnish this optimal solution $x^* = (0.375, 0.7842, 1.169, 1.6383, 2.118, 2.7588, 3.5)^T$ with optimality condition $\|d_N\| = 0.5802$.

Problem 11 ([26])
Problem (31) with $s = 0.4$ and additional equality constraints $-x_4 + x_6 = 1$.
We use $k_{sto} = 100$ and initial point $x^0 = (0.5, 1, 1.5, 2, 2.5, 3, 3.5)^T$. The Matlab code of our approach furnish this optimal solution $x^* = (0.4, 0.8198, 1.2199, 1.694, 2.094, 2.694, 3.5)^T$ with optimality condition $\|d_N\| = 0.3969$.

In table 2, we remark that our algorithm SPRGB find a global solution for all tested problems, algorithm IP fail for fined a global solution of problem 9, FW algorithm fail for solving problems 4, 6, 9-11 and we show that

$$f^*_{RGB} < f^*_{SPRGB}$$

except in problems 5 and 8, we have $f^*_{RGB} = f^*_{SPRGB}$.

Table 2 Comparing optimal values of IP, FW, RGB and SPRGB algorithms for medium-scale problems.

Problem				Algorithm			
#	n_1	n_2	n_c	f^*_{FW}	f^*_{IP}	f^*_{RGB}	f^*_{SPRGB}
1	5	5	2	4.0763e-09	1.1566e-18	1.8074e-09	1.9780e-09
2	3	6	4	-26626	-26626	-2.6375	-26626
3	8	25	15	1138.4	1138.4	1278.8	1138.4
4	10	20	13	-48.8891	-48.9110	-48.9065	-48.9110
5	5	14	9	-21.1219	-21.1219	-21.1219	-21.1219
6	10	25	15	5.7713e-10	7.7078e-14	0.3129	2.1280e-11
7	4	10	6	-13	-13	-8	-13
8	10	26	16	-15	-15	-15	-15
9	24	34	10	1.8270e+4	1.5990e+04	2.5287e+004	1.5639e+4
10	7	14	8	0.0954	0.0765	0.1336	0.0755
11	7	14	9	0.1262	0.1085	0.1720	0.1018

Table 3 Comparing results between FW, IP, RGB and SPRGB algorithms.

	Algorithm											
Pb.	FW			IP			RGB			SPRGB		
#	k_{iter}	f_{eval}	CPU	k_{iter}	funcCount	CPU	k_{iter}	f_{eval}	CPU	k_{iter}	f_{eval}	CPU
1	100	3100	1.95	9	62	1.09	47	1938	0.20	15	396	0.17
2	100	3100	1.95	12	97	1.19	150	503	0.23	5	410	0.11
3	100	3100	3.23	42	1130	1.70	10000	30000	56.48	200	24139	41.22
4	150	4650	3.22	26	573	1.14	500	10965	1.09	50	5585	2.28
5	2	62	1.47	15	245	1.08	9	10	0.14	9	10	0.12
6	249	7719	4.08	75	1976	1.29	200	9303	0.65	25	7958	2.53
7	4	124	0.73	20	231	1.06	300	1011	0.42	14	6529	0.94
8	2	62	0.76	18	515	1.11	16	2	0.17	17	18	0.25
9	3	93	0.94	35	900	2.73	64	2	0.55	175	17052	8.41
10	250	7750	10.02	130	2350	3.60	300	1007	2.76	30	2957	9.50
11	250	7750	9.62	38	698	1.92	500	1557	4.28	60	5979	14.58

5.2.2 Large-scale problems

Problem 12 ([1])
Given a positive integer n, the problem is defined as follows:

$$\begin{cases} \text{minimize} & \sum_{i=1}^{n} \cos(2\pi x_i \sin(\frac{\pi}{20})) \\ \text{subject to} & x_i - x_{i+1} = 0.4 \qquad i = 1, \dots, n-1, \end{cases}$$

Problem 13 ([9])
Consider Rastrigin's function

$$\begin{cases} \text{minimize} & \sum_{i=1}^{n} (x_i^2 - 10cos(2\pi x_i) + 10) \\ \text{subject to} & \sum_{i=1}^{n} x_i = 0 \\ & -5.12 \le x_i \le 5.12, \quad i = 1, 2, \dots, n. \end{cases}$$

Table 4 The results furnished by SPRGB algorithms for problem 12.

Problem			Algorithm			
n_1	n_2	n_c	CPU time	$k_{\max}$	k_{sto}	f^*_{SPRGB}
50	50	49	0.0625	2	1	-2.0140
100	100	99	0.1094	3	1	-3.7032
250	250	249	0.1250	3	1	-4.6087
500	500	499	0.1875	2	1	-4.0147
1000	1000	999	1.578	3	1	-4.9829
2500	2500	2499	13.25	2	1	-5.0117
5000	5000	4999	107.43	2	1	-2.0491
6000	6000	5999	165.84	2	1	-5.0372

We change a variable x by variable y such that:

$$y_i = x_i + 5.12 \qquad i = 1, \dots, n,$$

and penalized a bounded constraints.

Table 5 The results furnished by SPRGB algorithms for problem 13.

Problem			Algorithm			
n_1	n_2	n_c	CPU time	$k_{\max}$	k_{sto}	f^*_{SPRGB}
100	100	1	5.3125	46	50	2.75e-9
250	250	1	3.5156	20	75	9.72e-5
500	500	1	10.6875	55	75	3.53e-4
750	750	1	12.5156	15	300	3.00e-4
1000	1000	1	24.5313	16	500	3.10e-3
1500	1500	1	22.0781	11	550	1.14e-4
2000	2000	1	19.3438	7	700	9.80e-3

Problem 14 Consider the optimal control of a servomotor [13]

$$\begin{cases} \text{minimize} \quad J(\mathbf{x}_0, \mathbf{u}) = \int_0^{\infty} [\mathbf{x^T(t)Qx(t)} + \mathbf{ru^2(t)}]\mathbf{dt} \\ \text{subject to} \quad \frac{\partial \mathbf{x}}{\partial \mathbf{t}} = \begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \mathbf{u} \end{cases} \tag{32}$$

where $Q = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, r > 0.$

The state and control variables are parametrized by the proposed approximation method in [31].

$$\begin{cases} \text{minimize} \quad \frac{h}{2}(\sum_1^m \mathbf{y}_i^2 + r \sum_1^m \mathbf{u}_i^2) \\ \text{subject to :} \; y_1 = y_0 + \frac{h}{2} z_1 \\ \qquad y_i = y_{i-1} + \frac{h}{2}(z_i + z_{i-1}), i = 2, \dots, m \\ \qquad z_1 = z_0 + \frac{h}{2}(-z_1 + u_i) \\ \qquad z_i = z_{i-1} + \frac{h}{2}(-z_i + u_i - z_{i+1} + u_{i-1}), i = 2, \dots, m \end{cases} \tag{33}$$

where $\mathbf{x} = (\mathbf{y}, \mathbf{z})$ then the dimension of the problem $n = 3m$; the time interval $h = 2$.

We choose $r = 0.5$; we start by $\mathbf{u}_0 = 1000{*}ones(1, m)$ and $\mathbf{x}_0 = [3.25; 4.95]$.

Table 6 The results furnished by SPRGB algorithms for problem 14.

Problem			Algorithm			
n_1	n_2	n_c	CPU time	k_{iter}	k_{sto}	f^*_{SPRGB}
300	300	200	0.20	5	1	1.7801e-08
600	600	400	0.34	5	1	0.1505
900	900	600	0.73	6	1	9.2596e-13
3000	3000	2000	18.84	7	1	1.0755e-12
6000	6000	4000	134.83	7	1	0.0023
9000	9000	6000	427.53	7	1	0.4994

6 Conclusion and feature works

We have proposed a new feasible descent stochastic method, based on reduced gradient method and bisection algorithm, for linearly constrained smooth non-convex optimization. The global convergence of SPRGB algorithm is guaranteed for non-convex function by regularization of number of perturbation. The implementation and test of SPRGB algorithm proposed show that this approach is effective to calculate for non-convex optimization problems with linear constraints.

The SPRGB algorithm can be solved an integer programming problems, learning machine optimization and image regularization via penalty method. Also we can generalized SPRGB implementation for solving optimization problems with non-linear constraints and non-smooth optimization problems.

Acknowledgements We indebted to the anonymous Reviewers and Editor for many suggestions and stimulating comments to improve the original manuscript. Also the author thank Professor Pierre L'Ecuyer in Montreal University, that gave several insightful comments.

References

1. Li C. and Li D., An extension of the FletcherReeves method to linear equality constrained optimization problem. Applied Mathematics and Computation 219, 10909-10914, (2013)
2. Avramidis A.N., Chan W. and L'Ecuyer P., Staffing multi-skill call centers via search methods and a performance approximation. IIE Transactions,41(6), 483 - 497, (2009).
3. Baushev A.N. and Morozova E.Y., A multidimensional bisection method for minimizing function over simplex. Lectures notes in engineering and computer science, 2, 801–803, (2007).
4. Bazaraa M.S., Sherali H.D. and Shetty C.M., Nonlinear programming theory and application. 3rd ed. Wiley-Interscience, (2006).
5. Beck P., Lasdon L. and Engquist M., A Reduced gradient algorithm for nonlinear network problems. ACM Trans. Math. Softw. 9, 57–70, (1983).

6. Bihain A, Nguyen VH, Strodiot JJ. A reduced subgradient algorithm. Math. Program. Study. 1987;30:127149.
7. Bouhadi M., Ellaia R. and Souza de Cursi J.E., Random perturbations of the projected gradient for linearly constrained problems. Advances in convex analysis and global optimization. Honoring the memory of C. Caratheodory (1873–1950). Dordrecht, Kluwer Academic Publishers. Nonconvex Optim. Appl. 54, 487–499, (2001).
8. Carson Y. and Maria A., Simulation optimization: methods and applications. Proceeding of Winter Simulation Conference, USA, (1997).
9. Che H., Li C., He X. and Huang T., An intelligent method of swarm neural networks for equalities constrained nonconvex optimization. Neurocomputing 167, 569-577, (2015).
10. Dembo R.S., Dealing with degeneracy in reduced gradient algorithms, Mathematical Programming, 31, 375–363, (1985).
11. Dorea C.C.Y. Stopping rules for a random optimization method. SIAM Journal Control and Optimization, 28(4), 841–850, (1990).
12. El Mouatasim A., Ellaia R. and Al-Hossain A., A continuous approach to combinatorial optimization: application to water system pump operations. Optimization Letters Journal, 6(1), 177–198 (2012).
13. El Mouatasim A., Ellaia R. and Souza de Cursi J.E., Stochastic perturbation of reduced gradient & GRG methods for nonconvex programming problems. Journal of Applied Mathematics and Computation, 226, 198–211, (2014).
14. Floudas C.A. and Pardalos P.M., A collection of test problems for constrained global optimization algorithms. Lecture Notes in Computer Science 455. Springer-Verlag, Berlin, (1990).
15. Gould N.I.M., On modified factorization for large-scale linearly constrained optimization. SIAM J. OPTIM., 9(4), 10441–1063, (1999).
16. I. Griva, S. G. Nash and A. Sofer, Linear and Nonlinear Optimization. SIAM, ISBN 978-0-898716-61-0, (2009).
17. Hock W. and Schittkowski K., Test examples for nonlinear programming codes, Lecture Notes in Economics and Mathematical Systems, 187, Springer, (1981)
18. Horst T., Modification, implementation and comparison of three algorithms for globally solving linearly constrained concave minimization problems. Computing, 42, 271–289, (1989).
19. Huard P. Un algorithme général de gradient réduit [Ageneral reduced gradient algorithm]. Bulletin de la Direction des Etudes et Recherches, Electricité de France, Série C. 1982; 2:91109.
20. Kiwiel K.C., Free-steering relaxation methods for problems with strictly convex costs and linear constraints. Math. Oper. Res. 22(2), 326–349, (1997).
21. Kotel'nikov E.A., Applying a reduced gradient in quadratic programming. Numerical Analysis and Applications, 3(1), 17–24, (2010).
22. L'Ecuyer P. and Touzin R., On the Deng-Lin random number generators and related methods. Statistics and Computing, 14(1), 5–9, (2003).
23. Lacoste-Julien S., Convergence rate of Frank-Wolfe for non-convex objectives. arXiv:1607.00345v1, (2016).
24. Levy A. and Montalvo A., The tunneling algorithm for the global minimization of functions. *In Proceedings of the Conference on Numerical Analysis*, Dundee, Scotland, (1977).
25. J.M. Martinez, E.A. Pilotta, and M. Raydan, Spectral gradient methods for linearly constrained optimization, Journal of Optimization Theory and Applications, 125(3), 629–651, (2005).
26. Masden K. and Schjaer-Jacobsen H., Linearly constrained minimax optimization, Mathematical programming, 14, 208–223, (1978).
27. Nanuclef R., Frandi E., Sartori C. and Allende H., A novel FrankWolfe algorithm. Analysis and applications to large-scale SVM training. Information Sciences 285, 66-99, (2014).
28. Nikolaou N., Fast optimization of non-convex Machine Learning objectives. Master thesis, University of Edinburgh, (2012).
29. Sadegh P. and Spall J.C., Optimal random perturbations for stochastic approximation using a simultaneous perturbation gradient approximation. IEEE Trans. Autom. Control 44(1), 231–232, (1999).

30. Sumathi P. and Paulraj. S., Identification of redundant constraints in large scale linear programming problems with minimal computational effort. Appl. Math. Sci. 77(80), 3963–3974, (2013).
31. Wang F., Su C. and Liu Y., Computation of optimal feedforward and feedback control by a modified reduced gradient method. Applied Mathematics and Computation 104, 85–100, (1999).
32. Wolfe P., The reduced gradient method. Rand Document. Santa Monica (CA), (1962).
33. Ziadi R., Ellaia R. and Bencherif-Madani A., Global optimization through a stochastic perturbation of the Polack-Ribière conjugate gradient method. Journal of Computational and Applied Mathematics, 317, 672–684, (2017).