Annals of the University of Craiova, Mathematics and Computer Science Series
Volume 47(2), 2020, Pages 314–330
ISSN: 1223-6934

Stochastic perturbation of Frank-Wolfe method for nonconvex programming problems

Abdelkrim El Mouatasim and Abderrahmane Ettahiri

Abstract. In this paper, we present a random perturbation of Frank-Wolfe method (a.k.a. Conditional gradient method) for solving nonconvex differentiable programming under linear differentiable constraints. The perturbation avoids convergence to local minima. Theoretical results guarantee the convergence of the proposed method towards a global minimizer. Some numerical results of medium and large size problems are provided to show the effectiveness of our approach.

2010 Mathematics Subject Classification. Primary 90C26; Secondary 90C30.
Key words and phrases. Non-convex optimization, Nonlinear programming, Linear constraints, Large-scale problems, Frank-Wolfe algorithm, Stochastic perturbation.

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 nonconvex problems in these phenomena. There exist several application areas for non-convex optimization with linear constraints (NCOLC) problems like combinatorial optimization (water distribution [6], co-localization image and video), optimal control [7], integer programming of call center [2], machine learning [14, 15], and or learning neural networks based on parsimonious coding and Frank-Wolfe algorithm. This algorithm also known as the conditional gradient, was originally proposed by Marguerite Frank and Philip Wolfe in 1956 [10], 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 nonconvex optimization problems with linear equality or inequality constraints of the form

$$\begin{cases} \text{minimize} & f(x) \\ \text{subject to} & Ax \leq b \\ & \ell \leq x \leq \eta \end{cases} \tag{1}$$

where $f:\ \mathbb{R}^n \to \mathbb{R}$ is a twice continuously differentiable function, A is $m \times n$ matrix with rank m, b is an m-vector, and the lower and upper bound vectors, ℓ and η, may

Received August 26, 2019. Accepted March 15, 2020.

contain some infinite components;
and

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

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 Frank-Wolfe. The stochastic algorithms like the genetic algorithm and the simulated annealing algorithm are also ineffcients for solving this type of problems. For this reason, in order to solve this kind of problems, we try to perturb stocasicly the deterministic classic method.

The problem (2) can be numerically approached by using Frank-Wolfe (FW) 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 [17] for unconstrained minimization of continuously differentiable functions and adapted to linearly constrained problems in [5].

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

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

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 $\{\mathbf{X}^k\}_{k\geq 0}$ 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 Frank-Wolfe method is recalled in section 3. Then, in section 4, we present the stochastic perturbation of FW method. Finally, in section 5, we provide some numerical experiments.

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

$$S = \{x \in E \mid Ax = b, \quad x \geq 0\}.$$

The objective function is $f : E \to \mathbb{R}$, its lower bound on S is denoted by α^* i.e. $\alpha^* = \min_S f$.

Let us introduce

$$S_\lambda = C_\lambda \cap S; \qquad \text{where} \quad C_\lambda = \{x \in E \mid f(x) \le \lambda\}.$$

We assume that

$$f \text{ is twice continuously differentiable on } E, \tag{3}$$

$$\forall \lambda > \alpha^* \ : S_\lambda \text{ is not empty, closed and bounded}, \tag{4}$$

$$\forall \lambda > \alpha^* : meas(S_\lambda) > 0, \tag{5}$$

where $meas(S_\lambda)$ is the measure of S_λ.

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

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

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 Frank–Wolfe method.

Thus,

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

where ε, η are positive real numbers.

3. The Frank-Wolfe method

In this section, we recall Frank-Wolfe method for convex optimization, see Frank and Wolfe [10], as well as Demyanov and Rubinov [4], cited here for minimization problems. From now on, we consider a nonlinear programming problem with linear equality or inequality constraints of the form

$$\begin{cases} \min & f(x) \\ s.t & Ax = (or \ \le) \ b \\ & 0 \le x \end{cases}$$

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

3.1. Search direction & line search. In the Frank-Wolfe algorithm one determines d_k through the solution of the approximation of the problem (1) 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:

$$\begin{cases} \text{minimize} & \nabla f(x^k)^\top s \\ \text{subject to} & As = (or \ \leq)\ b \\ & 0 \leq s \end{cases}$$

This is an LP problem, and it gives an extreme point, s_k, as an optimal solution. The search direction is $d_k := s_k - x_k$, that is, the direction vector from the feasible point x_k towards the extreme point.
d_k is a descent direction. Indeed,

$$\begin{aligned} \nabla f(x_k)^\top d_k &= \nabla f(x_k)^\top (s_k - x_k) \\ &= \nabla f(x_k)^\top s_k - \nabla f(x_k)^\top x_k < 0 \end{aligned}$$

since $\nabla f(x_k)^\top s_k < \nabla f(x_k)^\top x_k$.

By using iterations of the general form:

$$\forall k \geq 0 : x^{k+1} = Q_k(x^k) = x^k + \eta_k d_k. \tag{7}$$

One determine a step length, η_k, such that

$$f(x_k + \eta_k d_k) \leq f(x_k).$$

Among the many different step size rules that the Frank-Wolfe algorithm admits, we detail two :
1. exact line-search: we choose η_k solution of the problem of one-dimensional minimization of the function $\eta \to f(x_k + \eta d_k)$ on $[0, 1]$, we have

$$f(x_k + \eta_k d_k) = \min_{\eta \in [0,1]} f(x_k + \eta d_k).$$

2. inexact line search: Armijo rule, we have

$$f(x_k + \eta_k d_k) \leq f(x_k) + \beta \eta_k d_k^\top \nabla f(x_k).$$

3.2. Algorithm of Frank-Wolfe. The Frank-Wolfe 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 $x^1, x^2, \ldots$ that converges towards a solution of the optimization problem. The algorithm is defined as follows (Algorithm 1):

3.3. Convergence of Frank-Wolfe 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,[12]):

Algorithm 1 Frank-Wolfe algorithm

1: Choose a feasible point $x^{(0)}$
2: **for** $k = 0 \dots K$ **do**
3: Compute $\mathbf{s}_k := \text{LMO}\ (\nabla f(x^{(k)}))$
4: Let $\mathbf{d}_k := \mathbf{s}_k - x^{(k)}$ (the FW direction)
5: Compute $g_k := \left\langle -\nabla f(x^{(k)}), \mathbf{d}_k \right\rangle$ (FW gap)
6: **if** $g_k < \varepsilon$ **then return** $x^{(k)}$
8: Step size by optimal line search
$$\alpha_t \in \arg\min_{\alpha\in[0,1]} f(x^{(t)} + \alpha \mathbf{d}_t)$$
9: Update $x^{(k+1)} := x^{(k)} + \eta_t \mathbf{d}_k$
10: **end for**
11: **return** $x^{(K)}$

Theorem 3.1. *(Convergence of FW 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 Frank-Wolfe gap:*

$$\min_{0\le i\le t} g_i \le \frac{\max\left\{2h_0, L diam(D)^2\right\}}{\sqrt{t+1}} \qquad for\ t \ge 0,$$

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

Proof. See, [12]. □

4. Stochastic perturbation of Frank-Wolfe method

From [10], it is well-known that if f is not convex, the global minimum can not be found using a FW algorithm. To overcome this difficulty, we propose an appropriate random perturbation. In the next, we will establish the convergence of SPFW to a global minimum for non-convex optimization problems.

The sequence of real numbers $\left\{x^k\right\}_{k\ge 0}$ is replaced by a sequence of random variables $\left\{\mathbf{X}^k\right\}_{k\ge 0}$ involving a random perturbation P_k of the deterministic iteration (7); then we have $\mathbf{X}^0 = x^0$;

$$\forall k \ge 0\ \mathbf{X}^{k+1} = Q_k(\mathbf{X}^k) + P_k = \mathbf{X}^k + \eta_k d^k + P_k = \mathbf{X}^k + \eta_k (d^k + \frac{P_k}{\eta_k}), \tag{8}$$

where $\eta_k \ne 0$ satisfied the Step 8 in FW algorithm, and

$$\forall k \ge 1 \quad P_k \text{ is independent from } (\mathbf{X}^{k-1}, \dots, \mathbf{X}^0).$$

and

$$\mathbf{X} \in S \Rightarrow Q_k(\mathbf{X}) + P_k \in S.$$

Equation (8) 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 (8) become

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

General properties defining convenient sequences of perturbation $\{P_k\}_{k\geq 0}$ can be found in the literature [5, 17]: 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 Φ_k and ϕ_k the cumulative distribution function and the probability density of Z_k, respectively.

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

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

and the condition probability density of $\mathbf{X}^{k+1}$ 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 η_k is determined by Step 8. Let $P_k = \xi_k Z_k$

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

It follow that

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

So, we have

$$f_{k+1}(y \mid \mathbf{X}^k = x) = \frac{1}{\xi_k^n}\phi_k\left(\frac{y - Q_k(x)}{\xi_k}\right), \quad y \in S. \tag{9}$$

The relation (6) shows that

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

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

$$y \in S_\lambda \Rightarrow \phi_k\left(\frac{y - Q_k(x)}{\xi_k}\right) \geq h_k\left(\frac{\beta(\lambda, \varepsilon)}{\xi_k}\right). \tag{10}$$

For simplicity, let

$$Z_k = \mathbf{1}_C(Z_k) Z_k, \tag{11}$$

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

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

$$\forall k \geq 0: \ \alpha^* \geq U_{k+1} \geq U_k. \tag{12}$$

Thus, there exists $U \leq \alpha^*$ such that

$$U_k \to U \quad for \ k \to +\infty.$$

Lemma 4.1. *Let $P_k = \xi_k Z_k$ and $\gamma = f(x^0)$ if Z_k is given by (11), then there exists $v > 0$ such that*

$$P(U_{k+1} > \theta | U_k \leq \theta) \geq \frac{meas(S_\gamma - S_\theta)}{\xi_k^n} h_k\left(\frac{\beta(\gamma, \varepsilon)}{\xi_k}\right) > 0 \qquad \forall \theta \in (\alpha^*, \alpha^* + v],$$

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 $\hat{S}_\theta$ is not empty and has a strictly positive measure.

If $meas(S - \hat{S}_\theta) = 0$ 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 $\varepsilon > 0$ such that $meas(S - \hat{S}_\theta) > 0$. For $\theta \in (\alpha^*, \alpha^* + \varepsilon]$, we have $\hat{S}_\theta \subset \hat{S}_\varepsilon$ and $meas(S - \hat{S}_\theta) > 0$.

$P(\mathbf{X}^k \notin \hat{S}_\theta) = P(\mathbf{X}^k \in S - \hat{S}_\theta) = \int_{S - \hat{S}_\theta} P(\mathbf{X}^k \in dx) > 0$ for any $\theta \in (\alpha^*, \alpha^* + \varepsilon]$, since the sequence $\{\mathbf{U}_i\}_{i \geq 0}$ is increasing, we have also

$$\left\{\mathbf{X}^i\right\}_{i \geq 0} \subset S_\gamma. \tag{13}$$

Thus

$$P(\mathbf{X}^k \notin \hat{S}_\theta) = P(\mathbf{X}^k \in S - \hat{S}_\theta) = \int_{S_\gamma - \hat{S}_\theta} P(\mathbf{X}^k \in dx) > 0 \;\text{ for any } \theta \in (\alpha^*, \alpha^* + \varepsilon].$$

Let $\theta \in (\alpha^*, \alpha^* + \varepsilon]$, we have from (12)

$$P(U_{k+1} > \theta \mid U_k \leq \theta) = P(\mathbf{X}^{k+1} \in \hat{S}_\theta \mid \mathbf{X}^i \notin \hat{S}_\theta, i = 0, \ldots, k).$$

But Markov chain yield that

$$P(\mathbf{X}^{k+1} \in \hat{S}_\theta \mid \mathbf{X}^i \notin \hat{S}_\theta, i = 0, \ldots, k) = P(\mathbf{X}^{k+1} \in \hat{S}_\theta \mid \mathbf{X}^k \notin \hat{S}_\theta).$$

By the conditional probability rule

$$P(\mathbf{X}^{k+1} \in \hat{S}_\theta \mid \mathbf{X}^k \notin \hat{S}_\theta) = \frac{P(\mathbf{X}^{k+1} \in \hat{S}_\theta, \mathbf{X}^k \notin \hat{S}_\theta)}{P(\mathbf{X}^k \notin \hat{S}_\theta)}.$$

Moreover

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

From (13) we have

$$P(\mathbf{X}^{k+1} \in \hat{S}_\theta \mid \mathbf{X}^k \notin \hat{S}_\theta) = \int_{S_\gamma - \hat{S}_\theta} P(\mathbf{X}^k \in dx) \int_{\hat{S}_\theta} f_{k+1}(y \mid \mathbf{X}^k = x) dy,$$

and

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

Thus

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

Taking (9) into account, we have

$$P(\mathbf{X}^{k+1} \in \hat{S}_\theta \mid \mathbf{X}^k \notin \hat{S}_\theta) \geq \frac{1}{\xi_k^n} \inf_{x \in S_\gamma - \hat{S}_\theta} \left\{ \int_{\hat{S}_\theta} \phi_k \left(\frac{y - Q_k(x)}{\xi_k} \right) dy \right\}.$$

The relation (6) shows that

$$\|y - Q_k(x)\| \leq \beta(\gamma, \varepsilon).$$

and (10) 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

$$P(\mathbf{X}^{k+1}\in\hat{S}_\theta|\mathbf{X}^k\notin\hat{S}_\theta) \geq \frac{1}{\xi_k^n}\inf_{x\in S_\gamma-\hat{S}_\theta}\int_{\hat{S}_\theta} h_k\left(\frac{\beta(\gamma,\varepsilon)}{\xi_k}\right)dy.$$

$$P(\mathbf{X}^{k+1}\in\hat{S}_\theta|\mathbf{X}^k\notin\hat{S}_\theta) \geq \frac{meas(S_\gamma-S_\theta)}{\xi_k^n}h_k\left(\frac{\beta(\gamma,\varepsilon)}{\xi_k}\right).$$

□

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 [17]):

Lemma 4.2. *Let $\{U_k\}_{k\geq 0}$ be a increasing sequence, upper bounded by α^*. 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 $\{\mathbf{c}_k(\theta)\}_{k\geq 0}$ such that*

$$\forall k\geq 0\ :\ P(U_{k+1}>\theta\mid U_k\leq\theta)\geq\mathbf{c}_k(\theta)>0 \quad and \quad \sum_{k=0}^{+\infty}\mathbf{c}_k(\theta)=+\infty.$$

Then U=α^* almost surely.

Proof. For instance, see [13, 17]. □

Theorem 4.3. *Let $\gamma=f(x^0)$, assume that $x^0\in S$, the sequence ξ_k is non increasing and*

$$\sum_{k=0}^{+\infty}h_k\left(\frac{\beta(\gamma,\varepsilon)}{\xi_k}\right)=+\infty. \tag{14}$$

Then U=α^* almost surely.

Proof. Let

$$\mathbf{c}_k(\theta)=\frac{meas(S_\gamma-S_\theta)}{\xi_k^n}h_k\left(\frac{\beta(\gamma,\varepsilon)}{\xi_k}\right)>0.$$

Since the sequence $\{\xi_k\}_{k\geq 0}$ is non increasing,

$$\mathbf{c}_k(\theta)\geq\frac{meas(S_\gamma-S_\theta)}{\xi_k^n}h_k\left(\frac{\beta(\gamma,\varepsilon)}{\xi_k}\right)>0.$$

Thus, Eq. (14) shows that

$$\sum_{k=0}^{+\infty}\mathbf{c}_k(\theta)\geq\frac{meas(S_\gamma-S_\theta)}{\xi_k^n}\sum_{k=0}^{+\infty}h_k\left(\frac{\beta(\gamma,\varepsilon)}{\xi_k}\right)=+\infty.$$

Using Lemmas 4.1 and 4.2 we have U=α^* almost surely. □

Theorem 4.4. *Let Z_k define by (11), 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,

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

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 4.4, we have $U{=}\alpha^*$ almost surely. □

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 $\mathbf{X}^0 = x^0 \in S$. At step $k \geq 0, \mathbf{X}^k$ is known and $\mathbf{X}^{k+1}$ is determined.

We generate k_{sto} the number of perturbation, the case $k_{sto} = 0$ corresponds to the unperturbed Frank-Wolfe method.

In our experiments, the Gaussian variates are obtained from calls to standard generators. We use

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

The methods in the tables have the following meanings:

(i) "FW" stands for the method of Frank-Wolfe.

(ii) "SPFW" stands for the method of stochastic perturbation of Frank-Wolfe.

The code of the proposed algorithm SPFW is written by using Matlab programming language. We test SPFW method and compare it with FW algorithm. This algorithms has been tested on some problems from [1, 9, 16, 18, 19, 20], where linear constraints are present with given initial feasible points x^0. The results are listed in Table 2 to Table 4, where n stands for the dimension of tested problem and n_c stands for the number of constraints. We will report the following results: theoptimal value f^* and the number of iteration $Iter$.

5.1. Small and medium scale problem. 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 best known minimum value f_{best}.

Problem 1. ([1])

$$\begin{cases} \text{minimize}: & x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1)\cos(4\pi x_2) + 0.3 \\ \text{subject to}: & -50 \leq x_1 \leq 50 \\ & -50 \leq x_2 \leq 50 \end{cases}$$

We use $k_{sto} = 3$ and initial point $x^0 = (20, 10)^T$. The Matlab code of our approach furnish this optimal solution $x^* = (0.00518, 0.00284)^T$ and $f^*_{SPFW} = f_{best} = 5.92e\text{-}04$.

Problem 2. ([1])

$$\begin{cases} \text{minimize}: & 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4 \\ \text{subject to}: & -5 \leq x_1 \leq 5 \\ & -5 \leq x_2 \leq 5 \end{cases}$$

We use $k_{sto} = 5$ and initial point $x^0 = (1, 1)^T$. The Matlab code of our approach furnish this optimal solution $x^* = (-0.13147, 0.7303)^T$ and $f^*_{SPFW} = f_{best} = -1.0232$.

Problem 3. ([1])

$$\begin{cases} \text{minimize}: & 10^5x_1^2 + x_2^2 - (x_1^2 + x_2^2)^2 + 10^{-5}(x_1^2 + x_2^2)^4 \\ \text{subject to}: & -20 \leq x_1 \leq 20 \\ & -20 \leq x_2 \leq 20 \end{cases}$$

We use $k_{sto} = 10$ and initial point $x^0 = (5, 5)^T$. The Matlab code of our approach furnish this optimal solution $x^* = (0.00067, 14.8546)^T$ and $f^*_{SPFW} = f_{best} = -24776.45$.

Problem 4. ([1])

$$\begin{cases} \text{minimize}: & -\cos(x_1)\cos(x_2)\exp(-(x_1-\pi)^2 - (x_2-\pi)^2) \\ \text{subject to}: & -10 \leq x_1 \leq 10 \\ & -10 \leq x_2 \leq 10 \end{cases}$$

We use $k_{sto} = 25$ and initial point $x^0 = (2, 1)^T$. The Matlab code of our approach furnish this optimal solution $x^* = (3.0928, 3.204)^T$ and $f^*_{SPFW} = f_{best} = -0.99062$.

Problem 5. ([1])

$$\begin{cases} \text{minimize}: & (\exp(x_1) - x_2)^4 + 100(x_2 - x_3)^6 + (\tan(x_3 - x_4))^4 + x_1^8 \\ \text{subject to}: & -1 \leq x_1 \leq 1 \\ & -1 \leq x_2 \leq 1 \\ & -1 \leq x_3 \leq 1 \\ & -1 \leq x_4 \leq 1 \end{cases}$$

We use $k_{sto} = 20$ and initial point $x^0 = (0.5, 0.9, 0.9, 0.9)^T$. The Matlab code of our approach furnish this optimal solution $x^* = (-0.15227, 0.84332, 0.81448, 0.81061)^T$ and $f^*_{SPFW} = f_{best} = 4.03\text{e-}07$.

Problem 6. ([1])

$$\begin{cases} \text{minimize}: 100(x_2 - x_1^2)^2 + (1 - x_1)^2 + 90(x_4 - x_3^2)^2 + (1 - x_3)^2 + 10.1((x_2 - 1)^2 \\ \qquad +(x_4 - 1)^2) + 19.8(x_2 - 1)(x_4 - 1) \\ \text{subject to}: \ -10 \le x_1 \le 10 \\ \qquad -10 \le x_2 \le 10 \\ \qquad -10 \le x_3 \le 10 \\ \qquad -10 \le x_4 \le 10 \end{cases}$$

We use $k_{sto} = 20$ and initial point $x^0 = (0.9, 1, 1.2, 1.2)^T$. The Matlab code of our approach furnish this optimal solution $x^* = (0.91266, 0.83238, 1.0803, 1.1674)^T$ and $f^*_{SPFW} = f_{best}$ =0.02534.

Problem 7. ([16])

$$\begin{cases} \text{minimize}: -x_1 + x_1x_2 - x_2 \\ \text{subject to}: \ -6x_1 + 8x_2 \le 3 \\ \qquad 3x_1 - x_2 \le 3 \\ \qquad 0 \le x_1, x_2 \le 5 \end{cases}$$

We use $k_{sto} = 15$ and initial point $x^0 = (0, 0)^T$. The Matlab code of our approach furnish this optimal solution $x^* = (1.1667, 0.50005)^T$ and $f^*_{SPFW} = f_{best} = -1.0833$.

Problem 8. ([18])

$$\begin{cases} \text{minimize}: -2x_1 - 6x_2 + x_1^3 + 8x_2^2 \\ \text{subject to}: \ x_1 + 6x_2 \le 6 \\ \qquad 5x_1 + 4x_2 \le 10 \\ \qquad 0 \le x_1 \le 2 \\ \qquad 0 \le x_2 \le 1 \end{cases}$$

We use $k_{sto} = 2$ and initial point $x^0 = (0, 1)^T$. The Matlab code of our approach furnish this optimal solution $x^* = (0.81618, 0.37523)^T$ and $f^*_{SPFW} = f_{best} = -2.2137$.

Problem 9. ([20])

$$\begin{cases} \text{minimize}: x_1^2 - 10x_1x_2 + 7x_1 + 7x_2 - 9 \\ \text{subject to}: \ -2x_1 + 3x_2 \le 6 \\ \qquad 4x_1 - 5x_2 \le 8 \\ \qquad 5x_1 + 3x_2 \le 15 \\ \qquad -4x_1 - 3x_2 \le -12 \\ \qquad x_1, x_2 \ge 0 \end{cases}$$

We use $k_{sto} = 10$ and initial point $x^0 = (1, 3)^T$. The Matlab code of our approach furnish this optimal solution $x^* = (1.547, 2.4188)^T$ and $f^*_{SPFW} = f_{best} = -16.27$.

Problem 10. ([19])

$$\begin{cases} \text{minimize}: 2x_1 - 2x_1^2 + 2x_1x_2 + 3x_2 - 2x_2^2 \\ \text{subject to}: \ -x_1 + x_2 \le 1 \\ \qquad x_1 - x_2 \le 1 \\ \qquad -x_1 + 2x_2 \le 3 \\ \qquad 2x_1 - x_2 \le 3 \\ \qquad x_1, x_2 \ge 0 \end{cases}$$

We use $k_{sto} = 10$ and initial point $x^0 = (0.5, 0.5)^T$. The Matlab code of our approach furnish this optimal solution $x^* = (3, 3)^T$ and $f^*_{SPFW} = f_{best} = -3$.

Problem 11. ([8])

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

We use $k_{sto} = 20$ and initial point $x^0 = (2, 3/2, 0, 3/2, 0)^T$. The Matlab code of our approach furnish this optimal solution $x^* = (1, 0.73395, 0.73401, 1.266, 1.266)^T$ and $f^*_{SPFW} = f_{best}$ =3.91e-09.

Problem 12. ([8])

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

We use $k_{sto} = 20$ and initial point $x^0 = (1, 0, 0)^T$. The Matlab code of our approach furnish this optimal solution $x^* = (0.61781, 0.3282, 0.053988)^T$ and $f^*_{SPFW} = f_{best} = -26250.46$.

Problem 13. ([8])

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

where:

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

and data of y is presented in table 5.1 (see [8]).

We use $k_{sto} = 50$ and initial point $x^0 = (0.1, 0.2, 180, 160, 210, 11.21, 3.21, 5.8)^T$. The Matlab code of our approach furnish this optimal solution $x^* = (0.5009916, 0.5009925, 137.247, 187.1867, 174.5884, 16.48846, 24.89633, 10.55855)^T$ and $f^*_{SPFW} = f_{best} = 1149.78$.

Problem 14. ([8])

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

TABLE 1. Data of y.

i	y_i	i	y_i	i	y_i
1	95	102-118	150	199-201	200
2	105	119-122	155	102-204	205
3-6	110	123-142	160	205-212	210
7-10	115	143-150	165	213	215
11-25	120	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

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

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

We use $k_{sto} = 1$ and initial point $x^0 = (2.9, 0, 0.8, 0.2, 1.7)^T$. The Matlab code of our approach furnish this optimal solution $x^* = (0.40964, 5.6011, 6.1354, 7.7007e-12, 0.4258)^T$ and $f^*_{SPFW} = f_{best} = -21.1304$.

Problem 15. ([8])

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

where $y_i = 1 + 0.25(x_i - 1), \quad i = 1, 2, \ldots, 10.$

We use $k_{sto} = 5$ and initial point $x^0 = (1, 1, 1, 1, 1, 1, 1, 1, 0.5, 0.5)^T$. The Matlab code of our approach furnish this optimal solution $x^* = (0.99561, 0.94154, 0.94154, 0.94154, 0.94154, 0.94154, 0.94154, 0.94154)^T$ and $f^*_{SPFW} = f_{best}$ =0.0063.

Problem 16. ([8])

$$\begin{cases} \text{minimize :} & x_1 - x_2 - x_3 - x_1x_3 + x_1x_4 + x_2x_3 - x_2x_4 \\ \text{subject to :} & x_1 + 2x_2 \leq 8 \\ & 4x_1 + x_2 \leq 12 \\ & 3x_1 + 4x_2 \leq 12 \\ & 2x_3 + x_4 \leq 8 \\ & x_3 + 2x_4 \leq 8 \\ & x_3 + x_4 \leq 5 \\ & 0 \leq x_i, \ i = 1, \ldots, 10. \end{cases}$$

We use $k_{sto} = 5$ and initial point $x^0 = (0, 0, 0, 0)^T$. The Matlab code of our approach furnish this optimal solution $x^* = (3, -7.2287e-12, 4, -3.3552e-09)^T$ and $f^*_{SPFW} = f_{best} = -13$.

Problem 17. ([8])

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

We use $k_{sto} = 1$ and initial point $x^0 = (0, 0, 0, 0, 0, 0, 0, 0, 0, 0)^T$. The Matlab code of our approach furnish this optimal solution $x^* = (1, 1, 1, 1, 1, 1, 1, 1, 1, 1)^T$ and $f^*_{SPFW} = f_{best} = -15$.

Problem 18. ([8])

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

where

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

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

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

$$a = (8, 24, 20, 24, 16, 12)^T$$
$$b = (29, 41, 13, 21)^T$$
$$c = \begin{pmatrix} 300 & 270 & 460 & 800 \\ 740 & 600 & 540 & 380 \\ 300 & 490 & 380 & 760 \\ 430 & 250 & 390 & 600 \\ 210 & 830 & 470 & 680 \\ 360 & 290 & 400 & 310 \end{pmatrix} \text{ and } \quad d = \begin{pmatrix} -7 & -4 & -6 & -8 \\ -12 & -9 & -14 & -7 \\ -13 & -12 & -8 & -4 \\ -7 & -9 & -16 & -8 \\ -4 & -10 & -21 & -13 \\ -17 & -9 & -8 & -4 \end{pmatrix}.$$

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 $k_{sto} = 15$ and initial point $x^0 = (2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)^T$. The Matlab code of our approach furnish this optimal solution $x^* = (5.9998, 2.0002, 0, 0, 0, 2.9998, 0, 21, 20, 0, 0, 0, 0, 24, 0, 0, 3.0002, 0, 12.9998, 0, 0, 12, 0, 0)^T$ and $f^*_{SPFW} = f_{best}$ =15639.

TABLE 2. Comparing results between FW, SPFW algorithms.

			Algorithm			
Problem			FW		SPFW	
#	n	n_c	f^*	*Iter*	f^*	*Iter*
1	2	4	0.4936	9	5.92e-04	5
2	2	4	-0.2154	18	-1.0232	4
3	2	4	-24757.98	1934	-24776.45	897
4	2	4	-5e-09	2	-0.99062	2
5	4	4	0.06923	2	4.03e-07	2
6	4	4	0.02647	6	0.02534	2
7	2	2	-0.9999	2	-1.0833	2
8	2	2	-2.2136	8	-2.2137	5
9	2	4	-16.25	75	-16.27	60
10	2	4	0	3	-3	2
11	5	2	2.15e-05	12	3.91e-09	7
12	3	1	-26247.10	600	-26250.46	17
13	8	15	1538.31	3	1149.78	2
14	5	9	-21.1304	3	-21.1304	2
15	10	15	0.70	5	0.0063	1
16	4	6	-13	9	-13	2
17	10	16	-15	3	-15	2
18	24	10	18270	5	15639	3

5.2. Large scale problems. The numerical results of large scale problems are listed in Table 3 and Table 4.

Problem 19. ([8])

$$\begin{cases} \text{minimize} : -0.1 \sum_{i=1}^{n} \cos(5\pi x_i) + \sum_{i=1}^{n} x_i^2 \\ \text{subject to} : \ -1 \leq x_i \leq 1, \quad i = 1, 2, \ldots, n \end{cases}$$

Problem 20. ([8])

$$\begin{cases} \text{minimize} : 1 - \cos\left(2\pi \|x\|\right) + 0.1 \|x\| \\ \text{subject to} : \ -100 \leq x_i \leq 100, \quad i = 1, 2, \ldots, n \end{cases}$$

where $\|x\| = \sqrt{\sum_{i=1}^{n} x_i^2}$.

TABLE 3. Comparing results between FW, SPFW algorithms for problem 19.

			Algorithm			
Problem 19			FW		SPFW	
k	n	n_c	f^*	*Iter*	f^*	*Iter*
100	100	200	-109.98	3	-110	1
200	200	400	-220	5	-220	1
300	300	600	-329.95	3	-330	1
500	500	1000	-550	5	-550	1
900	900	1800	-990	5	-990	1

TABLE 4. Comparing results between FW, SPFW algorithms for problem 20.

			Algorithm			
Problem 20			FW		SPFW	
k	n	n_c	f^*	*Iter*	f^*	*Iter*
100	100	200	1.8999	111	0.3999	1
200	200	400	3.0104	45	0.10811	1
300	300	600	0.7046	5	0.5998	2
500	500	1000	4.99	132	0.8	2
900	900	1800	0.717	48	0.99	3

From Table 2 above, we see that our algorithm SPFW can find a global solution with a small number of iterations, and the computation results illustrate that our algorithm SPFW executes well for those problems. In contrast to the numerical results of FW algorithm, the results in Table 3 and Table 4 show that when the number of variables increases, this benefit becomes extremely apparent. This shows the potential advantage of SPFW algorithm when applied to solving problems with large numbers of variables.

6. Conclusion

In this work, we have studied the behavior of the Frank-Wolfe in non-convex situations. The stochastic perturbation of the Frank-Wolfe method (SPFW) converges to the global minimum for all differential objective functions, but the FW method converges to the local minimum. The numerical experiments show that the method is effective to calculate the global optimum. However, we observe that the adjunction of the stochastic perturbation improves the result, with a larger number of evaluations of the objective function. The main difficulty in the practical use of the stochastic perturbation is connected to the tuning of the parameters $\hat{a}$ and K_{sto}.

References

[1] M.M. Ali, C. Khompatraporn and Z.B. Zabinsky, A numerical evaluation of several stochastic algorithms on selected continuous global optimization test problems, *Journal of global optimization* **31** (2005), no. 4, 635–672.

[2] A.N. Avramidis, W. Chan, P. L'Ecuyer, Staffing multi-skill call centers via search methods and a performance approximation, *IIE Transactions* **41** (2009), no. 6, 483–497.

[3] D.P. Bertsekas, *Nonlinear Programming, 2nd Edition*, Athena Scientific, Belmont, 1999.

[4] V. Demyanov, A. Rubinov, *Approximate Methods in Optimization Problems*, American Elsevier Publishing Company, New York, 1970).

[5] A. El Mouatasim, R. Ellaia, J.E. Souza de Cursi, Random perturbation of projected variable metric method for linear constraints nonconvex nonsmooth optimization, *International Journal of Applied Mathematics and Computer Science* **21** (2011), no. 2, 317–329.

[6] A. El Mouatasim, R. Ellaia, A. Al-Hossain, A continuous approach to combinatorial optimization: application to water system pump operations, *Optimization Letters* **6** (2012), no. 1, 177–198.

[7] A. El Mouatasim, R. Ellaia, J.E. Souza de Cursi, Stochastic perturbation of reduced gradient & GRG methods for nonconvex programming problems, *Applied Mathematics and Computation* **226** (2014), 198–211.

[8] A. El Mouatasim, Implementation of reduced gradient with bisection algorithms for non-convex optimization problem via stochastic perturbation, *Numerical Algorithms* **78** (2018), no. 1, 41–62.

[9] C.A. Floudas, P.M. Pardalos, *A collection of test problems for constrained global optimization algorithms*, Lecture Notes in Computer Science, Volume 455, Springer-Verlag, Berlin, 1990.

[10] M. Frank, P. Wolfe, An Algorithm for Quadratic Programming, *Naval Research Logistics Quarterly* **3** (1956), 95–110.

[11] D. Garber,E. Hazan, Faster rates for the Frank-Wolfe method over strongly-convex sets, Proceedings of the 32nd *International Conference on Machine Learning*, Lille, France, JMLR: W&CP volume 37 (2015).

[12] S. Lacoste-Julien, Convergence rate of Frank-Wolfe for non-convex objectives, arXiv:1607.00345 [math.OC] (2016).

[13] P. L'Ecuyer, R. Touzin, On the Deng-Lin random number generators and related methods, *Statistic and Computing* **14** (2003), no. 1, 5–9.

[14] R. Nanuclef, E. Frandi, C. Sartori, H. Allende, A novel Frank-Wolfe algorithm. Analysis and applications to large-scale SVM training, *Information Sciences* **285** (2014), 66–99.

[15] N. Nikolaou, Fast Optimization of Non-Convex Machine Learning Objectives. Master thesis, University of Edinburgh (2012).

[16] P.M. Pardalos, J.B. Rosen, *Constrained Global Optimization: Algorithms and Applications*, Springer, 1987.

[17] M. Pogu, J.E. Souza de Cursi, Global optimization by random perturbation of the gradient method with a fixed parameter, *Journal of Global Optimization* **5** (1994), no. 2, 159–180.

[18] H.E. Romeijn, R.L. Smith, Simulated annealing for constrained global optimization, *Journal of Global Optimization* **5** (1994), no. 2, 101–126.

[19] Y. Zhao, T. Zhao, Global optimization for generalized linear multiplicative programming using convex relaxation, *Mathematical Problems in Engineering* **2018** (2018), Article ID 9146309, 1–8.

[20] X.G. Zhou, B.Y. Cao, A simplicial branch and bound duality-bounds algorithm to linear multiplicative programming *Journal of Applied Mathematics* **2013** (2013), Article ID 984168.

(Abdelkrim El Mouatasim, Abderrahmane Ettahiri) Faculty of Polydisciplinary Ouarzazate (FPO), Ibn Zohr University, B.P. 284, Ouarzazate 45800, Morocco.
E-mail address: a.mouatasim@gmail.com, ett.abdo@gmail.com