UNIVERSITÀ DEGLI STUDI DI PADOVA

Università degli Studi di Padova

DIPARTIMENTO DI MATEMATICA "TULLIO LEVI-CIVITA"

Corso di Laurea Magistrale in Matematica

Analysis of Stochastic Frank-Wolfe methods for Product Domains

Relatore:
Prof. Francesco Rinaldi

Correlatore:
Damiano Zeffiro

Candidato:
Alessandro Da Villa
Matricola 2023832

---

21 Luglio 2023

Everything that happens
is from now on

J. Vernon

Contents

Introduction

Frank-Wolfe algorithm (usually abbreviated as FW) was first introduced in 1956 [6] for the purpose of providing a first order projection-free algorithm for constrained convex optimization. At each iteration, the method aims to minimize a linear approximation of the original objective function, moving towards a so-called descent direction.

Since the original algorithm has been shown to converge with a sublinear rate of $O(1/k)$ (which is worse than the linear rate of the projected gradient method), variants like the Away-Step and the Pairwise one have been introduced starting from [17]; [9] and [2] proved that these new approaches converge linearly when the feasible set $\mathcal{P}$ is a polytope.

Thanks to its immediacy from both a conceptual and an implemental point of view, the Frank-Wolfe algorithm and its variants have been applied to a wide range of problems in the field of Operations Research. Among them, we are interested in using Frank-Wolfe techniques to solve problems of regularized empirical risk minimization (ERM) like the following

$$\min_{x \in \mathcal{P}} F(\mathbf{x}) := \frac{1}{n} \sum_{i=1}^{n} f_i(\mathbf{x})$$

which can be easily reformulated as a stochastic optimization problem.

In many situations related to machine learning or statistics, the number of observations in ERM happens to be very large; consequently, the computation of the full gradient in every FW iteration becomes an extremely expensive task. In order to overcome this issue, stochastic variants of FW have been considered and an $O(1/k)$ rate of convergence in expectation has been proved in [11]. The main idea behind this techinques is to replace the full gradient with a cheaper stochastic one.

Over the past few years, big data has also led to the need of optimization algorithms which have low computational cost per iteration and are able to exploit conveniently the model structure. One of the most common situations we can find in real-world applications is the presence of a feasible set $\mathcal{P}$ which is block separable, that is

$$\mathcal{P} = \mathcal{P}_1 \times \mathcal{P}_2 \times \cdots \times \mathcal{P}_l$$

with $\mathcal{P}_j \subset \mathbb{R}^{p_j}$ for $j \in \{1, \dots, l\}$ and $\sum\limits_{j=1}^{l} p_j = p$.

Product domains of this kind can be found in many different contexts like, e.g., traffic assignement [12], structural SVMs [10], semi-relaxed optimal transport [7] or dictionary learning [4].

Starting from [10], many different block-coordinate variants of the classic FW method have been introduced, with the aim of decomposing the original problem in low-dimensional independent subproblems exploiting the block separable structure of the feasible set. However, the presence of short steps (iterations where, in order to guarantee feasibility, we don't obtain fair progresses) tends to slow down the algorithm and doesn't allow us to smoothly extend the standard convergence analysis for FW to the block-coordinate case. Recently, [3] made use of the Short Step Chain (SSC) procedure described in [15] to provide a brand new block-coordinate bad step free algorithmic framework which guarantees a local linear convergence rate under a particular angle condition (first introduced in [15]) and a Kurdyka-Łojasiewicz (KL) property (previously seen in [1]).

In this thesis we will describe and conduct the convergence analysis of some stochastic variants of the Frank-Wolfe method in the presence of a product domain. The work is organized as follows. In Chapter 1, after a quick overview of the classic FW, Away-Step FW and Pairwise FW methods, we will describe in detail two stochastic variants: the Away-Step Stochastic FW method (ASFW) and the Pairwise Stochastic FW method (PSFW). As done in [8], we will see that these two variants converge linearly in expectation (and also almost surely) developing a novel proof technique based on some concepts of empirical processes and concentration inequalities.
In Chapter 2 we will describe the Short Step Chain procedure first introduced in [15]; we will then move to the special case of product domains, formulating a Block-Coordinate Frank-Wolfe method with SSC which is proven to converge linearly under some angle condition and some KL property (as seen in [3]).
In Chapter 3 we present two stochastic variants of the BCFW method with SSC which can be used to solve the ERM problem on a product domain.
Taking inspiration from the techniques seen in [8] we will be able to prove that such algorithms converge linearly in expectation (and also almost surely) under some mild assumptions.

Notation and preliminaries

Herein, vectors are denoted with bold lower-case letters $\mathbf{a}, \mathbf{b}, \mathbf{c}, \dots$. For a block vector $\mathbf{x} \in \mathbb{R}^p = \mathbb{R}^{p_1} \times \cdots \times \mathbb{R}^{p_l}$, we denote by $\mathbf{x}_j \in \mathbb{R}^{p_j}$ the component related to the j-th block in such a way that $\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_l)$. For any iterative algorithm producing outputs in $\mathbb{R}^p$, we denote with $\mathbf{x}^{(i)}$ the output of the i-th iteration of the method. For any vector $\mathbf{x} \in \mathbb{R}^p \setminus \mathbf{0}$, $\hat{\mathbf{x}} = \frac{\mathbf{x}}{||\mathbf{x}||}$ and $\hat{\mathbf{x}} = \mathbf{0}$ otherwise.
For any $\mathcal{C}$ closed and convex subset of $\mathbb{R}^p$ we indicate with $\pi(\mathcal{C}, \mathbf{x})$ the projection of $\mathbf{x}$ over $\mathcal{C}$. We also denote with $T_{\mathcal{C}}(\mathbf{x})$ the tangent cone to $\mathcal{C}$ in $\mathbf{x}$. In order to shorten the notation, we will write $\pi_{\mathbf{x}}(\mathbf{g})$ instead of $||\pi(T_{\mathcal{C}}(\mathbf{x}), \mathbf{g})||$ for any vector $\mathbf{g} \in \mathbb{R}^p$. In the convergence analysis we will carry out, it will be useful to have some type of angle condition, which ensures that the slope of the descent selection detected by some projection-free method $\mathcal{A}$ is optimal, up to some constant $\tau > 0$.
First, for any $\mathbf{x} \in \mathcal{C}$ and $\mathbf{g} \in \mathbb{R}^p$ define the directional slope lower bound as

$$\mathrm{DSB}_{\mathcal{A}}(\mathcal{C}, \mathbf{x}, \mathbf{g}) = \inf_{\mathbf{d} \in \mathcal{A}(\mathbf{x}, \mathbf{g})} \frac{\langle \mathbf{g}, \mathbf{d} \rangle}{\pi_{\mathbf{x}}(\mathbf{g}) ||\mathbf{d}||} \tag{0.0.1}$$

In the case when $\mathbf{x}$ is stationary, we set $\mathrm{DSB}_{\mathcal{A}}(\mathcal{C}, \mathbf{x}, \mathbf{g}) = 1$.
Then, given $K \subset \mathcal{C}$, define the slope lower bound as

$$\mathrm{SB}_{\mathcal{A}}(\mathcal{C}, K) = \inf_{\substack{\mathbf{g} \in \mathbb{R}^n \\ \mathbf{x} \in K}} \mathrm{DSB}_{\mathcal{A}}(\mathcal{C}, \mathbf{x}, \mathbf{g}) = \inf_{\substack{\mathbf{g} : \pi_{\mathbf{x}}(\mathbf{g}) \neq 0 \\ \mathbf{x} \in K}} \mathrm{DSB}_{\mathcal{A}}(\mathcal{C}, \mathbf{x}, \mathbf{g}) \tag{0.0.2}$$

We say that the angle condition holds for the method $\mathcal{A}$ if

$$\mathrm{SB}_{\mathcal{A}}(\mathcal{C}) := \mathrm{SB}_{\mathcal{A}}(\mathcal{C}, \mathcal{C}) = \tau > 0 \tag{0.0.3}$$

This condition is satisfied with explicit bounds in many interesting frameworks.
It will be useful also to recall the so-called local Kurdyka-Łojasiewicz (KL) property.

Assumption. Given a stationary point $\mathbf{x}_* \in \mathcal{C}$, there exist $\eta, \delta > 0$ such that for every $\mathbf{x} \in B_\delta(\mathbf{x}_*)$ with $f(\mathbf{x}_*) < f(\mathbf{x}) < f(\mathbf{x}_*) + \eta$ we have

$$\pi_{\mathbf{x}}(-\nabla f(\mathbf{x})) \geq \sqrt{2\sigma}[f(\mathbf{x}) - f(\mathbf{x}_*)]^{\frac{1}{2}} \tag{0.0.4}$$

Such condition holds for σ-strongly convex functions, but also in many non-convex settings: for example, the condition is satisfied when f is (non-convex) quadratic, i.e. $f(\mathbf{x}) = \mathbf{x}^T Q \mathbf{x} + b^T \mathbf{x} + c$, and $\mathcal{P}$ is a polytope.

Chapter 1

Stochastic Frank-Wolfe methods

1.1 The Frank-Wolfe algorithm

We will start our discussion with a a brief overview of the classic Frank-Wolfe (FW) algorithm and its main variants, trying to examine its strengths and its weaknesses. The ideas and the techniques behind this method take inspiration from convex optimization, where our aim is to minimize a convex objective function on a convex set. Afterwards, FW algorithm has been applied to a wide range of non-convex problems.

1.1.1 Frank-Wolfe for convex optimization

Dealing with constrained optimization on convex sets, Frank-Wolfe method is surely one of the most famous and widely used techniques.
In many situations it is preferred over other methods (like Projected Gradient) because of its nice properties, especially for machine learning applications. Suppose we want to solve the following problem:

$$\min_{\mathbf{x}\in\mathcal{P}} F(\mathbf{x}) \tag{1.1.1}$$

with $F : \mathbb{R}^p \to \mathbb{R}$ convex function with Lipschitz continuous gradient having constant $L > 0$, and $\mathcal{P} \in \mathbb{R}^p$ convex and compact set. Frank-Wolfe (often abbreviated in FW) method is based on the minimization of the linear approximation of the objective function, making use of the necessary condition for local minima of the problem (1.1.1). The main features of FW method are displayed in Algorithm 1.

Algorithm 1 Frank-Wolfe Method

```
1: Initialization $\mathbf{x}^{(0)} \in \mathcal{P}$.
2: for $k = 0, 1, \dots$ do
3:     Set $\mathbf{p}^{(k)} = \arg\min_{\mathbf{x}\in\mathcal{P}} \langle \nabla F(\mathbf{x}^{(k)}), \mathbf{x} \rangle$
4:     Set $\mathbf{d}^{(k)} = \mathbf{p}^{(k)} - \mathbf{x}^{(k)}$
5:     if $\langle \nabla F(\mathbf{x}^{(k)}), \mathbf{d}^{(k)} \rangle = 0$ then STOP
6:     else Set $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha^{(k)}\mathbf{d}^{(k)}$ with $\alpha^{(k)}$ suitably chosen stepsize
7:     end if
8: end for
```

In most cases, when we work with real-world situations we don't aim for an exact minimizer, but we rather settle for a nice stopping condition, which ensures that we are sufficiently near to a stationary point.
As concerns the computation of the stepsize described in Step 4, we have plenty of options to choose from; some common rules are given below.

- Constant stepsize: fix $\alpha \in (0, 1]$ for any iteration.
- Exact line-search: $\alpha^{(k)}$ is the valued obtained by the exact minimization of F along the direction $\mathbf{d}^{(k)}$, i.e.

$$\alpha^{(k)} \in \arg\min_{\alpha} F(\mathbf{x}^{(k)} + \alpha\mathbf{d}^{(k)}).$$

- Armijo rule: Given parameters δ and γ, with $\delta \in (0, 1)$ and $\gamma \in (0, \frac{1}{2})$, and a starting stepsize $\Delta^{(k)}$, we try steps

$$\alpha = \delta^m \Delta^{(k)}$$

  for $m = 1, 2, \ldots$ until condition

$$f(\mathbf{x}^{(k)} + \alpha\mathbf{d}^{(k)}) \leq F(\mathbf{x}^{(k)}) + \gamma\alpha\langle\nabla F(\mathbf{x}^{(k)}), \mathbf{d}^{(k)}\rangle$$

  is satisfied; at that point, set $\alpha^{(k)} = \alpha$.
- Diminishing stepsize: choose $\alpha^{(k)} = \frac{2}{k+1}$.

Looking at Step 3, we see how each iteration involves just an optimization of a linear function (the linearization of the objective F) over the feasible set; this is one of the strengths of this method, and it explains its wide applicability. Projected gradient methods, for example, require the optimization of a quadratic function over $\mathcal{P}$ which is clearly an harder task.
Notice also that each iterate is built as a convex combination of the previous iterate and a feasible search corner $\mathbf{p}^{(k)}$; this characteristic brings to two other big advantages.
First, since F is convex, minimizing the linearization of F over $\mathcal{P}$ we immediately obtain a lower bound on $F(\mathbf{x}^*)$, the value the optimal solution of our problem.
Secondly, going through the method we can write each iterate as a convex combination of the starting point $\mathbf{x}^{(0)}$ and all the search corners $\mathbf{p}^{(k)}$ previously found. This means that the iterates of Frank-Wolfe method are sparse, another useful property especially when we work in high dimensions.

[6] and [5] proved that the classical Frank-Wolfe method shows a slow convergence rate, as we can see from the theorem below.

Theorem 1.1.1. *Let F be a convex function with Lipschitz continuous gradient having constant $L > 0$. Call $\mathbf{x}^* = \min_{\mathbf{x}\in\mathcal{P}} F(\mathbf{x})$ and D the diameter of the set $\mathcal{P}$. The Frank-Wolfe method with diminishing stepsize $\alpha^{(k)} = \frac{2}{k+1}$ satisfies the following inequality:*

$$F(\mathbf{x}^{(k+1)}) - F(\mathbf{x}^*) \leq \frac{2LD^2}{k+1} \tag{1.1.2}$$

for all $k \geq 1$.

This sublinear convergence rate represents a huge drawback for the Frank-Wolfe method and it is due to the linear minimization oracle (LMO). The phenomenom is even more evident when the feasible set is a polytope: as soon as the iterates become closer to an optimum on the boundary, search directions become more orthogonal to the optimal direction and we observe a zig-zagging effect which prevents us to find $\mathbf{x}^*$ in a fast way. In order to overcome this obstacle, some variants of the method have been proposed. We will focus on two of them: Away-Step Frank-Wolfe (AFW) and Pairwise Frank-Wolfe (PFW).

1.1.2 Frank-Wolfe variants

One popular way to avoid the above mentioned zig-zagging problem is to keep track of the vertices previously involved in the algorithm (usually called "active" vertices), and move away from the ones producing the worst results.
We call $U^{(k)}$ the set of active vertices at iteration k.

At each iteration k the algorithm computes

$$\mathbf{p}^{(k)} \in \arg\min_{\mathbf{s}\in\mathcal{P}} \langle \nabla F(\mathbf{x}^{(k)}), \mathbf{s}\rangle$$
$$\mathbf{u}^{(k)} \in \arg\max_{\mathbf{v}\in U^{(k)}} \langle \nabla F(\mathbf{x}^{(k)}), \mathbf{v}\rangle$$

We define the classic Frank-Wolfe direction as

$$\mathbf{d}^{FW} = \mathbf{p}^{(k)} - \mathbf{x}^{(k)}$$

and the so-called away direction as

$$\mathbf{d}^{AS} = \mathbf{x}^{(k)} - \mathbf{u}^{(k)}$$

Given $\mathbf{x}^{(\mathbf{k})} \in \mathcal{P}$ and the active set $U^{(k)}$, the directions chosen by the Away-Step Frank-Wolfe method (called AFW directions) at iteration k with respect to the active set $U^{(k)}$ are

$$\mathbf{d}^{AFW} \in \arg\min\{\langle \nabla F(\mathbf{x}^{(k)}), \mathbf{d}\rangle \mid \mathbf{d} \in \{\mathbf{d}^{FW}, \mathbf{d}^{AS}\}\}.$$

In this variant, we include directions pointing away from extreme points.

In the Pairwise Frank-Wolfe method, the chosen PFW directions are

$$\mathbf{d}^{PFW} = \mathbf{d}^{FW} + \mathbf{d}^{AS}$$

or, equivalently

$$\mathbf{d}^{PFW} = \mathbf{p}^{(k)} - \mathbf{u}^{(k)}$$

This second variant combines classic FW and away-step FW, summing their respective search directions.

When we work in some specific settings (for example on polytopes) and we use one of these variants, we immediately find the following (see [15] for the full proof).

Lemma 1.1.2. *When the feasible set $\mathcal{P}$ is a polytope, AFW and PWF both satisfy the angle condition (0.0.3).*

1.2 Stochastic variants of Frank-Wolfe algorithm

The term "empirical risk minimization" (ERM) embraces a large class of constrained optimization problems. In this chapter we will focus on the following minimization problem

$$\min_{x\in\mathcal{P}} F(\mathbf{x}) := \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x}) \tag{1.2.1}$$

In our analysis, $\mathcal{P}$ will always be a polytope which can be described as all the possible convex combinations of a finite set of vertices V. Alternatively, a polytope can also be written as $\mathcal{P} = \{\mathbf{x} \in \mathbb{R}^p : \mathbf{C}\mathbf{x} \le \mathbf{d}\}$ for some $\mathbf{C} \in \mathbb{R}^{m\times p}, \mathbf{d} \in \mathbb{R}^m$. For every $i = 1, \dots, n$, the function $f_i : \mathbb{R}^p \to \mathbb{R}$ is σ_i-strongly convex with L_i Lipschitz continuous gradient.
Proximal gradient method represent a popular approach for these kind of problems, but the fact that it relies on some kind of projection operation makes this method expensive in many situations.
For this reason, projection-free methods like Frank Wolfe have been largely used in the last years: when the feasible set is a polytope, this approach is substantially faster.
Many applications in statistics and data science need a large number of observations f_i: as a result, computing the exact gradient at each iteration can be computationally heavy. In this context, it can be useful to rely on some cheaper "stochastic" gradient.
Problem (1.2.1) can indeed be reformulated from a stochastic point of view: first, we can define a random variable ξ taking values in $\{1, \dots, n\}$ following a discrete uniform distribution. If we define $f(i, \mathbf{x}) = f_i(\mathbf{x})$ for every $i = 1, \dots, n$ and $\mathbf{x} \in \mathcal{P}$, the previous problem can be rewritten as follows:

$$\min_{\mathbf{x}\in\mathcal{P}} \quad \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x}) = \mathbb{E} f(\xi, \mathbf{x}) \tag{1.2.2}$$

It will be useful also to define $\nabla f(\xi, \mathbf{x}) = \nabla f_\xi(\mathbf{x})$.

In Algorithm 2 a possible implementation of the Stochastic Away-Step Frank Wolfe method is displayed.
At each iteration k we sample $m^{(k)}$ random variables taking values uniformly in $\{1, \dots, n\}$, which corresponds to restrict our computations to just $m^{(k)}$ observations $f_{\xi_1}, \dots, f_{\xi_{m^{(k)}}}$. Then, starting from Step 4, we focus on the ancillary problem

$$\min_{\mathbf{x}\in\mathcal{P}} F^{(k)}(\mathbf{x})$$

where $F^{(k)}(\mathbf{x}) = \frac{1}{m^{(k)}}\sum\limits_{i=1}^{m^{(k)}} f_{\xi_i}(\mathbf{x})$ and we solve it using the Away-Step Frank Wolfe method.
It is immediate to see that $F^{(k)}$ is strongly convex with constant $\sigma^{(k)} = \frac{1}{m^{(k)}}\sum\limits_{i=1}^{m^{(k)}} \sigma_{\xi_i}$ and Lipschitz continuous with constant $L^{(k)} = \frac{1}{m^{(k)}}\sum\limits_{i=1}^{m^{(k)}} L_{\xi_i}$.
We also define our stochastic gradient as $\mathbf{g}^{(k)} = \frac{1}{m^{(k)}}\sum\limits_{i=1}^{m^{(k)}} \nabla_{\mathbf{x}} f(\xi_i, \mathbf{x}^{(k)})$.
Once a descent direction and a suitable stepsize are chosen, in Step 13 we build

the current solution and in Step 14 we update the active set $U^{(k+1)}$ via a specific procedure called Vertex Representation Update (VRU).

Algorithm 2 Away-Step Stochastic Frank-Wolfe algorithm

```
    Initialization $\mathbf{x}^{(0)} \in V$, $f_i$, $L_i$ and $k = 0$.
1:  Set $\mu^{(0)}_{\mathbf{x}^{(0)}} = 1$, $\mu^{(0)}_{\mathbf{v}} = 0$ for any $\mathbf{v} \in V \setminus \{\mathbf{x}^{(0)}\}$ and $U^{(0)} = \{\mathbf{x}^{(0)}\}$.
2:  for $k = 0, 1, \dots$ do
3:      Sample $\xi_1, \dots, \xi_{m^{(k)}} \overset{\text{i.i.d.}}{\sim} \xi$
4:      Set $\mathbf{g}^{(k)} = \frac{1}{m^{(k)}} \sum_{i=1}^{m^{(k)}} \nabla_{\mathbf{x}} f(\xi_i, \mathbf{x}^{(k)})$ and $L^{(k)} = \frac{1}{m^{(k)}} \sum_{i=1}^{m^{(k)}} L_{\xi_i}$.
5:      Compute $\mathbf{p}^{(k)} \in \arg\min_{\mathbf{x} \in \mathcal{P}} \langle \mathbf{g}^{(k)}, \mathbf{x} \rangle$.
6:      Compute $\mathbf{u}^{(k)} \in \arg\max_{\mathbf{v} \in U^{(k)}} \langle \mathbf{g}^{(k)}, \mathbf{v} \rangle$.
7:      if $\langle \mathbf{g}^{(k)}, \mathbf{p}^{(k)} + \mathbf{u}^{(k)} - 2\mathbf{x}^{(k)} \rangle \leq 0$ then
8:          Set $\mathbf{d}^{(k)} = \mathbf{p}^{(k)} - \mathbf{x}^{(k)}$ and $\gamma^{(k)}_{\max} = 1$.
9:      else
10:         Set $\mathbf{d}^{(k)} = \mathbf{x}^{(k)} - \mathbf{u}^{(k)}$ and $\gamma^{(k)}_{\max} = \frac{\mu^{(k)}_{\mathbf{u}^{(k)}}}{1 - \mu^{(k)}_{\mathbf{u}^{(k)}}}$.
11:     end if
12:     Set $\gamma^{(k)} = \min\{-\frac{\langle \mathbf{g}^{(k)}, \mathbf{d}^{(k)} \rangle}{L^{(k)} ||\mathbf{d}^{(k)}||^2}, \gamma^{(k)}_{\max}\}$ or determine it by line-search.
13:     Set $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \gamma^{(k)} \mathbf{d}^{(k)}$.
14:     Update $U^{(k+1)}$ and $\mu^{(k+1)}$ by procedure VRU.
15: end for
```

In the Pairwise Stochastic Frank-Wolfe algorithm, the procedure remains the same except for the choice of the descent direction: lines from 7 to 11 are replaced by $\mathbf{d}^{(k)} = \mathbf{p}^{(k)} - \mathbf{u}^{(k)}$ and $\gamma^{(k)}_{\max} = \mu^{(k)}_{\mathbf{u}^{(k)}}$. In the following section we will prove that these methods converge linearly in expectation and, consequently, converge linearly almost surely.

1.3 Convergence analysis of ASFW and PSFW

We now present a brand new procedure, first introduced by Goldfarb, Iyengar and Zhou in [8], to prove that the Away-Step Stochastic Frank-Wolfe algorithm (ASFW) and the Pairwise Stochastic Frank-Wolfe algorithm (PSFW) converge linearly in expectation.
The first thing we need is the lemma below, which will be used in the proof of the final theorem. A proof of this lemma is contained in [2].

Lemma 1.3.1. *For any $\mathbf{x} \in \mathcal{P} \setminus \{\mathbf{x}^{(k)}_*\}$ that can be represented as $\mathbf{x} = \sum_{\mathbf{v} \in U^{(k)}} \mu_{\mathbf{v}} \mathbf{v}$ for some $U^{(k)} \subset V$ where $\sum_{\mathbf{v} \in U^{(k)}} \mu_{\mathbf{v}} = 1$ and $\mu_{\mathbf{v}} > 0$ for every $\mathbf{v} \in U^{(k)}$, it holds that*

$$\max_{\mathbf{u} \in U, \mathbf{p} \in V} \langle \nabla F^{(k)}(\mathbf{x}), \mathbf{u} - \mathbf{p} \rangle \geq \frac{\Omega_{\mathcal{P}}}{|U|} \frac{\langle \nabla F^{(k)}(\mathbf{x}), \mathbf{x} - \mathbf{x}^{(k)}_* \rangle}{||\mathbf{x} - \mathbf{x}^{(k)}_*||}$$

where $|U^{(k)}|$ denotes the cardinality of $U^{(k)}$, V is the set of extreme point of $\mathcal{P}$ and

$$\Omega_{\mathcal{P}} = \frac{\zeta}{\phi}$$

for

$$\zeta = \min_{\mathbf{v}\in V, i\in\{1,\dots,m\}: a_i > \mathbf{C}_i\mathbf{v}} (d_i - \mathbf{C}_i\mathbf{v})$$
$$\phi = \max_{i\in\{1,\dots,m\}\setminus I(V)} ||\mathbf{C}_i||$$

Before moving on, we need some notions which come from the empirical processes literature.

1.3.1 Some notions from empirical processes theory

We will start from the bracketing number, a quantity that measures the complexity of a function class.

Definition. Given two functions l and u, the bracket $[l, u]$ is the set of all functions f such that $l \leq f \leq u$. A bracket $[l, u]$ is called an ϵ-bracket in L_1 if $\mathbb{E}|u - l| < \epsilon$.

Definition. Let $\mathcal{F}$ be a class of functions. The bracketing number $N_{[]}(\epsilon, \mathcal{F}, L_1)$ is the minimum number of ϵ-brackets needed to cover $\mathcal{F}$.

We can now state a lemma which provides an upper bound for the bracketing number of a function class indexed by a finite dimensional bounded set.

Lemma 1.3.2. *Let $\mathcal{F} = \{f_\theta \mid \theta \in \Theta\}$ be a family of measurable functions indexed by a bounded subset $\Theta \subset \mathbb{R}^p$. Suppose that there exists a measurable function g such that*

$$|f_{\theta_1}(\xi) - f_{\theta_2}(\xi)| \leq g(\xi)||\theta_1 - \theta_2|| \tag{1.3.1}$$

for every $\theta_1, \theta_2 \in \Theta$. If $||g(\xi)||_1 < \infty$, then the bracketing number satisfies

$$N_{[]}(\epsilon||g||_1, \mathcal{F}, L_1) \leq (\frac{\sqrt{p}D_\Theta}{\epsilon})^p \tag{1.3.2}$$

for every $0 \leq \epsilon \leq D_\Theta$, where $D_\Theta = \sup\{||\theta_1 - \theta_2|| \mid \theta_1, \theta_2 \in \Theta\}$.

A proof of Lemma 1.3.2 can be found in [8].

1.3.2 Convergence analysis

Starting from the upper bound for $N_{[]}$ found in Lemma 1.3.2, we can now provide a concentration bound for the variable $\sup_{\mathbf{x}\in\mathcal{P}} |F^{(k)}(\mathbf{x}) - F(\mathbf{x})|$ which involves both the objective function of the main problem and its "reduced" version of the ancillary one.

Lemma 1.3.3. *For any $\delta > 0$ and $0 < \epsilon < \min\{D, \frac{\delta}{2L_F}\}$ it holds*

$$\mathbb{P}\{\sup_{\mathbf{x}\in\mathcal{P}} |F^{(k)}(\mathbf{x}) - F(\mathbf{x})| \geq \delta\} \leq 2K_{\mathcal{P}}(\frac{D}{\epsilon})^p \exp\{-\frac{m^{(k)}(\delta - 2L_F\epsilon)^2}{2(u_F - l_F)^2}\}$$

where $L_F = \min\{L_1, \dots, L_n\}, K_{\mathcal{P}} = (\sqrt{p})^p, u_F = \max\{\sup_{\mathbf{x}\in\mathcal{P}} f_i(\mathbf{x}) \mid i = 1, \dots, n\}$ and $l_F = \min\{\inf_{\mathbf{x}\in\mathcal{P}} f_i(\mathbf{x}) \mid i = 1, \dots, n\}$.

We can immediately state a corollary which provides useful upper bounds for the expected values of $\sup_{\mathbf{x}\in\mathcal{P}} |F^{(k)} - F(\mathbf{x})|$ and $|F^{(k)}(\mathbf{x}_*^{(k)}) - F(\mathbf{x}^*)|$.

Corollary 1.3.4. *When $m^{(k)} \geq 3$,*

$$\mathbb{E}\sup_{\mathbf{x}\in\mathcal{P}} |F^{(k)}(\mathbf{x}) - F(\mathbf{x})| \leq C_1\sqrt{\frac{\log m^{(k)}}{m^{(k)}}}$$

and

$$\mathbb{E}|F^{(k)}(\mathbf{x}_*^{(k)}) - F(\mathbf{x}^*)| \leq C_1\sqrt{\frac{\log m^{(k)}}{m^{(k)}}}$$

where

$$C_1 = 4(|u_F| + |l_F|)K_{\mathcal{P}}D^p \exp\{-p(\log \frac{u_F - l_F}{2\sqrt{2}L_F})\} + (u_F - l_F)\sqrt{p+1}.$$

Lastly, we will also need the following technical lemma.

Lemma 1.3.5. *Let $c_i \geq 0$ and $b_i \in \{0,1\}$ for $i = 1,\dots,n$. Assume that $\sum_{j=1}^{n} b_j = m < n$. Then for $0 < a < 1$ we have*

$$\sum_{k=1}^{n} a^{\sum_{j=k}^{n} b_j} c_k \leq \sum_{k=1}^{m} a^{m-k+1} c_k + \sum_{k=m+1}^{n} c_k. \tag{1.3.3}$$

All the previous proofs are contained in [8].

We are now ready to state the theorem about the linear convergence in expectation of the ASFW and PSFW method. We also report here the full proof, because it will be the starting point for the convergence analysis of the algorithms of Chapter 3.

Theorem 1.3.6. *Let $\{\mathbf{x}^{(k)}\}_{k\geq 1}$ be the sequence generated by Algorithm (2) for solving problem (1.2.1). Let N be the number of vertices used to represent $\mathbf{x}^{(k)}$ and F^* be the optimal value of the problem. Let $\rho = \min\{\frac{1}{2}, \frac{\Omega_{\mathcal{P}}^2\sigma_F}{16N^2L_FD^2}\}$ with $\sigma_F = \min\{\sigma_1,\dots,\sigma_n\}$, $L_F = \max\{L_1,\dots,L_n\}$. Set $m^{(i)} = \lceil \frac{1}{(1-\rho)^{2i+2}} \rceil$. Then for every $k \geq 1$*

$$\mathbb{E}\{F(\mathbf{x}^{(k+1)}) - F^*)\} \leq C_2(1-\beta)^{(k-1)/2}$$

with C_2 deterministic constant and $0 < \beta < \rho \leq \frac{1}{2}$.

Proof. In order to shorten notations, define $F_*^{(k)} = F^{(k)}(x_*^{(k)})$. The stochastic gradient is $\mathbf{g}^{(k)} = \nabla F^{(k)}(\mathbf{x})$. For the descent direction $\mathbf{d}^{(k)}$ chosen by the algorithm at iteration k, it holds

$$\langle \mathbf{g}^{(k)}, \mathbf{d}^{(k)}\rangle \leq \frac{1}{2}(\langle \mathbf{g}^{(k)}, \mathbf{p}^{(k)} - \mathbf{x}^{(k)}\rangle + \langle \mathbf{g}^{(k)}, \mathbf{x}^{(k)} - \mathbf{u}^{(k)}\rangle) = \frac{1}{2}\langle \mathbf{g}^{(k)}, \mathbf{p}^{(k)} - \mathbf{u}^{(k)}\rangle \leq 0.$$

Hence, we can lower bound $\langle \mathbf{g}^{(k)}, \mathbf{d}^{(k)} \rangle^2$ by

$$\begin{aligned}
\langle \mathbf{g}^{(k)}, \mathbf{d}^{(k)} \rangle^2 &\geq \frac{1}{4} \langle \mathbf{g}^{(k)}, \mathbf{u}^{(k)} - \mathbf{p}^{(k)} \rangle^2 \\
&\geq \frac{1}{4} \max_{\mathbf{p} \in V, \mathbf{u} \in U^{(k)}} \langle \mathbf{g}^{(k)}, \mathbf{u} - \mathbf{p} \rangle^2 \qquad \text{(by definition of } \mathbf{p}^{(k)} \text{ and } \mathbf{u}^{(k)}) \\
&= \frac{1}{4} \max_{\mathbf{p} \in V, \mathbf{u} \in U^{(k)}} \langle \nabla F^{(k)}(\mathbf{x}^{(k)}), \mathbf{u} - \mathbf{p} \rangle^2 \qquad \text{(by definition of } \mathbf{g}^{(k)}) \\
&\geq \frac{1}{4} \frac{\Omega_{\mathcal{P}}^2}{|U^{(k)}|^2} \frac{\langle \nabla F^{(k)}(\mathbf{x}^{(k)}), \mathbf{x}^{(k)} - \mathbf{x}_*^{(k)} \rangle^2}{||\mathbf{x}^{(k)} - \mathbf{x}_*^{(k)}||^2} \qquad \text{(by Lemma 1.3.1)} \\
&\geq \frac{\Omega_{\mathcal{P}}^2}{4N^2} \frac{(F^{(k)}(\mathbf{x}^{(k)}) - F_*^{(k)})^2}{||\mathbf{x}^{(k)} - \mathbf{x}_*^{(k)}||^2} \qquad \text{(since } F^{(k)} \text{ is convex)} \\
&\geq \frac{\Omega_{\mathcal{P}}^2 \sigma^{(k)}}{8N^2} (F^{(k)}(\mathbf{x}^{(k)}) - F_*^{(k)}) \qquad \text{(since } F^{(k)} \text{ is strongly convex)} \\
&\geq \frac{\Omega_{\mathcal{P}}^2 \sigma_F}{8N^2} (F^{(k)}(\mathbf{x}^{(k)}) - F_*^{(k)}) \qquad \text{(by definition of } \sigma_F)
\end{aligned}$$

We can also upper bound $\langle \mathbf{g}^{(k)}, \mathbf{d}^{(k)} \rangle$ in a similar way:

$$\begin{aligned}
\langle \mathbf{g}^{(k)}, \mathbf{d}^{(k)} \rangle &\leq \frac{1}{2} \langle \mathbf{g}^{(k)}, \mathbf{p}^{(k)} - \mathbf{u}^{(k)} \rangle \\
&\leq \frac{1}{2} \langle \mathbf{g}^{(k)}, \mathbf{x}_*^{(k)} - \mathbf{x}^{(k)} \rangle \qquad \text{(by definition of } \mathbf{p}^{(k)} \text{ and } \mathbf{u}^{(k)}) \\
&= \frac{1}{2} \langle \nabla F^{(k)}(\mathbf{x}^{(k)}), \mathbf{x}_*^{(k)} - \mathbf{x}^{(k)} \rangle \qquad \text{(by definition of } \mathbf{g}^{(k)}) \\
&\leq \frac{1}{2} (F_*^{(k)} - F^{(k)}(\mathbf{x}^{(k)})) \qquad \text{(since } F^{(k)} \text{ is convex)}
\end{aligned}$$

Let's now focus on the stepsize $\gamma^{(k)}$ chosen by the algorithm at iteration k. Four different scenarios are possible:

- $A^{(k)}$ $\gamma_{\max}^{(k)} \geq 1$ and $\gamma^{(k)} \leq 1$.
- $B^{(k)}$ $\gamma_{\max}^{(k)} \geq 1$ and $\gamma^{(k)} \geq 1$.
- $C^{(k)}$ $\gamma_{\max}^{(k)} < 1$ and $\gamma^{(k)} < \gamma_{\max}^{(k)}$.
- $D^{(k)}$ $\gamma_{\max}^{(k)} < 1$ and $\gamma^{(k)} = \gamma_{\max}^{(k)}$.

From the standard descent lemma, we have

$$\begin{aligned}
F^{(k)}(\mathbf{x}^{(k+1)}) &= F^{(k)}(\mathbf{x}^{(k)} + \gamma^{(k)} \mathbf{d}^{(k)}) \\
&\leq F^{(k)}(\mathbf{x}^{(k)}) + \gamma^{(k)} \langle \nabla F^{(k)}(\mathbf{x}^{(k)}), \mathbf{d}^{(k)} \rangle + \frac{L^{(k)} (\gamma^{(k)})^2}{2} ||\mathbf{d}^{(k)}||^2 \qquad (1.3.4) \\
&= F^{(k)}(\mathbf{x}^{(k)}) + \gamma^{(k)} \langle \mathbf{g}^{(k)}, \mathbf{d}^{(k)} \rangle + \frac{L^{(k)} (\gamma^{(k)})^2}{2} ||\mathbf{d}^{(k)}||^2.
\end{aligned}$$

Suppose that at iteration k first scenario $A^{(k)}$ occurs. Let $\delta_{A^{(k)}}$ denote the indicator function for such case. Then

$$\begin{aligned}
&\delta_{A^{(k)}}\{F^{(k)}(\mathbf{x}^{(k+1)}) - F_*^{(k)}\} \\
&\leq \delta_{A^{(k)}}\{F^{(k)}(\mathbf{x}^{(k)}) - F_*^{(k)} + \gamma^{(k)}\langle \mathbf{g}^{(k)}, \mathbf{d}^{(k)}\rangle + \frac{L^{(k)}(\gamma^{(k)})^2}{2}||\mathbf{d}^{(k)}||^2\} \\
&= \delta_{A^{(k)}}\{F^{(k)}(\mathbf{x}^{(k)}) - F_*^{(k)} - \frac{\langle \mathbf{g}^{(k)}, \mathbf{d}^{(k)}\rangle^2}{2L^{(k)}||\mathbf{d}^{(k)}||^2}\} \\
&\leq \delta_{A^{(k)}}\{(1 - \frac{\Omega_{\mathcal{P}}^2\sigma_F}{16N^2L^{(k)}D^2})(F^{(k)}(\mathbf{x}^{(k)}) - F_*^{(k)})\} \\
&\leq \delta_{A^{(k)}}\{(1 - \frac{\Omega_{\mathcal{P}}^2\sigma_F}{16N^2L_FD^2})(F^{(k)}(\mathbf{x}^{(k)}) - F_*^{(k)})\}.
\end{aligned}$$

Let's move now to case $B^{(k)}$. Since $\gamma^{(k)} > 1$, we have

$$-\frac{\langle \mathbf{g}^{(k)}, \mathbf{d}^{(k)}\rangle}{L^{(k)}||\mathbf{d}^{(k)}||^2} > 1$$

hence

$$-\langle \mathbf{g}^{(k)}, \mathbf{d}^{(k)}\rangle > L^{(k)}||\mathbf{d}^{(k)}||^2 \tag{1.3.5}$$

and also

$$\gamma^{(k)}\langle \mathbf{g}^{(k)}, \mathbf{d}^{(k)}\rangle + \frac{L^{(k)}(\gamma^{(k)})^2}{2}||\mathbf{d}^{(k)}||^2 \leq \langle \mathbf{g}^{(k)}, \mathbf{d}^{(k)}\rangle + \frac{L^{(k)}}{2}||\mathbf{d}^{(k)}||^2. \tag{1.3.6}$$

Using $\delta_{B^{(k)}}$ to denote the indicator function in this second case, we get

$$\begin{aligned}
&\delta_{B^{(k)}}\{F^{(k)}(\mathbf{x}^{(k+1)}) - F_*^{(k)}\} \\
&\leq \delta_{B^{(k)}}\{F^{(k)}(\mathbf{x}^{(k)}) - F_*^{(k)} + \gamma^{(k)}\langle \nabla F^{(k)}(\mathbf{x}^{(k)}), \mathbf{d}^{(k)}\rangle + \frac{L^{(k)}(\gamma^{(k)})^2}{2}||\mathbf{d}^{(k)}||^2\} \\
&= \delta_{B^{(k)}}\{F^{(k)}(\mathbf{x}^{(k)}) - F_*^{(k)} + \gamma^{(k)}\langle \mathbf{g}^{(k)}, \mathbf{d}^{(k)}\rangle + \frac{L^{(k)}(\gamma^{(k)})^2}{2}||\mathbf{d}^{(k)}||^2\} \\
&\leq \delta_{B^{(k)}}\{F^{(k)}(\mathbf{x}^{(k)}) - F_*^{(k)} + \langle \mathbf{g}^{(k)}, \mathbf{d}^{(k)}\rangle + \frac{L^{(k)}}{2}||\mathbf{d}^{(k)}||^2\} \\
&\leq \delta_{B^{(k)}}\{F^{(k)}(\mathbf{x}^{(k)}) - F_*^{(k)} + \frac{1}{2}\langle \mathbf{g}^{(k)}, \mathbf{d}^{(k)}\rangle\} \\
&\leq \delta_{B^{(k)}}\{\frac{1}{2}(F^{(k)}(\mathbf{x}^{(k)}) - F_*^{(k)})\}
\end{aligned}$$

As regards case $C^{(k)}$, computation are similar to the ones we performed for case $A^{(k)}$; therefore we get

$$\begin{aligned}
\delta_{C^{(k)}}\{F^{(k)}(\mathbf{x}^{(k+1)}) - F_*^{(k)}\} &\leq \delta_{C^{(k)}}\{F^{(k)}(\mathbf{x}^{(k)}) - F_*^{(k)} - \frac{\langle \mathbf{g}^{(k)}, \mathbf{d}^{(k)}\rangle^2}{2L^{(k)}||\mathbf{d}^{(k)}||^2}\} \\
&\leq \delta_{C^{(k)}}\{(1 - \frac{\Omega_{\mathcal{P}}^2\sigma_F}{16N^2L_FD^2})(F^{(k)}(\mathbf{x}^{(k)}) - F_*^{(k)})\}
\end{aligned}$$

In case $D^{(k)}$, also called "drop step", we have

$$\gamma^{(k)} = \gamma_{\max}^{(k)} \leq -\frac{\langle \mathbf{g}^{(k)}, \mathbf{d}^{(k)}\rangle}{L^{(k)}||\mathbf{d}^{(k)}||^2}$$

hence

$$\begin{aligned}
&\delta_{D^{(k)}}\{F^{(k)}(\mathbf{x}^{(k+1)}) - F_*^{(k)}\} \\
&\leq \delta_{D^{(k)}}\{F^{(k)}(\mathbf{x}^{(k)}) - F_*^{(k)} + \gamma^{(k)}\langle \nabla F^{(k)}(\mathbf{x}^{(k)}), \mathbf{d}^{(k)}\rangle + \frac{L^{(k)}(\gamma^{(k)})^2}{2}||\mathbf{d}^{(k)}||^2\} \\
&= \delta_{D^{(k)}}\{F^{(k)}(\mathbf{x}^{(k)}) - F_*^{(k)} + \gamma^{(k)}\langle \mathbf{g}^{(k)}, \mathbf{d}^{(k)}\rangle + \frac{L^{(k)}(\gamma^{(k)})^2}{2}||\mathbf{d}^{(k)}||^2\} \\
&\leq \delta_{D^{(k)}}\{F^{(k)}(\mathbf{x}^{(k)}) - F_*^{(k)} + \frac{\gamma^{(k)}}{2}\langle \mathbf{g}^{(k)}, \mathbf{d}^{(k)}\rangle\} \\
&\leq \delta_{D^{(k)}}\{F^{(k)}(\mathbf{x}^{(k)}) - F_*^{(k)}\}.
\end{aligned}$$

Define $\rho := \min\{\frac{1}{2}, \frac{\Omega_{\mathcal{P}}\sigma_F}{16N^2L_FD^2}\}$. We can immediately notice that ρ is a deterministic constant between 0 and 1. Hence

$$\begin{aligned}
F^{(k)}(\mathbf{x}^{(k+1)}) - F_*^{(k)} &\leq (1-\rho)^{\{1-\delta_{D^{(k)}}\}}(F^{(k)}(\mathbf{x}^{(k)}) - F_*^{(k)}) \\
&= (1-\rho)^{\{1-\delta_{D^{(k)}}\}}(F^{(k-1)}(\mathbf{x}^{(k)}) - F_*^{(k-1)}) + \\
&\quad + (1-\rho)^{\{1-\delta_{D^{(k)}}\}}(F^{(k)}(\mathbf{x}^{(k)}) - F_*^{(k)} - F^{(k-1)}(\mathbf{x}^{(k)}) + F_*^{(k-1)}) \\
&= (1-\rho)^{\{1-\delta_{D^{(k)}}\}}(F^{(k-1)}(\mathbf{x}^{(k)}) - F_*^{(k-1)}) + \\
&\quad + (1-\rho)^{\{1-\delta_{D^{(k)}}\}}\{F^{(k)}(\mathbf{x}^{(k)}) - F(\mathbf{x}^{(k)}) + F(\mathbf{x}^{(k)}) - F^{(k-1)}(\mathbf{x}^{(k)}) + \\
&\quad + F^* - F_*^{(k)} + F_*^{(k-1)} - F^*\} \\
&\leq (1-\rho)^{\{1-\delta_{D^{(k)}}\}}(F^{(k-1)}(\mathbf{x}^{(k)}) - F_*^{(k-1)}) + \\
&\quad + (1-\rho)^{\{1-\delta_{D^{(k)}}\}}\{|F^{(k)}(\mathbf{x}^{(k)}) - F(\mathbf{x}^{(k)})| + |F^{(k-1)}(\mathbf{x}^{(k)}) - F(\mathbf{x}^{(k)})| + \\
&\quad + |F_*^{(k)} - F^*| + |F_*^{(k-1)} - F^*|\} \\
&\leq (1-\rho)^{\sum_{i=1}^k\{1-\delta_{D^{(i)}}\}}(F^{(0)}(\mathbf{x}^{(1)}) - F_*^{(0)}) + \\
&\quad + \sum_{i=1}^k(1-\rho)^{\sum_{j=i}^k\{1-\delta_{D^{(j)}}\}}\{|F^{(i)}(\mathbf{x}^{(i)}) - F(\mathbf{x}^{(i)})| + \\
&\quad + |F^{(i-1)}(\mathbf{x}^{(i)}) - F(\mathbf{x}^{(i)})| + |F_*^{(i)} - F^*| + |F_*^{(i-1)} - F^*|\}
\end{aligned}$$

At iteration k, the drop steps can be at most $\frac{k+1}{2}$: it means that there are at most $\frac{k+1}{2}$ $\delta_{D^{(i)}}$'s which are equal to 1. Therefore, using Lemma 1.3.5 we have

$$\begin{aligned}
&\sum_{i=1}^k(1-\rho)^{\sum_{j=i}^k\{1-\delta_{D^{(j)}}\}}\{|F^{(i)}(\mathbf{x}^{(i)}) - F(\mathbf{x}^{(i)})| + |F^{(i-1)}(\mathbf{x}^{(i)}) - F(\mathbf{x}^{(i)})| + \\
&+ |F_*^{(i)} - F^*| + |F_*^{(i-1)} - F^*|\} \\
&\leq \sum_{i=k/2}^k \{|F^{(i)}(\mathbf{x}^{(i)}) - F(\mathbf{x}^{(i)})| + |F^{(i-1)}(\mathbf{x}^{(i)}) - F(\mathbf{x}^{(i)})| + |F_*^{(i)} - F^*| + |F_*^{(i-1)} - F^*|\} \\
&+ \sum_{i=1}^{k/2-1} (1-\rho)^{k/2-i}\{|F^{(i)}(\mathbf{x}^{(i)}) - F(\mathbf{x}^{(i)})| + |F^{(i-1)}(\mathbf{x}^{(i)}) - F(\mathbf{x}^{(i)})| + |F_*^{(i)} - F^*| \\
&+ |F_*^{(i-1)} - F^*|\}.
\end{aligned}$$

Recalling that $u_F = \max\{\sup_{\mathbf{x}\in\mathcal{P}} f_i(\mathbf{x}) \mid i = 1,\ldots,n\}$ and $l_F = \min\{\inf_{\mathbf{x}\in\mathcal{P}} f_i(\mathbf{x}) \mid i = 1,\ldots,n\}$ we finally get

$$\begin{aligned}F^{(k)}(\mathbf{x}^{(k+1)}) - F_*^{(k)} \leq &(1-\rho)^{\frac{k-1}{2}}(u_F - l_F)\\ &+ \sum_{i=k/2}^{k} \{|F^{(i)}(\mathbf{x}^{(i)}) - F(\mathbf{x}^{(i)})| + |F^{(i-1)}(\mathbf{x}^{(i)}) - F(\mathbf{x}^{(i)})| + |F_*^{(i)} - F^*|\\ &+ |F_*^{(i-1)} - F^*|\} + \sum_{i=1}^{k/2-1} (1-\rho)^{k/2-i}\{|F^{(i)}(\mathbf{x}^{(i)}) - F(\mathbf{x}^{(i)})|\\ &+ |F^{(i-1)}(\mathbf{x}^{(i)}) - F(\mathbf{x}^{(i)})| + |F_*^{(i)} - F^*| + |F_*^{(i-1)} - F^*|\}.\end{aligned}$$

Adding and subtracting, we also have

$$F^{(k)}(\mathbf{x}^{(k+1)}) - F_*^{(k)} = F(\mathbf{x}^{(k+1)}) - F^* + (F^{(k)}(\mathbf{x}^{(k+1)}) - F(\mathbf{x}^{(k+1)})) + (F^* - F_*^{(k)}).$$

Therefore

$$\begin{aligned}F(\mathbf{x}^{(k+1)}) - F^* \leq &(1-\rho)^{\frac{k-1}{2}}(u_F - l_F)\\ &+ \sum_{i=k/2}^{k+1} \{|F^{(i)}(\mathbf{x}^{(i)}) - F(\mathbf{x}^{(i)})| + |F^{(i-1)}(\mathbf{x}^{(i)}) - F(\mathbf{x}^{(i)})| + |F_*^{(i)} - F^*|\\ &+ |F_*^{(i-1)} - F^*|\} + \sum_{i=1}^{k/2-1} (1-\rho)^{k/2-i}\{|F^{(i)}(\mathbf{x}^{(i)}) - F(\mathbf{x}^{(i)})|\\ &+ |F^{(i-1)}(\mathbf{x}^{(i)}) - F(\mathbf{x}^{(i)})| + |F_*^{(i)} - F^*| + |F_*^{(i-1)} - F^*|\}.\end{aligned}$$

We can notice that for any deterministic $\mathbf{x} \in \mathcal{P}$, we have $\mathbb{E}\{F^{(k)}(\mathbf{x})\} = F(\mathbf{x})$. Using Corollary 1.3.4, we have that for every iteration k it holds

$$\mathbb{E}|F^{(k)}(\mathbf{x}^{(k)}) - F(\mathbf{x}^{(k)})| \leq \mathbb{E}\sup_{\mathbf{x}\in\mathcal{P}} |F^{(k)}(\mathbf{x}) - F(\mathbf{x})| \leq C_1\sqrt{\frac{\log m^{(k)}}{m^{(k)}}}$$

and

$$\mathbb{E}|F_*^{(k)} - F^*| \leq C_1\sqrt{\frac{\log m^{(k)}}{m^{(k)}}}.$$

Putting together what we found so far, and using $m^{(i)} = \lceil \frac{1}{(1-\rho)^{2i+2}} \rceil$, we have

$$\begin{aligned}
\mathbb{E}\{F(\mathbf{x}^{(k+1)}) - F^*\} \leq (1-\rho)^{\frac{k-1}{2}}(u_F - l_F) + 2C_1\{ \sum_{i=k/2}^{k+1} (\sqrt{\frac{\log m^{(i)}}{m^{(i)}}} + \sqrt{\frac{\log m^{(i-1)}}{m^{(i-1)}}}) \\
+ \sum_{i=1}^{k/2-1} (1-\rho)^{k/2-i} (\sqrt{\frac{\log m^{(i)}}{m^{(i)}}} + \sqrt{\frac{\log m^{(i-1)}}{m^{(i-1)}}})\} \\
\leq (1-\rho)^{\frac{k-1}{2}}(u_F - l_F) \\
+4C_1\{ \sum_{i=k/2}^{k+1} \sqrt{\frac{\log m^{(i-1)}}{m^{(i-1)}}} + \sum_{i=1}^{k/2-1} (1-\rho)^{k/2-i} (\sqrt{\frac{\log m^{(i-1)}}{m^{(i-1)}}}\} \\
\leq (1-\rho)^{\frac{k-1}{2}}(u_F - l_F) \\
+4C_1 \sqrt{2\log\frac{1}{1-\rho}} \{ \sum_{i=k/2}^{k+1} (1-\rho)^i \sqrt{i} + \sum_{i=1}^{k/2-1} (1-\rho)^{k/2}\sqrt{i}\}
\end{aligned} \tag{1.3.7}$$

where we used the fact that $\frac{\log x}{x}$ decreases for $x > e$.

From inequality (1.3.7) we immediately get that

$$\mathbb{E}\{F(x^{(k+1)}) - F^*\} \leq C_2(1-\beta)^{\frac{k-1}{2}}$$

for some constant C_2 and $0 < \beta < \rho < 1$.

$\square$

Obs. The proof of Theorem 1.3.6 contains stochastic arguments only in its last part. One could try to replicate such passages to prove the convergence of other similar stochastic algorithms, which is what we will do in Chapter 3.

From Theorem 1.3.6 we can also conclude that the ASFW method converges almost everywhere. The proof of such corollary can be found in [8].

Corollary 1.3.7. *Let $\{\mathbf{x}^{(k)}\}_{k\geq 1}$ be the sequence generated by Algorithm 2 for solving problem (1.2.1). Then*

$$\frac{F(\mathbf{x}^{(k)}) - F^*}{(1-\omega)^{\frac{k-1}{2}}} \to 0$$

almost surely as $k \to \infty$ for some $0 < \omega < \beta$. Therefore $F(\mathbf{x}^{(k)})$ converges linearly to F^* almost surely.

Obs. With the same line of reasoning, we can prove that the Pairwise Stochastic Frank Wolfe (PSFW) algorithm converges linearly in expectation and almost surely. Full proofs can be found in [8].

Chapter 2

Frank-Wolfe with SSC on product domains

In Chapter 1 we have already seen that the general Frank-Wolfe algorithm for minimizing a convex function over a polytope converges at an $O(1/k)$ rate ([6], [5]). This poor performances come from the presence of "short steps" in the algorithm: with this term we indicate iterations where, in order to guarantee feasibility, we are forced to take tiny steps which don't produce fair progresses.
Moreover, if we are in presence of a product domain and we put ourselves in a block-coordinate framework, the difficulties which derive from handling these bad iterations don't allow to easily extend the results of the standard convergence analysis for FW, AFW and PFW (as seen in [14]).
The Short Step Chain (SSC) method (introduced in [15]) aims to remove all the bad steps in the algorithm guaranteeing a sufficient advancement at each iteration; as we will see in the next pages, this procedure can be combined with any first order projection-free algorithm. Thanks to this supplementary tool, we will be able to provide a block-coordinate FW variant (as the one in [3]) which allows us to get rid of the bad steps and, at the same time, offers more flexibility in the block selection strategies and in the choice of the descent directions.
With these adjustments it will be possible to carry out the convergence analysis also in the block-coordinate case; a local linear convergence rate will be given for three different block selection strategies and for any FW-like direction under some hypotesis (the angle condition (0.0.3) and the KL property).

2.1 Frank-Wolfe with SSC

Before moving to product domains and to the related block-coordinate FW variants, we will discuss how the SSC method can be introduced in the classic FW algorithm in order to avoid the short steps phenomena and improve the convergence rate.

2.1.1 Short Step Chain (SSC) to avoid bad steps

In order to overcome the problems deriving from the bad steps (which can be very numerous) we use the so called Short Step Chain (SSC), which can be applied in general to any first order projection-free method.
The main idea behind this procedure is to get rid of these short steps by skipping gradient updates until we reach some specific conditions which ensure to have a

sufficient advancement. The main features of the Short Step Chain can be found in Algorithm 3.

Algorithm 3 Short Step Chain SSC($\bar{\mathbf{x}}, \mathbf{z}$)

```
    Initialization $\mathbf{y}^{(0)} = \bar{\mathbf{x}} \in \mathcal{P}$, $j = 0$.
    Phase I
 1: Select $\mathbf{d}^{(j)} \in \mathcal{A}(\mathbf{y}^{(j)}, \mathbf{w})$ and $\alpha_{\max}^{(j)} \in \alpha_{\max}(\mathbf{y}^{(j)}, \mathbf{d}^{(j)})$
 2: if $\mathbf{d}^{(j)} = 0$ then
 3:     return $\mathbf{y}^{(j)}$
 4: end if
    Phase II
 5: Compute $\beta^{(j)}$ auxiliary stepsize
 6: Let $\alpha^{(j)} = \min(\alpha_{\max}^{(j)}, \beta^{(j)})$
 7: $\mathbf{y}^{(j+1)} = \mathbf{y}^{(j)} + \alpha^{(j)}\mathbf{d}^{(j)}$
 8: if $\alpha^{(j)} = \beta^{(j)}$ then
 9:     return $\mathbf{y}^{(j+1)}$
10: end if
11: $j = j + 1$, go to Step 1.
```

The inputs needed by the SSC algorithm are some point $\bar{\mathbf{x}}$ and a vector $\mathbf{z}$. Therefore we can easily incorporate the code for SSC in any first order method: $\bar{\mathbf{x}}$ will be the iterate $\mathbf{x}^{(k)}$, and $\mathbf{z}$ will be the usual antigradient $-\mathbf{g} = -\nabla F(\mathbf{x}^{(k)})$.

Algorithm 4 First-order method with SSC

```
    Initialization $\mathbf{x}^{(0)} \in \mathcal{P}$, $k = 0$.
 1: while $\mathbf{x}^{(k)}$ is not stationary do
 2:     $\mathbf{g} = \nabla F(\mathbf{x}^{(k)})$
 3:     $\mathbf{x}^{(k+1)}$ =SSC($\mathbf{x}^{(k)}, -\mathbf{g}$)
 4: end while
```

Let's now analyze in detail what are the main features of the Short Step Chain method described in Algorithm 3.
In Phase I a stationarity check is performed on the descent directions $\mathbf{d}^{(j)}$.
In Phase II the stepsize $\alpha^{(j)}$ is finally computed, taking the minimum between the "usual" maximal stepsize $\alpha_{\max}^{(j)}$ and an ausiliary stepsize $\beta^{(j)}$ defined as follows:

$$\beta^{(j)} = \begin{cases} 0 & \text{if } \mathbf{y}^{(j)} \notin \Omega^{(j)} \\ \beta_{\max}(\Omega^{(j)}, \mathbf{y}^{(j)}, \mathbf{d}^{(j)}) & \text{if } \mathbf{y}^{(j)} \in \Omega^{(j)} \end{cases} \tag{2.1.1}$$

Notice that the point $\mathbf{y}^{(i+1)}$ generated after phase II is always feasible, because the stepsize $\alpha^{(j)}$ is by definition always smaller than the maximal stepsize $\alpha_{\max}^{(j)}$ along the chosen direction $\mathbf{d}^{(j)}$. In particular, $\beta_{\max}(\Omega^{(j)}, \mathbf{y}^{(j)}, \mathbf{d}^{(j)})$ represents the maximal feasible stepsize moving in the direction $\mathbf{d}^{(j)}$ starting from $\mathbf{y}^{(j)}$ with respect to the trust region

$$\Omega^{(j)} = \bar{B}_{||\mathbf{g}||/2L}(\bar{\mathbf{x}} - \frac{\mathbf{g}}{2L}) \cap \bar{B}_{\langle -\mathbf{g}, \hat{\mathbf{d}}^{(j)} \rangle / L}(\bar{\mathbf{x}}) \tag{2.1.2}$$

The choice of this peculiar trust region Ω_j guarantees a nice sufficient decrease condition for the objective function F, as proved in the lemma below (first introduced in [15]).

Lemma 2.1.1. *If we apply SSC algorithm with inputs $\bar{\mathbf{x}}$ and $-\mathbf{g}$, we get the following:*

$$F(\mathbf{y}^{(j)}) \leq F(\bar{\mathbf{x}}) - \frac{L}{2}||\bar{\mathbf{x}} - \mathbf{y}^{(j)}||^2 \tag{2.1.3}$$

for all $j = 1, \dots, T$ with T final iteration index in the SSC procedure.

Proof. Notice that the first ball $B_1 = \bar{B}_{||\mathbf{g}||/2L}(\bar{\mathbf{x}} - \frac{\mathbf{g}}{2L})$ involved in the definition of the trust region $\Omega^{(j)}$ does not depend on j; then, since $\mathbf{y}^{(0)} = \bar{\mathbf{x}}$ belongs to B_1, we have $\mathbf{y}^{(j)} \in B_1$ for every index $j = 1, \dots, T$. Applying the standard descent lemma and recalling the definition of B_1, we get the following chain of inequalities:

$$F(\mathbf{z}) \leq F(\bar{\mathbf{x}}) + \langle \mathbf{g}, \mathbf{z} - \bar{\mathbf{x}} \rangle + \frac{L}{2}||\bar{\mathbf{x}} - \mathbf{z}||^2 \leq F(\bar{\mathbf{x}}) - \frac{L}{2}||\bar{\mathbf{x}} - \mathbf{z}||^2 \quad \forall\, \mathbf{z} \in B_1$$

which proves the statement once we choose $\mathbf{z} = \mathbf{y}_j$. □

In [15] it is also underlined that the monotone decreasing behavior of the true objective function F during the SSC is guaranteed.

Lemma 2.1.2. *Assume $\mathbf{y}^{(j)} \in \bar{B}_{\langle -\mathbf{g}, \hat{\mathbf{d}}_j \rangle / L}(\bar{\mathbf{x}})$. Then we have*

$$F(\mathbf{y}^{(j)} + \beta^{(j)}\mathbf{d}^{(j)}) \leq F(\mathbf{y}^{(j)}).$$

Proof. Consider any $\beta \in [0, \beta^{(j)}]$. We have

$$\begin{aligned}\frac{d}{d\beta}F(\mathbf{y}^{(j)} + \beta\mathbf{d}^{(j)}) &= ||\mathbf{d}^{(j)}||\langle \nabla F(\mathbf{y}^{(j)} + \beta\mathbf{d}^{(j)}), \hat{\mathbf{d}}^{(j)} \rangle \\ &= ||\mathbf{d}^{(j)}||\langle (\nabla F(\mathbf{y}^{(j)} + \beta\mathbf{d}^{(j)}) - \mathbf{g}) + \mathbf{g}, \hat{\mathbf{d}}^{(j)} \rangle \\ &= ||\mathbf{d}^{(j)}||(\langle (\nabla F(\mathbf{y}^{(j)} + \beta\mathbf{d}^{(j)}) - \mathbf{g}), \hat{\mathbf{d}}^{(j)} \rangle + \langle \mathbf{g}, \hat{\mathbf{d}}^{(j)} \rangle)\end{aligned}$$

Since F has Lipschitz continuous gradient, it holds

$$\begin{aligned}\nabla F(\mathbf{y}^{(j)} + \beta\mathbf{d}^{(j)}) - \mathbf{g} &= \nabla F(\mathbf{y}^{(j)} + \beta\mathbf{d}^{(j)}) - \nabla F(\bar{\mathbf{x}}) \\ &\leq L||\bar{\mathbf{x}} - \mathbf{y}^{(j)} - \beta\mathbf{d}^{(j)}||\end{aligned}$$

and since $\mathbf{y}^{(j)} + \beta\mathbf{d}^{(j)} \in \bar{B}_{\langle -\mathbf{g}, \hat{\mathbf{d}}^{(j)} \rangle / L}(\bar{\mathbf{x}})$, we get

$$\frac{d}{d\beta}f(\mathbf{y}^{(j)} + \beta\mathbf{d}^{(j)}) \leq ||\mathbf{d}^{(j)}||(L||\bar{\mathbf{x}} - \mathbf{y}^{(j)} - \beta\mathbf{d}^{(j)}|| + \langle \mathbf{g}, \mathbf{d}^{(j)} \rangle) \leq 0.$$

□

In particular, when we incorporate the SSC procedure inside any first-order method like in Algorithm 4, we obtain the following

Proposition. *Consider the sequence $\mathbf{x}^{(k)}$ generated by Algorithm 4. Assume that*

- *The angle condition (0.0.3) holds*
- *The SSC procedure terminates in a finite number of steps*

Then

$$F(\mathbf{x}^{(k)}) - F(\mathbf{x}^{(k+1)}) \geq \frac{L}{2}||\mathbf{x}^{(k)} - \mathbf{x}^{(k+1)}||^2 \tag{2.1.4}$$

Proof. It is immediate from Lemma 2.1.1. □

2.1.2 SSC for Frank-Wolfe variants

We would now like to understand what happens when the descent directions generated from the first order method $\mathcal{A}$ inside the SSC are picked according to one of the Frank-Wolfe variants (Away-Step or Pairwise) seen in 1.1.2.
First, we report from [15] a general criterion that ensures the termination of the SSC when applied to any first-order projection free method.

Lemma 2.1.3. *Assume that the method $\mathcal{A}$ applied to any linear function $L_{\mathbf{g}} = \langle \mathbf{g}, \mathbf{x} \rangle$ on the feasible set $\mathcal{P}$ and with every stepsize maximal always terminates in at most T iterations with an optimal solution. Then the SSC applied to the method $\mathcal{A}$ on the feasible set $\mathcal{P}$ always terminates in at most T iterations.*

Proof. Since the method $\mathcal{A}$ applied to $L_{\mathbf{g}}$ terminates in at most T iterations, it means that it generates a sequence $\{\mathbf{y}_j\}_j$, for an index $j = 1, \dots, T'$ with $T' \leq T$ and $\mathbf{y}^{(T')} \in \arg\min_{\mathbf{x} \in \mathcal{P}} L_{\mathbf{g}}(\mathbf{x})$.
Assume now by contradiction that the SSC employs at least $T+1$ iterations, generating the sequence $\{\mathbf{y}^{(j)}\}_{j=1,\dots,T+1}$. If this is the case, the method must always do maximal steps for $j = 1, \dots, T$, because it terminates only when $\alpha^{(j)} = \beta^{(j)}$ (Step 8 of Algorithm 3), and when this happens it means that $\alpha^{(j)} < \alpha^{(j)}_{\max}$.
But in this case the assumption of the lemma tells us that $\mathcal{A}$ finds $\mathbf{y}_{T'} \in \arg\min_{\mathbf{x} \in \mathcal{C}} L_{\mathbf{g}}(\mathbf{x})$ for some $T' \leq T$. At this point we find a contradiction, because the method isn't able to find a feasible descent direction in Phase I and it terminates returning $\mathbf{y}^{(T')}$ after just $T' \leq T$ iterations. □

We can use Lemma 2.1.3 to bound the number of iterations of the SSC when combined with the Away-Step and the Pairwise Frank-Wolfe variants.

Proposition. *If $\mathcal{P}$ is a polytope described by its set of vertices V, the SSC always terminates in at most:*

- *$|V|$ iterations for the AFW*
- *$|V| - 1$ iterations for the PFW*

The proof is contained in [15].

2.2 Block-Coordinate Frank-Wolfe method with SSC

From now on, our work will mainly focus on minimizing some objective function in a so-called product domain. These kind of problems arise in many different contexts; we will see how Frank-Wolfe method with SSC can be a good choice for solving them, once we implement some smart strategies to exploit the peculiar structure of the feasible set. More precisely, we would like to study the problem

$$\min_{\mathbf{x} \in \mathcal{P}} F(\mathbf{x}) \tag{2.2.1}$$

where F has Lipschitz continuous gradient with constant L and $\mathcal{P} = \mathcal{P}_1 \times \mathcal{P}_2 \times \cdots \times \mathcal{P}_l$ where $\mathcal{P}_i \subset \mathbb{R}^{n_i}$ are closed and convex for $i = 1, \dots, l$.
In the case of F convex, one of the standard approaches used to solve such problem is represented by the Block-Coordinate Gradient Descent (BCGD) method. The main

idea behind this technique is to pick (according to some rule) some block of variables at each iteration and build a simple model for the objective function which involves only the chosen variables. At this point, the method performs a projection on the feasible set related to the chosen block.
These kind of strategies are easy to understand and to implement; the main problem lies in the projection, which can be very costly even when we work on structured sets.
For this reason, in the last years projection-free algorithms have become more popular in these kind of settings; among them, we focus on Frank-Wolfe variants that are able to effectively exploit the structure of the problem.

2.2.1 Randomized Block-Coordinate approach applied to the classical Frank-Wolfe

We have already described the main advantages in using the Frank-Wolfe method. However, dealing with product domains, this approach shows a meaningful drawback: each iteration requires a full pass through the data to determine the arg min of the linearized objective function, which corresponds to p calls to the minimization oracle. This issue can be very challenging when we work with high dimensional data. For this reason a randomized block-coordinate variant of the method has been proposed in 2013 by Lacost-Julien et al. [10], focusing on the case of structured support vector machines; we display it in Algorithm 5.
It is suited for product domains, with the intent of exploiting the structure of the problem.

Algorithm 5 Randomized Block-Coordinate FW on Product Domain

```
   Initialization $\mathbf{x}^{(0)} \in \mathcal{P} = \mathcal{P}_1 \times \mathcal{P}_2 \times \cdots \times \mathcal{P}_l$ and $k = 0$.
1: for $k = 0, 1, \dots$ do
2:     Pick $i$ at random in $\{1, \dots, l\}$
3:     Set $\mathbf{p}_i^{(k)} = \arg\min_{\mathbf{s} \in \mathcal{P}_i} \langle (\nabla_i F(\mathbf{x}^{(k)})), \mathbf{s} \rangle$
4:     Set $\mathbf{d}_i^{(k)} = \mathbf{p}_i^{(k)} - \mathbf{x}_i^{(k)}$
5:     if $\langle \nabla_i F(\mathbf{x}^{(k)}), \mathbf{d}_i^{(k)} = 0 \rangle$ then STOP
6:     else For all indices $j \neq i$, set $\mathbf{x}_j^{(k+1)} = \mathbf{x}_j^{(k)}$
7:         For index $i$, set $\mathbf{x}_i^{(k+1)} = \mathbf{x}_i^{(k)} + \alpha^{(k)} \mathbf{d}_i^{(k)}$ with $\alpha^{(k)}$ suitably chosen stepsize
8:     end if
9: end for
```

In Step 2, we randomly pick one index between 1 and l, which corresponds to the choice of one block of variables. Then, in Step 5 and 6, we update the variables related to the chosen block using the classic FW procedure, while all the other variables are left unchanged.
Many changes have been proposed in [14] in order to improve performances of such algorithm. In particular, the authors focused on adaptive block sampling strategies and using FW variants (like Pairwise or Away-Step).
Moreover, some parallel and distributed block coordinate FW algorithms have been studied in [16].
However, even with these adjustments, a significative issue remains: it's not possible

to directly extend the standard convergence analysis for the batch FW (and its variants) to the block-coordinate case. As seen in [14], the primary cause is to be sought in the presence of the short steps previously mentioned and, as a result, novel proof techniques are required to fill the gap between FW and BCFW.
Notice also that all these procedures are designed for convex programming problems and focus only on random sampling variants in the block selection process.
In the following section we will propose a new, general framework which can be applied also to the non-convex case and explores different block selection strategies.

2.2.2 BCFW with SSC

We will now present a new approach to our problem, motivated by a twofold purpose: fix the bad steps issue and guarantee sufficient flexibility in the choice of FW-like directions and also block selection strategies.
In order to avoid bad steps in the FW procedure, we proceed as follows. At each iteration, after having picked at least one block according to a certain rule, we make use of the SSC algorithm seen in 2.1.1: this allows us to skip gradient computations until proper conditions are satisfied.

As regards the block selection strategy, we can take inspiration from the Block-Coordinate gradient descent (BCGD) method and focus on three different options:

- Parallel or Jacobi-like selection: at iteration k, choose $\mathcal{M}^{(k)} = \{1, \dots, l\}$.
  In this case, SSC is performed separately on all blocks. This clearly represents a cheaper alternative with respect to the use of SSC on the whole set of variables, especially if used on multicore architectures which allow to solve those problems in parallel.

- Gauss-Southwell (GS) selection: choose $\mathcal{M}^{(k)} = \{i(k)\}$ with

$$i(k) \in \underset{i \in \{1,\dots,l\}}{\arg\max} \langle -\mathbf{g}_i^{(k)}, \text{SSC}(\mathbf{x}_i^{(k)}, -\nabla_i F(\mathbf{x}^{(k)})) - \mathbf{x}_i^{(k)} \rangle$$

  In this case, we perform SSC on all blocks and then we select a block which maximally violates optimality conditions.
  Using this strategy we expect better progress in terms of the objective function, but also a heavier computational burden.

- Random sampling: choose $\mathcal{M}^{(k)} = \{i(k)\}$ with $i(k)$ chosen uniformly at random in $\{1, \dots, l\}$.
  In this case, at each iteration we randomly generate an index with uniform probability distribution.

Obviously, as first order projection-free method $\mathcal{A}$ inside SSC we can choose the one which appears to be most convenient for our case.

Taking into account what we have seen so far, we can formulate a more general block-coordinate procedure, summarized in Algorithm 6.

Each iteration of the algorithm presents two main operations. In Step 2, we pick a subset of blocks according to one of the rules we described so far. Then, in Step 3 and Step 4, we apply the SSC method in order to update the variables related to the chosen blocks, while all the other variables remain unchanged.

Algorithm 6 Block-coordinate method with SSC

```
   Initialization $\mathbf{x}^{(0)} \in \mathcal{P}, k = 0$.
1: if $\mathbf{x}^{(k)}$ is stationary then STOP
2: end if
3: Choose $\mathcal{M}^{(k)} \in \{1, \dots, l\}$
4: For all $i \notin \mathcal{M}^{(k)}$ set $\mathbf{x}_i^{(k+1)} = \mathbf{x}_i^{(k)}$
5: For all $i \in \mathcal{M}^{(k)}$ set $\mathbf{x}_i^{(k+1)} = \text{SSC}(\mathbf{x}_i^{(k)}, -\nabla_i F(\mathbf{x}^{(k)}))$
6: Set $k = k + 1$ and go to Step 1
```

Obs. Is it important to underline once more that these method do not require a convex function F. As a matter of fact, his new algorithmic framework applies also to non-convex programming.

Inside SSC we can apply, in principle, any first order projection-free method $\mathcal{A}$ in order to obtain descent directions of any kind.
From now on, we suppose to choose FW-like directions (like AFW or PFW directions): in this case, the algorithm takes the name of Block-Coordinate Frank-Wolfe with Short Step Chain (BCFW with SSC).
In the next section we will analyze its convergence properties under some mild assumptions on the objective function F and on the chosen first order method $\mathcal{A}$.

2.3 Convergence analysis

In this section we report the main convergence result for the Block-Coordinate method with SSC described in Algorithm 6.
In our examination we will consider the three different block selection strategies described in 2.2.2.
Full proof of the following theorem can be found in [3].

Theorem 2.3.1. *Let $\mathbf{x}_* \in \mathcal{P}$ be a stationary point and consider the sequence $\mathbf{x}^{(k)}$ generated by Algorithm 6. Assume that:*

- *the local KL condition (0.0.4) holds at $\mathbf{x}_*$;*
- *the angle condition (0.0.3) holds in every block for the same $\tau > 0$;*
- *the SSC procedure always terminates in a finite number of steps;*
- *$F(\mathbf{x}_*)$ is a minimum in the connected component of $\{\mathbf{x} \in \mathcal{P} : F(\mathbf{x}) \le F(\mathbf{x}^{(0)})\}$ containing $\mathbf{x}^{(0)}$.*

Then, there exists $\tilde{\delta} > 0$ such that, if $\mathbf{x}^{(0)} \in B_{\tilde{\delta}}(\mathbf{x}_*)$:

- *If at iteration k we follow the parallel block selection strategy, we have*

$$F(\mathbf{x}^{(k)}) - F(\mathbf{x}_*) \le (q_P)^k [F(\mathbf{x}^{(0)}) - F(\mathbf{x}_*)] \tag{2.3.1}$$

and $\mathbf{x}^{(k)} \to \tilde{\mathbf{x}}_*$ with

$$||\mathbf{x}^{(k)} - \tilde{\mathbf{x}}_*|| \le \frac{\sqrt{2 - 2q_P}}{\sqrt{L}(1 - \sqrt{q_P})} (q_P)^{\frac{k}{2}} [F(\mathbf{x}^{(0)}) - F(\tilde{\mathbf{x}}_*)] \tag{2.3.2}$$

with

$$q_P = 1 - \frac{\sigma\tau^2}{4L(1+\tau)^2}. \tag{2.3.3}$$

- *If at iteration k we follow the GS block selection strategy, we have*

  $$F(\mathbf{x}^{(k)}) - F(\mathbf{x}_*) \leq (q_{GS})^k [F(\mathbf{x}^{(0)}) - F(\mathbf{x}_*)] \tag{2.3.4}$$

  and $\mathbf{x}^{(k)} \to \tilde{\mathbf{x}}_*$ with

  $$||\mathbf{x}^{(k)} - \tilde{\mathbf{x}}_*|| \leq \frac{\sqrt{2-2q_{GS}}}{\sqrt{L}(1-\sqrt{q_{GS}})} (q_{GS})^{\frac{k}{2}} [F(\mathbf{x}^{(0)}) - F(\tilde{\mathbf{x}}_*)] \tag{2.3.5}$$

  with

  $$q_{GS} = 1 - \frac{\sigma\tau^2}{4lL(1+\tau)^2}. \tag{2.3.6}$$

- *If at iteration k we follow the random block selection strategy we have, under the additional condition that*

  $$\min\{F(x) : ||\mathbf{x} - \mathbf{x}_*|| = \delta\} > F(\mathbf{x}_*) \tag{2.3.7}$$

  holds for some $\delta > 0$, that

  $$\mathbb{E}\{F(\mathbf{x}^{(k)}) - F(\mathbf{x}_*)\} \leq (q_R)^k [F(\mathbf{x}^{(0)}) - F(\mathbf{x}_*)] \tag{2.3.8}$$

  and $\mathbf{x}^{(k)} \to \tilde{\mathbf{x}}_*$ almost surely with

  $$\mathbb{E}||\mathbf{x}^{(k)} - \tilde{\mathbf{x}}_*|| \leq \frac{\sqrt{2-2q_R}}{\sqrt{L}(1-\sqrt{q_R})} (q_R)^{\frac{k}{2}} [F(\mathbf{x}^{(0)}) - F(\tilde{\mathbf{x}}_*)] \tag{2.3.9}$$

  with $q_R = q_{GS}$.

Obs. Lemma 1.1.2 ensures that Pairwise FW and Away-Step FW both satisfy the angle condition (0.0.3).
Furthermore, in Lemma 2.1.2 we proved that SSC applied to AFW and PFW always terminates in a finite number of steps.
This ensures that, if the local KL property is satisfied by the objective function F at $\mathbf{x}_*$, our Block-Coordinate Frank-Wolfe method with SSC described in Algorithm 6 converges linearly in the sense of the theorem. σ-strongly convex functions, for example, satisfy this condition for any point $\mathbf{x}$. In this case, our algorithm converges globally with the rates given in Theorem 2.3.1.

Obs. Notice that the parallel strategy and the Gauss-Southwell one give the same convergence rate.
The random block selection strategy gives in turn the same rate, but in expectation; also, a further assumption on $\mathbf{x}_*$ is needed.
Even if the convergence rate is the same for the three approaches, the constants involved are different: the constant for the GS and the random block selection strategy involves also the number of blocks l. Consequently, the larger the number of blocks,

the worse the rate.
Such constant is larger than the one found for the parallel case, because for $l > 1$ we have

$$\frac{\mu\tau^2}{4lL(1+\tau)^2} < \frac{\mu\tau^2}{4L(1+\tau)^2}$$

and hence

$$q_{GS} = 1 - \frac{\mu\tau^2}{4lL(1+\tau)^2} > 1 - \frac{\mu\tau^2}{4L(1+\tau)^2} = q_P.$$

Chapter 3

Analysis of Stochastic Block-Coordinate Frank-Wolfe method with SSC for product domains

In this last chapter we will describe and study the convergence of the stochastic variants of two kinds of Block-Coordinate Frank-Wolfe algorithms with SSC seen in 2.2.2.
We will prove that such algorithms converge linearly in expectation and almost everywhere using the technique introduced by Golfarb, Iyengar and Zhou in [8], previously seen in Chapter 1.

3.1 Stochastic variants of BCFW with SSC

We are interested in solving an ERM-type problem on a feasible set which takes the form of a product domain. In particular, we consider the following minimization problem

$$\min_{\mathbf{x}\in\mathcal{P}} F(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x}) \tag{3.1.1}$$

where $\mathcal{P} \subset \mathbb{R}^p$ is not only a polytope, but also block separable: it means that

$$\mathcal{P} = \mathcal{P}_1 \times \mathcal{P}_2 \times \cdots \times \mathcal{P}_l$$

with $\mathcal{P}_j \subset \mathbb{R}^{p_j}$ polytope for $j \in \{1, \dots, l\}$ and $\sum\limits_{j=1}^{l} p_j = p$.
We also suppose that for every $i = 1, \dots, n$, $f_i : \mathbb{R}^p \to \mathbb{R}$ is strongly convex with constant σ_i and has Lipschitz continuous gradient with constant L_i.
We combine what we already saw in the first two chapters to find a stochastic version of the Block-Coordinate Frank-Wolfe algorithm with SSC. A possible implementation is shown in Algorithm 7.
The basic structure is similar to Algorithm 2: first, at each iteration we sample $m^{(k)}$ variables ξ_i, and then we build the function $F^{(k)}$ which has to be minimized at iteration k in order to find $\mathbf{x}^{(k+1)}$.
Since we work on a product domain, at each iteration we can pick a suitable set of

blocks and perform SSC only on them, leaving the other variables unchanged.
We already know that inside SSC we can use any first order projection-free method; however, we restrict our analysis to AFW and PFW directions. With this choice, we are sure that SSC terminates in a finite number of steps (see [15] for the full proof).

Algorithm 7 Stochastic BCFW algorithm with SSC

```
   Initialization $\mathbf{x}^{(0)} \in V$, $f_i$, $L_i$ and $k = 0$.
1: for $k = 0, 1, \dots$ do
2:     Sample $\xi_1, \dots, \xi_{m^{(k)}} \overset{\text{i.i.d.}}{\sim} \xi$
3:     Set $F^{(k)}(\mathbf{x}) = \frac{1}{m^{(k)}} \sum_{i=1}^{m^{(k)}} f(\xi_i, \mathbf{x})$ and $L^{(k)} = \frac{1}{m^{(k)}} \sum_{i=1}^{m^{(k)}} L_{\xi_i}$
4:     Choose $\mathcal{M}^{(k)} \subset \{1, \dots, l\}$
5:     For all indexes $j \notin \mathcal{M}^{(k)}$, set $\mathbf{x}_j^{(k+1)} = \mathbf{x}_j^{(k)}$
6:     For all indexes $j \in \mathcal{M}^{(k)}$, set $\mathbf{x}_j^{(k+1)} = \text{SSC}(\mathbf{x}_j^{(k)}, (-\nabla_j F^{(k)}(\mathbf{x}^{(k)}))$
7: end for
```

In particular, if we decide to pick our blocks according to the Jacobi rule (which means $\mathcal{M}^{(k)} = \{1, \dots, l\}$), the procedure becomes the one described in Algorithm 8. We will refer to this variant as Stochastic BCFW with SSC and parallel selection, or Stochastic parallel BCFW with SSC.

Algorithm 8 Stochastic BCFW algorithm with SSC and parallel selection

```
   Initialization $\mathbf{x}^{(0)} \in V$, $f_i$, $L_i$ and $k = 0$.
1: for $k = 0, 1, \dots$ do
2:     Sample $\xi_1, \dots, \xi_{m^{(k)}} \overset{\text{i.i.d.}}{\sim} \xi$
3:     Set $F^{(k)}(\mathbf{x}) = \frac{1}{m^{(k)}} \sum_{i=1}^{m^{(k)}} f(\xi_i, \mathbf{x})$ and $L^{(k)} = \frac{1}{m^{(k)}} \sum_{i=1}^{m^{(k)}} L_{\xi_i}$
4:     for $j = 1, \dots, l$ do
5:         Set $\mathbf{x}_j^{(k+1)} = \text{SSC}(\mathbf{x}_j^{(k)}, -\nabla_j F^{(k)}(\mathbf{x}^{(k)}))$
6:     end for
7: end for
```

If we pick our blocks according to the Gauss-Southwell rule, the procedure becomes the one described in Algorithm 9. We will refer to this variant as Stochastic BCFW with SSC and GS selection, or Stochastic GS BCFW with SSC.

3.2 Convergence analysis

In this section we will prove that the Stochastic parallel BCFW algorithm with SSC and the Stochastic GS BCFW with SSC converge linearly in expectation.
Define $F^{(k)} = \frac{1}{m^{(k)}} \sum_{i=1}^{m^{(k)}} f(\xi_i, \mathbf{x}) = \frac{1}{m^{(k)}} \sum_{i=1}^{m^{(k)}} f_{\xi_i}(\mathbf{x})$, $L^{(k)} = \frac{1}{m^{(k)}} \sum_{i=1}^{m^{(k)}} L_{\xi_i}$ and $\sigma^{(k)} = \frac{1}{m^{(k)}} \sum_{i=1}^{m^{(k)}} \sigma_{\xi_i}$. Recall also that $\mathbf{g}^{(k)} = \nabla_{\mathbf{x}} F^{(k)} = \frac{1}{m^{(k)}} \sum_{i=1}^{m^{(k)}} \nabla f_{\xi_i}(\mathbf{x})$.

First of all, we need some lemmas from [3] which are useful to prove the linear convergence of BCFW with SSC. We can immediately extend their statements and

Algorithm 9 Stochastic BCFW algorithm with SSC and GS selection

```
   Initialization $\mathbf{x}^{(0)} \in V$, $f_i$, $L_i$ and $k = 0$.
1: for $k = 0, 1, \dots$ do
2:     Sample $\xi_1, \dots, \xi_{m^{(k)}} \overset{\text{i.i.d.}}{\sim} \xi$
3:     Set $F^{(k)}(\mathbf{x}^{(k)}) = \frac{1}{m^{(k)}} \sum_{i=1}^{m^{(k)}} f(\xi_i, \mathbf{x}^{(k)})$ and $L^{(k)} = \frac{1}{m^{(k)}} \sum_{i=1}^{m^{(k)}} L_{\xi_i}$
4:     Set $j(k) \in \arg\max_{j \in \{1,\dots,l\}} \langle -\mathbf{g}_j^{(k)}, \text{SSC}(\mathbf{x}_j^{(k)}, -\nabla_j F(\mathbf{x}^{(k)})) - \mathbf{x}_j^{(k)} \rangle$
5:     For all indexes $j \neq j(k)$, set $\mathbf{x}_j^{(k+1)} = \mathbf{x}_j^{(k)}$
6:     For $j = j(k)$, set $\mathbf{x}_j^{(k+1)} = \text{SSC}(\mathbf{x}_j^{(k)}, -\nabla_j F^{(k)}(\mathbf{x}^{(k)}))$
7: end for
```

their proofs to the stochastic case: it is sufficient to replace F with $F^{(k)}$ and $\mathbf{g}$ with $\mathbf{g}^{(k)}$ in the statements, all the procedures remain the same.

Lemma 3.2.1. *Let $\{\mathbf{x}^{(k)}\}$ be a sequence generated by Algorithm 7. For a fixed $j \in \{1, \dots, l\}$, let $\{\mathbf{w}^{(k)}\} = \{\mathbf{x}_j^{(k)}\}$ and let $\mathbf{w}^{(k+1)} = SSC(\mathbf{w}^{(k)}, -\mathbf{g}^{(k)})$. Then there exists $\tilde{\mathbf{w}}^{(k)} \in \{\mathbf{y}^{(t)}\}_{t=0}^{T}$ such that*

$$||\mathbf{w}^{(k+1)} - \mathbf{w}^{(k)}|| \geq \frac{\tau}{L} ||\pi(T_{\mathcal{P}_j}(\tilde{\mathbf{w}}^{(k)}), -\mathbf{g}^{(k)})|| \tag{3.2.1}$$

and

$$\begin{aligned} &||\tilde{\mathbf{w}}^{(k)} - \mathbf{w}^{(k)}|| \leq ||\mathbf{w}^{(k+1)} - \mathbf{w}^{(k)}|| \\ &\langle -\mathbf{g}^{(k)}, \tilde{\mathbf{w}}^{(k)} - \mathbf{w}^{(k)} \rangle \leq \langle -\mathbf{g}^{(k)}, \mathbf{w}^{(k+1)} - \mathbf{w}^{(k)} \rangle. \end{aligned} \tag{3.2.2}$$

We also have

$$L^{(k)} ||\mathbf{y} - \mathbf{w}^{(k)}||^2 \leq \langle -\mathbf{g}^{(k)}, \mathbf{y} - \mathbf{w}^{(k)} \rangle \tag{3.2.3}$$

for $\mathbf{y} \in \{\mathbf{w}^{(k+1)}, \tilde{\mathbf{w}}^{(k)}\}$.

From now on, we denote by $\overline{\text{SSC}}(\mathbf{w}^{(k)}, -\mathbf{g}^{(k)})$ a point $\tilde{\mathbf{w}}^{(k)}$ with the properties stated in Lemma 3.2.1.

Lemma 3.2.2. *Let $\{\mathbf{x}^{(k)}\}$ be a sequence generated by Algorithm 7, and assume that the angle condition holds for the method $\mathcal{A}_j$ with constant τ (same for each block index $j \in \{1, \dots, l\}$). Let $\bar{\mathbf{x}}^{(k)} = [\overline{SSC}(\mathbf{x}_j^{(k)}, -\nabla_j F^{(k)}(\mathbf{x}^{(k)}))]_{j=1}^{l}$ and $\tilde{\mathbf{x}}^{(k+1)} = [SSC(\mathbf{x}_j^{(k)}, -\nabla_j F^{(k)}(\mathbf{x}^{(k)}))]_{j=1}^{l}$.*
If the KL property holds at $\bar{\mathbf{x}}^{(k)}$, we then have, abbreviating $\mathbf{g}^{(k)} = \nabla F^{(k)}(\mathbf{x}^{(k)})$ and defining $F_*^{(k)} = \min_{\mathbf{x} \in \mathcal{P}} F^{(k)}(\mathbf{x})$:

$$\begin{aligned} ||\tilde{\mathbf{x}}^{(k+1)} - \mathbf{x}^{(k)}||^2 &\geq \frac{\tau^2}{2(1+\tau^2)(L^{(k)})^2} ||\pi(T_{\mathcal{P}}(\bar{\mathbf{x}}^{(k)}), -\nabla F^{(k)}(\bar{\mathbf{x}}^{(k)}))||^2 \\ &\geq \frac{\tau^2 \sigma^{(k)}}{(L^{(k)})^2 (1+\tau^2)} [F^{(k)}(\bar{\mathbf{x}}^{(k)}) - F_*^{(k)}] \end{aligned} \tag{3.2.4}$$

and

$$\frac{1}{2} \langle -\mathbf{g}^{(k)}, \tilde{\mathbf{x}}^{(k+1)} - \mathbf{x}^{(k)} \rangle \geq \frac{1}{3} [F^{(k)}(\mathbf{x}^{(k)}) - F^{(k)}(\tilde{\mathbf{x}}^{(k)})] \tag{3.2.5}$$

Lemma 3.2.3. *Let $\{\mathbf{x}^{(k)}\}$ be a sequence generated by Algorithm 7, and assume that the angle condition holds for the method $\mathcal{A}_j$ with constant τ (same for each block index $j \in \{1,\dots,l\}$). Let $\bar{\mathbf{x}}^{(k)} = [\overline{SSC}(\mathbf{x}_j^{(k)}, -\nabla_j F^{(k)}(\mathbf{x}^{(k)}))]_{j=1}^{l}$.*
If the KL property holds at $\bar{\mathbf{x}}^{(k)}$, for parallel updates (like in Algorithm 8)

$$F^{(k)}(\mathbf{x}^{(k)}) - F^{(k)}(\mathbf{x}^{(k+1)}) \geq \frac{\tau^2\sigma^{(k)}}{2L^{(k)}(1+\tau^2)}(F^{(k)}(\bar{\mathbf{x}}^{(k)}) - F_*^{(k)}) \tag{3.2.6}$$

and for GS updates (like in Algorithm 9)

$$F^{(k)}(\mathbf{x}^{(k)}) - F^{(k)}(\mathbf{x}^{(k+1)}) \geq \frac{\tau^2\sigma^{(k)}}{2L^{(k)}(1+\tau^2)}\frac{1}{l}(F^{(k)}(\bar{\mathbf{x}}^{(k)}) - F_*^{(k)}). \tag{3.2.7}$$

Lemma 3.2.4. *Let $\{\mathbf{x}^{(k)}\}$ be a sequence generated by Algorithm 7, and assume that the angle condition holds for the method $\mathcal{A}_j$ with constant τ (same for each block index $j \in \{1,\dots,l\}$). Let $\bar{\mathbf{x}}^{(k)} = [\overline{SSC}(\mathbf{x}_i^{(k)}, -\nabla_j F^{(k)}(\mathbf{x}^{(k)}))]_{j=1}^{l}$. Then for parallel updates (like in Algorithm 8)*

$$F^{(k)}(\mathbf{x}^{(k)}) - F^{(k)}(\mathbf{x}^{(k+1)}) \geq \frac{1}{3}[F^{(k)}(\mathbf{x}^{(k)}) - F^{(k)}(\bar{\mathbf{x}}^{(k)})] \tag{3.2.8}$$

and for GS updates (like in Algorithm 9)

$$F^{(k)}(\mathbf{x}^{(k)}) - F^{(k)}(\mathbf{x}^{(k+1)}) \geq \frac{1}{3l}[F^{(k)}(\mathbf{x}^{(k)}) - F^{(k)}(\bar{\mathbf{x}}^{(k)})]. \tag{3.2.9}$$

All the previous lemmas allow us to find an upper bound for $F^{(k)}(\mathbf{x}^{(k+1)}) - F_*^{(k)}$, from which we are going to start proving the linear convergence in expectation of our method.

Lemma 3.2.5. *Let $\{\mathbf{x}^{(k)}\}$ be a sequence generated by Algorithm 7, and assume that the angle condition holds for the method $\mathcal{A}_j$ with constant τ (same for each block index $j \in \{1,\dots,l\}$). If the KL property holds at $\mathbf{x}^{(k)}$, then:*

- *for parallel updates (like in Algorithm 8)*

$$F^{(k)}(\mathbf{x}^{(k+1)}) - F_*^{(k)} \leq q_P^{(k)}(F^{(k)}(\mathbf{x}^{(k)}) - F_*^{(k)}) \tag{3.2.10}$$

  with $q_P^{(k)} = 1 - \frac{\sigma^{(k)}\tau^2}{4L^{(k)}(1+\tau)^2}$.

- *for GS updates (like in Algorithm 9)*

$$F^{(k)}(\mathbf{x}^{(k+1)}) - F_*^{(k)} \leq q_{GS}^{(k)}(F^{(k)}(\mathbf{x}^{(k)}) - F_*^{(k)}) \tag{3.2.11}$$

  with $q_{GS}^{(k)} = 1 - \frac{\sigma^{(k)}\tau^2}{4lL^{(k)}(1+\tau)^2}$.

Obs. Since $\sigma_i \leq L_i$, then it must be also $\sigma^{(k)} \leq L^{(k)}$ for each iteration k. Also, since $\tau \in (0,1]$ we have

$$0 < \frac{\sigma^{(k)}\tau^2}{2L^{(k)}(1+\tau)^2} \leq \frac{\sigma^{(k)}\tau^2}{2L^{(k)}(1+\tau^2)} \leq \frac{\tau}{3L^{(k)}} \leq \frac{1}{3}$$

similarly to what is done in [3] in the proof of Lemma 3.2.5. Then

$$q_P^{(k)} = 1 - \frac{\sigma^{(k)}\tau^2}{4L^{(k)}(1+\tau)^2} \geq 1 - \frac{1}{6} = \frac{5}{6} \tag{3.2.12}$$

In particular, we get that $0 < q_P^{(k)} < 1$ for each iteration k.
In a similar way, we can prove that

$$q_{GS}^{(k)} \geq 1 - \frac{1}{6l} = \frac{6l-1}{6l} \tag{3.2.13}$$

which tells us that also $0 < q_{GS}^{(k)} < 1$ for each iteration k.

We are now ready to prove the linear convergence of the stochastic variants of the Block-coordinate Frank-Wolfe with SSC described in Algorithm 8 and 9. We will start from the parallel case: the other one will be similar.

Theorem 3.2.6. *Let $\{\mathbf{x}^{(k)}\}$ be the sequence generated by Algorithm 8 for solving problem (3.1.1). Suppose that for every $i = 1, \dots, n$, the functions f_i are strongly convex with constant σ_i and have Lipschitz continuous gradient with constant L_i. Let F^* be the optimal value of the problem. Set $\bar{q}_P = \max_k q_P^{(k)}$ and $m^{(i)} = \lceil 1/(\bar{q}_P)^{4i+4} \rceil$. Assume also that:*

- *angle condition holds for the method $\mathcal{A}_j$ with constant τ (same for each block index $j \in \{1, \dots, l\}$);*
- *the SSC procedure always terminates in a finite number of steps.*

Then for every $k \geq 1$

$$\mathbb{E}\{F(\mathbf{x}^{(k+1)}) - F^*\} \leq \bar{C}(\bar{q}_P)^k \tag{3.2.14}$$

with $\bar{C}$ deterministic constant and $\bar{q}_P = \max_{i=1,\dots,k} q_P^{(i)}$.

Proof. First of all, recall that since every f_i is strongly convex with constant σ_i, then for every $k \geq 1$ the stochastic function $F^{(k)}$ is strongly convex with constant $\sigma^{(k)} = \frac{1}{m^{(k)}} \sum_{i=1}^{m^{(k)}} \sigma_{\xi_i}$. We already noticed that KL property holds globally for strongly convex functions; consequently, all lemmas from 3.2.2 to 3.2.5 can be applied. In particular, inequality (3.2.10) of lemma 3.2.5 holds for each iteration, which means that for every $k \geq 1$ we have

$$F^{(k)}(\mathbf{x}^{(k+1)}) - F_*^{(k)} \leq q_P^{(k)}(F^{(k)}(\mathbf{x}^{(k)}) - F_*^{(k)}).$$

Adding and subtracting appropriately, we get the following chain of inequalities for every $k \geq 1$:

$$\begin{aligned}
&F^{(k)}(\mathbf{x}^{(k+1)}) - F_*^{(k)} \leq q_P^{(k)}[F^{(k)}(\mathbf{x}^{(k)}) - F_*^{(k)}] \\
&= q_P^{(k)}[F^{(k-1)}(\mathbf{x}^{(k)}) - F_*^{(k-1)} + F^{(k)}(\mathbf{x}^{(k)}) - F_*^{(k)} - F^{(k-1)}(\mathbf{x}^{(k)}) + F_*^{(k-1)}] \\
&= \ q_P^{(k)}[F^{(k-1)}(\mathbf{x}^{(k)}) - F_*^{(k-1)}] + q_P^{(k)}[F^{(k)}(\mathbf{x}^{(k)}) - F_*^{(k)} - F^{(k-1)}(\mathbf{x}^{(k)}) + F_*^{(k-1)}] \\
&= q_P^{(k)}[F^{(k-1)}(\mathbf{x}^{(k)}) - F_*^{(k-1)}] + q_P^{(k)}[F^{(k)}(\mathbf{x}^{(k)}) - F(\mathbf{x}^{(k)}) + F(\mathbf{x}^{(k)}) - F^{(k-1)}(\mathbf{x}^{(k)}) \\
&\quad + F^* - F_*^{(k)} + F_*^{(k-1)} - F^*] \\
&\leq q_P^{(k)}[F^{(k-1)}(\mathbf{x}^{(k)}) - F_*^{(k-1)}] + q_P^{(k)}\{|F^{(k)}(\mathbf{x}^{(k)}) - F(\mathbf{x}^{(k)})| + |F^{(k-1)}(\mathbf{x}^{(k)}) - F(\mathbf{x}^{(k)})| \\
&\quad + |F_*^{(k)} - F^*| + |F_*^{(k-1)} - F^*|\} \\
&\leq \prod_{i=1}^{k} q_P^{(i)}[F^{(0)}(\mathbf{x}^{(1)}) - F_*^{(0)}] + \sum_{i=1}^{k}\prod_{j=i}^{k} q_P^{(j)}\{|F^{(i)}(\mathbf{x}^{(i)}) - F(\mathbf{x}^{(i)})| + |F^{(i-1)}(\mathbf{x}^{(i)}) - F(\mathbf{x}^{(i)})| \\
&\quad + |F_*^{(i)} - F^*| + |F_*^{(i-1)} - F^*|\} \\
&\leq \prod_{i=1}^{k} q_P^{(i)}(u_F - l_F) + \sum_{i=1}^{k}\prod_{j=i}^{k} q_P^{(j)}\{|F^{(i)}(\mathbf{x}^{(i)}) - F(\mathbf{x}^{(i)})| + |F^{(i-1)}(\mathbf{x}^{(i)}) - F(\mathbf{x}^{(i)})| + \\
&\quad + |F_*^{(i)} - F^*| + |F_*^{(i-1)} - F^*|\}
\end{aligned}$$

We also have that

$$F^{(k)}(\mathbf{x}^{(k+1)}) - F_*^{(k)} = F(\mathbf{x}^{(k+1)}) - F^* + (F^{(k)}(\mathbf{x}^{(k+1)}) - F(\mathbf{x}^{(k+1)})) + (F^* - F_*^{(k)})$$

So we get

$$\begin{aligned}
F(\mathbf{x}^{(k+1)}) - F^* &= F^{(k)}(\mathbf{x}^{(k+1)}) - F_*^{(k)} - (F^{(k)}(\mathbf{x}^{(k+1)}) - F(\mathbf{x}^{(k+1)})) - (F^* - F_*^{(k)}) \\
&\leq F^{(k)}(\mathbf{x}^{(k+1)}) - F_*^{(k)} + |F^{(k)}(\mathbf{x}^{(k+1)}) - F(\mathbf{x}^{(k+1)})| + |F^* - F_*^{(k)}| \\
&\leq \prod_{i=1}^{k} q_P^{(i)}(u_F - l_F) + \sum_{i=1}^{k+1}\prod_{j=i}^{k} q_P^{(j)}\{|F^{(i)}(\mathbf{x}^{(i)}) - F(\mathbf{x}^{(i)})| \\
&\quad + |F^{(i-1)}(\mathbf{x}^{(i)}) - F(\mathbf{x}^{(i)})| + |F_*^{(i)} - F^*| + |F_*^{(i-1)} - F^*|\}
\end{aligned} \tag{3.2.15}$$

with the convention that a nullary product like $\prod_{j=k+1}^{k} q_P^{(j)}$ is equal to 1.

As in Chapter 1, we can notice that for any iteration k and any deterministic $\mathbf{x} \in \mathcal{P}$, we have $\mathbb{E}\{F^{(k)}(\mathbf{x})\} = F(\mathbf{x})$. In addition, by Corollary 1.3.4, we have the following bounds for every iteration k:

$$\mathbb{E}|F^{(k)}(\mathbf{x}^{(k)}) - F(\mathbf{x}^{(k)})| \leq \mathbb{E}\sup_{\mathbf{x}\in\mathcal{P}}|F^{(k)}(\mathbf{x}) - F(\mathbf{x})| \leq C_1\sqrt{\frac{\log m^{(k)}}{m^{(k)}}}$$

and

$$\mathbb{E}|F_*^{(k)} - F^*| \leq C_1\sqrt{\frac{\log m^{(k)}}{m^{(k)}}}$$

Combining all bounds, we have

$$
\begin{aligned}
\mathbb{E}\{F(\mathbf{x}^{(k+1)}) - F^*\} =& \prod_{i=1}^{k} q_P^{(i)}(u_F - l_F) + \sum_{i=1}^{k+1}\prod_{j=i}^{k} q_P^{(j)}\{\mathbb{E}|F^{(i)}(\mathbf{x}^{(i)}) - F(\mathbf{x}^{(i)})| + \\
& + \mathbb{E}|F^{(i-1)}(\mathbf{x}^{(i)}) - F(\mathbf{x}^{(i)})| + \mathbb{E}|F_*^{(i)} - F^*| + \mathbb{E}|F_*^{(i-1)} - F^*|\} \\
\leq& \prod_{i=1}^{k} q_P^{(i)}(u_F - l_F) + \sum_{i=1}^{k+1}\prod_{j=i}^{k} q_P^{(j)}\{2C_1\sqrt{\frac{\log m^{(i)}}{m^{(i)}}} + 2C_1\sqrt{\frac{\log m^{(i-1)}}{m^{(i-1)}}}\} \\
\leq& \prod_{i=1}^{k} q_P^{(i)}(u_F - l_F) + 4C_1\sum_{i=1}^{k+1}\prod_{j=i}^{k} q_P^{(j)}\sqrt{\frac{\log m^{(i-1)}}{m^{(i-1)}}}
\end{aligned}
$$

where in the last inequality we use the fact that $\frac{\log x}{x}$ is decreasing for $x > e$.
If we define $\bar{q}_P = \max_{i=1,\dots,k} q_P^{(i)}$ and we use $m^{(i)} = \lceil 1/(\bar{q}_P)^{4i+4}\rceil$ we get

$$
\begin{aligned}
\mathbb{E}\{F(\mathbf{x}^{(k+1)}) - F^*\} &\leq (\bar{q}_P)^k(u_F - l_F) + 4C_1\sum_{i=1}^{k+1}\prod_{j=i}^{k}\bar{q}_P\sqrt{\frac{\log m^{(i-1)}}{m^{(i-1)}}} \\
&= (\bar{q}_P)^k(u_F - l_F) + 4C_1\sum_{i=1}^{k+1}(\bar{q}_P)^{k-i+1}2\sqrt{\log\left(\frac{1}{\bar{q}_P}\right)}\,\sqrt{i}\,(\bar{q}_P)^{2i} \\
&= (\bar{q}_P)^k(u_F - l_F) + 8C_1\sum_{i=1}^{k+1}(\bar{q}_P)^{k+1}\sqrt{\log\left(\frac{1}{\bar{q}_P}\right)}\,(\bar{q}_P)^i\sqrt{i} \\
&= (\bar{q}_P)^k\left[(u_F - l_F) + 8C_1\bar{q}_P\sqrt{\log\left(\frac{1}{\bar{q}_P}\right)}\sum_{i=1}^{k+1}(\bar{q}_P)^i\sqrt{i}\right] \\
&\leq \bar{C}(\bar{q}_P)^k
\end{aligned}
$$

for

$$
\begin{aligned}
\bar{C} &= (u_F - l_F) + 8C_1\bar{q}_P\sqrt{\log\left(\frac{1}{\bar{q}_P}\right)}\sum_{i=1}^{k+1}(\bar{q}_P)^i\sqrt{i} \\
&\leq (u_F - l_F) + 8C_1\bar{q}_P\sqrt{\log\left(\frac{1}{\bar{q}_P}\right)}\sum_{i=1}^{k+1}(\bar{q}_P)^i i \\
&\leq (u_F - l_F) + 8C_1\bar{q}_P\sqrt{\log\left(\frac{1}{\bar{q}_P}\right)}\frac{\bar{q}_P}{(1-\bar{q}_P)^2}
\end{aligned}
$$

where in the last inequality we use the fact that $\sum_{k=1}^{+\infty} q^i i = \frac{q}{(1-q)^2}$ for any $|q| < 1$.
Since $0 < \bar{q}_P < 1$ and $\bar{C}$ is a finite constant independent from k, we get our result. $\square$

Obs. If we adopt AFW or PFW directions, we are sure that SSC terminates in a finite number of steps. This means that Theorem 3.2.6 gives us linear convergence in expectation for the Stochastic BCFW with parallel selection.

As we previously saw in Chapter 1, from the linear convergence in expectation we are also able to prove that Algorithm 8 converges linearly almost everywhere. The procedure is almost the same as the one for the ASFW and PSFW algorithms: full proof can be found in [8].

Corollary 3.2.7. *Let $\{\mathbf{x}^{(k)}\}$ be the sequence generated by Algorithm 8 for solving problem (3.1.1). Then*

$$\frac{F(\mathbf{x}^{(k)}) - F^{(*)}}{\omega^k} \to 0$$

almost surely as $k \to +\infty$ for some $\bar{q}_P < \omega < 1$. Therefore $F(\mathbf{x}^{(k)})$ linearly converges to F^* almost surely.

As regards the variant with Gauss-Southwell block selection described in Algorithm 9, the results are similar.

Theorem 3.2.8. *Let $\{\mathbf{x}^{(k)}\}$ be the sequence generated by Algorithm 9 for solving problem (3.1.1). Suppose that for every $i = 1, \dots, n$, the functions f_i are strongly convex with constant σ_i and have Lipschitz continuous gradient with constant L_i. Let F^* be the optimal value of the problem. Set $\bar{q}_{GS} = \max_k q_{GS}^{(k)}$ and $m^{(i)} = \lceil 1/(\bar{q}_{GS})^{4i+4} \rceil$. Assume also that:*

- *angle condition holds for the method $\mathcal{A}_j$ with constant τ (same for each block index $j \in \{1, \dots, l\}$);*
- *the SSC procedure always terminates in a finite number of steps.*

Then for every $k \geq 1$

$$\mathbb{E}\{F(\mathbf{x}^{(k+1)}) - F^*\} \leq \tilde{C}(\bar{q}_{GS})^k \tag{3.2.16}$$

with $\tilde{C}$ deterministic constant and $\bar{q}_{GS} = \max_{i=1,\dots,k} q_{GS}^{(i)}$.

Corollary 3.2.9. *Let $\{\mathbf{x}^{(k)}\}$ be the sequence generated by Algorithm 9 for solving problem (3.1.1). Then*

$$\frac{F(\mathbf{x}^{(k)}) - F^{(*)}}{\omega^k} \to 0$$

almost surely as $k \to +\infty$ for some $\bar{q}_{GS} < \omega < 1$. Therefore $F(\mathbf{x}^{(k)})$ linearly converges to F^* almost surely.

Conclusions

In this thesis, we proposed two new stochastic Frank-Wolfe variants for product domains, starting from the parallel and GS Block-coordinate Frank-Wolfe algorithm with Short Step Chain proposed by Bomze, Rinaldi and Zeffiro.
We proved that these stochastic variants converge linearly in expectation and almost surely under some assumptions (strong convexity, Lipschitz continuous gradient and a tailored angle condition).
For the proof, we took inspiration by the techniques used by Goldfarb, Iyengar and Zhou for the convergence analysis of the Stochastic Away-Step FW and the Pairwise Away-Step FW algorithm. For this purpose we made use of some basic concepts of empirical processes and concentration inequalities, which could be useful to analyze the convergence of other stochastic algorithms.
Future work could include extending these algorithms to problems with non-strongly convex functions and introduce stochastic variance reduced gradients (SVRG) in the procedure.
Besides, it remains to study the convergence of the stochastic variant of the Block-Coordinate Frank-Wolfe algorithm with SSC and with random block selection.

Bibliography

[1] Hédy Attouch et al. “Proximal alternating minimization and projection methods for nonconvex problems: An approach based on the Kurdyka-Łojasiewicz inequality”. In: *Mathematics of operations research* 35.2 (2010), pp. 438–457.

[2] Amir Beck and Shimrit Shtern. “Linearly convergent away-step conditional gradient for non-strongly convex functions”. In: *Mathematical Programming* 164 (2017), pp. 1–27.

[3] Immanuel Bomze, Francesco Rinaldi, and Damiano Zeffiro. “Projection free methods on product domains”. In: *arXiv preprint arXiv:2302.04839* (2023).

[4] Nicolas Boumal. *An introduction to optimization on smooth manifolds.* Cambridge University Press, 2023.

[5] Michael D Canon and Clifton D Cullum. “A tight upper bound on the rate of convergence of Frank-Wolfe algorithm”. In: *SIAM Journal on Control* 6.4 (1968), pp. 509–516.

[6] Marguerite Frank and Philip Wolfe. “An algorithm for quadratic programming”. In: *Naval research logistics quarterly* 3.1-2 (1956), pp. 95–110.

[7] Takumi Fukunaga and Hiroyuki Kasai. “Fast block-coordinate Frank-Wolfe algorithm for semi-relaxed optimal transport”. In: *arXiv preprint arXiv:2103.05857* (2021).

[8] Donald Goldfarb, Garud Iyengar, and Chaoxu Zhou. “Linear convergence of stochastic frank wolfe variants”. In: *Artificial Intelligence and Statistics*. PMLR. 2017, pp. 1066–1074.

[9] Jacques Guélat and Patrice Marcotte. “Some comments on Wolfe’s ‘away step’”. In: *Mathematical Programming* 35.1 (1986), pp. 110–119.

[10] Simon Lacoste-Julien et al. “Block-coordinate Frank-Wolfe optimization for structural SVMs”. In: *International Conference on Machine Learning*. PMLR. 2013, pp. 53–61.

[11] Jean Lafond, Hoi-To Wai, and Eric Moulines. “Convergence analysis of a stochastic projection-free algorithm”. In: *arXiv preprint arXiv:1510.01171* 77 (2015).

[12] Larry J LeBlanc, Edward K Morlok, and William P Pierskalla. “An efficient approach to solving the road network equilibrium traffic assignment problem”. In: *Transportation research* 9.5 (1975), pp. 309–318.

[13] Julie Nutini, Issam Laradji, and Mark Schmidt. “Let’s Make Block Coordinate Descent Converge Faster: Faster Greedy Rules, Message-Passing, Active-Set Complexity, and Superlinear Convergence”. In: *arXiv preprint arXiv:1712.08859* (2017).

[14] Anton Osokin et al. "Minding the gaps for block Frank-Wolfe optimization of structured SVMs". In: *international conference on machine learning*. PMLR. 2016, pp. 593–602.

[15] Francesco Rinaldi and Damiano Zeffiro. "Avoiding bad steps in Frank-Wolfe variants". In: *Computational Optimization and Applications* 84.1 (2023), pp. 225–264.

[16] Yu-Xiang Wang et al. "Parallel and distributed block-coordinate frank-wolfe algorithms". In: *International Conference on Machine Learning*. PMLR. 2016, pp. 1548–1557.

[17] G Zoutendijk and J Abadie. "Integer and nonlinear programming". In: *North-Holland, Amsterdam. INDEX: VOLUMES* 1 (1970), pp. 1–410.