UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

String-averaging incremental subgradient methods for constrained convex optimization problems

Rafael Massambone de Oliveira

Tese de Doutorado do Programa de Pós-Graduação em Ciências de
Computação e Matemática Computacional (PPG-CCMC)

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

Rafael Massambone de Oliveira

String-averaging incremental subgradient methods for constrained convex optimization problems

Doctoral dissertation submitted to the Institute of Mathematics and Computer Sciences – ICMC-USP, in partial fulfillment of the requirements for the degree of the Doctorate Program in Computer Science and Computational Mathematics. *FINAL VERSION*

Concentration Area: Computer Science and Computational Mathematics

Advisor: Prof. Dr. Elias Salomão Helou Neto

USP – São Carlos
September 2017

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados fornecidos pelo(a) autor(a)

Oliveira, Rafael Massambone de
048s String-averaging incremental subgradient methods
for constrained convex optimization problems /
Rafael Massambone de Oliveira; orientador Elias
Salomão Helou Neto. -- São Carlos, 2017.
111 p.

Tese (Doutorado - Programa de Pós-Graduação em
Ciências de Computação e Matemática Computacional) --
Instituto de Ciências Matemáticas e de Computação,
Universidade de São Paulo, 2017.

1. Otimização. 2. Métodos (iterativos,
matemática). 3. Métodos de subgradientes
incrementais. 4. Algoritmos de médias das
sequências (cadeias). I. Helou Neto, Elias Salomão,
orient. II. Título.

Rafael Massambone de Oliveira

**Média das sequências e algoritmos de subgradientes
incrementais para problemas de otimização convexa com
restrições**

Tese apresentada ao Instituto de Ciências
Matemáticas e de Computação – ICMC-USP,
como parte dos requisitos para obtenção do título
de Doutor em Ciências – Ciências de Computação e
Matemática Computacional. *VERSÃO REVISADA*

Área de Concentração: Ciências de Computação e
Matemática Computacional

Orientador: Prof. Dr. Elias Salomão Helou Neto

USP – São Carlos
Setembro de 2017

Dedicated to Reinaldo de Oliveira (in memoriam).

ACKNOWLEDGEMENTS

First of all, I would like to thank my wife Amandha de Oliveira for her patience and support throughout this process.

I thank my mother Fátima Maria Massambone de Oliveira for everything she has always done for me.

I thank my brother Rodolfo Massambone de Oliveira for all the help, especially during some difficult times. Thank you also to all the family members who have helped me in recent months.

I thank my advisor Prof. Elias Salomão Helou Neto and Prof. Eduardo Fontoura Costa for all the knowledge that was transmitted very respectfully.

I thank my friends Vinícius Bertogna and João Pedro Bertogna for all the help they have given me.

I thank all the friends of the Laboratory of Optimization (LOt) for the long conversations and the good moments.

I thank the CeMEAI for all the structure provided to carry out this work.

I thank the FAPESP for the financial support. Grant 2015/10171-2.

ABSTRACT

OLIVEIRA, R. M. **String-averaging incremental subgradient methods for constrained convex optimization problems.** 2017. 111 p. Tese (Doutorado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2017.

In this doctoral thesis, we propose new iterative methods for solving a class of convex optimization problems. In general, we consider problems in which the objective function is composed of a finite sum of convex functions and the set of constraints is, at least, convex and closed. The iterative methods we propose are basically designed through the combination of incremental subgradient methods and string-averaging algorithms. Furthermore, in order to obtain methods able to solve optimization problems with many constraints (and possibly in high dimensions), generally given by convex functions, our analysis includes an operator that calculates approximate projections onto the feasible set, instead of the Euclidean projection. This feature is employed in the two methods we propose; one deterministic and the other stochastic. A convergence analysis is proposed for both methods and numerical experiments are performed in order to verify their applicability, especially in large scale problems.

Keywords: Incremental subgradient methods, string-averaging algorithms, convex optimization, stochastic optimization.

RESUMO

OLIVEIRA, R. M. **Média das sequências e algoritmos de subgradientes incrementais para problemas de otimização convexa com restrições.** 2017. 111 p. Tese (Doutorado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2017.

Nesta tese de doutorado, propomos novos métodos iterativos para a solução de uma classe de problemas de otimização convexa. Em geral, consideramos problemas nos quais a função objetivo é composta por uma soma finita de funções convexas e o conjunto de restrições é, pelo menos, convexo e fechado. Os métodos iterativos que propomos são criados, basicamente, através da junção de métodos de subgradientes incrementais e do algoritmo de média das sequências. Além disso, visando obter métodos flexíveis para soluções de problemas de otimização com muitas restrições (e possivelmente em altas dimensões), dadas em geral por funções convexas, a nossa análise inclui um operador que calcula projeções aproximadas sobre o conjunto viável, no lugar da projeção Euclideana. Essa característica é empregada nos dois métodos que propomos; um determinístico e o outro estocástico. Uma análise de convergência é proposta para ambos os métodos e experimentos numéricos são realizados a fim de verificar a sua aplicabilidade, principalmente em problemas de grande escala.

Palavras-chave: Métodos de subgradientes incrementais, algoritmo de média das sequências, otimização convexa, otimização estocástica.

LIST OF FIGURES

Figure 1 – Subgradient projection method	23
Figure 2 – Incremental subgradient method	33
Figure 3 – Distributed subgradient method	40
Figure 4 – A schematic view of the string-averaging incremental subgradient method	44
Figure 5 – The feasibility operator $\mathcal{V}_X$	54
Figure 6 – Artificial image, projections and sinogram	61
Figure 7 – Shepp-Logan phantom with resolution 256×256	63
Figure 8 – Sinograms from limited data	63
Figure 9 – Objective function, total variation and relative squared error as functions of the computational time in a noise-free condition	66
Figure 10 – Reconstructed Shepp-Logan phantom	68
Figure 11 – Test with 17.8% of relative noise	69
Figure 12 – Test with 8.78% of relative noise	70
Figure 13 – Test with 5.65% of relative noise	71
Figure 14 – Reconstructed images obtained in the tests with Poisson noise	72
Figure 15 – Sinogram obtained by synchrotron radiation illumination of eggs of fishes.	73
Figure 16 – Total variation as function of the residual ℓ_1 -norm for the experiment using eggs of fishes.	73
Figure 17 – Reconstructed images from sinogram given in Figure 15	74
Figure 18 – Profile lines from images in Figure 17.	74
Figure 19 – Performance profiles of the tested methods by using $k_{\max} = 2000$	98
Figure 20 – Performance profiles of the tested methods by using $k_{\max} = 5000$	99
Figure 21 – Computational time of the tested methods by using $k_{\max} = 5000$	99

LIST OF ALGORITHMS

Algorithm 1 – Method of simple dual averages	27
Algorithm 2 – Method of weighted dual averages	27
Algorithm 3 – Nesterov’s method for strongly convex functions and simple convex sets	28
Algorithm 4 – FISTA	29
Algorithm 5 – OSGA	31
Algorithm 6 – Stochastic subgradient mirror-descent method	32
Algorithm 7 – Helou and De Pierro incremental subgradient method	36
Algorithm 8 – String-averaging incremental subgradient algorithm	55
Algorithm 9 – SAISSA	81

LIST OF TABLES

Table 1 – Running time and total variation obtained by the tested method with Poisson relative noise	67
Table 2 – Parameters of the objective function used in tests for SAISSA	94

CONTENTS

1	INTRODUCTION	21
1.1	Organization of this thesis	21
1.2	The problem, literature and related methods	21
1.2.1	<i>Subgradient projection methods</i>	23
1.2.2	<i>Incremental subgradient methods</i>	33
1.2.3	<i>Distributed subgradient methods</i>	39
1.3	The proposed approach	43
2	STRING-AVERAGING INCREMENTAL SUBGRADIENTS FOR CON- STRAINED CONVEX OPTIMIZATION WITH APPLICATIONS TO RECONSTRUCTION OF TOMOGRAPHIC IMAGES	45
2.1	Introduction	45
2.2	Preliminary theory	49
2.3	Proposed algorithm	52
2.4	Convergence analysis	56
2.5	Numerical experiments	60
2.5.1	<i>Experimental work</i>	61
2.5.2	<i>Image reconstruction analysis</i>	65
2.5.3	<i>Tests with real data</i>	69
2.6	Final comments	70
3	STRING-AVERAGING INCREMENTAL STOCHASTIC SUBGRA- DIENT ALGORITHMS	75
3.1	Introduction	75
3.2	Proposed algorithm	79
3.3	Convergence analysis	81
3.3.1	<i>Notation and main hypothesis</i>	81
3.3.2	<i>Auxiliary lemmas</i>	82
3.3.3	<i>Convergence results</i>	88
3.4	Numerical experiments	91
3.4.1	<i>Motivation</i>	92
3.4.2	<i>The numerical experiment</i>	93
3.4.2.1	<i>Experimental setup</i>	93

3.4.2.2	<i>Comparison of the methods</i>	97
3.5	Final comments	100
4	CONCLUSION	101
	BIBLIOGRAPHY	103

INTRODUCTION

1.1 Organization of this thesis

The subject of this doctoral thesis is directly related to the content of two articles: the first one has already been published (see (OLIVEIRA; HELOU; COSTA, 2016)) and is reproduced in Chapter 2; the second one is in preparation for submission, and the current version of the manuscript can be found in Chapter 3. A conclusion is made in Chapter 4.

In the content that follows in this introduction, Section 1.2 defines the general problem of interest and provides a discussion about its solution using iterative methods. For each method, the assumptions made on the problem are identified. We have selected some of the main methods existing in the literature for discussion, but we focus on those that have a certain connection with the approach we propose in section 1.3. We highlight the main features of each method, so that the reader can identify the advantages and disadvantages of its use for solving a specific problem.

We have sought to maintain the format of the articles in full and, as a consequence, each of the chapters of the thesis can be read independently. There are no dependencies on previous chapters. However, to maintain this reading pattern, a few mathematical definitions and/or concepts have been repeated in different chapters.

1.2 The problem, literature and related methods

Throughout this thesis, we deal with the following general optimization problem:

$$\begin{aligned} \min \quad & f(\mathbf{x}) = \sum_{i=1}^m f_i(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{x} \in \mathbf{X} \subseteq \mathbb{R}^n, \end{aligned} \tag{1.1}$$

where $\mathbf{X}$ is a non-empty, closed and convex set and the $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $i \in \{1, 2, \dots, m\}$, are convex functions. Problems that can be modeled as in (1.1) appear frequently in practice and have been studied often in the literature: for instance, we can mention *tomographic image reconstruction problems* (BROWNE; PIERRO, 1996; NETO; PIERRO, 2005; NETO; PIERRO, 2009; OLIVEIRA; HELOU; COSTA, 2016), *assignment problems* (NEDIĆ; BERTSEKAS, 2001b), *least squares problems* (BERTSEKAS, 1997), *distributed optimization problems in wireless sensor networks* (BLATT; HERO; GAUCHMAN, 2007) and *neural network training* (BERTSEKAS; TSITSIKLIS, 1996).

It is common to find in the literature connections involving models such as (1.1) with *multi-agent networks*, where a collection of agents $\mathcal{I} = \{1, \dots, m\}$ cooperate with each other to seek the minimizer of the aggregate cost $\sum_{i=1}^m f_i(\mathbf{x})$, such that each component function f_i is known only to a particular agent i . An important question is: when using an iterative method to solve (1.1), how much information can each agent offer and receive at iteration k ? In other words, how does the network make information available for use and how does the method make use of available information in each iteration? In this sense, issues such as network topology and connectivity can be crucial when choosing a method to solve (1.1).

We dedicate the next section to make a brief review about some of the main methods in the literature able to solve (1.1) giving special attention to optimization problems in multi-agent networks, not only because we want to compare the potential practical applications of the mentioned methods with the approach we propose, but because multi-agent networks involve some intricate assumptions.

Notation: The subdifferential set of a convex function $\phi : \mathbb{R}^n \rightarrow \mathbb{R}$ at $\mathbf{x}$ is defined as

$$\partial\phi(\mathbf{x}) = \{\mathbf{g} \in \mathbb{R}^n \mid \langle \mathbf{g}, \mathbf{z} - \mathbf{x} \rangle \leq \phi(\mathbf{z}) - \phi(\mathbf{x}), \forall \mathbf{z} \in \mathbb{R}^n\}. \quad (1.2)$$

If ϕ is differentiable, then $\partial\phi(\mathbf{x}) = \{\nabla\phi(\mathbf{x})\}$. We denote $\mathcal{P}_{\mathbf{X}}$ as the Euclidean projection onto the non-empty closed and convex set $\mathbf{X}$. We define also

$$\begin{aligned} \phi^* &= \inf_{\mathbf{x} \in \mathbf{X}} \phi(\mathbf{x}) \quad \text{and} \\ \mathbf{X}^* &= \{\mathbf{x} \in \mathbf{X} \mid \phi(\mathbf{x}) = \phi^*\}, \end{aligned}$$

and $d_{\mathbf{X}}(\mathbf{x}) = \min_{\mathbf{y} \in \mathbf{X}} \|\mathbf{x} - \mathbf{y}\|$ denotes the Euclidean distance from a point $\mathbf{x}$ to the closed and convex set $\mathbf{X}$. We will always suppose that $\mathbf{X}^* \neq \emptyset$. Unless it is otherwise mentioned, we will always consider $\|\cdot\|$ as the Euclidean norm in $\mathbb{R}^n$. Denote yet $\|\mathbf{x}\|_* = \sup_{\|\mathbf{y}\| \leq 1} \langle \mathbf{x}, \mathbf{y} \rangle$ as the dual norm associated to $\|\cdot\|$.

1.2.1 Subgradient projection methods

Although not created specifically to solve problems as (1.1), such methods are very popular and have several variants. For this class of methods, we consider that $\mathcal{P}_X$ is simple to compute. Since $\mathbf{g} := \sum_{i=1}^m \mathbf{g}_i \in \partial f(\mathbf{x})$, where $\mathbf{g}_i \in \partial f_i(\mathbf{x})$ (see Theorem 4.1.1 in (HIRIART-URRUTY; LEMARÉCHAL, 1993a)), these methods can make use of $\mathbf{g}$ as an update direction. A standard approach can be found in (BERTSEKAS, 1999; SHOR, 1985; POLYAK, 1987; BERTSEKAS, 2015) and is described as

$$\mathbf{x}^{k+1} = \mathcal{P}_X(\mathbf{x}^k - \lambda_k \mathbf{g}^k), \quad (1.3)$$

where $\mathbf{x}^0 \in \mathbb{R}^n$, $\{\lambda_k\}$ is a sequence of positive step-sizes and $\mathbf{g}^k = \sum_{i=1}^m \mathbf{g}_i^k$ with $\mathbf{g}_i^k \in \partial f_i(\mathbf{x}^k)$. When we use such methods to solve (1.1) in a network optimization context, we implicitly assume the existence of a central coordinator node c that has access to information of all agents in each iteration k , in such a manner that the network topology can be defined from a graph partially connected with nodes $\{1, \dots, m, c\}$. Figure 1 shows a scheme of this situation.

Figure 1 – Iteration k of subgradient projection method applied to problem (1.1).

The next four theorems establish the main convergence results about subgradient projection methods. For this, we will assume the following assumption: for some scalar $C > 0$, we have

$$\sup\{\|\mathbf{g}^k\| \mid k = 0, 1, \dots\} \leq C. \quad (1.4)$$

Theorem 1.1 (Constant step-sizes). Let (1.4) hold and suppose that $\lambda_k := \lambda > 0$.

(i) If $f^* = -\infty$, then $\liminf_{k \rightarrow \infty} f(\mathbf{x}^k) = f^*$.

(ii) If $f^* > -\infty$, then $\liminf_{k \rightarrow \infty} f(\mathbf{x}^k) \leq f^* + \frac{\lambda C^2}{2}$.

(iii) Define $f_K^* = \min_{k=0,\dots,K} f(\mathbf{x}^k)$. Then for any positive scalar ϵ , we have

$$f_{K'}^* \leq f^* + \frac{\lambda C^2 + \epsilon}{2},$$

$$\text{where } K' = \left\lceil \frac{d_{\mathbf{X}^*}^2(\mathbf{x}^0)}{\lambda \epsilon} \right\rceil.$$

Proof. See Propositions 3.2.3 and 3.2.4 in (BERTSEKAS, 2015). $\square$

In a general way, Theorem 1.1 states that we cannot expect convergence to a minimizer without additional assumptions. We may only ensure that we will asymptotically approach a neighborhood of $\mathbf{X}^*$, whose size will depend on λ . Part (iii) gives the number of necessary iterations to guarantee an ϵ -optimality up to the threshold tolerance $\lambda C^2/2$, which depends on the distance of the initial point $\mathbf{x}^0$ to $\mathbf{X}^*$.

The next theorem shows that, with an additional assumption, we can obtain a *linear rate of convergence* to the set of all $\mathbf{x} \in \mathbf{X}$, such that $d_{\mathbf{X}^*}^2(\mathbf{x}) \leq \frac{\lambda C^2}{2\gamma}$, $\gamma > 0$. We recall that a nonnegative scalar sequence $\{a_k\}$ *converges* (at least) *linearly* or *geometrically* if there exist scalars $b > 0$ and $\rho \in (0, 1)$ such that $a_k \leq b\rho^k$ for all k (see (BERTSEKAS, 2015) p. 57).

Theorem 1.2 (Constant step-sizes). If in addition to the assumption described in (1.4) we assume that for some $\gamma > 0$,

$$f(\mathbf{x}) - f^* \geq \gamma d_{\mathbf{X}^*}^2(\mathbf{x}), \quad \forall \mathbf{x} \in \mathbf{X}, \quad (1.5)$$

and that $\lambda \leq 1/2\gamma$, then we have for all k

$$d_{\mathbf{X}^*}^2(\mathbf{x}^{k+1}) \leq (1 - 2\lambda\gamma)^{k+1} d_{\mathbf{X}^*}^2(\mathbf{x}^0) + \frac{\lambda C^2}{2\gamma}.$$

Proof. See Proposition 3.2.5 in (BERTSEKAS, 2015). $\square$

Assumption (1.5) holds, for instance, when f is strongly convex on $\mathbb{R}^n$ or if f is polyhedral and $\mathbf{X}$ is polyhedral and compact (BERTSEKAS, 2015).

An important characteristic of the subgradient projection method (1.3) is that the new iterate may not improve the objective for any value of the step-size. However, if the step-size is sufficiently small, the distance of the current iterate to the optimal solution is reduced. Indeed, using the nonexpansion property of the projection $\mathcal{P}_{\mathbf{X}}$, we

have for all $\mathbf{z} \in \mathbf{X}$ and $k \geq 0$,

$$\begin{aligned} \|\mathbf{x}^{k+1} - \mathbf{z}\|^2 &\leq \|\mathcal{P}_{\mathbf{X}}(\mathbf{x}^k - \lambda_k \mathbf{g}^k) - \mathbf{z}\|^2 \\ &\leq \|\mathbf{x}^k - \lambda_k \mathbf{g}^k - \mathbf{z}\|^2 \\ &\leq \|\mathbf{x}^k - \mathbf{z}\|^2 - 2\lambda_k \langle \mathbf{g}^k, \mathbf{x}^k - \mathbf{z} \rangle + \lambda_k^2 \|\mathbf{g}^k\|^2 \\ &\leq \|\mathbf{x}^k - \mathbf{z}\|^2 - 2\lambda_k (f(\mathbf{x}^k) - f(\mathbf{z})) + \lambda_k^2 \|\mathbf{g}^k\|^2, \end{aligned} \quad (1.6)$$

where the last inequality follows from the subgradient definition. Taking $\mathbf{z} = \mathbf{x}^* \in \mathbf{X}^*$ and choosing

$$\lambda_k < \frac{f(\mathbf{x}^k) - f^*}{\|\mathbf{g}^k\|^2}, \quad (1.7)$$

we have $\|\mathbf{x}^{k+1} - \mathbf{x}^*\| < \|\mathbf{x}^k - \mathbf{x}^*\|$.

Theorem 1.3 (Dynamic step-sizes). If $\{\lambda_k\}$ is determined by the dynamic step-size rule (1.7), then:

- (i) The sequence $\{\mathbf{x}^k\}$ generated by subgradient projection method (1.3) converges to some optimal solution $\mathbf{x}^* \in \mathbf{X}^*$.
- (ii) $\lim_{k \rightarrow \infty} \inf \sqrt{k}(f(\mathbf{x}^k) - f^*) = 0$.
- (iii) If there is $\alpha > 0$ such that $f(\mathbf{x}) - f^* \geq \alpha \|\mathbf{x} - \mathbf{x}^*\|$ for all $\mathbf{x} \in \mathbf{X}$, then

$$\|\mathbf{x}^{k+1} - \mathbf{x}^*\| \leq \rho^k \|\mathbf{x}^0 - \mathbf{x}^*\|, \quad (1.8)$$

where $\rho = \sqrt{1 - \alpha^2/C^2}$ and C is an upper bound of $\|\mathbf{g}^k\|$.

- (iv) $f_k^* - f^* \leq \frac{CD}{\sqrt{k}}$, where D is an upper bound for $\|\mathbf{x}^1 - \mathbf{x}^*\|$, $\mathbf{x} \in \mathbf{X}$.

Proof. For the proof of (i)-(iii) see Theorem 2, p. 142 in (POLYAK, 1987). (iv) By replacing (1.7) in inequality (1.6) and taking $\mathbf{z} = \mathbf{x}^*$ we obtain,

$$\begin{aligned} \|\mathbf{x}^{k+1} - \mathbf{x}^*\|^2 &\leq \|\mathbf{x}^k - \mathbf{x}^*\|^2 - \frac{(f(\mathbf{x}^k) - f^*)^2}{\|\mathbf{g}^k\|^2} \\ &\leq \|\mathbf{x}^1 - \mathbf{x}^*\|^2 - \sum_{j=1}^k \frac{(f(\mathbf{x}^j) - f^*)^2}{\|\mathbf{g}^j\|^2}. \end{aligned}$$

Since $\|\mathbf{x}^{k+1} - \mathbf{x}^*\|^2 \geq 0$, the previous inequality provides

$$\frac{(f_k^* - f^*)^2}{C^2} k = \frac{\left(\min_{j=1, \dots, k} f(\mathbf{x}^j) - f^* \right)^2}{C^2} k \leq \sum_{j=1}^k \frac{(f(\mathbf{x}^j) - f^*)^2}{C^2} \leq \|\mathbf{x}^1 - \mathbf{x}^*\|^2 \leq D^2,$$

proving the result. □

Theorem 1.3 (iii) states that the sequence $\{\|\mathbf{x}^k - \mathbf{x}^*\|\}$ converges linearly¹. Unfortunately, the step-size rule (1.7) requires that we know f^* , which is not available in many practical situations. Alternative dynamic step-size sequences can be obtained by replacing f^* in (1.7) by an approximation (BERTSEKAS, 2015; NEDIĆ; BERTSEKAS, 2001b). Another approach consists in determining a step-size sequence that diminishes to zero, but satisfies $\sum_{k=0}^{\infty} \lambda_k = \infty$.

Theorem 1.4 (Diminishing step-sizes). Let assumption (1.4) hold. If $\{\lambda_k\}$ satisfies $\lambda_k \rightarrow 0$ and $\sum_{k=0}^{\infty} \lambda_k = \infty$, then

$$\liminf_{k \rightarrow \infty} f(\mathbf{x}^k) = f^*.$$

Moreover if $\sum_{k=0}^{\infty} \lambda_k^2 < \infty$, then $\{\mathbf{x}^k\}$ converges to some optimal solution $\mathbf{x}^* \in \mathbf{X}^*$.

Proof. See Proposition 3.2.6 in (BERTSEKAS, 2015). □

Now we pass to discuss some important variants of the subgradient projection method (1.3). For the case when each f_i is non-differentiable, the *primal dual subgradient methods* by Nesterov (NESTEROV, 2009) achieve the optimal rate of convergence $O(1/\sqrt{k})$ (for a description of the lower bounds on the complexity of first order methods in convex optimization, see (NEMIROVSKI; YUDIN, 1983) or (NESTEROV, 2004)). When applied to solve (1.1), the author analyzes two explicit versions: *method of simple dual averages* and *method of weighted dual averages*. Descriptions of these methods can be found in Algorithms 1 and 2. For that, consider a distance-generating function $\omega(\mathbf{x})$, that is, a continuously differentiable function with domain containing $\mathbf{X}$ and strongly convex on $\mathbf{X}$ (with respect to $\|\cdot\|$) with strong convexity parameter $\sigma \geq 0$. Two basic examples are: $\omega(\mathbf{x}) = \frac{1}{2}\|\mathbf{x}\|^2$ with $\|\cdot\|$ being the Euclidean norm and

$$\omega(\mathbf{x}) = \sum_{i=1}^n [\mathbf{x}]_i \ln [\mathbf{x}]_i - \sum_{i=1}^n [\mathbf{x}]_i. \quad (1.9)$$

Define,

$$\pi_{\beta}(\mathbf{y}) = \arg \min_{\mathbf{x} \in \mathbf{X}} \{\beta \omega(\mathbf{x}) - \langle \mathbf{y}, \mathbf{x} \rangle\}.$$

Let us define yet the following sequence:

$$\begin{aligned} \hat{\beta}_0 = \hat{\beta}_1 = 1, \quad \hat{\beta}_{j+1} = \hat{\beta}_j + \frac{1}{\hat{\beta}_j}, \quad j \geq 1 \quad \text{and thus} \\ \hat{\beta}_{k+1} = \sum_{j=0}^k \frac{1}{\hat{\beta}_j}, \quad k \geq 0. \end{aligned}$$

¹ It is common to find expressions such as “the linearly convergent algorithm” when inequalities as in (1.8) hold or when $f(\mathbf{x}^k) - f^* \leq b\rho^k$ for all k (see e.g., (NEUMAIER, 2016; NESTEROV, 2004)).

Algorithm 1: METHOD OF SIMPLE DUAL AVERAGES (NESTEROV, 2009)

input : $\gamma > 0$.
₁ Initialization: Set $\mathbf{y}^0 = \mathbf{0}$.
₂ **for** $k \geq 0$ **do**
₃ Compute $\mathbf{g}^k$ and set $\mathbf{y}^{k+1} = \mathbf{y}^k + \mathbf{g}^k$.
₄ Choose $\beta_{k+1} = \gamma \hat{\beta}_{k+1}$. Set $\mathbf{x}^{k+1} = \pi_{\beta_{k+1}}(-\mathbf{y}^{k+1})$.

Algorithm 2: METHOD OF WEIGHTED DUAL AVERAGES (NESTEROV, 2009)

input : $\rho > 0$.
₁ Initialization: Set $\mathbf{y}^0 = \mathbf{0}$.
₂ **for** $k \geq 0$ **do**
₃ Compute $\mathbf{g}^k$ and set $\mathbf{y}^{k+1} = \mathbf{y}^k + \mathbf{g}^k / \|\mathbf{g}^k\|_*$.
₄ Choose $\beta_{k+1} = \frac{\hat{\beta}_{k+1}}{\rho \sqrt{\sigma}}$. Set $\mathbf{x}^{k+1} = \pi_{\beta_{k+1}}(-\mathbf{y}^{k+1})$.

Algorithms 1 and 2 use implicitly the weights $\lambda_k = 1$ and $\lambda_k = 1/\|\mathbf{g}^k\|_*$, respectively. Defining,

$$\hat{\mathbf{x}}^{k+1} = \frac{\sum_{j=0}^k \lambda_j \mathbf{x}^j}{\sum_{j=0}^k \lambda_j} \quad \text{and} \quad f_L^* = \min\{f(\mathbf{x}) \mid \mathbf{x} \in \mathbf{X} \text{ and } \omega(\mathbf{x}) \leq L\},$$

and assuming that there is $C_* > 0$ such that

$$\|\mathbf{g}\|_* \leq C_* \quad \forall \mathbf{g} \in \partial f(\mathbf{x}), \quad \forall \mathbf{x} \in \mathbf{X},$$

then the following estimate holds to Algorithm 1:

$$f(\hat{\mathbf{x}}^{k+1}) - f_L^* \leq \frac{0.5 + \sqrt{2k+1}}{k+1} \left(\gamma L + \frac{C_*^2}{2\sigma\gamma} \right).$$

If it is possible to estimate C_* and L , then we can use the optimal choice of γ :

$$\gamma^* = C_* / \sqrt{2\sigma L}.$$

For Algorithm 2, the following estimate holds:

$$f(\hat{\mathbf{x}}^{k+1}) - f_L^* \leq \frac{0.5 + \sqrt{2k+1}}{(k+1)\sqrt{\sigma}} C_* \left(\frac{1}{\rho} L + \frac{\rho}{2} \right).$$

But now, in order to choose a reasonable value of ρ , we need a reasonable estimate only for L . The optimal choice of ρ is as follows:

$$\rho^* = \sqrt{2L}.$$

When $f = \sum_{i=1}^m f_i$ is a strongly convex function with Lipschitz continuous gradients and associated Lipschitz constant $\mathcal{L}$, the method (2.2.19) in (NESTEROV,

Algorithm 3: NESTEROV'S $\mathcal{L}$ -GRADIENT METHOD FOR μ -STRONGLY CONVEX FUNCTIONS AND SIMPLE SETS $\mathbf{X}$ (NESTEROV, 2004)

input : $\mathbf{x}^0 \in \mathbb{R}^n$, $\alpha_0 \in (0, 1)$, the strong convex parameter $\mu \geq 0$ and the Lipschitz constant $\mathcal{L} > 0$ of ∇f .

1 Initialization: Set $\mathbf{y}^0 = \mathbf{x}^0$ and $q = \frac{\mu}{\mathcal{L}}$.

2 **for** $k \geq 0$ **do**

3 Compute $f(\mathbf{y}^k)$, $\nabla f(\mathbf{y}^k)$ and set

$$\mathbf{x}^{k+1} = \arg \min_{\mathbf{x} \in \mathbf{X}} \{f(\mathbf{y}^k) + \langle \nabla f(\mathbf{y}^k), \mathbf{x} - \mathbf{y}^k \rangle + \frac{\mathcal{L}}{2} \|\mathbf{x} - \mathbf{y}^k\|^2\}. \quad (1.10)$$

4 Compute $\alpha_{k+1} \in (0, 1)$ from equation

$$\alpha_{k+1}^2 = (1 - \alpha_{k+1})\alpha_k^2 + q\alpha_{k+1},$$

 and set $\zeta_{k+1} = \frac{\alpha_k(1-\alpha_k)}{\alpha_k^2 + \alpha_{k+1}}$.

5 Set

$$\mathbf{y}^{k+1} = \mathbf{x}^{k+1} + \zeta_k(\mathbf{x}^{k+1} - \mathbf{x}^k).$$

2004) is linearly convergent. See a description of the method in Algorithm 3. This method achieves the following estimate (Theorem 2.2.3 in (NESTEROV, 2004)): if $\alpha_0 \geq \sqrt{q}$, then

$$f(\mathbf{x}^k) - f^* \leq \min \left\{ (1 - \sqrt{q})^k, \frac{4\mathcal{L}}{(2\sqrt{\mathcal{L}} + k\sqrt{\zeta_0})^2} \right\} (f(\mathbf{x}^0) - f^* + \frac{\zeta_0}{2} \|\mathbf{x}^0 - \mathbf{x}^*\|^2), \quad (1.11)$$

where $\zeta_0 = \frac{\alpha_0(\alpha_0\mathcal{L} - \mu)}{1 - \alpha_0}$.

It is clear that the strong convexity hypothesis is stronger than the one made for problem (1.1). However, when $\mu = 0$ (and thus we are considering that f is convex with Lipschitz continuous gradients), then (1.11) ensures that the function values $f(\mathbf{x}^k) - f^*$ are more accurate than $O(k^{-2})$ after k iterations. This method can be seen as a variant of the subgradient (in this case the subgradient is a gradient) projection method in the sense that the subgradient is computed at an intermediary point $\mathbf{y}^k$ (easily computable) and is used to obtain $\mathbf{x}^{k+1}$ in equation (1.10) by minimizing a quadratic approximation of f at $\mathbf{y}^k$. Furthermore, if $\mathbf{X} = \mathbb{R}^n$ we can note that for all $\mathbf{z} \in \mathbb{R}^n$ and $\eta > 0$,

$$\arg \min_{\mathbf{x} \in \mathbb{R}^n} \{f(\mathbf{z}) + \langle \nabla f(\mathbf{z}), \mathbf{x} - \mathbf{z} \rangle + \frac{\eta}{2} \|\mathbf{x} - \mathbf{z}\|^2\} = \mathbf{z} - \frac{1}{\eta} \nabla f(\mathbf{z}),$$

that is in the basic form of iteration (1.3) with a constant step-size $\frac{1}{\eta}$.

To the next approach, we consider the problem (1.1) with: $\mathbf{X} = \mathbb{R}^n$ and $m = 2$, i.e., $f(\mathbf{x}) = f_1(\mathbf{x}) + f_2(\mathbf{x})$, where f_1 is convex and continuously differentiable with Lipschitz continuous gradients and f_2 is convex (possibly nonsmooth). To solve this

problem, a popular algorithm called FISTA (Fast Iterative Shrinkage-Thresholding Algorithm) (BECK; TBOULLE, 2009b) stands out by achieving an $O(1/k^2)$ complexity result which is “optimal” for smooth problems in the sense of Nemirovsky and Yudin (NEMIROVSKI; YUDIN, 1983). The description of this method can be viewed in the Algorithm 4 below.

Algorithm 4: FAST ITERATIVE SHRINKAGE-THRESHOLDING ALGORITHM (FISTA) (BECK; TBOULLE, 2009b)

input : A Lipschitz constant $\mathcal{L} > 0$ of ∇f_1 and $\mathbf{x}^0 \in \mathbb{R}^n$.
 1 Initialization: Set $\mathbf{y}^1 = \mathbf{x}^0$ and $t_1 = 1$.
 2 **for** $k \geq 1$ **do**
 3
$$\mathbf{x}^k = \arg \min_{\mathbf{x} \in \mathbb{R}^n} \{f_1(\mathbf{y}^k) + \langle \mathbf{x} - \mathbf{y}^k, \nabla f_1(\mathbf{y}^k) \rangle + \frac{\mathcal{L}}{2} \|\mathbf{x} - \mathbf{y}^k\|^2 + f_2(\mathbf{x})\}, \quad (1.12)$$

 4
$$t_{k+1} = \frac{1 + \sqrt{1 + 4t_k^2}}{2},$$

 5
$$\mathbf{y}^{k+1} = \mathbf{x}^k + \left(\frac{t_k - 1}{t_{k+1}} \right) (\mathbf{x}^k - \mathbf{x}^{k-1}).$$

This method is useful when (1.12) can be computed analytically or by a low cost scheme. For instance, when $f_2(\mathbf{x}) = \varphi \|\mathbf{x}\|_1$, $\varphi > 0$, (used as a regularizer in many image restauration problems) then

$$\mathbf{x}^k = \mathcal{T}_{\varphi/\mathcal{L}}(\mathbf{y}^k - \frac{1}{\mathcal{L}} \nabla f_1(\mathbf{y}^k)),$$

where $\mathcal{T}_\alpha : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is the *shrinkage operator* given by (see e.g., (COMBETTES; PESQUET, 2011)):

$$[\mathcal{T}_\alpha(\mathbf{x})]_j = [[\mathbf{x}]_j - \alpha]_+ \operatorname{sgn}([\mathbf{x}]_j), \quad j = 1, \dots, n,$$

where $[x]_+$ denotes the positive part of $x \in \mathbb{R}$ and $\operatorname{sgn}(\cdot)$ is the sign function. In general, Algorithm 4 can be used to solve constrained problems ($\mathbf{X} \subset \mathbb{R}^n$ is closed and convex), but the computation of (1.12) might require intensive computation, unless $\mathbf{X}$ is simple. Theorem 4.4 in (BECK; TBOULLE, 2009b) provides the following complexity result for FISTA:

$$f(\mathbf{x}^k) - f^* \leq \frac{2\mathcal{L}\|\mathbf{x}^0 - \mathbf{x}^*\|^2}{(k+1)^2}, \quad \text{for any } k \geq 1 \quad \text{and } \forall \mathbf{x}^* \in \mathbf{X}^*.$$

A possible drawback of Algorithms 3 and 4 is that the Lipschitz constant $\mathcal{L}$ is not always easy to estimate. However, a backtracking scheme can be incorporated in FISTA, in such a manner knowledge if the Lipschitz constant is not necessary. Both methods belong to the class of the *proximal-type algorithms*, which always need more

information than just subgradients in each iteration. Furthermore, if $\mathbf{X}$ is not simple, then (1.10) and (1.12) can demand high computational cost. In chapters 2 and 3, the reader will note that we focus in algorithms that can be applied in large scale convex optimization problems, such those where $\mathbf{X}$ is, for instance, the union of many sub-level sets of convex functions.

An important adaptive proximal-type subgradient method is analyzed in (NEUMAIER, 2016). The algorithm is based on monotonically reducing bounds on the error $f(\mathbf{x}^k) - f^*$, at the “currently best point” $\mathbf{x}^k$ ($\mathbf{x}^0 \in \mathbf{X}$), through the scheme that follows. Recall that ω is a distance-generating function, that is, continuously differentiable and strongly convex (with strong convex parameter σ) over $\mathbf{X}$. Define $E : \mathbb{R} \times \mathbb{R}^n \rightarrow \mathbb{R}$ as

$$E(\gamma, \mathbf{y}) = - \inf_{\mathbf{x} \in \mathbf{X}} \frac{\gamma + \langle \mathbf{y}, \mathbf{x} \rangle}{\omega(\mathbf{x})}. \quad (1.13)$$

The method requires that $E(\gamma, \mathbf{y})$ be reached for each $\gamma \in \mathbb{R}$ and $\mathbf{y} \in \mathbb{R}^n$ at some $\mathbf{z} := \mathbf{z}(\gamma, \mathbf{y}) \in \mathbf{X}$. Furthermore, the method assumes that if f is strongly convex, we may know a number $\mu > 0$ such that $f - \mu\omega$ is still convex. Proposition 1.1.2 in (HIRIART-URRUTY; LEMARÉCHAL, 1993a) shows that if $\mu > 0$ is the strong convexity constant of f , then this assumption holds for $\omega(\mathbf{x}) = \frac{1}{2}\|\mathbf{x}\|^2$. Thus, with this assumption being satisfied for the chosen function ω , the subgradient definition (1.2) provides for all $\mathbf{x}, \mathbf{v} \in \mathbf{X}$

$$f(\mathbf{x}) - \mu\omega(\mathbf{x}) \geq f(\mathbf{v}) - \mu\omega(\mathbf{v}) + \langle \mathbf{g} - \mu\nabla\omega(\mathbf{v}), \mathbf{x} - \mathbf{v} \rangle, \quad (1.14)$$

where $\mathbf{g} \in \partial f(\mathbf{v})$. Note that (1.14) allows us to construct relaxations of the form

$$f(\mathbf{x}) \geq \gamma + \langle \mathbf{y}, \mathbf{x} \rangle + \mu\omega(\mathbf{x}), \quad \forall \mathbf{x} \in \mathbf{X}. \quad (1.15)$$

Indeed, taking $\mathbf{v} = \mathbf{x}^0 \in \mathbf{X}$ and replacing it in (1.14), then (1.15) holds with

$$\mathbf{y} = \mathbf{y}^0 := \mathbf{g}^0 - \mu\nabla\omega(\mathbf{x}^0) \quad \text{and} \quad \gamma = \gamma_0 := f(\mathbf{x}^0) - \mu\omega(\mathbf{x}^0) - \langle \mathbf{y}^0, \mathbf{x}^0 \rangle,$$

where $\mathbf{g}^0 \in \partial f(\mathbf{x}^0)$. Now, given $\bar{\mathbf{x}} \in \mathbf{X}$ and $\lambda \in [0, 1]$, define

$$\bar{\mathbf{y}} = \mathbf{y} + \lambda(\bar{\mathbf{g}}_{\mu, \omega} - \mathbf{y}) \quad \text{and} \quad \bar{\gamma} = \gamma + \lambda(f(\bar{\mathbf{x}}) - \mu\omega(\bar{\mathbf{x}}) - \langle \bar{\mathbf{g}}_{\mu, \omega}, \bar{\mathbf{x}} \rangle - \gamma), \quad (1.16)$$

where $\bar{\mathbf{g}}_{\mu, \omega} = \bar{\mathbf{g}} - \nabla\omega(\bar{\mathbf{x}})$ and $\bar{\mathbf{g}} \in \partial f(\bar{\mathbf{x}})$. Proposition 3.2 in (NEUMAIER, 2016) ensures that if $f - \mu\omega$ is convex and (1.15) holds for given $\mathbf{y} \in \mathbb{R}^n$ and $\gamma \in \mathbb{R}$, then (1.15) also holds with $\bar{\mathbf{y}}$ and $\bar{\gamma}$ defined in (1.16). Proposition 3.3 in (NEUMAIER, 2016) shows that given $\mathbf{x} \in \mathbf{X}$, if $\gamma \in \mathbb{R}$ and $\mathbf{y} \in \mathbb{R}^n$ satisfy (1.15),

$$\hat{\gamma} := \gamma - f(\mathbf{x}) \quad \text{and} \quad \eta := E(\hat{\gamma}, \mathbf{y}) - \mu, \quad (1.17)$$

then (1.15) provides

$$0 \leq f(\mathbf{x}) - f^* \leq \eta\omega(\mathbf{x}^*). \quad (1.18)$$

This result follows basically from (1.13) for $\mathbf{z} = \mathbf{x}^*$ and by noticing that $\inf_{\mathbf{x} \in \mathbf{X}} \omega(\mathbf{x}) > 0$. The method consists of finding, in each iteration k , an appropriate $\mathbf{x}^k$ such that a strong upper bound on the error fact η can be achieved. In this way, the update is done by moving from the best current point $\mathbf{x}^k$ into the direction of the point $\mathbf{z}^k = \mathbf{z}(\gamma_k, f(\mathbf{x}^k))$ used to determine the error bound η with (1.13) (possibly calculated by using (1.16)). The measure of the progress on the reduction of the error in (1.18) is computed in terms of the quantity $R = (\eta - \bar{\eta}) / \alpha \lambda \eta$, $\alpha \in (0, 1)$, where $\bar{\eta}$ is the error factor tentatively modified in the current iteration. The method is detailed in Algorithm 5.

Algorithm 5: NEUMAIER'S OPTIMAL SUBGRADIENT ALGORITHM (OSGA) (NEUMAIER, 2016)

input : $\mu \geq 0$. Global tuning parameters: $\alpha \in (0, 1)$, $\lambda_{max} \in (0, 1)$, $\kappa > 0$ and $\kappa' \in (0, \kappa]$.

- 1 Initialization: Choose $\mathbf{x}^0 \in \mathbf{X}$. Set
- 2 $\mathbf{y}^0 = \mathbf{g}^0 - \mu \nabla \omega(\mathbf{x}^0)$.
- 3 $\gamma_0 = f(\mathbf{x}^0) - \mu \omega(\mathbf{x}^0) - \langle \mathbf{y}^0, \mathbf{x}^0 \rangle$.
- 4 $\mathbf{z}^0 = \mathbf{z}(\gamma_0 - f(\mathbf{x}^0), \mathbf{y}^0)$; $\eta = E(\gamma_0 - f(\mathbf{x}^0), \mathbf{y}^0) - \mu$.
- 5 $\lambda = \lambda_{max}$.
- 6 **for** $k \geq 0$ **do**
- 7 Compute: $\bar{\mathbf{x}}^k = \mathbf{x}^k + \lambda(\mathbf{z}^k - \mathbf{x}^k)$; $\bar{\mathbf{g}}_{\mu, \omega}^k = \bar{\mathbf{g}}^k - \mu \nabla \omega(\bar{\mathbf{x}}^k)$, $\bar{\mathbf{g}}^k \in \partial f(\bar{\mathbf{x}}^k)$;
- 8 $\bar{\mathbf{y}}^k = \mathbf{y}^k + \lambda(\bar{\mathbf{g}}_{\mu, \omega}^k - \mathbf{y}^k)$; $\bar{\gamma}_k = \gamma_k + \lambda(f(\bar{\mathbf{x}}^k) - \mu \omega(\bar{\mathbf{x}}^k) - \langle \bar{\mathbf{g}}_{\mu, \omega}^k, \bar{\mathbf{x}}^k \rangle - \gamma_k)$;
- 9 $\hat{\mathbf{x}}^k = \arg \min_{\mathbf{r} \in \{\mathbf{x}^k, \bar{\mathbf{x}}^k\}} f(\mathbf{r})$; $\hat{\gamma}_k = \bar{\gamma}_k - f(\hat{\mathbf{x}}^k)$; $\hat{\mathbf{z}}^k = \mathbf{z}(\hat{\gamma}_k, \bar{\mathbf{y}}^k)$;
- 10 $\mathbf{w}^k = \mathbf{x}^k + \lambda(\hat{\mathbf{z}}^k - \mathbf{x}^k)$;
- 11 Choose $\bar{\mathbf{w}}^k$ with $f(\bar{\mathbf{w}}^k) \leq \min(f(\hat{\mathbf{x}}^k), f(\mathbf{w}^k))$;
- 12 $\tilde{\gamma}_k = \bar{\gamma}_k - f(\bar{\mathbf{w}}^k)$; $\bar{\mathbf{z}}^k = \mathbf{z}(\tilde{\gamma}_k, \bar{\mathbf{y}}^k)$; $\bar{\eta} = E(\tilde{\gamma}_k, \bar{\mathbf{y}}^k) - \mu$;
- 13 $\mathbf{x}^{k+1} = \bar{\mathbf{w}}^k$;
- 14 $R = \frac{1}{\alpha \lambda} \frac{(\eta - \bar{\eta})}{\eta}$;
- 15 **if** $R < 1$ **then**
- 16 $\bar{\lambda} = \lambda e^{-\kappa}$;
- 17 **else**
- 18 $\bar{\lambda} = \min(\lambda e^{\kappa'(R-1)}, \lambda_{max})$;
- 19 $\lambda = \bar{\lambda}$;
- 20 **if** $\bar{\eta} < \eta$ **then**
- 21 $\mathbf{y}^{k+1} = \bar{\mathbf{y}}^k$; $\gamma_{k+1} = \bar{\gamma}_k$; $\mathbf{z}^{k+1} = \bar{\mathbf{z}}^k$; $\eta = \bar{\eta}$;
- 22 **else**
- 23 $\mathbf{y}^{k+1} = \mathbf{y}^0$, $\gamma_{k+1} = \gamma_0$, $\mathbf{z}^{k+1} = \mathbf{z}^0$;

Theorem 5.1 in (NEUMAIER, 2016) shows that, asymptotically, the number of iterations needed by the OSGA matches the lower bounds on the complexity given in (NEMIROVSKI; YUDIN, 1983). If f is nonsmooth and Lipschitz continuous in $\mathbf{X}$, the total number of iterations needed for OSGA to reach a point with $f(\mathbf{x}^k) - f^* \leq \epsilon$, $\epsilon > 0$, is at most $O((\epsilon^2 + \mu\epsilon)^{-1})$. If f is smooth and has Lipschitz continuous gradients

with Lipschitz constant $\mathcal{L}$, the same precision is attained in at most $O(\epsilon^{-1/2})$ steps if $\mu = 0$ and at most $O(|\log \epsilon| \sqrt{\mathcal{L}/\mu})$ if $\mu > 0$. However, as in Algorithms 3 and 4, the method requires more than the computation of one subgradient by iteration (which is considered in (NEMIROVSKI; YUDIN, 1983) and sometimes called first-order black box oracle (NESTEROV, 2004)). Each iteration of OSGA requires, at least, the computation of two function values $f(\bar{\mathbf{x}}^k)$ and $f(\mathbf{w}^k)$ and one subgradient $\bar{\mathbf{g}}^k$.

To finalize the discussion of variants of the subgradient projection method (1.3), we will consider a convex stochastic problem. Suppose that

$$f_i(\mathbf{x}) = \mathbb{E}[F_i(\mathbf{x}, \epsilon)], \quad \forall i = 1, \dots, m, \quad (1.19)$$

where ϵ is a random vector and the expectation $\mathbb{E}[\cdot]$ is well defined and finite valued for every $\mathbf{x} \in \mathbf{X}$. For each $i = 1, \dots, m$, $F_i(\cdot, \epsilon)$ is convex for all ϵ and hence, it follows that $f_i(\cdot)$ is convex. An important approach that is able to solve (1.1) with (1.19) is the *stochastic subgradient mirror-descent method* (NEMIROVSKI; YUDIN, 1983; NEMIROVSKI *et al.*, 2009; NEDIĆ; LEE, 2014). See a description in Algorithm 6.

Algorithm 6: STOCHASTIC SUBGRADIENT MIRROR-DESCENT METHOD

input : $\mathbf{x}^0 \in \mathbf{X}$ and $\{\lambda_k\}$.
1 for $k \geq 0$ **do**
2 Get a sample ϵ^k of the random variable ϵ .
3 Compute

$$\mathbf{x}^{k+1} = \arg \min_{\mathbf{z} \in \mathbf{X}} \{ \lambda_k \langle \bar{\mathbf{g}}^k, \mathbf{z} - \mathbf{x}^k \rangle + \omega(\mathbf{z}) - [\omega(\mathbf{x}^k) + \langle \nabla \omega(\mathbf{x}^k), \mathbf{z} - \mathbf{x}^k \rangle] \}, \quad (1.20)$$

 where $\bar{\mathbf{g}}^k$ is a subgradient of $\sum_{i=1}^m F_i(\mathbf{x}, \epsilon^k)$ at $\mathbf{x} = \mathbf{x}^k$.

Important contributions can be found in (NEMIROVSKI *et al.*, 2009) and (NEDIĆ; LEE, 2014). Both consider analyzing bounds for $\mathbb{E}[f(\tilde{\mathbf{x}}^k) - f^*]$ with $\{\tilde{\mathbf{x}}^k\}$ being the sequence of weighted-averages of the iterates in the form $\tilde{\mathbf{x}}^k = \sum_{j=0}^k \Lambda_j \mathbf{x}^j$, where $\Lambda_0, \Lambda_1, \dots, \Lambda_k$ are non-negative scalars with sum equal to 1, generally defined in terms of the step-size values $\lambda_0, \lambda_1, \dots, \lambda_k$. In (NEDIĆ; LEE, 2014) different step-sizes are considered by analyzing two distinct cases: (i) f is strongly convex with strong convexity parameter $\mu > 0$ and $\mathbf{X}$ is closed and convex; (ii) $\mathbf{X}$ is compact and convex (f is just convex on $\mathbb{R}^n$). For (i), equation (1.20) is replaced by

$$\mathbf{x}^{k+1} = \arg \min_{\mathbf{z} \in \mathbf{X}} \left\{ \frac{\lambda_k}{\mu} \langle \bar{\mathbf{g}}^k, \mathbf{z} - \mathbf{x}^k \rangle + \omega(\mathbf{z}) - [\omega(\mathbf{x}^k) + \langle \nabla \omega(\mathbf{x}^k), \mathbf{z} - \mathbf{x}^k \rangle] \right\},$$

and for the step-sizes two specific choices are considered: one of them is

$$\lambda_0 = 1 \quad \text{and} \quad \lambda_k = \frac{2}{k+1}, \quad \forall k \geq 1,$$

and the other one is $\lambda_k = \frac{1}{t_{k+1}}$ where t_{k+1} is the same used in FISTA (see Algorithm 4) ($t_0 = 1$). The weighted-average points are defined as follows:

$$\tilde{\mathbf{x}}^k = \frac{1}{\sum_{j=0}^k \frac{1}{\lambda_j}} \sum_{j=0}^k \frac{1}{\lambda_j} \mathbf{x}^j, \quad \forall k \geq 0.$$

When $\omega(\mathbf{z}) - [\omega(\mathbf{x}) + \langle \nabla \omega(\mathbf{x}), \mathbf{z} - \mathbf{x} \rangle] \leq \frac{1}{2} \|\mathbf{x} - \mathbf{z}\|^2$, (that holds, for example, if $\omega(\mathbf{x}) = \frac{1}{2} \|\mathbf{x}\|^2$ with $\|\cdot\|$ being the Euclidean norm) then the method gives a rate of convergence of $O(1/k)$ (for any of the step-sizes defined above). For (ii), choosing $\lambda_k = a/\sqrt{k+1}$ with $a > 0$, the method (as the method by (NEMIROVSKI *et al.*, 2009)) provides a rate of convergence of $O(1/\sqrt{k})$.

1.2.2 Incremental subgradient methods

The first occurrence of incremental subgradient methods seems to be due to Kibardin (KIBARDIN, 1979), where (1.1) is considered in the unconstrained case. Years later, Solodov and Zavriev (SOLODOV; ZAVRIEV, 1998) analyzed the convergence of the method for the constrained problem (with $\mathbf{X}$ convex and compact) when bounded perturbations are present at the subgradient computations. To solve (1.1), a well-known approach to the incremental subgradient method, due to Nedić and Bertsekas (NEDIĆ; BERTSEKAS, 2001b), can be described by

$$\mathbf{x}_0^k = \mathbf{x}^k, \tag{1.21}$$

$$\mathbf{x}_i^k = \mathcal{P}_{\mathbf{X}}(\mathbf{x}_{i-1}^k - \lambda_k \mathbf{g}_i^k), \quad i = 1, \dots, m, \quad \mathbf{g}_i^k \in \partial f_i(\mathbf{x}_{i-1}^k), \tag{1.22}$$

$$\mathbf{x}^{k+1} = \mathbf{x}_m^k. \tag{1.23}$$

In these methods, no central coordinator node is required, since cyclically the information that each agent contains is passed from one to the other, until all agents have collaborated to update $\mathbf{x}^k$. Therefore, at every instant $t = km + i$, $k = 0, 1, \dots$, the information of the agent i must be available for use. Figure 2 illustrates this situation.

Figure 2 – Iteration k of the incremental subgradient method.

The next two propositions provide the main convergence results about the method (1.21)-(1.23), for constant and diminishing step-size rules. For that, we assume the following: there exists a scalar C_i such that

$$\|\mathbf{g}\| \leq C_i, \quad \forall \mathbf{g} \in \partial f_i(\mathbf{x}^k) \cup \partial f_i(\mathbf{x}_{i-1}^k), \quad i = 1, \dots, m, \quad k = 0, 1, \dots, \quad (1.24)$$

and $C = \sum_{i=1}^m C_i$.

Proposition 1.5 (Constant step-sizes). Let (1.24) hold and let $\{\mathbf{x}^k\}$ be a sequence generated by the method (1.21)-(1.23) with $\lambda_k = \lambda > 0$.

(i) If $f^* = -\infty$, then $\lim_{k \rightarrow \infty} \inf f(\mathbf{x}^k) = -\infty$. If f^* is finite, then

$$\liminf_{k \rightarrow \infty} f(\mathbf{x}^k) \leq f^* + \frac{\lambda m^2 C^2}{2}.$$

(ii) Given $\epsilon > 0$ we have

$$\min_{k=0, \dots, K} f(\mathbf{x}^k) \leq f^* + \frac{\lambda m^2 C^2 + \epsilon}{2}$$

where

$$K = \left\lceil \frac{d_{\mathbf{x}^*}^2(\mathbf{x}^0)}{\lambda \epsilon} \right\rceil.$$

(iii) Suppose there exists a positive scalar γ such that (1.5) holds. Then, if $\lambda \leq \frac{1}{2\gamma}$, we have

$$d_{\mathbf{x}^*}^2(\mathbf{x}^k) \leq (1 - 2\lambda\gamma)^k d_{\mathbf{x}^*}^2(\mathbf{x}^0) + \frac{\lambda m^2 C^2}{2\gamma}.$$

Proof. (i) See Proposition 2.1 in (NEDIĆ; BERTSEKAS, 2001b). (ii) and (iii) See Propositions 2.3 and 2.4 in (NEDIĆ; BERTSEKAS, 2001a). $\square$

As the subgradient method (1.3), Proposition 1.5 shows that if a constant step-size is used, only convergence to a neighborhood of the optimum can be guaranteed.

Proposition 1.6 (Diminishing step-sizes). Let (1.24) hold and suppose that

$$\lambda_k \rightarrow 0_+, \quad \sum_{k=0}^{\infty} \lambda_k = \infty.$$

For the sequence $\{\mathbf{x}^k\}$ generated by (1.21)-(1.23) we have:

(i) $\lim_{k \rightarrow \infty} \inf f(\mathbf{x}^k) = f^*$.

(ii) If $\mathbf{X}^*$ is compact, then

$$\lim_{k \rightarrow \infty} d_{\mathbf{X}^*}(\mathbf{x}^k) = 0, \quad \lim_{k \rightarrow \infty} f(\mathbf{x}^k) = f^*.$$

(iii) If in addition we have $\sum_{k=0}^{\infty} \lambda_k^2 < \infty$, then $\mathbf{x}^k \rightarrow \mathbf{x}^*$ for some $\mathbf{x}^* \in \mathbf{X}^*$.

(iv) If for some $\gamma > 0$, (1.5) holds and $\lambda_k = \frac{R}{k+1}$ where $R > 0$, then

$$\begin{cases} d_{\mathbf{x}^*}^2(\mathbf{x}^{k+1}) \leq \frac{1}{(k+2)^p} \left(d_{\mathbf{x}^*}^2(\mathbf{x}^0) + 2^p (RmC)^2 \frac{2-p}{1-p} \right) & \text{if } p \in (0,1), \\ d_{\mathbf{x}^*}^2(\mathbf{x}^{k+1}) \leq \frac{1+\ln(k+1)}{k+1} (RmC)^2 & \text{if } p = 1, \\ d_{\mathbf{x}^*}^2(\mathbf{x}^{k+1}) \leq \frac{1}{(p-1)(k+2)} \left((RmC)^2 + \frac{(p-1)d_{\mathbf{x}^*}^2(\mathbf{x}^0) - (RmC)^2}{(k+2)^{p-1}} \right) & \text{if } p > 1, \end{cases}$$

where $p = 2\gamma R$.

Proof. For (i)-(iii) see Propositions 2.2, 2.3 and 2.4 in (NEDIĆ; BERTSEKAS, 2001b), respectively. For (iv) see Proposition 2.8 in (NEDIĆ; BERTSEKAS, 2001a). $\square$

When f^* is known, if the variant of the step-size sequence (1.7)

$$\lambda_k = \alpha_k \frac{f(\mathbf{x}^k) - f^*}{m^2 C^2}, \quad 0 < \underline{\alpha} \leq \alpha_k \leq \bar{\alpha} < 2, \quad \forall k \geq 0,$$

is used, then the method (1.21)-(1.23) converges to some optimal solution (see (NEDIĆ; BERTSEKAS, 2001b)) such that, the number of iterations K required for achieving $\min_{k=0,\dots,K} f(\mathbf{x}^k) - f^* \leq \epsilon$ is $O(\epsilon^{-2})$ (NEDIĆ; BERTSEKAS, 2001a). Furthermore, if (1.5) holds, then $d_{\mathbf{x}^*}(\mathbf{x}^k) \rightarrow 0$ linearly. When f^* is unknown, the *path-based incremental target level algorithm* in (NEDIĆ; BERTSEKAS, 2001b) considers step-sizes of the form:

$$\lambda_k = \alpha_k \frac{f(\mathbf{x}^k) - f_k^{lev}}{C^2}, \quad 0 < \underline{\alpha} \leq \alpha_k \leq \bar{\alpha} < 2, \quad \forall k \geq 0,$$

where f_k^{lev} is an estimate of f^* . The algorithm guarantees that $f_k^{lev} \rightarrow f^*$ and $\inf_{k \geq 0} f(\mathbf{x}^k) = f^*$.

Let us now discuss some variants of the incremental subgradient method (1.21)-(1.23). In particular, one is due to Helou and De Pierro in (NETO; PIERRO, 2009). The analyzed methods (see Sections 3.1 and 3.2 in (NETO; PIERRO, 2009)) fit into a general framework, in which several other methods (incremental or not) can be included. In each iteration of the algorithm, two operations are performed: an *optimality step*, which, at least approximately, directs the iterate toward the minimizer of objective function, followed by a *feasibility step* that drives the iteration in the direction of feasibility.

For concreteness, we consider an instance of the method proposed in (NETO; PIERRO, 2009) where the optimality operator provides, at iteration k , a vector $\mathbf{x}^{k+1/2}$ (a subiteration) by performing (1.21)-(1.23) (the authors also analyzed another optimality operator called aggregated incremental subgradient operator), the feasible set $\mathbf{X} = \bigcap_{j=1}^t \{\mathbf{x} \mid h_j(\mathbf{x}) \leq 0\}$ where $h_j : \mathbb{R}^n \rightarrow \mathbb{R}$ are convex for all $j = 1, \dots, t$ (t is finite) and the feasibility operator performs sequential applications of the *subgradient projections*

(BAUSCHKE; COMBETTES; KRUK, 2006; COMBETTES, 1997; YAMADA; OGURA, 2005a; YAMADA; OGURA, 2005b). Algorithm 7 gives a complete description of this method. The set $\mathbf{Y}$ must be chosen such that projection onto $\mathbf{Y}$ in (1.26) is simple.

Algorithm 7: HELOU AND DE PIERRO INCREMENTAL SUBGRADIENT METHOD

input : $\mathbf{x}^0 \in \mathbf{Y} \supset \mathbf{X}$, $\{\lambda_k\}$, $\sigma \in (0, 1)$, $\{v_1, \dots, v_t\}$ such that $v_j \in (\sigma, 2 - \sigma)$ for all j .

1 **for** $k \geq 0$ **do**

2 Obtain the subiteration $\mathbf{x}^{k+1/2}$ by computing

$$\mathbf{x}_0^k = \mathbf{x}^k, \quad (1.25)$$

$$\mathbf{x}_i^k = \mathcal{P}_{\mathbf{Y}}(\mathbf{x}_{i-1}^k - \lambda_k \mathbf{g}_i^k), \quad i = 1, \dots, m, \quad (1.26)$$

$$\mathbf{x}^{k+1/2} = \mathbf{x}_m^k. \quad (1.27)$$

3 Set $\mathbf{y}_0^k = \mathbf{x}^{k+1/2}$.

4 **for** $j = 1, \dots, t$ **do**

5
$$\mathbf{y}_j^k = \begin{cases} \mathbf{y}_{j-1}^k - v_j \frac{(h_j(\mathbf{y}_{j-1}^k))_+}{\|\mathbf{h}_j^k\|^2} \mathbf{h}_j^k, & (\mathbf{h}_j^k \in \partial h_j(\mathbf{y}_{j-1}^k)) \text{ if } \mathbf{h}_j^k \neq \mathbf{0}, \\ \mathbf{y}_{j-1}^k, & \text{otherwise.} \end{cases} \quad (1.28)$$

6 Set $\mathbf{x}^{k+1} = \mathbf{y}_t^k$.

Equations (1.25)-(1.27) represent the optimality step, whereas (1.28) represents the feasibility step. Convergence results can be obtained for all classes of methods that fit into the analyzed framework (including Algorithm 7) if we assume some general assumptions on the optimality and feasibility operators (see Assumptions 1-4 and Theorem 2.5 in (NETO; PIERRO, 2009) or Theorem 2.1 in Chapter 2). The algorithm was applied to a large scale optimization problem: *tomographic image reconstruction from limited data*.

In (RAM; NEDIĆ; VEERAVALLI, 2009) the authors consider that only stochastic subgradients are available. This situation can occur, for instance, when the component functions are as in (1.19). Once the deterministic subgradients are replaced by stochastic subgradients $\tilde{\mathbf{g}}_i$ (e.g., $\tilde{\mathbf{g}}_i \in \partial_{\mathbf{x}} F_i(\mathbf{x}, \epsilon)$), then we can consider subgradients with stochastic errors of form $\tilde{\mathbf{g}}_i = \mathbf{g}_i + \epsilon_i$. The proposed method is identical to (1.21)-(1.23), with the exception that $\mathbf{g}_i + \epsilon_i$ is used instead of $\mathbf{g}_i$. Convergence results are provided for constant and diminishing step-sizes by considering boundedness assumptions on the subdifferentials $\partial f_i(\mathbf{x})$ ($\mathbf{x} \in \mathbf{X}$) and on the second moments of the subgradient errors. In a similar fashion to the ordinary subgradient method (1.3) and the incremental subgradient method (1.21)-(1.23), we cannot guarantee the convergence of the iterates for constant step-sizes. However, the authors provide bounds for $\lim_{k \rightarrow \infty} \inf \mathbb{E}[f(\mathbf{x}^k)] -$

f^* and $\inf_{k \geq 0} f(\mathbf{x}^k) - f^*$ that hold with probability 1. For diminishing step-sizes with $\sum_{k=0}^{\infty} \lambda_k = \infty$, $\sum_{k=0}^{\infty} \lambda_k^2 < \infty$ and an additional assumption involving λ_k and the bounds for the second moments of the subgradient errors, it was proved that $\{\mathbf{x}^k\}$ converges to some $\mathbf{x}^* \in \mathbf{X}^*$ with probability 1.

The incremental subgradient methods discussed above have a *cyclic* structure, which can be seen by equation (1.22), where the information is passed agent-by-agent for all $i \in \mathcal{I}$ in a ring format, as seen in Figure 2. Thus, $\mathbf{x}^{k+1} = \mathbf{x}_m^k$ (equation (1.23)) represents the end of the cycle (or “iteration”) k , developed through of m “subiterations” given by (1.22). We can discard the subiterations-type format present in the method (1.21)-(1.23), in order to obtain a simpler form given by

$$\mathbf{x}^{k+1} = \mathcal{P}_{\mathbf{X}}(\mathbf{x}^k - \lambda_k \mathbf{g}_{i_k}^k), \quad \mathbf{g}_{i_k}^k \in \partial f_{i_k}(\mathbf{x}^k). \quad (1.29)$$

Making $i_k = i \in \mathcal{I}$ for each $k = tm + i - 1$, where $t \in \mathbb{N}$, then (1.29) reduces to (1.21) - (1.23) for $k = m, 2m, 3m, \dots$, i.e., m iterations of (1.29) correspond to one iteration of (1.21)-(1.23).

This remark has inspired a significant amount of *non-cyclic* methods (NEDIĆ; BERTSEKAS, 2001b; JOHANSSON; RABI; JOHANSSON, 2010; RAM; NEDIĆ; VEER-AVALI, 2009; BERTSEKAS, 2011), where the order of processing is not necessarily equal to described above. In (NEDIĆ; BERTSEKAS, 2001b), a randomized version of the incremental subgradient method was analyzed by imposing that i_k is chosen through a random variable ω_k with uniform distribution in $\mathcal{I}$. An improvement in the rate of convergence is clearly noted in the numerical experiments carried out, which considered an instance of a dual version of the assignment problem. In a context of multi-agent networks, it is highly desirable that all agents are available to update the information received in an iteration k , since the probability that an agent is chosen in a given iteration is the same ($1/m$) for all agents.

A generalization of the randomized incremental subgradient method proposed in (NEDIĆ; BERTSEKAS, 2001b), named *Markovian incremental subgradient method*, was made in (JOHANSSON; RABI; JOHANSSON, 2010). In this paper, i_k is chosen through a *time-homogeneous Markov chain* that is irreducible, aperiodic and its stationary distribution is uniform. It is clear that the cyclic and randomized methods described above can be reformulated from the use of a Markov chain, simply by requiring that the transition probability matrix P of the chain be

$$P = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ 1 & 0 & 0 & \cdots & 0 \end{bmatrix} \quad \text{and} \quad P = \begin{bmatrix} \frac{1}{m} & \frac{1}{m} & \cdots & \frac{1}{m} \\ \frac{1}{m} & \frac{1}{m} & \cdots & \frac{1}{m} \\ \vdots & \vdots & \cdots & \vdots \\ \frac{1}{m} & \frac{1}{m} & \cdots & \frac{1}{m} \end{bmatrix}$$

respectively. The authors consider a probability transition matrix such that the iterate $\mathbf{x}^k$ only jumps to an adjacent node ($[P]_{i,j} = 0$ if $(i,j) \notin E$, where E is the set of links in the graph $G(\mathcal{I}, E)$). For that, they used the so-called *Metropolis-Hastings scheme*. In some practical situations, this type of communication can be advantageous. In wireless sensor networks, it is a well-known fact that the amount of energy consumed for a single wireless communication can be quite significant. In general, the amount of energy required for a single hop communication between two nodes i and j , through the link (i,j) , depends (among other things) on the distance between them (see, e.g., (RABBAT; NOWAK, 2004)). Therefore, the total amount of energy needed for multi-hop communications in a wireless sensor network can be a drawback when we assume the topology imposed by the randomized incremental subgradient algorithm, since the method allows hop communications between nodes placed in a long distance, at all iterations.

In (JOHANSSON; RABI; JOHANSSON, 2010), upper bounds are provided for the expected number of iterations needed to reach some accuracy for the cyclic (1.29), randomized and Markovian incremental subgradient methods by using constant step-sizes. In this case, the randomized method stands out and the comparisons between the cyclic and Markovian incremental subgradient methods depend on the topology of the network as well as the transition probability matrix of the Markov chain. However, in (JOHANSSON; RABI; JOHANSSON, 2007; JOHANSSON; CARRETTI; JOHANSSON, 2008), numerical experiments show that the Markovian incremental subgradient method has favorable properties regarding the ease of implementation and energy consumption in wireless sensor networks.

Another important generalization of incremental subgradient methods can be found in (RAM; NEDIĆ; VEERAVALLI, 2009). The authors generalize the Markov incremental subgradient method proposed in (JOHANSSON; RABI; JOHANSSON, 2010) by using a non time-homogeneous Markov chain (with similar properties) and stochastic errors in the subgradient computation. Convergence results and error bounds are provided for diminishing and constant step-size rules, respectively.

To finalize, an *incremental proximal subgradient method* was proposed in (BERTSEKAS, 2011), for functions f_i of the form

$$f_i(\mathbf{x}) = g_i(\mathbf{x}) + h_i(\mathbf{x}), \quad g_i, h_i : \mathbb{R}^n \rightarrow \mathbb{R},$$

where g_i and h_i are convex functions for all $i \in \mathcal{I}$. One of the proposed methods has the form

$$\begin{aligned} \mathbf{z}^k &= \arg \min_{\mathbf{x} \in \mathbf{X}} \left\{ g_{i_k}(\mathbf{x}) + \frac{1}{2\lambda_k} \|\mathbf{x} - \mathbf{x}^k\|^2 \right\}, \\ \mathbf{x}^{k+1} &= \mathcal{P}_{\mathbf{X}}(\mathbf{z}^k - \lambda_k \mathbf{h}_{i_k}^k), \quad \mathbf{h}_{i_k}^k \in \partial h_{i_k}(\mathbf{z}^k). \end{aligned}$$

Convergence results and error bounds were provided for cyclic and randomized orders of processing. These algorithms are well-suited for problems in which there is flexibility to separate the component functions into parts, that can be conveniently handled by proximal iterations (possibly in closed forms) and the remaining parts to be handled by subgradient iterations.

1.2.3 Distributed subgradient methods

This approach is of special interest in the analysis of large-scale networks, which consist of multiple agents with different objectives. Each agent generates and maintains estimates of the optimal solution based on information concerning his own cost function f_i , and on exchanges of these estimates directly or indirectly with the other agents in the network. In (NEDIĆ; OZDAGLAR, 2009), the authors focus on the distributed control of a network consisting of m agents over a time-varying topology, where each agent processes his/her local information and shares the information with his/her neighbors. The global objective is defined as in (1.1) for $\mathbf{X} = \mathbb{R}^n$. They suppose that the agents update and send information from and to neighbors at discrete times $t_0, t_1, t_2, \dots$. The neighbors of an agent i in the k -th iteration are the agents j able to communicate through a directed link (j, i) . The vector $\mathbf{x}_i^k$ denotes the information state of agent i at time t_k . They assume yet that each agent i has a vector of weights $\mathbf{a}_i^k \in \mathbb{R}^m$ at any time t_k . For each j , the scalar $a_{i,j}^k$ is the weight that agent i assigns to $\mathbf{x}_j$ obtained from a neighboring agent j , when the information is received during the time interval (t_k, t_{k+1}) . Notice that this statement allows asynchronous algorithmic versions, since at an instant t_k , it is possible that agent i receives an information $\mathbf{x}_j^r$ from agent j such that $r < k$, i.e., agent i receives the information with a delay $k - r$. See an illustrative example in Figure 3.

The method proposed in (NEDIĆ; OZDAGLAR, 2009), when no communication delay is assumed (the method is synchronous), can be described by

$$\mathbf{x}_i^{k+1} = \sum_{j=1}^m a_{i,j}^k \mathbf{x}_j^k - \lambda_i^k \mathbf{g}_i^k, \quad \mathbf{g}_i^k \in \partial f_i(\mathbf{x}_i^k), \quad (1.30)$$

where $\lambda_i^k > 0$ is a stepsize used by agent i .

The distributed subgradient method (1.30) can be seen as a generalization of the *consensus algorithm*, by considering $f_i = 0$ for all $i = 1, \dots, m$, i.e., the method solves the problem in which each agent must find a common vector $\mathbf{x} \in \mathbb{R}^n$ (a common decision or agreement) only through a sequence of local estimate updates and local information exchanges (see (JADBABAIE; LIN; MORSE, 2003) and (BLONDEL *et al.*, 2005)). When each agent i has constraints that can be represented by a convex set $\mathbf{X}_i$, then this problem is reduced, for each agent, to a convex feasibility problem, i.e., after a fixed

Figure 3 – Distributed subgradient method for a 7-agent network and links configuration in the time interval (t_k, t_{k+1}) . Note that $r \leq k$ may occur. We highlight the update iteration of the agents 1 and 5. Possible choices for $\mathbf{a}_1^k$ and $\mathbf{a}_5^k$ could be: $\mathbf{a}_1^k = [0.3 \ 0.25 \ 0.2 \ 0.25 \ 0 \ 0 \ 0]^T$ and $\mathbf{a}_5^k = [0 \ 0 \ 0.2 \ 0.15 \ 0.4 \ 0.25 \ 0]^T$. Updates $\mathbf{x}_1^{k+1}$ and $\mathbf{x}_5^{k+1}$ can be obtained by (1.30).

time t_k , each agent i must produce $\mathbf{x}_i^k \approx \mathbf{x} \in \cap_{i=1}^m \mathbf{X}_i$. In (NEDIĆ; OZDAGLAR; PARRILO, 2010), this problem was dealt with setting $f_i(\mathbf{x}) = \frac{1}{2} \|\mathbf{x} - \mathcal{P}_{\mathbf{X}_i}(\mathbf{x})\|^2$ and $\mathbf{X} = \mathbb{R}^n$ in (1.1). Making $\mathbf{w}_i^k = \sum_{j=1}^m a_{i,j}^k \mathbf{x}_j^k$, the analyzed algorithm was $\mathbf{x}_i^{k+1} = \mathbf{w}_i^k - \nabla f_i(\mathbf{w}_i^k)$, which can be interpreted as a distributed gradient algorithm. A consensus algorithm that considers the presence of delays can be found in (NEDIĆ; OZDAGLAR, 2010).

Important convergence results for distributed subgradient methods can be derived from weighted averages of $\mathbf{x}_i^k$ for each i . For (1.30), error bounds of the objective function are analyzed over the following vector:

$$\hat{\mathbf{x}}_i^k = \frac{1}{k} \sum_{l=0}^{k-1} \mathbf{x}_i^l, \quad \forall i = 1, \dots, m.$$

The main convergence result is obtained from verification of the Assumptions 1-8 in (NEDIĆ; OZDAGLAR, 2009): they refer to *rules over the weights, connectivity, intercommunication interval, information exchange, subgradient boundedness and optimal solution set*. We highlight two of them, because they are important for understanding the results in the next propositions.

- *Weights rule*: there is $\eta \in (0, 1)$ such that $a_{i,i}^k \geq \eta$ for all $k \geq 0$; $a_{i,j}^k \geq \eta$ for all $k \geq 0$ and all agents j communicating directly with agent i in the interval (t_k, t_{k+1}) , otherwise, $a_{i,j}^k = 0$; for all i and k , $\sum_{j=1}^m a_{i,j}^k = 1$.
- *Bounded intercommunication interval*: Define $E_k = \{(j, i) \mid a_{i,j}^k > 0\}$. There exists a scalar Q such that the graph $(\mathcal{I}, \cup_{l=1}^Q E_{k+l})$ is strongly connected for all k .

Proposition 1.7. Let Assumptions 1-8 in (NEDIĆ; OZDAGLAR, 2009) hold. Let $\mathbf{x}_i^0$ be the initial vector of agent i and suppose that $\max_{i=1,\dots,m} \|\mathbf{x}_i^0\| \leq \lambda C$, where C is an upper bound of the subgradients (Assumption 7) and the constant λ is a positive step-size. Then

$$f(\hat{\mathbf{x}}_i^k) - f^* \leq \frac{md_{\mathbf{x}^*}^2(\frac{1}{m} \sum_{j=1}^m \mathbf{x}_j^0)}{2\lambda k} + \lambda C, \quad \forall k \geq 1, \quad (1.31)$$

where,

$$C = \frac{C_{subg}^2(1 + 8mC_{net})}{2} + 2mC_{subg}\bar{C}_{subg}C_{net}, \quad C_{net} = 1 + \frac{m}{1 - (1 - \eta^{Q_0})^{1/Q_0}} \frac{1 + \eta^{-Q_0}}{1 - \eta^{Q_0}},$$

$Q_0 = (m - 1)Q$, C_{subg} is an upper bound of the subgradients of f_j at $\mathbf{x}_i^k$ and $\bar{C}_{subg}$ is an upper bound of the subgradients of f_j at $\hat{\mathbf{x}}_i^k$.

Proof. See Proposition 3 (b) in (NEDIĆ; OZDAGLAR, 2009). $\square$

Proposition above shows that there is a trade-off between choosing a larger λ and obtaining faster convergence (to a neighborhood of the optimum) and selecting a smaller λ in order to obtain convergence to a better neighborhood. Diminishing step-size rules were considered, e.g., in (RAM; NEDIĆ; VEERAVALLI, 2010; JAKOVETIĆ; XAVIER; MOURA, 2014; LOBEL; OZDAGLAR, 2008).

For our work, the *distributed stochastic subgradient projection algorithm* described in (RAM; NEDIĆ; VEERAVALLI, 2010) deserves to be noted. This is because, as in the incremental stochastic subgradient method discussed in (RAM; NEDIĆ; VEERAVALLI, 2009), stochastic errors are assumed in the subgradient computation at each iteration. This situation will also be considered in the formulation of the method that we will present in Chapter 3. Furthermore, it is suitable for solving the general problem (1.1). In this method, each agent i generates its iterate sequence $\mathbf{x}_i^k$ according to the following relation:

$$\mathbf{x}_i^{k+1} = \mathcal{P}_{\mathbf{X}} \left[\sum_{j=1}^m a_{i,j}^k \mathbf{x}_j^k - \lambda_k (\bar{\mathbf{g}}_i^k + \boldsymbol{\epsilon}_i^{k+1}) \right], \quad \bar{\mathbf{g}}_i^k \in \partial f_i \left(\sum_{j=1}^m a_{i,j}^k \mathbf{x}_j^k \right), \quad (1.32)$$

where $\boldsymbol{\epsilon}_i^{k+1}$ is the stochastic error in the subgradient evaluation of the agent i . The main convergence results of this method can be seen in the next two propositions.

Proposition 1.8 (Convergence in mean). Suppose that the norm of the subgradient errors has uniformly bounded second moments, i.e., there are scalars ν_i such that

$$\mathbb{E}[\|\boldsymbol{\epsilon}_i^{k+1}\|^2] \leq \nu_i^2, \quad \forall i \in \mathcal{I} \quad \text{and} \quad k \geq 0. \quad (1.33)$$

In addition, suppose that the assumptions on weights and intercommunication interval made above hold. Then:

- (i) If $\lambda_k \rightarrow 0$, then $\lim_{k \rightarrow \infty} \mathbb{E}[\|\mathbf{x}_i^k - \mathbf{y}^k\|] = 0$ for all $i \in \mathcal{I}$, where $\mathbf{y}^k = \frac{1}{m} \sum_{i=1}^m \mathbf{x}_i^k$.
- (ii) If $\mathbf{X}$ is bounded and $\lambda_k \rightarrow \bar{\lambda} \geq 0$ with $\sum_{k=0}^{\infty} \lambda_k = \infty$ when $\bar{\lambda} = 0$ then, for all $i \in \mathcal{I}$,
- $$\liminf_{k \rightarrow \infty} \mathbb{E}[f(\mathbf{x}_i^k)] \leq f^* + \max_{\mathbf{x}, \mathbf{y} \in \mathbf{X}} \|\mathbf{x} - \mathbf{y}\| \sum_{j=1}^m \mu_j + m\bar{\lambda} (\max_{j \in \mathcal{I}} \{C_j + \nu_j\})^2 \left(\frac{9}{2} + \frac{2m\theta\beta}{1-\beta} \right),$$
- where $\mu_j = \lim_{k \rightarrow \infty} \sup \|\mathbb{E}[\epsilon_j^{k+1}]\|$, C_j is an upper-bound on the subgradient norms of f_j over the set $\mathbf{X}$, $\theta = \left(1 - \frac{\eta}{4m^2}\right)^{-2}$ and $\beta = \left(1 - \frac{\eta}{4m^2}\right)^{1/Q}$.
- (iii) Define $\mathbf{z}_i^k = \frac{\sum_{l=1}^k \lambda_l \mathbf{x}_i^l}{\sum_{l=1}^k \lambda_l}$. Let the same conditions from (ii) hold. Then, we have for all $i \in \mathcal{I}$ and $k \geq 1$,

$$\limsup_{k \rightarrow \infty} \mathbb{E}[f(\mathbf{z}_i^k)] \leq f^* + \max_{\mathbf{x}, \mathbf{y} \in \mathbf{X}} \|\mathbf{x} - \mathbf{y}\| \sum_{j=1}^m \mu_j + m\bar{\lambda} (\max_{j \in \mathcal{I}} \{C_j + \nu_j\})^2 \left(\frac{9}{2} + \frac{2m\theta\beta}{1-\beta} \right).$$

Proof. See Corollary 5.2 and Propositions 5.3 and 5.4 in (RAM; NEDIĆ; VEERAVALLI, 2010). $\square$

Proposition 1.8 (i) states that there is an asymptotic consensus in mean. In (ii), notice that $\lim_{k \rightarrow \infty} \sup \|\mathbb{E}[\epsilon_j^{k+1}]\| < \infty$ due to (1.33) and by Jensen's inequality. Notice that, when $\|\mathbb{E}[\epsilon_j^{k+1}]\| \rightarrow 0$ and $\lambda_k \rightarrow 0$, (ii) and (iii) yield respectively

$$\liminf_{k \rightarrow \infty} \mathbb{E}[f(\mathbf{x}_i^k)] = f^* \quad \text{and} \quad \lim_{k \rightarrow \infty} \mathbb{E}[f(\mathbf{z}_i^k)] = f^*.$$

The next proposition provides sufficient conditions over the step-sizes and expected error so that agents reach a consensus and the iterate sequences of agents converge to a common optimal point with probability 1 and in mean square.

Proposition 1.9 (Almost sure and mean square convergence). Consider that there are scalars ν_i such that

$$\mathbb{E}[\|\epsilon_i^{k+1}\|^2 | F_k] \leq \nu_i^2, \quad \forall i \in \mathcal{I} \quad \text{and} \quad \forall k \geq 0, \quad (1.34)$$

hold with probability 1, where F_k denotes the σ -algebra generated by the errors in the agent system up to iteration k . Suppose that the assumptions on weights and intercommunication interval hold. Suppose yet that $\|\mathbf{g}_i\| \leq C_i$, $\mathbf{g}_i \in \partial f_i(\mathbf{x})$, for all $\mathbf{x} \in \mathbf{X}$ and $\mathbf{y}^k = \frac{1}{m} \sum_{i=1}^m \mathbf{x}_i^k$.

- (i) If $\sum_{k=0}^{\infty} \lambda_k^2 < \infty$, then for all $i \in \mathcal{I}$ we have

$$\lim_{k \rightarrow \infty} \|\mathbf{y}^k - \mathbf{x}_i^k\| = 0$$

with probability 1 and in mean square, i.e.,

$$\lim_{k \rightarrow \infty} \mathbb{E}[\|\mathbf{y}^k - \mathbf{x}_i^k\|^2] = 0.$$

- (ii) If $\sum_{k=0}^{\infty} \lambda_k = \infty$, $\sum_{k=0}^{\infty} \lambda_k^2 < \infty$ and $\sum_{k=0}^{\infty} \|\mathbb{E}[\epsilon_i^{k+1} | F_k]\|^2 < \infty$, then each $\mathbf{x}_i^k \rightarrow \mathbf{x}^* \in \mathbf{X}^*$ (the iterate sequence of each agent converges to the same optimal point) with probability 1 and in mean square.

Proof. See Theorems 6.1 and 6.2 in (RAM; NEDIĆ; VEERAVALLI, 2010). $\square$

1.3 The proposed approach

In this thesis, the methods we propose to solve (1.1) are composed by incremental (cyclic) subgradient iterations and by *string-averaging algorithm*. For this reason, we named them as *string-averaging incremental subgradient algorithm* and *string-averaging incremental stochastic subgradient algorithm*, for the stochastic case. String-averaging algorithms were created by Censor et. al. in (CENSOR; ELFVING; HERMAN, 2001) and applied to convex feasibility problems with algorithms that use projection methods. Its idea is simple and basically consists of splitting a set of indices in subsets (the strings) and calculating independent sequences, only with informations of each subset. An operator combining all sequences (usually performing a weighted average) is applied to obtain an output. Promising results were obtained using this technique, mainly due to the flexibility of computing the sequences in parallel (see, e.g., (CENSOR; TOM, 2003; PENFOLD et al., 2010; CENSOR; ZASLAVSKI, 2013; CENSOR; ZASLAVSKI, 2014; HELOU et al., 2014a)).

The methods that will be discussed in Chapters 2 and 3 are applied to large scale optimization problems, i.e., when $\mathbf{x}$ has high dimensions and/or $\mathbf{X}$ is a feasible set constructed from many constraints. In Chapter 2, we analyze a deterministic method, and apply it to a tomographic image reconstruction problem from limited data. In Chapter 3, we analyze a stochastic method from addition of the stochastic approximation approach to compute inexact subgradients. The method was applied to a fair rate allocation problem and tested in a set of 10^3 instances with increasing dimensions and number of constraints. However, the methods we will discuss in Chapters 2 and 3 are also suitable for applications to sensor networks with some particular features, such as:

1. The entire network $\mathcal{I} = \bigcup_{\ell=1}^P S_{\ell}$ such that $S_{\ell} \cap S_j = \emptyset$ for any $\ell, j \in \{1, \dots, P\}$ with $\ell \neq j$.
2. Let E_{ℓ}^k be the set of all links (i, j) at instant t_k , such that $i, j \in S_{\ell}$. Then, for each ℓ and t_k , the directed graph $G(S_{\ell}, E_{\ell}^k)$ is connected.
3. Each S_{ℓ} has a weight $w_{\ell} \in [0, 1]$ over the entire network $\mathcal{I}$, such that $\sum_{\ell=1}^P w_{\ell} = 1$.
4. The objective is similar to (1.1) with conditions above.

These features allow us to deal with multi-agent networks composed by P subnets, each containing $m(\ell)$ agents, which can monitor distinct environments in such a manner that agents only exchange information with the agents of their own subnet. Another point is that the weights on the subnets suggest weighted costs $w_\ell f_{i_s^\ell}$ for the objective function in (1.1), where $f_{i_s^\ell}$ is the local cost calculated by agent $i_s^\ell \in S_\ell$, with $s \in \{1, \dots, m(\ell)\}$. In this way, the method we consider generates, in each instant t_k (synchronously), vectors $\mathbf{x}_\ell^k$ only with information of the subnet S_ℓ by performing an incremental subgradient iteration. A central node receives these vectors from each subnet and processes them in order to provide an output for the entire network. Figure 4 shows how the proposed methods work.

Figure 4 – Scheme representing the proposed method operating in a sensor network composed of 4 subnets at time interval (t_k, t_{k+1}) , each with a different weight over the entire network. Each subnet operates an incremental subgradient method at time t_k . The outputs generated by each subnet are processed by the central node and redistributed to each subnet at time t_{k+1} .

Note that the presence of the central node can be a drawback from the energetic point of view if the network topology is not carefully adjusted. However, the methods discussed in the previous sections are not (at least directly) suitable for solving (1.1) with the addition of these assumptions, since there is no communication between agents allocated at different subnets. In addition, the methods we propose can be applied normally, as we will see in the applications in Chapters 2 and 3, in the conventional case $P = 1$, since we can choose the sets S_ℓ appropriately and thus benefiting from the parallel processing that can be carried out in each S_ℓ . Therefore, the approaches we propose not only generalize some methods of the literature, but also open the door for the development of algorithms that adapt to some realistic contexts.

STRING-AVERAGING INCREMENTAL SUBGRADIENTS FOR CONSTRAINED CONVEX OPTIMIZATION WITH APPLICATIONS TO RECONSTRUCTION OF TOMOGRAPHIC IMAGES

We present a method for non-smooth convex minimization which is based on subgradient directions and string-averaging techniques. In this approach, the set of available data is split into sequences (strings) and a given iterate is processed independently along each string, possibly in parallel, by an incremental subgradient method (ISM). The end-points of all strings are averaged to form the next iterate. The method is useful to solve sparse and large-scale non-smooth convex optimization problems, such as those arising in tomographic imaging. A convergence analysis is provided under realistic, standard conditions. Numerical tests are performed in a tomographic image reconstruction application, showing good performance for the convergence speed when measured as the decrease ratio of the objective function, in comparison to classical ISM.

2.1 Introduction

A fruitful approach to solve an inverse problem is to recast it as an optimization problem, leading to a more flexible formulation that can be handled with different techniques. The *reconstruction of tomographic images* is a classical example of a problem that has been explored by optimization methods, among which the well-known *incremental subgradient method* (ISM) (SOLODOV; ZAVRIEV, 1998; NEDIĆ; BERTSEKAS, 2001b;

NETO; PIERRO, 2009), that is a variation of the *subgradient method* (DEM'YANOV; VASIL'EV, 1985; SHOR, 1985; POLYAK, 1987), features nice performance in terms of convergence speed. There are many papers that discuss incremental gradient/-subgradient algorithms for convex/non-convex objective functions (smooth or not) with applications to several fields (BERTSEKAS, 1997; BERTSEKAS; TSITSIKLIS, 2000; BLATT; HERO; GAUCHMAN, 2007; NEDIĆ; BERTSEKAS, 2001b; RAM; NEDIĆ; VEERAVALLI, 2010; RAM; NEDIĆ; VEERAVALLI, 2009; SOLODOV, 1998; SOLODOV; ZAVRIEV, 1998; TANAKA; KUDO, 2003; TSENG, 1998). Some examples of applications to tomographic image reconstruction are found in (AHN; FESSLER, 2003; BROWNE; PIERRO, 1996; PIERRO; YAMAGISHI, 2001; HUDSON; LARKIN, 1994; TANAKA; KUDO, 2003). In this paper, we consider a rather general optimization problem that can be addressed by ISM and is useful for tomographic reconstruction and other problems, including to find solutions for ill-conditioned and/or large-scale linear systems. This problem consists of determining:

$$\begin{aligned} \mathbf{x} &\in \arg \min f(\mathbf{x}) \\ \text{s.t. } \mathbf{x} &\in \mathbf{X} \subset \mathbb{R}^n, \end{aligned} \tag{2.1}$$

where:

- (i) $f(\mathbf{x}) := P \sum_{i=1}^m \zeta_i f_i(\mathbf{x})$ in which $f_i: \mathbb{R}^n \rightarrow \mathbb{R}$ are convex (and possibly non-differentiable) functions;
- (ii) $\mathbf{X}$ is a non-empty, convex and closed set;
- (iii) The set $\mathcal{I} = \{S_1, \dots, S_P\}$ is a *partition* of $\{1, \dots, m\}$, i.e., $S_\ell \cap S_j = \emptyset$ for any $\ell, j \in \{1, \dots, P\}$ with $\ell \neq j$ and $\bigcup_{\ell=1}^P S_\ell = \{1, \dots, m\}$;
- (iv) Given $w_\ell \in [0, 1]$ and $S_\ell \in \mathcal{I}$, $\ell = 1, \dots, P$, the weights ζ_i satisfy $\zeta_i = w_\ell$ for all $i \in S_\ell$;
- (v) $\sum_{\ell=1}^P w_\ell = 1$.

Problem (2.1) with conditions (i)-(v) is reduced to the classical problem of minimizing $\sum_{i=1}^m f_i(\mathbf{x})$, s.t. $\mathbf{x} \in \mathbf{X}$, when $w_\ell = 1/P$ for all $\ell = 1, \dots, P$. The reason why we write the problem in this more complex way is twofold. On one hand, it is common to find problems in which a set of fixed weights are used to prioritize the contribution of some component functions. For instance, in the context of distributed networks, the component functions f_i (also called “agents”) can be affected by external conditions, network topology, traffic, etc., making possible that some sets of agents have a prevalent role on the network, which can be modeled by the weights (related to the corresponding subnets). On the other hand, the weights and the partition, which bring flexibility to the model and could possibly be explored aiming for instance at faster convergence, will fit naturally in our algorithmic framework.

We consider an approach that mixes ISM and *string-averaging algorithms* (SA algorithms). The general form of the SA algorithm was proposed initially in (CENSOR; ELFVING; HERMAN, 2001) and applied in solving *convex feasibility problems* (CFPs) with algorithms that use *projection methods* (CENSOR; ELFVING; HERMAN, 2001; CENSOR; TOM, 2003; PENFOLD *et al.*, 2010). Strings are created so that ISM (more generally, any ϵ -incremental subgradient method) can be processed in an independent form for each string (by step operators). Then, an average of string iterations is computed (combination operator), guiding the iterations towards the solution. To complete, approximate projections are used to maintain feasibility. We provide an analysis of convergence under reasonable assumptions, such as diminishing step-size rule and boundedness of the subgradients.

Some previous works in the literature have improved the understanding and practical efficiency of ISM by creating more general algorithmic structures, enabling a broader analysis of convergence and making them more robust and accurate (NEDIĆ; BERTSEKAS, 2001b; NETO; PIERRO, 2009; RAM; NEDIĆ; VEERAVALLI, 2009; RAM; NEDIĆ; VEERAVALLI, 2010; JOHANSSON; RABI; JOHANSSON, 2010). We improve on those results by adding a string-averaging structure to the ISM that allows for an efficient parallel processing of a complete iteration which, consequently, can lead to fast convergence and suitable approximate solutions. Furthermore, the presented techniques present better smoothing properties in practice, which is good for imaging tasks. These features are desirable, especially when we seek to solve ill-conditioned/large scale problems. As mentioned at the beginning of this section, one of our goals is to obtain an efficient method for solving problems of reconstruction of tomographic images from incompletely sampled data.

Although our work is closely linked to ISM, it is important to mention other classes of methods that can be applied to convex optimization problems. Under reasonable assumptions, problem (2.1) can be solved using *proximal-type algorithms* (for a description of some of the main methods, see (COMBETTES; PESQUET, 2011)). For instance, in (COMBETTES; PESQUET, 2008) the authors propose a proximal decomposition method derived from the Douglas-Rachford algorithm and establish its weak convergence in Hilbert spaces. Some variants and generalizations of such methods can be found in (BREDIES, 2009; RAGUET; FADILI; PEYRÉ, 2013; COMBETTES; PESQUET, 2016; CHEN; ROCKAFELLAR, 1997; COMBETTES; PESQUET, 2011). The *bundle approach* (HIRIART-URRUTY; LEMARÉCHAL, 1993b) is often used for numerically solving non-differentiable convex optimization problems. Also, *first order accelerated techniques* (BECK; TEBOULLE, 2009b; BECK; TEBOULLE, 2009a) form yet another family of popular techniques for convex optimization problem endowed with a certain separability property. The advantage of ISM over the aforementioned techniques lies in its lightweight iterations from the computational viewpoint, not even

requiring sufficient decrease properties to be checked. Besides, ISM presents a fast practical convergence rate in the first iterations which enables this technique to achieve good reconstructions within a small amount of time even for the huge problem sizes that appear, for example, in tomography.

The *tomographic image reconstruction problem* consists in finding an image described by a function $\psi : \mathbb{R}^2 \rightarrow \mathbb{R}$ from samples of its *Radon Transform*, which can be recast into solving the linear system

$$R\mathbf{x} = \mathbf{b}, \quad (2.2)$$

where R is the discretization of the Radon Transform operator, $\mathbf{b}$ contains the collected samples of the Radon Transform and the desired image is represented by the vector $\mathbf{x}$. We consider solving the problem (2.2), rewriting it as a minimization problem, as in (2.1). Solving problem (2.2) from an optimization standpoint is not a new idea. In particular, (NETO; PIERRO, 2009) illustrates the application of some of the methods arising from a general framework to tomographic image reconstruction with a limited number of views. For the discretized problem (2.2), iterative methods such as ART (Algebraic Reconstruction Technique) (NATTERER; WÜBBELING, 2001), POCS (Projection Onto Convex Sets) (BAUSCHKE; BORWEIN, 1996; COMBETTES, 1997), and Cimmino (COMBETTES, 1994) have been widely used in the past.

Tomographic image reconstruction is an inverse problem in the sense that the image ψ is to be obtained from the inversion of a linear compact operator, which is well known to be an ill-conditioned problem. While the specific case of Radon inversion has an analytical solution, that was published in 1917 by Johann Radon (for details see (NATTERER, 1986)), both such analytical techniques and the aforementioned iterative methods for linear systems of equations suffer from amplification of the statistical noise which, in practice, is always present in the right-hand side of (2.2). Therefore, methods designed to deal with noisy data have been developed, based on a maximum likelihood approach, among which EM (Expectation Maximization) (SHEPP; VARDI, 1982; VARDI; SHEPP; KAUFMAN, 1985), OS-EM (Ordered Subsets Expectation Maximization) (HUDSON; LARKIN, 1994), RAMLA (Row-Action Maximum Likelihood Algorithm) (BROWNE; PIERRO, 1996), BSREM (Block Sequential Regularized Expectation Maximization) (PIERRO; YAMAGISHI, 2001), DRAMA (Dynamic RAMLA) (TANAKA; KUDO, 2003; NETO; PIERRO, 2005), modified BSREM and relaxed OS-SPS (Ordered Subset-Separable Paraboloidal Surrogates) (AHN; FESSLER, 2003) are some of the best known in the literature. In (HELOU *et al.*, 2014a), a variant of the EM algorithm was introduced, called *String-Averaging Expectation-Maximization* (SAEM) *algorithm*. The SAEM algorithm was used in problems of image reconstruction in *Single-Photon Emission Computerized Tomography* (SPECT) and showed good performance in simulated and real data studies. High-contrast images, with less noise and

clearer object boundaries were reconstructed without incurring in more computation time. Besides the BSREM, DRAMA, modified BSREM and relaxed OS-SPS, that are relaxed algorithms for (penalized) maximum-likelihood image reconstruction in tomography, the method introduced in (DEWARAJA; KORAL; FESSLER, 2010) considers an approach, based in OS-SPS, in which extra anatomical boundary information is used. Other methods that use penalized models can be found in (HARMANY; MARCIA; WILLETT, 2012; CHOUZENOUX *et al.*, 2013). Proximal methods were used in (ANTHOINE *et al.*, 2012) to reconstruct images obtained via *Cone Beam Computerized Tomography* (CBCT) and *Positron Emission Tomography* (PET). In (CHOUZENOUX *et al.*, 2013), the *Majorize-Minimize Memory Gradient algorithm* (CHOUZENOUX *et al.*, 2011; CHOUZENOUX *et al.*, 2013) is studied and applied to imaging tasks.

The paper is organized as follows: Section 2.2 contains some preliminary theory involving incremental subgradient methods, optimality and feasibility operators and string-averaging algorithm; Section 2.3 discusses the proposed algorithm to solve (2.1), (i)-(v); Section 2.4 shows theoretical convergence results; in Section 2.5 numerical tests are performed with reconstruction of tomographic images. Final considerations are given in Section 2.6.

2.2 Preliminary theory

Throughout the text, we will use the following notations: bold-type notations e.g. $\mathbf{x}$, $\mathbf{x}_i$ and $\mathbf{x}_i^k$ are vectors whereas x is a number. We denote x_i as the i th coordinate of vector $\mathbf{x}$. Moreover,

$$\mathcal{P}_{\mathbf{X}}(\mathbf{x}) := \arg \min_{\mathbf{y} \in \mathbf{X}} \|\mathbf{y} - \mathbf{x}\|, \quad d_{\mathbf{X}}(\mathbf{x}) := \|\mathbf{x} - \mathcal{P}_{\mathbf{X}}(\mathbf{x})\|, \\ [x]_+ := \max\{0, x\}, \quad f^* = \inf_{\mathbf{x} \in \mathbf{X}} f(\mathbf{x}) \quad \text{and} \quad \mathbf{X}^* = \{\mathbf{x} \in \mathbf{X} \mid f(\mathbf{x}) = f^*\},$$

where we assume that $\mathbf{X}^* \neq \emptyset$.

One of the main methods for solving (2.1) is the *subgradient method*, whose extensive theory can be found in (DEM'YANOV; VASIL'EV, 1985; SHOR, 1985; POLYAK, 1987; HIRIART-URRUTY; LEMARÉCHAL, 1993b; BERTSEKAS, 1999),

$$\mathbf{x}^{k+1} = \mathcal{P}_{\mathbf{X}} \left(\mathbf{x}^k - \lambda_k \sum_{i=1}^m \mathbf{g}_i^k \right), \quad \lambda_k > 0, \quad \mathbf{g}_i^k \in \partial f_i(\mathbf{x}^k), \quad (2.3)$$

where the *subdifferential* of $f : \mathbb{R}^n \rightarrow \mathbb{R}$ at $\mathbf{x}$ (the set of all subgradients) can be defined by

$$\partial f(\mathbf{x}) := \{\mathbf{g} \mid f(\mathbf{x}) + \langle \mathbf{g}, \mathbf{z} - \mathbf{x} \rangle \leq f(\mathbf{z}), \forall \mathbf{z}\}. \quad (2.4)$$

A similar approach to (2.3), known as *incremental subgradient method*, was studied firstly by Kibardin in (KIBARDIN, 1979) and then analyzed by Solodov and Zavriev in

(SOLODOV; ZAVRIEV, 1998), in which a complete iteration of the algorithm can be described as follows:

$$\begin{aligned} \mathbf{x}_0^k &= \mathbf{x}^k \\ \mathbf{x}_i^k &= \mathbf{x}_{i-1}^k - \lambda_k \mathbf{g}_i^k, \quad i = 1, \dots, m, \quad \mathbf{g}_i^k \in \partial f_i(\mathbf{x}_{i-1}^k) \\ \mathbf{x}^{k+1} &= \mathcal{P}_{\mathbf{X}}(\mathbf{x}_m^k). \end{aligned} \quad (2.5)$$

A variant of this algorithm that uses projection onto $\mathbf{X}$ to compute the sub-iterations $\mathbf{x}_i^k$ was analyzed in (NEDIĆ; BERTSEKAS, 2001b).

The method we propose in this paper for solving the problem given in (2.1), (i)-(v) has the following general form described in (NETO; PIERRO, 2009):

$$\begin{aligned} \mathbf{x}^{k+1/2} &= \mathcal{O}_f(\lambda_k, \mathbf{x}^k); \\ \mathbf{x}^{k+1} &= \mathcal{V}_{\mathbf{X}}(\mathbf{x}^{k+1/2}). \end{aligned} \quad (2.6)$$

In the above equations, $\mathcal{O}_f$ is called *optimality operator* and $\mathcal{V}_{\mathbf{X}}$ is the *feasibility operator*. This framework was created to handle quite general algorithms for convex optimization problems. The basic idea consists in dividing an iterate in two parts: an optimality step which tries to guide the iterate towards the minimizer of the objective function (but not necessarily in a descent direction), followed by the feasibility step that drives the iterate in the direction of feasibility.

Next we enunciate a result due to Helou and De Pierro (see (NETO; PIERRO, 2009, Theorem 2.5)), establishing convergence of the method (2.6) under some conditions. This result is the key for the convergence analysis of the algorithm we propose in Section 2.3.

Theorem 2.1. The sequence $\{\mathbf{x}^k\}$ generated by the method described in (2.6) converges in the sense that

$$d_{\mathbf{X}^*}(\mathbf{x}^k) \rightarrow 0 \quad \text{and} \quad \lim_{k \rightarrow \infty} f(\mathbf{x}^k) = f^*,$$

if all of the following conditions hold:

Condition 1 (Properties of optimality operator). *For every $\mathbf{x} \in \mathbf{X}$ and for all sequence $\lambda_k \geq 0$, there exist $\alpha > 0$ and a sequence $\rho_k \geq 0$ such that the optimality operator $\mathcal{O}_f$ satisfies for all $k \geq 0$*

$$\|\mathcal{O}_f(\lambda_k, \mathbf{x}^k) - \mathbf{x}\|^2 \leq \|\mathbf{x}^k - \mathbf{x}\|^2 - \alpha \lambda_k (f(\mathbf{x}^k) - f(\mathbf{x})) + \lambda_k \rho_k. \quad (2.7)$$

We further assume that the error term in the above inequality vanishes, i.e., $\rho_k \rightarrow 0$ and we consider a boundedness property for the optimality operator: there is $\gamma > 0$ such that

$$\|\mathbf{x}^k - \mathcal{O}_f(\lambda_k, \mathbf{x}^k)\| \leq \lambda_k \gamma. \quad (2.8)$$

Condition 2 (Property of feasibility operator). *For the feasibility operator $\mathcal{V}_{\mathbf{X}}$, we impose that for all $\delta > 0$, exists $\epsilon_\delta > 0$ such that, if $d_{\mathbf{X}}(\mathbf{x}^{k+1/2}) \geq \delta$ and $\mathbf{x} \in \mathbf{X}$ we have*

$$\|\mathcal{V}_{\mathbf{X}}(\mathbf{x}^{k+1/2}) - \mathbf{x}\|^2 \leq \|\mathbf{x}^{k+1/2} - \mathbf{x}\|^2 - \epsilon_\delta. \quad (2.9)$$

Moreover, for all $\mathbf{x} \in \mathbf{X}$, $\mathcal{V}_{\mathbf{X}}(\mathbf{x}) = \mathbf{x}$, i.e., $\mathbf{x}$ is a fixed point of $\mathcal{V}_{\mathbf{X}}$.

Condition 3 (Diminishing step-size rule). *The sequence $\{\lambda_k\}$ satisfies*

$$\lambda_k \rightarrow 0^+, \quad \sum_{k=0}^{\infty} \lambda_k = \infty. \quad (2.10)$$

Condition 4. *The optimal set $\mathbf{X}^*$ is bounded, $\{d_{\mathbf{X}}(\mathbf{x}^k)\}$ is bounded and*

$$[f(\mathcal{P}_{\mathbf{X}}(\mathbf{x}^k)) - f(\mathbf{x}^k)]_+ \rightarrow 0.$$

Remark 2.2. Regarding the requirement $[f(\mathcal{P}_{\mathbf{X}}(\mathbf{x}^k)) - f(\mathbf{x}^k)]_+ \rightarrow 0$, it holds if there is a bounded sequence $\{\mathbf{v}^k\}$ where $\mathbf{v}^k \in \partial f(\mathcal{P}_{\mathbf{X}}(\mathbf{x}^k))$ and $d_{\mathbf{X}}(\mathbf{x}^k) \rightarrow 0$. Indeed,

$$\langle \mathbf{v}^k, \mathbf{y} - \mathcal{P}_{\mathbf{X}}(\mathbf{x}^k) \rangle \leq f(\mathbf{y}) - f(\mathcal{P}_{\mathbf{X}}(\mathbf{x}^k)), \quad \forall \mathbf{y} \in \mathbb{R}^n.$$

By Cauchy-Schwarz inequality, we have $\|\mathbf{v}^k\| \|\mathcal{P}_{\mathbf{X}}(\mathbf{x}^k) - \mathbf{y}\| \geq [f(\mathcal{P}_{\mathbf{X}}(\mathbf{x}^k)) - f(\mathbf{y})]_+$. Taking $\mathbf{y} = \mathbf{x}^k$, then $d_{\mathbf{X}}(\mathbf{x}^k) \rightarrow 0$ ensures that $[f(\mathcal{P}_{\mathbf{X}}(\mathbf{x}^k)) - f(\mathbf{x}^k)]_+ \rightarrow 0$. Therefore, under this mild boundedness assumption on the subdifferentials $\partial f(\mathcal{P}_{\mathbf{X}}(\mathbf{x}^k))$, proving that $d_{\mathbf{X}}(\mathbf{x}^k) \rightarrow 0$ also ensures that $[f(\mathcal{P}_{\mathbf{X}}(\mathbf{x}^k)) - f(\mathbf{x}^k)]_+ \rightarrow 0$.

Concerning the assumption $d_{\mathbf{X}}(\mathbf{x}^k) \rightarrow 0$, Proposition 2.1 in (NETO; PIERRO, 2009) shows that it holds if $\{d_{\mathbf{X}}(\mathbf{x}^k)\}$ is bounded, $\lambda_k \rightarrow 0^+$, and Equation (2.8) plus Condition 2 hold. Since Condition 4 requires $\{d_{\mathbf{X}}(\mathbf{x}^k)\}$ to be bounded, then we have that $[f(\mathcal{P}_{\mathbf{X}}(\mathbf{x}^k)) - f(\mathbf{x}^k)]_+ \rightarrow 0$ just under the boundedness assumption on $\partial f(\mathcal{P}_{\mathbf{X}}(\mathbf{x}^k))$. Furthermore, Corollary 2.7 in (NETO; PIERRO, 2009) states that $d_{\mathbf{X}}(\mathbf{x}^k) \rightarrow 0$ if $\lambda_k \rightarrow 0^+$, Conditions 1 and 2 hold and there is f_l such that $f(\mathbf{x}^k) \geq f_l$ for all k . Basically, the hypotheses of this corollary allow to show that $\{d_{\mathbf{X}}(\mathbf{x}^k)\}$ is bounded and result follow by Proposition 2.1. This is the situation that occurs in our numerical experiment. Such remarks are important to show how the hypotheses of our main convergence result (see Corollary 2.8 in Section 2.4) can be reasonable. $\square$

To state our algorithm in next section, we need to define the operators $\mathcal{O}_f$ and $\mathcal{V}_{\mathbf{X}}$. Below we present the last ingredient of our operator $\mathcal{O}_f$, the *String-Averaging* (SA) algorithm. Originally formulated in (CENSOR; ELFVING; HERMAN, 2001), SA algorithm consists of dividing an index set $I = \{1, 2, \dots, \eta\}$ into *strings* in the following manner

$$\Delta_\ell := \{i_1^\ell, i_2^\ell, \dots, i_{m(\ell)}^\ell\}, \quad (2.11)$$

where $m(\ell)$ represents the number of elements in the string Δ_ℓ and $\ell \in \{1, 2, \dots, N\}$. Let us consider $\mathcal{X}$ and $\mathcal{Y}$ as subsets of $\mathbb{R}^n$ where $\mathcal{Y} \subseteq \mathcal{X}$. The basic idea behind the

method consists in the sequential application of *step operators* $\mathcal{F}^{i_s^\ell} : \mathcal{X} \rightarrow \mathcal{Y}$, for each $s = 1, 2, \dots, m(\ell)$ over each string Δ_ℓ , producing N vectors $\mathbf{y}_\ell^k \in \mathcal{Y}$. Next, a *combination operator* $\mathcal{F} : \mathcal{Y}^N \rightarrow \mathcal{Y}$ mixes, usually by weighted average, all vectors $\mathbf{y}_\ell^k$ to obtain $\mathbf{y}^{k+1}$. We refer to the index s as the *step* and the index k as the *iteration*. Therefore, given $\mathbf{x}^0 \in \mathcal{X}$ and strings $\Delta_1, \dots, \Delta_N$ of $\mathbf{I}$, a complete iteration of the SA algorithm is computed, for each $k \geq 0$, by equations

$$\mathbf{y}_\ell^k := \mathcal{F}^{i_{m(\ell)}^\ell} \circ \dots \circ \mathcal{F}^{i_2^\ell} \circ \mathcal{F}^{i_1^\ell} (\mathbf{x}^k), \quad (2.12)$$

$$\mathbf{y}^{k+1} := \mathcal{F}((\mathbf{y}_1^k, \dots, \mathbf{y}_N^k)). \quad (2.13)$$

The main advantage of this approach is to allow for computation of each vector $\mathbf{y}_\ell^k$ in *parallel* at each iteration k , which is possible because the step operators $\mathcal{F}^{i_1^\ell}, \dots, \mathcal{F}^{i_{m(\ell)}^\ell}$ act along each string independently.

2.3 Proposed algorithm

Now we are ready to define $\mathcal{O}_f$ and $\mathcal{V}_\mathbf{X}$. Let us start by defining the optimality operator $\mathcal{O}_f : \mathbb{R}_+ \times \mathbf{Y} \rightarrow \mathbf{Y}$, where $\mathbf{Y}$ is a non-empty, closed and convex set such that $\mathbf{X} \subset \mathbf{Y} \subseteq \mathbb{R}^n$. For this, let $\mathcal{F}^{i_s^\ell} : \mathbb{R}_+ \times \mathbf{Y} \rightarrow \mathbf{Y}$ and $\mathcal{F} : \mathbf{Y}^P \rightarrow \mathbf{Y}$. Consider the set of strings $\Delta_1 = S_1, \dots, \Delta_P = S_P$ and the weight set $\{w_\ell\}_{\ell=1}^P$ as defined in the problem given in (2.1) with conditions (iii)-(v). Then, given $\mathbf{x} \in \mathbf{Y}$ and $\lambda \in \mathbb{R}_+$, we define

$$\mathbf{x}_{i_0^\ell} := \mathbf{x}, \quad \text{for all } \ell = 1, \dots, P, \quad (2.14)$$

$$\mathbf{x}_{i_s^\ell} := \mathcal{F}^{i_s^\ell}(\lambda, \mathbf{x}_{i_{s-1}^\ell}) := \mathbf{x}_{i_{s-1}^\ell} - \lambda \mathbf{g}_{i_s^\ell}, \quad s = 1, \dots, m(\ell), \quad (2.15)$$

$$\mathbf{x}_\ell := \mathbf{x}_{i_{m(\ell)}^\ell}, \quad \ell = 1, \dots, P, \quad (2.16)$$

$$\mathcal{O}_f(\lambda, \mathbf{x}) := \mathcal{F}((\mathbf{x}_1, \dots, \mathbf{x}_P)) := \sum_{\ell=1}^P w_\ell \mathbf{x}_\ell, \quad (2.17)$$

where $\mathbf{g}_{i_s^\ell} \in \partial f_{i_s^\ell}(\mathbf{x}_{i_{s-1}^\ell})$. Operators $\mathcal{F}^{i_s^\ell}$ in (2.15) correspond to the step operators in equation (2.12) of the SA algorithm and its definition is motivated by equation (2.5) of the incremental subgradient method. Function $\mathcal{F}$ in (2.17) corresponds to the combination operator in (2.13) and performs a weighted average of the end-points $\mathbf{x}_\ell$, completing the definition of the operator $\mathcal{O}_f$.

Now we need to define a feasibility operator $\mathcal{V}_\mathbf{X}$. For that, we use the *subgradient projection* (BAUSCHKE; COMBETTES; KRUK, 2006; COMBETTES, 1997; YAMADA;

OGURA, 2005a; YAMADA; OGURA, 2005b). Let us start noticing that every convex set $\mathbf{X} \neq \emptyset$ can be written as

$$\mathbf{X} = \bigcap_{i=1}^t \text{lev}_0(h_i), \quad (2.18)$$

where $\text{lev}_0(h_i) := \{\mathbf{x} \mid h_i(\mathbf{x}) \leq 0\}$. Each function $h_i : \mathbb{R}^n \rightarrow \mathbb{R}$ (t is finite) is supposed to be convex. The feasibility operator $\mathcal{V}_{\mathbf{X}} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is defined in (NETO; PIERRO, 2009) in the following form:

$$\mathcal{V}_{\mathbf{X}} := \mathcal{S}_{h_t}^{\nu_t} \circ \mathcal{S}_{h_{t-1}}^{\nu_{t-1}} \circ \dots \circ \mathcal{S}_{h_1}^{\nu_1}. \quad (2.19)$$

This definition assumes that there is $\sigma \in (0, 1]$ such that $\nu_i \in [\sigma, 2 - \sigma]$ for all i . Each operator $\mathcal{S}_h^\nu : \mathbb{R}^n \rightarrow \mathbb{R}^n$ in the previous definition is constructed using a ν -relaxed version of the subgradient projection with Polyak-type step-sizes, i.e.,

$$\mathcal{S}_h^\nu(\mathbf{x}) := \begin{cases} \mathbf{x} - \nu \frac{[h(\mathbf{x})]_+}{\|\mathbf{h}\|^2} \mathbf{h}, & \text{if } \mathbf{h} \neq \mathbf{0}; \\ \mathbf{x}, & \text{otherwise,} \end{cases} \quad (2.20)$$

where $\nu \in (0, 2)$ and $\mathbf{h} \in \partial h(\mathbf{x})$.

In order to get a better understanding of the behavior of our feasibility operator, Figure 5 shows the trajectory taken by successive applications of the operator $\mathcal{V}_{\mathbf{X}}$. The feasible set $\mathbf{X}$ is the intersection of the zero sublevel sets of the following convex functions: $h_1(\mathbf{x}) = \langle \mathbf{a}, \mathbf{x} \rangle + 2\|\mathbf{x}\|_1 - 1$, $h_2(\mathbf{x}) = 3\|\mathbf{x}\|_\infty - 2.5$ and $h_3(\mathbf{x}) = \|A\mathbf{x} - \mathbf{a}\|_1 + 2\|B\mathbf{x} - \mathbf{c}\|_2 - 10$ where

$$A = \begin{bmatrix} 2 & 1 \\ -1 & 3 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 0 \\ -2 & 2 \end{bmatrix}, \quad \mathbf{a} = [2 \quad 1]^T \quad \text{and} \quad \mathbf{c} = [1 \quad -2]^T.$$

To obtain $\mathcal{V}_{\mathbf{X}}(\mathbf{x}) = \mathcal{S}_{h_3}^{\nu_3} \circ \mathcal{S}_{h_2}^{\nu_2} \circ \mathcal{S}_{h_1}^{\nu_1}(\mathbf{x})$, we compute the subgradients $\mathbf{h}_i \in \partial h_i(\mathbf{s}_{i-1})$, $i = 1, \dots, 3$, such that $\mathbf{s}_0 := \mathbf{x}$ and $\mathbf{s}_i := \mathcal{S}_{h_i}^{\nu_i}(\mathbf{s}_{i-1})$. We choose $[-3 \quad -2.5]^T$ as an initial point and the following relaxation parameters: $\nu_1 = 0.5$, $\nu_2 = 0.6$ and $\nu_3 = 0.7$.

Remark 2.3. A string-averaging version of the feasibility operator can easily be derived in the following manner. Consider Q strings $V_j := \{i_1^j, i_2^j, \dots, i_{\kappa(j)}^j\} \subset \{1, \dots, t\}$ such that $\bigcup_{j=1}^Q V_j = \{1, \dots, t\}$, where $\kappa(j)$ is the number of elements in the string V_j . Then, for each $j = 1, \dots, Q$, we define the *string feasibility operator* $\mathcal{V}_j$ as

$$\mathcal{V}_j := \mathcal{S}_{h_{i_{\kappa(j)}^j}}^{\nu_{i_{\kappa(j)}^j}} \circ \mathcal{S}_{h_{i_{\kappa(j)-1}^j}}^{\nu_{i_{\kappa(j)-1}^j}} \circ \dots \circ \mathcal{S}_{h_{i_1^j}}^{\nu_{i_1^j}},$$

each satisfying for $\mathbf{y} \in \mathbf{X}_j := \bigcap_{i \in V_j} \text{lev}_0(h_i)$ and every $\mathbf{x}$ with $d_{\mathbf{X}_j}(\mathbf{x}) \geq \delta$:

$$\|\mathcal{V}_j(\mathbf{x}) - \mathbf{y}\|^2 \leq \|\mathbf{x} - \mathbf{y}\|^2 - \epsilon_\delta^j. \quad (2.21)$$

Figure 5 – Ten successive applications of the feasibility operator $\mathcal{V}_X$. The circles represent the points obtained in each application, starting from the point $[-3 \ -2.5]^T$. The dashed line describes the trajectory formed by these points.

We can then average these operators to obtain a new feasibility operator as $\tilde{\mathcal{V}}_X := 1/Q \sum_{j=1}^Q \mathcal{V}_j$. Making use of the triangle inequality and $(\sum_{i=1}^n a_i)^2 \leq n \sum_{i=1}^n a_i^2$, we have

$$\begin{aligned} \|\tilde{\mathcal{V}}_X(\mathbf{x}) - \mathbf{y}\|^2 &= \left\| \frac{1}{Q} \sum_{j=1}^Q [\mathcal{V}_j(\mathbf{x}) - \mathbf{y}] \right\|^2 = \frac{1}{Q^2} \left\| \sum_{j=1}^Q [\mathcal{V}_j(\mathbf{x}) - \mathbf{y}] \right\|^2 \\ &\leq \frac{1}{Q^2} \left(\sum_{j=1}^Q \|\mathcal{V}_j(\mathbf{x}) - \mathbf{y}\| \right)^2 \leq \frac{1}{Q} \sum_{j=1}^Q \|\mathcal{V}_j(\mathbf{x}) - \mathbf{y}\|^2. \end{aligned}$$

Now we notice that if $\mathbf{X}$ is bounded, $d_X(\mathbf{x}) \geq \delta$ implies that $\max\{d_{X_j}(\mathbf{x})\} \geq \tilde{\delta} > 0$ (for weaker conditions under which the same holds, see the results in (HOFFMANN, 1992)). Therefore, by using (2.21) in the inequality above we obtain, for $\mathbf{y} \in \mathbf{X} = \bigcap_{j=1}^Q \mathbf{X}_j$ and every $\mathbf{x}$ with $d_X(\mathbf{x}) \geq \delta$:

$$\|\tilde{\mathcal{V}}_X(\mathbf{x}) - \mathbf{y}\|^2 \leq \frac{1}{Q} \sum_{j=1}^Q \left\{ \|\mathbf{x} - \mathbf{y}\|^2 - \epsilon_{\delta_j}^j \right\},$$

where $\delta_j := d_{X_j}(\mathbf{x})$. Therefore:

$$\|\tilde{\mathcal{V}}_X(\mathbf{x}) - \mathbf{y}\|^2 \leq \|\mathbf{x} - \mathbf{y}\|^2 - \tilde{\epsilon}_\delta,$$

where $\tilde{e}_\delta = \min_{j \in \{1, \dots, Q\}} \{e_\delta^j\}$.

The above argument suggests that if the operators $\mathcal{V}_j$ satisfy Condition 2 with $\mathbf{X}$ replaced by $\mathbf{X}_j$, then its average also will satisfy Condition 2 with $\mathbf{X}$ replaced by $\bigcap_{j=1}^Q \mathbf{X}_j$. To the best of our knowledge, the previous discussion presents a first step to generalize some of the results from (CENSOR; ELFVING; HERMAN, 2001; CENSOR; ZASLAVSKI, 2013; CENSOR; ZASLAVSKI, 2014) towards averaging strings of inexact projections, or more specifically, averaging of Fejér-monotone operators. We do not make use of averaged feasibility operators in this paper for clarity of presentation and also because our numerical examples can be handled in the classical way, without string averaging, since our model has few constraints. $\square$

With the optimality and feasibility operators already defined, we present a complete description of the algorithm we propose to solve the problem defined in (2.1), (i)-(v).

Algorithm 8: STRING-AVERAGING INCREMENTAL SUBGRADIENT ALGORITHM

input : An initial vector $\mathbf{x}^0 \in \mathbf{Y}$, $K > 0$ and a sequence of step-sizes $\lambda_k \geq 0$.

1 **for** $k = 0$ **to** K **do**

2 (Step operators) Compute independently for each $\ell = 1, \dots, P$:

$$\begin{aligned} \mathbf{x}_{i_0^\ell}^k &= \mathbf{x}^k, \\ \mathbf{x}_{i_s^\ell}^k &= \mathcal{F}^{i_s^\ell}(\lambda_k, \mathbf{x}_{i_{s-1}^\ell}^k), \quad s = 1, \dots, m(\ell), \\ \mathbf{x}_\ell^k &= \mathbf{x}_{i_{m(\ell)}^\ell}^k, \end{aligned} \tag{2.22}$$

3 where $i_s^\ell \in \Delta_\ell := S_\ell$ for each $s = 1, \dots, m(\ell)$ and $\mathcal{F}^{i_s^\ell}$ is defined in (2.15).
 (Combination operator) Use the end-points $\mathbf{x}_\ell^k$ obtained in (2.15) and the optimality operator $\mathcal{O}_f$ defined in (2.17) to obtain:

$$\mathbf{x}^{k+1/2} = \mathcal{O}_f(\lambda_k, \mathbf{x}^k). \tag{2.23}$$

4 Apply feasibility operator $\mathcal{V}_\mathbf{X}$ defined in (2.19) on the sub-iteration $\mathbf{x}^{k+1/2}$ to obtain:

$$\mathbf{x}^{k+1} = \mathcal{V}_\mathbf{X}(\mathbf{x}^{k+1/2}). \tag{2.24}$$

5 **return** $\mathbf{x}^{K+1}$.

2.4 Convergence analysis

Along this section, we denote $F_{S_\ell}(\mathbf{x}) = \sum_{s=1}^{m(\ell)} f_{i_s^\ell}(\mathbf{x})$ for each $\ell = 1, \dots, P$. The following subgradient boundedness assumption is key in this paper: for all ℓ and s ,

$$C_{i_s^\ell} = \sup_{k \geq 0} \left\{ \|\mathbf{g}\| \mid \mathbf{g} \in \partial f_{i_s^\ell}(\mathbf{x}^k) \cup \partial f_{i_s^\ell}(\mathbf{x}_{i_{s-1}^\ell}^k) \right\} < \infty. \quad (2.25)$$

Recall that Theorem 2.1 is the main tool for the convergence analysis, so we will show that each of its conditions are valid under assumption (2.25). We present auxiliary results in the next two lemmas.

Lemma 2.4. Let $\{\mathbf{x}^k\}$ be the sequence generated by Algorithm 8 and suppose that subgradient boundedness assumption (2.25) holds. Then, for each ℓ and s and for all $k \geq 0$, we have

(i)

$$f_{i_s^\ell}(\mathbf{x}^k) - f_{i_s^\ell}(\mathbf{x}_{i_{s-1}^\ell}^k) \leq C_{i_s^\ell} \|\mathbf{x}_{i_{s-1}^\ell}^k - \mathbf{x}^k\|. \quad (2.26)$$

(ii)

$$\|\mathbf{x}_{i_s^\ell}^k - \mathbf{x}^k\| \leq \lambda_k \sum_{r=1}^s C_{i_r^\ell}. \quad (2.27)$$

(iii) For all $\mathbf{y} \in \mathbb{R}^n$, we have

$$\left\langle \sum_{s=1}^{m(\ell)} \mathbf{g}_{i_s^\ell}^k, \mathbf{y} - \mathbf{x}^k \right\rangle \leq F_{S_\ell}(\mathbf{y}) - F_{S_\ell}(\mathbf{x}^k) + 2\lambda_k \sum_{s=2}^{m(\ell)} C_{i_s^\ell} \left(\sum_{r=1}^{s-1} C_{i_r^\ell} \right), \quad (2.28)$$

where $\mathbf{g}_{i_s^\ell}^k \in \partial f_{i_s^\ell}(\mathbf{x}_{i_{s-1}^\ell}^k)$.

Proof. (i) By definition of the subdifferential $\partial f_{i_s^\ell}(\mathbf{x}^k)$, we have

$$f_{i_s^\ell}(\mathbf{x}^k) - f_{i_s^\ell}(\mathbf{x}_{i_{s-1}^\ell}^k) \leq -\langle \mathbf{v}_{i_s^\ell}^k, \mathbf{x}_{i_{s-1}^\ell}^k - \mathbf{x}^k \rangle,$$

where $\mathbf{v}_{i_s^\ell}^k \in \partial f_{i_s^\ell}(\mathbf{x}^k)$. The result follows from the Cauchy-Schwarz inequality and the subgradient boundedness assumption (2.25).

(ii) Developing the equation $\mathbf{x}_{i_s^\ell}^k = \mathbf{x}_{i_{s-1}^\ell}^k - \lambda_k \mathbf{g}_{i_s^\ell}^k$ for each $s = 1, \dots, m(\ell)$ yields,

$$\begin{aligned} \|\mathbf{x}_{i_1^\ell}^k - \mathbf{x}^k\| &= \|\mathbf{x}^k - \lambda_k \mathbf{g}_{i_1^\ell}^k - \mathbf{x}^k\| && \leq \lambda_k C_{i_1^\ell}, \\ \|\mathbf{x}_{i_2^\ell}^k - \mathbf{x}^k\| &\leq \|\mathbf{x}_{i_1^\ell}^k - \mathbf{x}^k\| + \lambda_k \|\mathbf{g}_{i_2^\ell}^k\| && \leq \lambda_k (C_{i_1^\ell} + C_{i_2^\ell}), \\ &\vdots && \vdots \\ \|\mathbf{x}_{i_s^\ell}^k - \mathbf{x}^k\| &\leq \|\mathbf{x}_{i_{s-1}^\ell}^k - \mathbf{x}^k\| + \lambda_k \|\mathbf{g}_{i_s^\ell}^k\| && \leq \lambda_k \sum_{r=1}^s C_{i_r^\ell}. \end{aligned}$$

(iii) By Cauchy-Schwarz inequality and definition of the subdifferential $\partial f_{i_s^\ell}(\mathbf{x}_{i_{s-1}^\ell}^k)$ we have,

$$\begin{aligned}
\left\langle \sum_{s=1}^{m(\ell)} \mathbf{g}_{i_s^\ell}^k, \mathbf{y} - \mathbf{x}^k \right\rangle &= \sum_{s=1}^{m(\ell)} \langle \mathbf{g}_{i_s^\ell}^k, \mathbf{x}_{i_{s-1}^\ell}^k - \mathbf{x}^k \rangle + \sum_{s=1}^{m(\ell)} \langle \mathbf{g}_{i_s^\ell}^k, \mathbf{y} - \mathbf{x}_{i_{s-1}^\ell}^k \rangle \\
&\leq \sum_{s=1}^{m(\ell)} \|\mathbf{g}_{i_s^\ell}^k\| \|\mathbf{x}^k - \mathbf{x}_{i_{s-1}^\ell}^k\| + \sum_{s=1}^{m(\ell)} (f_{i_s^\ell}(\mathbf{y}) - f_{i_s^\ell}(\mathbf{x}_{i_{s-1}^\ell}^k)) \\
&= \sum_{s=2}^{m(\ell)} \|\mathbf{g}_{i_s^\ell}^k\| \|\mathbf{x}^k - \mathbf{x}_{i_{s-1}^\ell}^k\| + F_{S_\ell}(\mathbf{y}) - F_{S_\ell}(\mathbf{x}^k) \\
&\quad - \sum_{s=2}^{m(\ell)} (f_{i_s^\ell}(\mathbf{x}_{i_{s-1}^\ell}^k) - f_{i_s^\ell}(\mathbf{x}^k)).
\end{aligned}$$

By eqs. (2.26), (2.27) and the subgradient boundedness assumption (2.25) we obtain,

$$\begin{aligned}
\left\langle \sum_{s=1}^{m(\ell)} \mathbf{g}_{i_s^\ell}^k, \mathbf{y} - \mathbf{x}^k \right\rangle &\leq \sum_{s=2}^{m(\ell)} \|\mathbf{g}_{i_s^\ell}^k\| \|\mathbf{x}^k - \mathbf{x}_{i_{s-1}^\ell}^k\| + F_{S_\ell}(\mathbf{y}) - F_{S_\ell}(\mathbf{x}^k) \\
&\quad + \sum_{s=2}^{m(\ell)} \|\mathbf{v}_{i_s^\ell}^k\| \|\mathbf{x}^k - \mathbf{x}_{i_{s-1}^\ell}^k\| \\
&\leq F_{S_\ell}(\mathbf{y}) - F_{S_\ell}(\mathbf{x}^k) + \sum_{s=2}^{m(\ell)} (\|\mathbf{g}_{i_s^\ell}^k\| + \|\mathbf{v}_{i_s^\ell}^k\|) \left(\lambda_k \sum_{r=1}^{s-1} C_{i_r^\ell} \right) \\
&\leq F_{S_\ell}(\mathbf{y}) - F_{S_\ell}(\mathbf{x}^k) + 2\lambda_k \sum_{s=2}^{m(\ell)} C_{i_s^\ell} \left(\sum_{r=1}^{s-1} C_{i_r^\ell} \right).
\end{aligned}$$

□

The following Lemma is useful to analyze the convergence of the Algorithm 8.

Lemma 2.5. Let $\{\mathbf{x}^k\}$ be the sequence generated by Algorithm 8 and suppose that assumption (2.25) holds. Then, there is a positive constant C such that, for all $\mathbf{y} \in \mathbf{Y} \supset \mathbf{X}$ and for all $k \geq 0$ we have

$$\|\mathcal{O}_f(\lambda_k, \mathbf{x}^k) - \mathbf{y}\|^2 \leq \|\mathbf{x}^k - \mathbf{y}\|^2 - \frac{2}{P} \lambda_k (f(\mathbf{x}^k) - f(\mathbf{y})) + C \lambda_k^2, \quad (2.29)$$

Proof. Initially, we can develop equation (2.22) for each $\ell = 1, \dots, P$ and obtain $\mathbf{x}_\ell^k = \mathbf{x}^k - \lambda_k \sum_{s=1}^{m(\ell)} \mathbf{g}_{i_s^\ell}^k$, where $\mathbf{g}_{i_s^\ell}^k \in \partial f_{i_s^\ell}(\mathbf{x}_{i_{s-1}^\ell}^k)$. Thus, from equation (2.23) we have for all $k \geq 0$,

$$\begin{aligned}
\mathcal{O}_f(\lambda_k, \mathbf{x}^k) &= \sum_{\ell=1}^P w_\ell \mathbf{x}_\ell^k \\
&= \sum_{\ell=1}^P w_\ell \left(\mathbf{x}^k - \lambda_k \sum_{s=1}^{m(\ell)} \mathbf{g}_{i_s^\ell}^k \right) \\
&= \mathbf{x}^k - \lambda_k \sum_{\ell=1}^P w_\ell \sum_{s=1}^{m(\ell)} \mathbf{g}_{i_s^\ell}^k.
\end{aligned}$$

Using the above equation we obtain for all $\mathbf{y} \in \mathbf{Y}$ and for all $k \geq 0$,

$$\begin{aligned} \|\mathcal{O}_f(\lambda_k, \mathbf{x}^k) - \mathbf{y}\|^2 &= \left\| \mathbf{x}^k - \mathbf{y} - \lambda_k \sum_{\ell=1}^P w_\ell \sum_{s=1}^{m(\ell)} \mathbf{g}_{i_s^\ell}^k \right\|^2 \\ &= \|\mathbf{x}^k - \mathbf{y}\|^2 - 2 \left\langle \mathbf{x}^k - \mathbf{y}, \lambda_k \sum_{\ell=1}^P w_\ell \sum_{s=1}^{m(\ell)} \mathbf{g}_{i_s^\ell}^k \right\rangle + \left\| \lambda_k \sum_{\ell=1}^P w_\ell \sum_{s=1}^{m(\ell)} \mathbf{g}_{i_s^\ell}^k \right\|^2 \\ &= \|\mathbf{x}^k - \mathbf{y}\|^2 + 2\lambda_k \sum_{\ell=1}^P w_\ell \left\langle \sum_{s=1}^{m(\ell)} \mathbf{g}_{i_s^\ell}^k, \mathbf{y} - \mathbf{x}^k \right\rangle + \lambda_k^2 \left\| \sum_{\ell=1}^P w_\ell \sum_{s=1}^{m(\ell)} \mathbf{g}_{i_s^\ell}^k \right\|^2. \end{aligned}$$

Now, using Lemma 2.4 (iii), triangle inequality and $P \sum_{\ell=1}^P w_\ell F_{S_\ell}(\mathbf{x}) = f(\mathbf{x})$ we have,

$$\begin{aligned} \|\mathcal{O}_f(\lambda_k, \mathbf{x}^k) - \mathbf{y}\|^2 &\leq \|\mathbf{x}^k - \mathbf{y}\|^2 - 2\lambda_k \sum_{\ell=1}^P w_\ell \left[F_{S_\ell}(\mathbf{x}^k) - F_{S_\ell}(\mathbf{y}) - 2\lambda_k \sum_{s=2}^{m(\ell)} C_{i_s^\ell} \left(\sum_{r=1}^{s-1} C_{i_r^\ell} \right) \right] \\ &\quad + \lambda_k^2 \left\| \sum_{\ell=1}^P w_\ell \sum_{s=1}^{m(\ell)} \mathbf{g}_{i_s^\ell}^k \right\|^2 \\ &\leq \|\mathbf{x}^k - \mathbf{y}\|^2 - 2\lambda_k \left(\sum_{\ell=1}^P w_\ell F_{S_\ell}(\mathbf{x}^k) - \sum_{\ell=1}^P w_\ell F_{S_\ell}(\mathbf{y}) \right) \\ &\quad + 4\lambda_k^2 \sum_{\ell=1}^P w_\ell \left[\sum_{s=2}^{m(\ell)} C_{i_s^\ell} \left(\sum_{r=1}^{s-1} C_{i_r^\ell} \right) \right] + \lambda_k^2 \left(\sum_{\ell=1}^P w_\ell \sum_{s=1}^{m(\ell)} \|\mathbf{g}_{i_s^\ell}^k\| \right)^2 \\ &= \|\mathbf{x}^k - \mathbf{y}\|^2 - 2\frac{\lambda_k}{P} (f(\mathbf{x}^k) - f(\mathbf{y})) + 4\lambda_k^2 \sum_{\ell=1}^P w_\ell \left[\sum_{s=2}^{m(\ell)} C_{i_s^\ell} \left(\sum_{r=1}^{s-1} C_{i_r^\ell} \right) \right] \\ &\quad + \lambda_k^2 \left(\sum_{\ell=1}^P w_\ell \sum_{s=1}^{m(\ell)} \|\mathbf{g}_{i_s^\ell}^k\| \right)^2. \end{aligned}$$

Finally, by subgradient boundedness assumption (2.25), we obtain for all $\mathbf{y} \in \mathbf{Y}$ and for all $k \geq 0$,

$$\begin{aligned} \|\mathcal{O}_f(\lambda_k, \mathbf{x}^k) - \mathbf{y}\|^2 &\leq \|\mathbf{x}^k - \mathbf{y}\|^2 - 2\frac{\lambda_k}{P} (f(\mathbf{x}^k) - f(\mathbf{y})) \\ &\quad + \lambda_k^2 \left[4 \sum_{\ell=1}^P w_\ell \left[\sum_{s=2}^{m(\ell)} C_{i_s^\ell} \left(\sum_{r=1}^{s-1} C_{i_r^\ell} \right) \right] + \left(\sum_{\ell=1}^P w_\ell \sum_{s=1}^{m(\ell)} C_{i_s^\ell} \right)^2 \right] \\ &= \|\mathbf{x}^k - \mathbf{y}\|^2 - \frac{2}{P} \lambda_k (f(\mathbf{x}^k) - f(\mathbf{y})) + C \lambda_k^2. \end{aligned}$$

□

The next two propositions aim at showing that, under some mild additional hypothesis, $\mathcal{O}_f$ and $\mathcal{V}_X$ satisfy Conditions 1-2.

Proposition 2.6. Let $\{\mathbf{x}^k\}$ be the sequence generated by Algorithm 8 and suppose that subgradient boundedness assumption (2.25) holds. Then, if $\lambda_k \rightarrow 0^+$, the optimality operator $\mathcal{O}_f$ satisfies Condition 1 of Theorem 2.1.

Proof. Lemma 2.5 ensures that for all $\mathbf{x} \in \mathbf{X} \subset \mathbf{Y}$ we have,

$$\|\mathcal{O}_f(\lambda_k, \mathbf{x}^k) - \mathbf{x}\|^2 \leq \|\mathbf{x}^k - \mathbf{x}\|^2 - \frac{2}{P} \lambda_k (f(\mathbf{x}^k) - f(\mathbf{x})) + C \lambda_k^2.$$

Defining $\alpha = 2/P$ and $\rho_k = \lambda_k C \geq 0$, equation (2.7) is satisfied and $\rho_k \rightarrow 0$. Furthermore, by triangle inequality and subgradient boundedness assumption (2.25) we have,

$$\begin{aligned} \|\mathcal{O}_f(\lambda_k, \mathbf{x}^k) - \mathbf{x}^k\| &= \left\| \sum_{\ell=1}^P w_\ell \mathbf{x}_\ell^k - \mathbf{x}^k \right\| \\ &= \left\| \mathbf{x}^k - \lambda_k \sum_{\ell=1}^P w_\ell \sum_{s=1}^{m(\ell)} \mathbf{g}_{i_s^\ell}^k - \mathbf{x}^k \right\| \\ &= \lambda_k \left\| \sum_{\ell=1}^P w_\ell \sum_{s=1}^{m(\ell)} \mathbf{g}_{i_s^\ell}^k \right\| \\ &\leq \lambda_k \sum_{\ell=1}^P w_\ell \sum_{s=1}^{m(\ell)} C_{i_s^\ell}, \end{aligned}$$

implying that equation (2.8) is satisfied with $\gamma = \sum_{\ell=1}^P w_\ell \sum_{s=1}^{m(\ell)} C_{i_s^\ell}$. Therefore, Condition 1 is satisfied. $\square$

Proposition 2.7. ((NETO; PIERRO, 2009), Proposition 3.4) Let $\mathbf{x}^{k+1/2}$ given in (2.23) be the first element $\mathbf{s}_0^k$ of the sequence $\{\mathbf{s}_i^k\}$, $i = 1, \dots, t$, given as $\mathbf{s}_i^k := \mathcal{S}_{h_i}^{v_i}(\mathbf{s}_{i-1}^k)$. In this sense, consider that $\mathbf{h}_i^k \in \partial h_i(\mathbf{s}_{i-1}^k)$. Suppose that for some index j , the set $\text{lev}_0(h_j)$ is bounded. In addition, consider that all sequences $\{\mathbf{h}_i^k\}$ are bounded. Then, $\mathcal{V}_\mathbf{X}$ satisfies Condition 2 of Theorem 2.1.

The main result of the paper is given next.

Corollary 2.8. Let $\{\mathbf{x}^k\}$ be the sequence generated by Algorithm 8 and suppose that subgradient boundedness assumption (2.25) holds. In addition, suppose that $\text{lev}_0(h_j)$ is bounded for some j and all sequences $\{\mathbf{h}_i^k\}$ are bounded. Then, if Conditions 3-4 of Theorem 2.1 hold, we have

$$d_{\mathbf{X}^*}(\mathbf{x}^k) \rightarrow 0 \quad \text{and} \quad \lim_{k \rightarrow \infty} f(\mathbf{x}^k) = f^*.$$

Proof. Propositions 2.6 and 2.7 state that operators $\mathcal{O}_f$ and $\mathcal{V}_\mathbf{X}$ satisfy Conditions 1-2. Therefore, the result is obtained applying Theorem 2.1. $\square$

Recall that we discuss the reasonability of the Condition 4 as a hypothesis for this corollary in Remark 2.2.

2.5 Numerical experiments

In this section, we apply the problem formulation (2.1), (i)-(v) and the method given in Algorithm 8 to the reconstruction of tomographic images from few views, and we explore results obtained from simulated and real data to show that the method is competitive when compared with the classic incremental subgradient algorithm. Let us start with a brief description of the problem. The task of reconstructing tomographic images is related to the mathematical problem of finding a function $\psi : \mathbb{R}^2 \rightarrow \mathbb{R}$ from its line integrals along straight lines. More specifically, we desire to find ψ given the following function:

$$\mathcal{R}[\psi](\theta, t) := \int_{\mathbb{R}} \psi(t(\cos \theta, \sin \theta)^T + s(-\sin \theta, \cos \theta)^T) ds. \quad (2.30)$$

The application $\psi \mapsto \mathcal{R}[\psi]$ is so-called *Radon Transform* and for a fixed θ , $\mathcal{R}_\theta[\psi](t)$ is known as a *projection* of ψ . For a detailed discussion about the physical and mathematical aspects involving tomography and the definition in (2.30), see, for example (NATTERER, 1986; NATTERER; WÜBBELING, 2001; HERMAN, 2009).

We now provide an example to better understand the geometric meaning of the definition of Radon transform. We can display ψ as a picture if we assign to each value in $[0, 1]$, a grayscale such as in Figure 6-(a). Here we use an artificial image made up of a sum of indicator functions of ellipses. The bar on the right indicates the grayscale used. We also show the axes t , x , y and the integration path for a given pair (θ, t') , which appears as the dashed line segment. The t -axis directions vary according to the number of angles adopted for the reconstruction process. In general, $\theta \in [0, \pi)$ because $\mathcal{R}[\psi](\theta + \pi, -t) = \mathcal{R}[\psi](\theta, t)$. For a fixed angle θ , the projection $\mathcal{R}_\theta[\psi](t)$ is computed for each $t' \in [-1, 1]$. Figure 6-(b) shows the projections obtained for three fixed angles θ_1 , θ_2 and θ_3 . Its representation given in Figure 6-(c) as an image in the $\theta \times t$ coordinate system is called *sinogram*. We also call the Radon transform at a fixed angle a *view* or *projection*.

The Radon transform models the data in a tomographic image reconstruction problem. That is, for reconstruction of the function ψ , we must go from Figure 6-(c) to the desired image in Figure 6-(a), i.e., it would be desirable to calculate the inverse $\mathcal{R}^{-1}$. However, as already mentioned, the Radon Transform is a compact operator and therefore its inversion is an ill-conditioned problem. In fact, for $n = 2$ and $n = 3$, Radon obtained inversion formulas involving first and second order differentiation of the data (NATTERER; WÜBBELING, 2001), respectively, implying in an unstable process due the increase of the error propagation in presence of perturbed data (when noise is present, which may occur due to width of the x-ray beam, scatter, hardening of the beam, photon statistics, detector inaccuracies, etc (HERMAN, 2009)). Other difficulties arise when using analytical solutions in practice due to, for example, the limited

Figure 6 – (a) An artificial image and the integration path for a given (θ, t') used to compute a projection $\mathcal{R}_\theta[\psi](t)$. (b) Projections for three fixed angles. (c) Sinogram obtained from image in (a).

number of views that often occurs. This is why more sophisticated optimization models are useful, and it is desirable to use an objective function and constraints that forces the consistency of the solution to the data and guarantees stability of the solution.

2.5.1 Experimental work

In what follows we provide a detailed description of the experimental setup.

- a) *The problem:* we consider the task of reconstructing an image from few views. We use a model based in the ℓ_1 -norm of the residual associated to the linear system $R\mathbf{x} = \mathbf{b}$, where R is the $m \times n$ Radon matrix, obtained through discretization of the Radon transform in (2.30), $\mathbf{x} \in \mathbb{R}^n$ is the solution that we want to find, $\mathbf{b} \approx R\mathbf{x}^* \in \mathbb{R}^m$ represents the data that we have for the reconstruction (sinogram), $\mathbf{x}^* \in \mathbb{R}_+^n$ is the original image and $m \ll n$. The choice of the ℓ_1 -norm serves as a way to promote robustness to the error $\mathbf{b} - R\mathbf{x}^*$, which in the case of synchrotron illuminated tomography has relatively few very large components and many

smaller ones. The small errors are related with the Poisson nature of the data, while the outliers happen because of systematic detection failure either due to dust in the ray path or to, e.g., failed detector pixels. In this manner, the following optimization problem has suitable features for the use of the Algorithm 8:

$$\begin{aligned} \min \quad & f(\mathbf{x}) = \|\mathbf{R}\mathbf{x} - \mathbf{b}\|_1 \\ \text{s.t.} \quad & h(\mathbf{x}) = TV(\mathbf{x}) - \tau \leq 0, \\ & \mathbf{x} \in \mathbb{R}_+^n. \end{aligned} \tag{2.31}$$

Note that the objective function $f(\mathbf{x}) = \sum_{i=1}^m |\langle \mathbf{r}_i, \mathbf{x} \rangle - b_i| = \sum_{i=1}^m f_i(\mathbf{x})$, where $\mathbf{r}_i$ represents the i -th row of $\mathbf{R}$. In comparison to (2.1) (i)-(v), model (2.31) suggests constant weights $w_\ell = 1/P$ for all $\ell = 1, \dots, P$ to satisfy conditions (iv) and (v). In our tests, we use $P = 1, \dots, 6$ and, to build the sets S_ℓ , we ordered the indices of the data randomly and then distributed in P equally sized sets (or as close to it as possible) aiming at satisfying condition (iii). We also assume that the image $\mathbf{x}^*$ to be reconstructed has large approximately constant areas, as is often the case in tomographic images. Operator $TV : \mathbb{R}^n \rightarrow \mathbb{R}_+$ is called *total variation* and is defined by

$$TV(\mathbf{x}) = \sum_{i=1}^{r_2} \sum_{j=1}^{r_1} \sqrt{(x_{i,j} - x_{i-1,j})^2 + (x_{i,j} - x_{i,j-1})^2},$$

where $\mathbf{x} = [x_q]^T$, $q \in \{1, \dots, n\}$, $n = r_1 r_2$ and $x_{i,j} := x_{(i-1)r_1 + j}$. We have also used the boundary conditions $x_{0,j} = x_{i,0} = 0$ and $\tau = TV(\mathbf{x}^*)$.

- b) *Data generation*: for this simulated experiment, we have considered the reconstruction of the Shepp-Logan phantom (KAK; SLANEY, 1988). In Figure 7, we show this image using a grayscale version with resolution of 256×256 . This resolution will also be used for the reconstruction. For the vector $\mathbf{b}$ that contains the data to be used in the reconstruction, we need an efficient routine for calculating the product $\mathbf{R}\mathbf{x}$. We consider 24 equally spaced angular projections with each sampled at 256 equally spaced points. We also consider reconstruction of images affected by Poisson noise, i.e., we execute the algorithms using data that was generated as samples of a Poisson random variable having as parameter the exact Radon Transform of the scaled phantom:

$$\mathbf{b} \sim \text{Poisson}(\kappa \mathcal{R}[\psi](\theta, t)), \tag{2.32}$$

where the scale factor $\kappa > 0$ is used to control the simulated photon count, i.e., the noise level. Figure 8 shows the result obtained for $\mathbf{b} = \mathbf{R}\mathbf{x}^*$, where $\mathbf{x}^*$ is the Shepp-Logan phantom, in both cases, i.e., with and without noise in the data.

- c) *Initial image*: for the initial image $\mathbf{x}^0$, we seek an uniform vector that somehow has information from data obtained by the Radon Transform of Shepp-Logan

Figure 7 – Shepp-Logan phantom with resolution 256×256 .

Figure 8 – Sinograms obtained from Radon transform of Shepp-Logan phantom. Only 24 equally spaced angular projections are taken, each sampled at 256 equally spaced points.

head phantom. For that, we use an initial image that satisfy $\sum_{i=1}^m \langle \mathbf{r}_i, \mathbf{x}^0 \rangle = \sum_{i=1}^m b_i$. Therefore, supposing $x_j^0 = \zeta$ for all $j = 1, \dots, n$, we can compute ζ by

$$\zeta = \frac{\sum_{i=1}^m b_i}{\sum_{i=1}^m \langle \mathbf{r}_i, \mathbf{1} \rangle}, \quad (2.33)$$

where $\mathbf{1}$ is the n -vector whose components are all equal to 1.

d) *Applying Algorithm 8*: step-size sequence $\{\lambda_k\}$ was determined by the formula

$$\lambda_k = (1 - \rho c_k) \frac{\lambda_0}{\alpha k^s / P + 1}, \quad (2.34)$$

where the sequence c_k starts with $c_0 = 0$ and the following terms are given by

$$c_k = \frac{\langle \mathbf{x}^{k-1/2} - \mathbf{x}^{k-1}, \mathbf{x}^k - \mathbf{x}^{k-1/2} \rangle}{\|\mathbf{x}^{k-1/2} - \mathbf{x}^{k-1}\| \|\mathbf{x}^k - \mathbf{x}^{k-1/2}\|}.$$

Each c_k is the cosine of the angle between the directions taken by optimality and feasibility operators in the previous iteration. Thus, the factor $(1 - \rho c_k)$ serves as an empirical way to prevent oscillations. Finally, we use $\lambda_0 = \mu \|\mathbf{R}\mathbf{x}^0 - \mathbf{b}\|_1 / \|\mathbf{g}^0\|^2$, where μ is the number of parcels in which the sum is divided and $\mathbf{g}^0$ is a subgradient of objective function in $\mathbf{x}^0$. Other free parameters in (2.34) were tuned and set to: $\rho = 0.999$, $s = 0.51$ and $\alpha = 1.0$.

Now we need to calculate the subgradients for the objective function and TV. Since the vector

$$\mathbf{sign}(\mathbf{x}) = [u_i]^T, \quad \text{such that} \quad u_i := \begin{cases} 1, & \text{if } x_i > 0, \\ 0, & \text{if } x_i = 0, \\ -1, & \text{otherwise} \end{cases}$$

belongs to the set $\partial \|\mathbf{x}\|_1$, then the theorem 4.2.1, p. 263 in (HIRIART-URRUTY; LEMARÉCHAL, 1993b) guarantees that

$$\mathbf{R}^T \mathbf{sign}(\mathbf{R}\mathbf{x} - \mathbf{b}) \in \partial \|\mathbf{R}\mathbf{x} - \mathbf{b}\|_1,$$

and this subgradient will be used in our experiments. In particular, for each $k \geq 0$, $\ell = 1, \dots, P$ and $s = 1, \dots, m(\ell)$ we use

$$\mathbf{g}_{i_s}^k = \begin{cases} \mathbf{r}_{i_s}^\ell, & \text{if } \langle \mathbf{r}_{i_s}^\ell, \mathbf{x}_{i_{s-1}}^k \rangle - b_{i_s}^\ell > 0, \\ -\mathbf{r}_{i_s}^\ell, & \text{if } \langle \mathbf{r}_{i_s}^\ell, \mathbf{x}_{i_{s-1}}^k \rangle - b_{i_s}^\ell < 0, \\ 0, & \text{otherwise.} \end{cases} \quad (2.35)$$

A subgradient $\mathbf{h} = [t_{i,j}]$ for $h(\mathbf{x}) = TV(\mathbf{x}) - \tau$ can be computed by

$$t_{i,j} = \frac{2x_{i,j} - x_{i,j-1} - x_{i-1,j}}{\sqrt{(x_{i,j} - x_{i,j-1})^2 + (x_{i,j} - x_{i-1,j})^2}} + \frac{x_{i,j} - x_{i,j+1}}{\sqrt{(x_{i,j+1} - x_{i,j})^2 + (x_{i,j+1} - x_{i-1,j+1})^2}} + \frac{x_{i,j} - x_{i+1,j}}{\sqrt{(x_{i+1,j} - x_{i,j})^2 + (x_{i+1,j} - x_{i+1,j-1})^2}}, \quad (2.36)$$

where, if any denominator is zero, we annul the correspondent parcel.

Once we have determined the strings by setting $\Delta_1 = S_1, \dots, \Delta_P = S_P$ and weights ($w_\ell = 1/P$ for all ℓ), the step-size sequence λ_k in (2.34), initial image $\mathbf{x}^0$ in (2.33) and subgradients $\mathbf{g}_{i_s^\ell}^k \in \partial f_{i_s^\ell}(\mathbf{x}_{i_{s-1}^\ell}^k)$ in (2.35), optimality operator $\mathcal{O}_f$ (2.14)-(2.17) can be applied. By considering the subdifferential of $\|\mathbf{x}\|_1$, it is clear that $\partial f(\mathbf{x})$ is uniformly bounded, ensuring that assumption (2.25) holds. Moreover, since $\rho \in [0, 1]$, $\alpha > 0$, $s \in (0, 1]$ and $c_k \in [-1, 1]$, by Cauchy-Schwarz inequality we can ensure that $\lambda_k > 0$ and that Condition 3 of Theorem 2.1 holds.

Using $\mathbf{h} \in \partial h(\mathbf{x})$ defined in (2.36), operator $\mathcal{S}_h^\nu$ can be computed by equation (2.20), such that, in our tests, we use $\nu = 1$. The feasibility operator is thus given by $\mathcal{V}_\mathbf{X} = \mathcal{P}_{\mathbb{R}_+^n} \circ \mathcal{S}_h^\nu$ (see equation (2.19)). The projection step can be regarded as a special case of the operator $\mathcal{S}_g^\nu$ with $\nu = 1$ and $g = d_{\mathbb{R}_+^n}$. It is easy to see that $\mathcal{V}_\mathbf{X}$ defined in this way satisfies the conditions established in the Proposition 2.7.

In conclusion, once $\|\mathbf{R}\mathbf{x} - \mathbf{b}\|_1 \geq 0$, Corollary 2.7 in (NETO; PIERRO, 2009) implies that $\{d_\mathbf{X}(\mathbf{x}^k)\} \rightarrow 0$ (the sequence is bounded). Since $\partial f(\mathbf{x})$ is uniformly bounded, we have that $[f(\mathcal{P}_\mathbf{X}(\mathbf{x}^k)) - f(\mathbf{x}^k)]_+ \rightarrow 0$ and Condition 4 of Theorem 2.1 holds. Thus, Corollary 2.8 can be applied ensuring convergence of the Algorithm 8.

2.5.2 Image reconstruction analysis

To run the experiments, we used a computer with processor Intel Core i7-4790 CPU @ 3.60 GHz x8 and 16 GB of memory. The operating system used was Linux and the implementation was realized in C++. Figure 9 shows the decrease of the objective function with respect to computation time to compare convergence speed and image quality in the performed reconstructions. Furthermore, in order to obtain a more meaningful analysis, we consider the graphic of the total variation $TV(\mathbf{x})$ and the *relative squared error*,

$$RSE(\mathbf{x}) = \frac{\|\mathbf{x} - \mathbf{x}^*\|^2}{\|\mathbf{x}^*\|^2}.$$

Figure 9 – Decrease of the objective function, TV and RSE in a noise-free condition. Comparing ISM (1 string) with algorithms that use 2-6 strings, executed in parallel, lower values are reached for the objective and RSE functions. For TV , we get an oscillation with lower intensity (especially when we used 4-6 strings). Note that solid horizontal lines on TV graph represent the target value $\tau = TV(\mathbf{x}^*)$. The solid vertical lines show, for a fixed computation time, that both functions values are decreasing with respect to the number of strings P . Figure 10 shows the reconstructed images by the algorithms for this fixed computation time. In the bottom right figure, note that TV values appear to represent, at many fixed levels of residual ℓ_1 -norm, a decreasing function of the number of strings.

Note that the RSE metric requires information on the desired image. Also we show graphs of $TV(\mathbf{x}^k)$ as function of $f(\mathbf{x}^k)$.

When we use $P = 2, \dots, 6$ (algorithm is executed with 2-6 strings), it is possible to observe a faster decrease in the values of the objective function $f(\mathbf{x}^k)$ as the running time increases, if compared to the case where $P = 1$. Since there is no guarantee that algorithm produces a descent direction in each iteration, it is important to note that, in some of the tests, the intensity of the oscillation, i.e., $f(\mathbf{x}^{k+1}) - f(\mathbf{x}^k)$ for $f(\mathbf{x}^{k+1}) > f(\mathbf{x}^k)$, decreases as the number of strings increases (note, for example, the algorithm performance with 5 and 6 strings, from 0 to about 40 seconds). A similar behavior can be noted for values $TV(\mathbf{x}^k)$. For 4-6 strings, the algorithm is able to

provide images with a more appropriate TV level, approaching the feasible region more quickly. Even if closer to satisfying the constraints, for methods with a larger number of strings, the values of $RSE(\mathbf{x}^k)$ and of the objective function decrease with lower intensity oscillations and reach lower values within the same computation time. Interestingly, the experiments with noisy data show that the algorithm generates a sharp decrease in the intensity of oscillation precisely where image quality seems to reach a good level. The study of conditions under which we can establish a stopping criterion for the algorithm are left for future research, perhaps taking advantage of this kind of phenomenon.

The quality of the reconstruction is significantly affected by the increase in the number of strings. Figure 10 shows the reconstructed images obtained in the experiments. There is a clear difference in the quality of reconstruction for ISM and algorithms with 2-6 strings. For 5 and 6 strings the reconstruction is visually perfect.

Figures 11, 12 and 13 show plots similar to those in Figure 9 but now under different relative noise levels, which was computed as $\|\mathbf{b} - \mathbf{b}^+\| / \|\mathbf{b}^+\|$, where $\mathbf{b}^+$ is the vector that contains the ideal data. We can note that the behavior of algorithm is similar to the previous case. Algorithms with larger number of strings reach results that the ISM takes longer to reach. Furthermore, oscillations with lower intensity can be noted, especially for 5 and 6 strings. Figure 14 shows the reconstructed images obtained by ISM and method with 6 strings according to the following rule: we set an objective function value and seek the first iteration to fall below this threshold for each method. Table 1 provides the running time and total variation for each case. These data confirm a good performance of the algorithm with string averaging, in the sense that, for fixed values of objective function, algorithm running with 6 strings provides images in which quality appears to be improved (or at least is similar) with lower running time, if compared against ISM.

Method / Noise / $f(\mathbf{x})$	Time (s)	$TV(\mathbf{x})$
ISM / 17.8% / 3.194×10^4	2.45×10^2	2.7×10^5
$P = 6$ / 17.8% / 3.191×10^4	6.0×10^1	1.82×10^5
ISM / 8.78% / 6.070×10^4	7.11×10^2	6.91×10^5
$P = 6$ / 8.78% / 6.093×10^4	1.5×10^2	5.87×10^5
ISM / 5.65% / 9.889×10^4	1.87×10^3	1.48×10^6
$P = 6$ / 5.65% / 9.889×10^4	2.2×10^2	1.36×10^6

Table 1 – Running time and total variation obtained by ISM and method with 6 strings under conditions of Poisson relative noise used in the tests for some fixed values of objective function.

Figure 10 – Reconstructed images obtained in the computation time mentioned in the Figure 9.

Figure 11 – Test with 17.8% of relative noise.

2.5.3 Tests with real data

Tomographic data was obtained by synchrotron radiation illumination of eggs of fishes of the species *Prochilodus lineatus* collected at the Madeira river's bed at Brazilian National Synchrotron Light Laboratory's (LNLS) facility. The eggs had been previously embedded in formaldehyde in order to prevent decay, but were later fixed in water within a borosilicate capillary tube for the scan. The sample was placed between the x-ray source and a photomultiplier coupled to a CCD capable of recording the images. After each radiographic measurement, the sample was rotated by a fixed amount and a new measurement was made.

A monochromator was added to the experiment to filter out low energy photons and avoid overheating of the soft eggs and the embedding water. This leads to a low photon flux, which increased the required exposure time to 20 seconds for each projection measurement. Each of this radiographic image was 2048×2048 pixels covering an area of $0.76 \times 0.76 \text{ mm}^2$. Given the high measurement duration of each projection, for the experiment to have a reasonable time span, we have collected only 200 views in the 180° range, leading to slice tomographic datasets (sinograms) each of

Figure 12 – Test with 8.78% of relative noise.

dimension 2048×200 (see Figure 15).

In this experiment, we use $\tau = 5 \times 10^4$, $\nu = 1.5$ and the other parameters ρ , α and s were the same as in the previous experiment. Furthermore, to avoid high step-size values, we multiply the initial step-size λ_0 by 0.25. Figure 16 shows the plot of TV as function of residual ℓ_1 -norm. Better quality reconstructions are generated by the algorithm that uses 6 strings. Figure 17 shows the images obtained by reconstruction. By considering that the eggs were immersed in water, which has homogeneous attenuation value, we can conclude that image in Figure 17-(b) has less artifacts. Figure 18 shows profile lines of the reconstructed images in Figure 17. Note that algorithm running with 6 strings presents a reconstruction with less overshoot and more smoothness than ISM.

2.6 Final comments

We have presented a new String-Averaging Incremental Subgradients family of algorithms. The theoretical convergence analysis of the method was established and experiments were performed in order to assess the effectiveness of the algorithm. The

Figure 13 – Test with 5.65% of relative noise.

method featured a good performance in practice, being able to reconstruct images with superior quality when compared to classical incremental subgradient algorithms. Furthermore, algorithmic parameters selection was shown to be robust across a range of tomographic experiments. The discussed theory involves solving non-smooth constrained convex optimization problems and, in this sense, more general models can be numerically addressed by the presented method. Future work may be related to the application of the string-averaging technique in incremental subgradient algorithms with stochastic errors, such as those that appear in (RAM; NEDIĆ; VEERAVALLI, 2009) and (RAM; NEDIĆ; VEERAVALLI, 2010).

Acknowledgements

We would like to thank the LNLS for providing the beam time for the tomographic acquisition, obtained under proposal number 17338. We also want to thank Prof. Marcelo dos Anjos (Federal University of Amazonas) for kindly providing the fish egg samples used in the presented experimentation. Rafael M. de Oliveira was funded by FAPESP Grant 2015/10171-2. Elias S. Helou was partially funded by FAPESP Grants

Figure 14 – Reconstructed images obtained in the tests with Poisson noise. Items (a)-(f) exhibit: method / relative noise / $f(\mathbf{x})$. Table 1 shows running time and total variation obtained in each case.

Figure 15 – Sinogram obtained by synchrotron radiation illumination of eggs of fishes.

Figure 16 – Total variation as function of the residual ℓ_1 -norm for the experiment using eggs of fishes. Note that the method with 6 strings provides lower values for total variation with a lower oscillation level.

2013/16508-3 and 2013/07375-0 and CNPq Grant 311476/2014-7. Eduardo F. Costa was funded by FAPESP under Grant 13/19380-8 and CNPq under Grant 311290/2013-2.

Figure 17 – Reconstructed images from sinogram given in Figure 15. The vertical solid red lines show where the profiles of Figure 18 were taken from.

Figure 18 – Profile lines from images in Figure 17.

STRING-AVERAGING INCREMENTAL STOCHASTIC SUBGRADIENT ALGORITHMS

We present a method to solve constrained convex stochastic optimization problems when the objective is a finite sum of convex functions f_i . Our method is based on Incremental Stochastic Subgradient Algorithms (ISSA) and String-Averaging (SA) techniques, with an assumption that the subgradient directions are affected by random errors in each iteration. Our analysis allows the method to perform approximate projections onto the feasible set in each iteration. We provide convergence results for the case where a diminishing step-size rule is used. We test our method in a large set of random instances of a stochastic convex programming problem and we compare its performance with the robust mirror descent stochastic approximation algorithm proposed in (NEMIROVSKI *et al.*, 2009).

3.1 Introduction

Consider the stochastic optimization problem that consists in determining:

$$\begin{aligned} \mathbf{x} \in \arg \min f(\mathbf{x}) \\ \text{s.t. } \mathbf{x} \in \mathbf{X}, \end{aligned} \tag{3.1}$$

where:

(i) $\mathbf{X} = [(\cap_{j=1}^r \mathbf{X}_j) \cap \mathbf{Y}] \neq \emptyset$ where each set $\mathbf{X}_j \subset \mathbb{R}^n$ is non-empty, convex and closed and $\mathbf{Y} \subset \mathbb{R}^n$ is a non-empty, compact and convex set. We always assume that projection onto $\mathbf{Y}$ is simple to compute;

(ii) $f(\mathbf{x}) := P \sum_{i=1}^m v_i f_i(\mathbf{x})$ in which $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$ are functions of the form:

$$f_i(\mathbf{x}) = \mathbb{E}[F_i(\mathbf{x}, \boldsymbol{\theta})], \quad (3.2)$$

where $\boldsymbol{\theta} = \zeta(\boldsymbol{\omega}) \in \mathbb{R}^d$, and $\zeta : \boldsymbol{\Omega} \rightarrow \mathbb{R}^d$ is a *random vector* defined on the space $\boldsymbol{\Omega} \subseteq \mathbb{R}^d$. We assume that the *expectation* $\mathbb{E}[\cdot]$ in (3.2) is well defined and finite valued for every $\mathbf{x} \in \mathbf{X}$. For each $i = 1, \dots, m$, $F_i : \mathbb{R}^n \times \mathbb{R}^d \rightarrow \mathbb{R}$ is a convex function for all $\boldsymbol{\theta} \in \mathbb{R}^d$. It follows that $f_i(\cdot)$ is convex and hence (3.1) is a convex optimization problem. The scalars P and v_i are as follows;

(iii) P is the number of subsets of $\{1, \dots, m\}$ forming a partition $\mathcal{I} = \{S_1, \dots, S_P\}$;

(iv) v_i are weights satisfying $v_i = w_\ell$ if $i \in S_\ell$, where the scalars $w_\ell \in [0, 1]$ are given;

(v) $\sum_{\ell=1}^P w_\ell = 1$.

Notice that when $w_\ell = 1/P$ for all $\ell = 1, \dots, P$, then $f(\mathbf{x}) = \sum_{i=1}^m f_i(\mathbf{x})$.

The description of the optimization problem approach (3.1), with conditions (i)-(v), that we deal in this paper offers the flexibility of weighting the local functions f_i . This can be useful in a practical context when, for example, agents in a distributed network are affected by external factors, network topology, traffic, etc.

The literature on methods that use subgradient directions to solve deterministic/stochastic convex unconstrained/constrained optimization problems is very broad. In the deterministic case, the *subgradient algorithms* (DEM'YANOV; VASIL'EV, 1985; SHOR, 1985; POLYAK, 1987) are the best known and formed the basis for several variations, including the methods by Nesterov (NESTEROV, 2005; NESTEROV, 2004; NESTEROV, 2009) that are well known for achieving the optimal complexity orders, as proven by Nemirovski and Yudin in (NEMIROVSKI; YUDIN, 1983). When the objective function is smooth and convex, the method in (NESTEROV, 2005) presents a rate of convergence for the objective values with complexity $O(k^{-2})$ where k is the iteration number. Methods in (NESTEROV, 2009) and (NESTEROV, 2004) achieve the optimal complexities $O(k^{-1/2})$ and $O(k^{-1})$ when the objective is a non-smooth convex and non-smooth strongly convex function, respectively. In all three preceeding cases, the algorithms need as input the knowledge of global constants: a Lipschitz constant for the objective functions in the non-smooth case, a Lipschitz constant for the gradient in the smooth case and the constant of strong convexity in the non-smooth strongly convex case. A recent paper of Neumaier (NEUMAIER, 2016) presents a fully adaptive optimal first order method, in the sense that it can be used in any of the three cases mentioned above (through the use of an auxiliary update scheme if no strong convexity is assumed). Another important class of methods is formed by the *proximal-type algorithms*, where the main methods can be found in (COMBETTES; PESQUET, 2011)

as well as some variants and generalizations in (BREDIES, 2009; RAGUET; FADILI; PEYRÉ, 2013; COMBETTES; PESQUET, 2016; CHEN; ROCKAFELLAR, 1997). Beck and Teboulle (BECK; TEBOULLE, 2009c) developed an optimal adaptive proximal point algorithm that is popular in convex optimization problems endowed with a certain separability property. In this case, like all proximal-type algorithms, the method needs more information about the objective function than just subgradients.

When the objective is a sum of local convex functions, the methods that stand out are the *incremental subgradient algorithms* (ISA) (see, e.g., (KIBARDIN, 1979; SOLODOV; ZAVRIEV, 1998)). Nedić and Bertsekas (NEDIĆ; BERTSEKAS, 2001b) establish the convergence properties of a number of variants of ISA, including a particular one in which a randomization is performed on the order of selection of the component functions. In this case, the convergence rate is substantially improved. With or without randomization, ISA performed substantially better than the pure subgradient method when tested in the dual of the generalized assignment problem. Helou and De Pierro (NETO; PIERRO, 2009) proposed a method to solve constrained non-smooth convex optimization problems, which is composed by an optimality operator and a feasibility operator. The optimality operator tries to guide the iterates towards the minimizer of the objective function by using an ISA, whereas the feasibility operator aims at ultimately achieving feasibility. The idea of using alternative operators at the feasibility step becomes rather useful in many large scale constrained optimization problems, because the computation of the Euclidean projection, that is frequently required, can be very expensive. We can highlight that the method of Helou and De Pierro has a certain similarity to the *inexact restoration methods* (MARTÍNEZ; PILOTTA, 2000; MARTÍNEZ; PILOTTA, 2005), in the sense that, in both methods, the iteration is composed by two phases, one related to the optimality and the other related to feasibility. The algorithm by Helou and De Pierro has proved to be suitable for applications to tomographic image reconstruction problems from limited data. Another incremental gradient/subgradient algorithms for convex/non-convex objective functions (smooth or not) with applications to several fields can be found in (BROWNE; PIERRO, 1996; BERTSEKAS, 1997; SOLODOV, 1998; TSENG, 1998; BERTSEKAS; TSITSIKLIS, 2000; TANAKA; KUDO, 2003; BLATT; HERO; GAUCHMAN, 2007).

An important class of *stochastic subgradient algorithms* arose from a computational approach known as *stochastic approximation methods*, which was pioneered in a paper by Robbins and Monro (ROBBINS; MONRO, 1951) and resulted in a series of papers in the stochastic optimization field, e.g. (BERTSEKAS, 1973; GAIVORONSKII, 1978; ERMOLIEV, 1983; RUSZCZYŃSKI; SYSKI, 1986; PFLUG, 1996; NEMIROVSKI *et al.*, 2009; RAM; NEDIĆ; VEERAVALLI, 2009; YOUSEFIAN; NEDIĆ; SHANBHAG, 2010; TOURI; NEDIĆ; RAM, 2010; RAM; NEDIĆ; VEERAVALLI, 2010; YOUSEFIAN; NEDIĆ; SHANBHAG, 2012; NEDIĆ; LEE, 2014; COMBETTES; PESQUET, 2016). In these

methods, we need to be able to calculate, for each pair $(\mathbf{x}, \boldsymbol{\theta}) \in \mathbb{R}^n \times \mathbb{R}^d$, a stochastic subgradient $\tilde{\mathbf{g}} \in \partial_{\mathbf{x}} F(\mathbf{x}, \boldsymbol{\theta})$ (supposing $f(\mathbf{x}) = \mathbb{E}[F(\mathbf{x}, \boldsymbol{\theta})]$, in the same sense as (3.2)), such that $\mathbb{E}[\tilde{\mathbf{g}}] \in \partial f(\mathbf{x})$. This is always possible when $F(\cdot, \boldsymbol{\theta})$ is convex and $f(\cdot)$ is finite valued in a neighborhood of a point $\mathbf{x}$ (see, e.g., (STRASSEN, 1965; BERTSEKAS, 1973)), that is, $\partial f(\mathbf{x}) = \mathbb{E}[\partial_{\mathbf{x}} F(\mathbf{x}, \boldsymbol{\theta})]$. Ram and Nedić (RAM; NEDIĆ; VEERAVALLI, 2009) analyzed a method called *incremental stochastic subgradient algorithm* (ISSA) that is a particular instance of the methods mentioned above in this paragraph: in each iteration, the method requires stochastic subgradients of the form $\tilde{\mathbf{g}}_i := \mathbf{g}_i + \boldsymbol{\epsilon}_i$ where $\mathbf{g}_i$ belongs to the subdifferential of f_i at a subiteration of index $i - 1$ and $\boldsymbol{\epsilon}_i$ is a random vector for all $i = 1, \dots, m$. By adding a certain assumption of boundedness on the errors sequence, they provided an almost sure convergence result when a diminishing step-size is used, and they obtained almost sure bounds on the algorithm's performance when a constant step-size is used.

In this paper we analyze a method that is a combination of ISSA with *string-averaging algorithms* (SA algorithms), called *string-averaging incremental stochastic subgradient algorithm* (SAISSA). The general form of the SA algorithm was proposed initially in (CENSOR; ELFVING; HERMAN, 2001) and since then has been applied in convex feasibility problems (CENSOR; ELFVING; HERMAN, 2001; CENSOR; TOM, 2003; CENSOR; ZASLAVSKI, 2013), proton computed tomography image reconstruction (PENFOLD *et al.*, 2010), sparse common fixed-point problems (CENSOR; SEGAL, 2009) and constrained minimization problems (CENSOR; ZASLAVSKI, 2014). The behaviour of the string-averaging algorithm also was analyzed when applied in incremental algorithms to solve smooth/non-smooth convex optimization problems. Helou *et al.* (HELOU *et al.*, 2014b) presented a variant of the *row-action maximum likelihood algorithm* (RAMLA), called *string-averaging expectation-maximization* (SAEM) *algorithm*. This method was applied in problems of image reconstruction in single-photon emission computed tomography (SPECT) and showed good performance in simulated and real data studies. Oliveira *et al.* (OLIVEIRA; HELOU; COSTA, 2016) analyzed a string-averaging version of the incremental-subgradient-type algorithm proposed in (NETO; PIERRO, 2009). The method was also applied in problems of image reconstruction in tomography and featured a good performance in practice (both in simulated and real cases), being able to reconstruct images with superior quality when compared to the classical approach (cyclic, with just one string).

The SAISSA approach sets a fixed number of strings (subsets of data indices) in such a manner that makes possible to run optimality steps in each iteration, through an ISSA, in parallel. An average of the end-points is computed in order to guide the iterations towards the minimizer. Since ISSA can be viewed as decentralized stochastic optimization algorithm, where each of m agents of a distributed network assesses individually a particular f_i as to minimize $\sum_{i=1}^m f_i$, then the addition of the SA scheme

can be viewed as the introduction of a central node that distributes the workload to subnets of agents and computes the averages received from them. In this paper, we are interested in methods able to solve large scale constrained convex optimizations problems, where the Euclidean projection onto the feasible set is not necessarily simple to compute. In this sense, we apply to the output an approximate projection operator (the feasibility step) aiming to converge to the feasible set. An almost sure convergence analysis is carried out similarly to (RAM; NEDIĆ; VEERAVALLI, 2009) where we consider the assumptions of diminishing step-size rule and boundedness on the subgradients and stochastic errors. Our results improve over existing ones, including (RAM; NEDIĆ; VEERAVALLI, 2009), in that we develop an analysis that takes into account the inclusion of SA techniques, a general feasibility operator instead the projection, and our results are endorsed by numerical experiments.

The paper is organized as follows: Section 3.2 provides a brief description of ISSA and SA algorithms and presents the proposed algorithm SAISSA to solve (3.1), (i)-(v); Section 3.3 shows the theoretical convergence results; in Section 3.4 we discuss some numerical experiments; final considerations are given in Section 3.5.

3.2 Proposed algorithm

Let us start by defining the *subdifferential set* of $f : \mathbb{R}^n \rightarrow \mathbb{R}$ at $\mathbf{x}$ (the set of all subgradients):

$$\partial f(\mathbf{x}) := \{\mathbf{g} \mid f(\mathbf{x}) + \langle \mathbf{g}, \mathbf{z} - \mathbf{x} \rangle \leq f(\mathbf{z}), \forall \mathbf{z}\}.$$

To solve the problem in (3.1)-(i)-(v), important contributions can be found in (RAM; NEDIĆ; VEERAVALLI, 2009) and among them is the *incremental stochastic subgradient algorithm* (ISSA), of which a complete iteration is computed as

$$\begin{aligned} \mathbf{x}_0^k &= \mathbf{x}^k \\ \mathbf{x}_i^k &= \mathcal{P}_{\mathbf{X}}(\mathbf{x}_{i-1}^k - \lambda_k \tilde{\mathbf{g}}_i^k), \quad i = 1, \dots, m, \\ \mathbf{x}^{k+1} &= \mathbf{x}_m^k, \end{aligned} \tag{3.3}$$

where $\tilde{\mathbf{g}}_i^k := \mathbf{g}_i^k + \epsilon_i^{k+1}$, $\mathbf{g}_i^k \in \partial f_i(\mathbf{x}_{i-1}^k)$, $\mathcal{P}_{\mathbf{X}}$ is the Euclidean projection onto $\mathbf{X}$ and ϵ_i^{k+1} is a vector of random variables that represents the error when calculating $\tilde{\mathbf{g}}_i^k$. This algorithm can be thought as a model where signals are processed sequentially and cyclically in a sensor network, in order to accomplish some task that are related to minimizing a cost functional, as described in (3.1).

The method we propose is based on ISSA and *string-averaging* (SA) *algorithm*. Originally formulated in (CENSOR; ELFVING; HERMAN, 2001), the SA algorithm consists of dividing an index set $\mathbf{I} = \{1, \dots, \eta\}$ into *strings* in the following manner

$$\Delta_\ell := \{i_1^\ell, \dots, i_{\eta(\ell)}^\ell\},$$

where $\eta(\ell)$ represents the number of elements in the string Δ_ℓ and $\ell \in \{1, \dots, N\}$. To provide a description of this method, let us consider following operators: $\mathcal{F}_s^{i_s^\ell} : \mathbb{R}^n \rightarrow \mathbb{R}^n$, for each $s = 1, \dots, \eta(\ell)$ and $\mathcal{F} : \mathbb{R}^{Nn} \rightarrow \mathbb{R}^n$. Given $\mathbf{x}^0 \in \mathbb{R}^n$ and strings $\Delta_1, \dots, \Delta_N$ of $\mathbf{I}$, a complete iteration of the SA algorithm is computed, for each $k \geq 0$, by equations

$$\mathbf{x}_\ell^{k+1} := \mathcal{F}_{m(\ell)}^{i_{m(\ell)}^\ell} \circ \dots \circ \mathcal{F}_2^{i_2^\ell} \circ \mathcal{F}_1^{i_1^\ell}(\mathbf{x}^k), \quad \ell = 1, \dots, N, \quad (3.4)$$

$$\mathbf{x}^{k+1} := \mathcal{F}(\mathbf{x}_1^{k+1}, \dots, \mathbf{x}_N^{k+1}). \quad (3.5)$$

The main advantage of this approach is to allow for computation of each vector $\mathbf{x}_\ell^k$ in parallel at each iteration k , which is possible because the operators $\mathcal{F}_1^{i_1^\ell}, \dots, \mathcal{F}_{m(\ell)}^{i_{m(\ell)}^\ell}$ used in equation (3.4) act along each string independently. In our algorithm (see Algorithm 9 below), we consider that each operator $\mathcal{F}_s^{i_s^\ell}$ performs the step i_s^ℓ of sub-iteration (3.3) of the ISSA (without performing the projection onto $\mathbf{X}$).

Regarding the operator $\mathcal{F}$ used in (3.5), we impose that it has the form $\mathcal{F} := \mathcal{H} \circ \mathcal{C}$. Operator $\mathcal{C} : \mathbb{R}^{Nn} \rightarrow \mathbb{R}^n$ represents a combination that mixes, usually by weighted average, all vectors $\mathbf{x}_\ell^k$. After applying the operator $\mathcal{C}$, the optimization step of our algorithm ends. The application of operator $\mathcal{H}$ is related to the feasibility step of the method. Throughout this paper, we assume the following assumption on the operator $\mathcal{H}$.

Assumption 1. Let $\mathbf{y} \in \mathbb{R}^n$. The operator $\mathcal{H} : \mathbb{R}^n \rightarrow \mathbf{Y}$ is defined in such a manner that, for every $\delta > 0$, there is $\epsilon_\delta > 0$ such that if $d_{\mathbf{X}}(\mathbf{y}) \geq \delta$ and $\mathbf{x} \in \mathbf{X}$, then

$$\|\mathcal{H}(\mathbf{y}) - \mathbf{x}\|^2 \leq \|\mathbf{y} - \mathbf{x}\|^2 - \epsilon_\delta. \quad (3.6)$$

Further, for all $\mathbf{x} \in \mathbf{X}$, $\mathcal{H}(\mathbf{x}) = \mathbf{x}$.

In particular, we can notice that if an operator $\mathcal{H}$ satisfies Assumption 1, then $\mathcal{H}$ is a $\mathbf{X}$ -Fejer operator (SHOR, 1985). We will use operators that satisfy Assumption 1 aiming that, in each iteration k , the sequence can take a step towards the feasible set $\mathbf{X}$. When $\mathcal{H}$ is continuous, Fejer-type algorithms of the form $\mathbf{x}^{k+1} = \mathcal{H}(\mathbf{x}^k)$, $k = 0, 1, \dots$, are known to produce sequences which converge to the feasible set $\mathbf{X}$ (cf. Theorem 2.10 in (SHOR, 1985)). In section 3.4, we define $\mathcal{H}$ inspired by a Polyak version of the gradient (or subgradient) method (POLYAK, 1969), which are essentially Fejer-type algorithms. It is precisely in this sense that often these types of operators are also called *approximate projections* onto $\mathbf{X}$ and, for example, have already been successfully used in incremental subgradients methods applied in tomographic image reconstruction problems (NETO; PIERRO, 2009; OLIVEIRA; HELOU; COSTA, 2016).

Next we provide a complete description of our algorithm to solve (3.1)-(i)-(v) (see Algorithm 9). By interpreting our algorithm in a sensor network environment, the

Algorithm 9: STRING-AVERAGING INCREMENTAL STOCHASTIC SUBGRADIENT ALGORITHM - SAISSA

input : $K > 0$. An initial vector $\mathbf{x}^0 \in \mathbb{R}^n$ and a sequence of step-sizes $\{\lambda_k\}$.
 Initialize the strings making $\Delta_1 = S_1, \dots, \Delta_P = S_P$, where $\mathcal{I} = \{S_1, \dots, S_P\}$ is the partition defined in (3.1)-(iii).

1 **for** $k = 0$ **to** K **do**

2 Compute independently for each $\ell = 1, \dots, P$:

$$\begin{aligned} \mathbf{x}_{i_0}^k &= \mathbf{x}^k, \\ \mathbf{x}_{i_s}^k &= \mathbf{x}_{i_{s-1}}^k - \lambda_k (\mathbf{g}_{i_s}^k + \boldsymbol{\epsilon}_{i_s}^{k+1}), \\ \mathbf{x}_\ell^k &= \mathbf{x}_{i_{m(\ell)}}^k, \end{aligned} \tag{3.7}$$

where $s = 1, \dots, m(\ell)$ and $\mathbf{g}_{i_s}^k \in \partial f_{i_s}^k(\mathbf{x}_{i_{s-1}}^k)$.

3 Use the string iterations $\mathbf{x}_\ell^k$ obtained in (3.7) and the weight set $\{w_1, \dots, w_P\}$ defined in (3.1)-(iv)-(v) to obtain:

$$\mathbf{x}^{k+1} = \mathcal{H} \left(\sum_{\ell=1}^P w_\ell \mathbf{x}_\ell^k \right), \tag{3.8}$$

where $\mathcal{H}$ satisfies Assumption 1.

4 **return** $\mathbf{x}^{K+1}$.

addition of the string-averaging concept in the ISSA can be visualized as an algorithmic version of a network in which subnets process independent strings being coordinated by a central node that calculates and distributes the averages.

3.3 Convergence analysis

In this section, we study the convergence of the sequence $\{\mathbf{x}^k\}$ generated by SAISSA by considering a diminishing step-size rule.

3.3.1 Notation and main hypothesis

As already mentioned, we denote by $\mathcal{P}_X$ the Euclidean projection onto X . Furthermore, we consider the following notations:

$$f^* = \inf_{\mathbf{x} \in X} f(\mathbf{x}) \quad \text{and} \quad X^* = \{\mathbf{x} \in X \mid f(\mathbf{x}) = f^*\},$$

and

$$d_X(\mathbf{x}) = \|\mathbf{x} - \mathcal{P}_X(\mathbf{x})\|, \quad \text{and} \quad [x]_+ = \max\{0, x\}, \quad x \in \mathbb{R},$$

where $\|\cdot\|$ is the standard Euclidean norm in $\mathbb{R}^n$.

The stochastic error when computing the subgradient $\mathbf{g}_i^k$ is denoted by ϵ_i^{k+1} and we assume the processes $\{\mathbf{x}^t\}$ and ϵ_i^t , $1 \leq i \leq m$, $0 \leq t \leq k$ are observed at iteration k . For ease of notation we define the random variable $G_k = \{\mathbf{x}^0, \dots, \mathbf{x}^k, \epsilon_1^1, \dots, \epsilon_m^1, \dots, \epsilon_1^k, \dots, \epsilon_m^k\}$. One can think of G_k as the available information until the end of the iteration k . We assume that the errors have uniformly bounded second momentum for all $i = 1, \dots, m$, i.e., we assume the following assumption:

Assumption 2. There is a sequence of positive scalars ν_k such that, for all i and k ,

$$\mathbb{E}[\|\epsilon_i^{k+1}\|^2 | G_k] \leq \nu_{k+1}^2,$$

holds with probability 1.

Sentences involving random variables and expectations are valid in the “surely” sense as usual (it is in this way that occurs in Lemma 3.1 below, for example), unless otherwise indicated (as in the Assumption 2 above).

A subgradient boundedness assumption is key for the development of the results of this paper.

Assumption 3. For all $\ell = 1, \dots, P$ and $s = 1, \dots, m(\ell)$ we have

$$C_{i_s^\ell} = \sup_{k \geq 0} \left\{ \|\mathbf{g}\| \mid \mathbf{g} \in \partial f_{i_s^\ell}(\mathbf{x}^k) \cup \partial f_{i_s^\ell}(\mathbf{x}_{i_{s-1}^\ell}^k) \right\} < \infty.$$

3.3.2 Auxiliary lemmas

The next lemma establishes some basic properties inherent to the proposed method.

Lemma 3.1. Let $\{\mathbf{x}^k\}$ be the sequence generated by the Algorithm 9 and suppose that Assumption 3 holds. Then

(i) For each ℓ and s , we have

$$f_{i_s^\ell}(\mathbf{x}^k) - f_{i_s^\ell}(\mathbf{x}_{i_{s-1}^\ell}^k) \leq C_{i_s^\ell} \|\mathbf{x}_{i_{s-1}^\ell}^k - \mathbf{x}^k\|. \quad (3.9)$$

(ii) For each ℓ and s , we have

$$\|\mathbf{x}_{i_{s-1}^\ell}^k - \mathbf{x}^k\| \leq \lambda_k \sum_{r=1}^{s-1} (C_{i_r^\ell} + \|\epsilon_{i_r^\ell}^{k+1}\|). \quad (3.10)$$

(iii) Define the operator $F_\ell : \mathbb{R}^n \rightarrow \mathbb{R}$ as

$$F_\ell(\mathbf{x}) = \sum_{s=1}^{m(\ell)} f_{i_s^\ell}(\mathbf{x}). \quad (3.11)$$

Then, for all $\mathbf{z} \in \mathbb{R}^n$ and for each ℓ , we have

$$\begin{aligned} \left(\sum_{s=1}^{m(\ell)} \mathbf{g}_{i_s^\ell}^k \right)^T (\mathbf{z} - \mathbf{x}^k) &\leq F_\ell(\mathbf{z}) - F_\ell(\mathbf{x}^k) \\ &\quad + 2\lambda_k \sum_{s=2}^{m(\ell)} C_{i_s^\ell} \sum_{r=1}^{s-1} (C_{i_r^\ell} + \|\epsilon_{i_r^\ell}^{k+1}\|). \end{aligned}$$

Proof. (i) Supposing that $\hat{\mathbf{g}}_{i_s^\ell}^k \in \partial f_{i_s^\ell}(\mathbf{x}^k)$, we have

$$f_{i_s^\ell}(\mathbf{x}^k) - f_{i_s^\ell}(\mathbf{x}_{i_{s-1}^\ell}^k) \leq - \left(\hat{\mathbf{g}}_{i_s^\ell}^k \right)^T (\mathbf{x}_{i_{s-1}^\ell}^k - \mathbf{x}^k),$$

and the result follows from Cauchy-Schwarz inequality and Assumption 3.

(ii) It is easy to check that $\mathbf{x}_{i_{s-1}^\ell}^k = \mathbf{x}^k - \lambda_k \sum_{r=1}^{s-1} (\mathbf{g}_{i_r^\ell}^k + \epsilon_{i_r^\ell}^{k+1})$. Thus,

$$\begin{aligned} \|\mathbf{x}_{i_{s-1}^\ell}^k - \mathbf{x}^k\| &= \|\mathbf{x}^k - \lambda_k \sum_{r=1}^{s-1} (\mathbf{g}_{i_r^\ell}^k + \epsilon_{i_r^\ell}^{k+1}) - \mathbf{x}^k\| \\ &= \lambda_k \left\| \sum_{r=1}^{s-1} (\mathbf{g}_{i_r^\ell}^k + \epsilon_{i_r^\ell}^{k+1}) \right\| \\ &\leq \lambda_k \sum_{r=1}^{s-1} (\|\mathbf{g}_{i_r^\ell}^k\| + \|\epsilon_{i_r^\ell}^{k+1}\|) \\ &\leq \lambda_k \sum_{r=1}^{s-1} (C_{i_r^\ell} + \|\epsilon_{i_r^\ell}^{k+1}\|). \end{aligned}$$

(iii) Using the Cauchy-Schwarz inequality and $\mathbf{g}_{i_s^\ell}^k \in \partial f_{i_s^\ell}(\mathbf{x}_{i_{s-1}^\ell}^k)$ we have,

$$\begin{aligned} \left(\sum_{s=1}^{m(\ell)} \mathbf{g}_{i_s^\ell}^k \right)^T (\mathbf{z} - \mathbf{x}^k) &= \left(\sum_{s=1}^{m(\ell)} \mathbf{g}_{i_s^\ell}^k \right)^T (\mathbf{x}_{i_{s-1}^\ell}^k - \mathbf{x}^k) + \left(\sum_{s=1}^{m(\ell)} \mathbf{g}_{i_s^\ell}^k \right)^T (\mathbf{z} - \mathbf{x}_{i_{s-1}^\ell}^k) \\ &\leq \sum_{s=1}^{m(\ell)} \|\mathbf{g}_{i_s^\ell}^k\| \|\mathbf{x}^k - \mathbf{x}_{i_{s-1}^\ell}^k\| + \sum_{s=1}^{m(\ell)} (f_{i_s^\ell}(\mathbf{z}) - f_{i_s^\ell}(\mathbf{x}_{i_{s-1}^\ell}^k)) \\ &= \sum_{s=2}^{m(\ell)} \|\mathbf{g}_{i_s^\ell}^k\| \|\mathbf{x}^k - \mathbf{x}_{i_{s-1}^\ell}^k\| + F_\ell(\mathbf{z}) - F_\ell(\mathbf{x}^k) \\ &\quad - \sum_{s=2}^{m(\ell)} (f_{i_s^\ell}(\mathbf{x}_{i_{s-1}^\ell}^k) - f_{i_s^\ell}(\mathbf{x}^k)). \end{aligned}$$

Therefore, by equations (3.9) and (3.10) we conclude that

$$\begin{aligned}
\left(\sum_{s=1}^{m(\ell)} \mathbf{g}_{i_s^\ell}^k \right)^T (\mathbf{z} - \mathbf{x}^k) &\leq F_\ell(\mathbf{z}) - F_\ell(\mathbf{x}^k) + \sum_{s=2}^{m(\ell)} \|\mathbf{g}_{i_s^\ell}^k\| \|\mathbf{x}^k - \mathbf{x}_{i_{s-1}^\ell}^k\| \\
&\quad + \sum_{s=2}^{m(\ell)} C_{i_s^\ell} \|\mathbf{x}^k - \mathbf{x}_{i_{s-1}^\ell}^k\| \\
&\leq F_\ell(\mathbf{z}) - F_\ell(\mathbf{x}^k) + 2 \sum_{s=2}^{m(\ell)} C_{i_s^\ell} \|\mathbf{x}^k - \mathbf{x}_{i_{s-1}^\ell}^k\| \\
&\leq F_\ell(\mathbf{z}) - F_\ell(\mathbf{x}^k) + 2\lambda_k \sum_{s=2}^{m(\ell)} C_{i_s^\ell} \sum_{r=1}^{s-1} (C_{i_r^\ell} + \|\epsilon_{i_r^\ell}^{k+1}\|),
\end{aligned}$$

completing the proof. $\square$

Lemma 3.2. Let $a_j, j = 1, \dots, N$, be real numbers. Then

$$\left(\sum_{j=1}^N a_j \right)^2 \leq N \sum_{j=1}^N a_j^2,$$

Proof. It will be made by induction on N . The proof is trivial for $N = 1$. Now, assume that the result holds for some $N > 1$. Then

$$\begin{aligned}
\left(\sum_{j=1}^{N+1} a_j \right)^2 &= \left(\sum_{j=1}^N a_j \right)^2 + 2a_{N+1} \sum_{j=1}^N a_j + a_{N+1}^2 \\
&\stackrel{\text{ind. hypothesis}}{\leq} N \sum_{j=1}^N a_j^2 + 2a_{N+1} \sum_{j=1}^N a_j + a_{N+1}^2 \\
&\leq N \sum_{j=1}^N a_j^2 + \sum_{j=1}^N (a_j^2 + a_{N+1}^2) + a_{N+1}^2 \\
&= (N+1) \sum_{j=1}^{N+1} a_j^2,
\end{aligned}$$

where the second inequality follows from $2ab \leq a^2 + b^2$ for all $a, b \in \mathbb{R}$. $\square$

Lemma 3.3. Suppose that Assumptions 1 and 3 hold. Then, for all $\mathbf{z} \in \mathbf{X}$ and $k \geq 0$, the sequence $\{\mathbf{x}^k\}$ generated by Algorithm 9 satisfies

$$\begin{aligned}
\mathbb{E}[\|\mathbf{x}^{k+1} - \mathbf{z}\|^2 | G_k] &\leq \|\mathbf{x}^k - \mathbf{z}\|^2 - \frac{2}{P} \lambda_k (f(\mathbf{x}^k) - f(\mathbf{z})) \\
&\quad + 4\lambda_k^2 \sum_{\ell=1}^P w_\ell \sum_{s=2}^{m(\ell)} C_{i_s^\ell} \sum_{r=1}^{s-1} (C_{i_r^\ell} + v_{k+1}) \\
&\quad + 2C_\ell \lambda_k v_{k+1} \|\mathbf{x}^k - \mathbf{z}\| \\
&\quad + \lambda_k^2 \sum_{\ell=1}^P w_\ell m(\ell) \sum_{s=1}^{m(\ell)} (C_{i_s^\ell} + v_{k+1})^2,
\end{aligned}$$

with probability 1, where $C_\ell = \sum_{\ell=1}^P w_\ell m(\ell)$.

Proof. Given $k \geq 0$, if $\sum_{\ell=1}^P w_\ell \mathbf{x}_\ell^k \notin \mathbf{X}$, then we can take $0 < \delta_k \leq d_{\mathbf{X}}(\sum_{\ell=1}^P w_\ell \mathbf{x}_\ell^k)$ in such a manner that Assumption 1 ensures that there is $\epsilon_{\delta_k} > 0$ such that for all $\mathbf{z} \in \mathbf{X}$,

$$\|\mathbf{x}^{k+1} - \mathbf{z}\|^2 = \left\| \mathcal{H} \left(\sum_{\ell=1}^P w_\ell \mathbf{x}_\ell^k \right) - \mathbf{z} \right\|^2 \leq \left\| \sum_{\ell=1}^P w_\ell \mathbf{x}_\ell^k - \mathbf{z} \right\|^2 - \epsilon_{\delta_k},$$

otherwise $\mathcal{H}(\sum_{\ell=1}^P w_\ell \mathbf{x}_\ell^k) = \sum_{\ell=1}^P w_\ell \mathbf{x}_\ell^k$ and

$$\|\mathbf{x}^{k+1} - \mathbf{z}\|^2 = \left\| \mathcal{H} \left(\sum_{\ell=1}^P w_\ell \mathbf{x}_\ell^k \right) - \mathbf{z} \right\|^2 = \left\| \sum_{\ell=1}^P w_\ell \mathbf{x}_\ell^k - \mathbf{z} \right\|^2.$$

These two possibilities provide, together with (3.1)-(iv)-(v) and convexity of the squared norm

$$\|\mathbf{x}^{k+1} - \mathbf{z}\|^2 \leq \left\| \sum_{\ell=1}^P w_\ell (\mathbf{x}_\ell^k - \mathbf{z}) \right\|^2 \leq \sum_{\ell=1}^P w_\ell \|\mathbf{x}_\ell^k - \mathbf{z}\|^2.$$

By developing equation (3.7), we have for each ℓ

$$\mathbf{x}_\ell^k = \mathbf{x}^k - \lambda_k \sum_{s=1}^{m(\ell)} (\mathbf{g}_{i_s^\ell}^k + \boldsymbol{\epsilon}_{i_s^\ell}^{k+1}),$$

and together with the triangle inequality we obtain

$$\begin{aligned} \|\mathbf{x}^{k+1} - \mathbf{z}\|^2 &\leq \sum_{\ell=1}^P w_\ell \|\mathbf{x}^k - \lambda_k \sum_{s=1}^{m(\ell)} (\mathbf{g}_{i_s^\ell}^k + \boldsymbol{\epsilon}_{i_s^\ell}^{k+1}) - \mathbf{z}\|^2 \\ &= \sum_{\ell=1}^P w_\ell \left[\|\mathbf{x}^k - \mathbf{z}\|^2 - 2\lambda_k \left[\sum_{s=1}^{m(\ell)} (\mathbf{g}_{i_s^\ell}^k + \boldsymbol{\epsilon}_{i_s^\ell}^{k+1}) \right]^T (\mathbf{x}^k - \mathbf{z}) \right. \\ &\quad \left. + \lambda_k^2 \left\| \sum_{s=1}^{m(\ell)} (\mathbf{g}_{i_s^\ell}^k + \boldsymbol{\epsilon}_{i_s^\ell}^{k+1}) \right\|^2 \right] \\ &\leq \sum_{\ell=1}^P w_\ell \left[\|\mathbf{x}^k - \mathbf{z}\|^2 + 2\lambda_k \left(\sum_{s=1}^{m(\ell)} \mathbf{g}_{i_s^\ell}^k \right)^T (\mathbf{z} - \mathbf{x}^k) \right. \\ &\quad \left. - 2\lambda_k \left(\sum_{s=1}^{m(\ell)} \boldsymbol{\epsilon}_{i_s^\ell}^{k+1} \right)^T (\mathbf{x}^k - \mathbf{z}) + \lambda_k^2 \left(\sum_{s=1}^{m(\ell)} \|\mathbf{g}_{i_s^\ell}^k + \boldsymbol{\epsilon}_{i_s^\ell}^{k+1}\| \right)^2 \right]. \end{aligned} \quad (3.12)$$

Lemma 3.2 ensures that for each ℓ and k , if we set $a_s = \|\mathbf{g}_{i_s^\ell}^k + \boldsymbol{\epsilon}_{i_s^\ell}^{k+1}\|$, then

$$\left(\sum_{s=1}^{m(\ell)} \|\mathbf{g}_{i_s^\ell}^k + \boldsymbol{\epsilon}_{i_s^\ell}^{k+1}\| \right)^2 = \left(\sum_{s=1}^{m(\ell)} a_s \right)^2 \leq m(\ell) \sum_{s=1}^{m(\ell)} a_s^2 = m(\ell) \sum_{s=1}^{m(\ell)} \|\mathbf{g}_{i_s^\ell}^k + \boldsymbol{\epsilon}_{i_s^\ell}^{k+1}\|^2.$$

By the previous inequality and Lemma 3.1-(iii), we can rewrite (3.12) as

$$\begin{aligned}
\|\mathbf{x}^{k+1} - \mathbf{z}\|^2 &\leq \sum_{\ell=1}^P w_{\ell} \left[\|\mathbf{x}^k - \mathbf{z}\|^2 - 2\lambda_k (F_{\ell}(\mathbf{x}^k) - F_{\ell}(\mathbf{z})) \right. \\
&\quad + 4\lambda_k^2 \sum_{s=2}^{m(\ell)} C_{i_s^{\ell}} \sum_{r=1}^{s-1} (C_{i_r^{\ell}} + \|\mathbf{e}_{i_r^{\ell}}^{k+1}\|) \\
&\quad - 2\lambda_k \left(\sum_{s=1}^{m(\ell)} \mathbf{e}_{i_s^{\ell}}^{k+1} \right)^T (\mathbf{x}^k - \mathbf{z}) \\
&\quad \left. + \lambda_k^2 m(\ell) \sum_{s=1}^{m(\ell)} \|\mathbf{g}_{i_s^{\ell}}^k + \mathbf{e}_{i_s^{\ell}}^{k+1}\|^2 \right],
\end{aligned}$$

and using the definition of F_{ℓ} in (3.11) we have

$$\begin{aligned}
\|\mathbf{x}^{k+1} - \mathbf{z}\|^2 &\leq \|\mathbf{x}^k - \mathbf{z}\|^2 - \frac{2}{P} \lambda_k (f(\mathbf{x}^k) - f(\mathbf{z})) \\
&\quad + 4\lambda_k^2 \sum_{\ell=1}^P w_{\ell} \sum_{s=2}^{m(\ell)} C_{i_s^{\ell}} \sum_{r=1}^{s-1} (C_{i_r^{\ell}} + \|\mathbf{e}_{i_r^{\ell}}^{k+1}\|) \\
&\quad - 2\lambda_k \left(\sum_{\ell=1}^P w_{\ell} \sum_{s=1}^{m(\ell)} \mathbf{e}_{i_s^{\ell}}^{k+1} \right)^T (\mathbf{x}^k - \mathbf{z}) \\
&\quad + \lambda_k^2 \sum_{\ell=1}^P w_{\ell} m(\ell) \sum_{s=1}^{m(\ell)} (\|\mathbf{g}_{i_s^{\ell}}^k\|^2 + 2(\mathbf{g}_{i_s^{\ell}}^k)^T \mathbf{e}_{i_s^{\ell}}^{k+1} + \|\mathbf{e}_{i_s^{\ell}}^{k+1}\|^2).
\end{aligned}$$

We can use again the Cauchy-Schwarz inequality, triangle inequality, besides Assumption 3 to obtain

$$\begin{aligned}
\|\mathbf{x}^{k+1} - \mathbf{z}\|^2 &\leq \|\mathbf{x}^k - \mathbf{z}\|^2 - \frac{2}{P} \lambda_k (f(\mathbf{x}^k) - f(\mathbf{z})) \\
&\quad + 4\lambda_k^2 \sum_{\ell=1}^P w_{\ell} \sum_{s=2}^{m(\ell)} C_{i_s^{\ell}} \sum_{r=1}^{s-1} (C_{i_r^{\ell}} + \|\mathbf{e}_{i_r^{\ell}}^{k+1}\|) \\
&\quad + 2\lambda_k \sum_{\ell=1}^P w_{\ell} \sum_{s=1}^{m(\ell)} \|\mathbf{e}_{i_s^{\ell}}^{k+1}\| \|\mathbf{x}^k - \mathbf{z}\| \\
&\quad + \lambda_k^2 \sum_{\ell=1}^P w_{\ell} m(\ell) \sum_{s=1}^{m(\ell)} (C_{i_s^{\ell}}^2 + 2C_{i_s^{\ell}} \|\mathbf{e}_{i_s^{\ell}}^{k+1}\| + \|\mathbf{e}_{i_s^{\ell}}^{k+1}\|^2). \tag{3.13}
\end{aligned}$$

Taking expectation with respect to G_k in the inequality (3.13) we have

$$\begin{aligned}
\mathbb{E}[\|\mathbf{x}^{k+1} - \mathbf{z}\|^2 | G_k] &\leq \|\mathbf{x}^k - \mathbf{z}\|^2 - \frac{2}{P} \lambda_k (f(\mathbf{x}^k) - f(\mathbf{z})) \\
&\quad + 4\lambda_k^2 \sum_{\ell=1}^P w_\ell \sum_{s=2}^{m(\ell)} C_{i_s^\ell} \sum_{r=1}^{s-1} (C_{i_r^\ell} + \mathbb{E}[\|\boldsymbol{\epsilon}_{i_r^\ell}^{k+1}\| | G_k]) \\
&\quad + 2\lambda_k \|\mathbf{x}^k - \mathbf{z}\| \sum_{\ell=1}^P w_\ell \sum_{s=1}^{m(\ell)} \mathbb{E}[\|\boldsymbol{\epsilon}_{i_s^\ell}^{k+1}\| | G_k] \\
&\quad + \lambda_k^2 \sum_{\ell=1}^P w_\ell m(\ell) \sum_{s=1}^{m(\ell)} (C_{i_s^\ell}^2 + 2C_{i_s^\ell} \mathbb{E}[\|\boldsymbol{\epsilon}_{i_s^\ell}^{k+1}\| | G_k] \\
&\quad + \mathbb{E}[\|\boldsymbol{\epsilon}_{i_s^\ell}^{k+1}\|^2 | G_k]), \tag{3.14}
\end{aligned}$$

Now we need to find an estimate for $\mathbb{E}[\|\boldsymbol{\epsilon}_{i_s^\ell}^{k+1}\| | G_k]$. On the one hand, Assumption 2 provides $\sqrt{\mathbb{E}[\|\boldsymbol{\epsilon}_{i_s^\ell}^{k+1}\|^2 | G_k]} \leq \nu_{k+1}$ holds with probability 1. On the other hand, concavity of the squared root and the Jensen inequality ensure that

$$\mathbb{E}[\|\boldsymbol{\epsilon}_{i_s^\ell}^{k+1}\| | G_k] = \mathbb{E}[\sqrt{\|\boldsymbol{\epsilon}_{i_s^\ell}^{k+1}\|^2} | G_k] \leq \sqrt{\mathbb{E}[\|\boldsymbol{\epsilon}_{i_s^\ell}^{k+1}\|^2 | G_k]} \leq \nu_{k+1}, \tag{3.15}$$

also holds with probability 1. Using Assumption 2 and inequality (3.15), we can rewrite (3.14) as

$$\begin{aligned}
\mathbb{E}[\|\mathbf{x}^{k+1} - \mathbf{z}\|^2 | G_k] &\leq \|\mathbf{x}^k - \mathbf{z}\|^2 - \frac{2}{P} \lambda_k (f(\mathbf{x}^k) - f(\mathbf{z})) \\
&\quad + 4\lambda_k^2 \sum_{\ell=1}^P w_\ell \sum_{s=2}^{m(\ell)} C_{i_s^\ell} \sum_{r=1}^{s-1} (C_{i_r^\ell} + \nu_{k+1}) \\
&\quad + 2\lambda_k \|\mathbf{x}^k - \mathbf{z}\| \sum_{\ell=1}^P w_\ell \sum_{s=1}^{m(\ell)} \nu_{k+1} \\
&\quad + \lambda_k^2 \sum_{\ell=1}^P w_\ell m(\ell) \sum_{s=1}^{m(\ell)} (C_{i_s^\ell}^2 + 2C_{i_s^\ell} \nu_{k+1} + \nu_{k+1}^2),
\end{aligned}$$

concluding that,

$$\begin{aligned}
\mathbb{E}[\|\mathbf{x}^{k+1} - \mathbf{z}\|^2 | G_k] &\leq \|\mathbf{x}^k - \mathbf{z}\|^2 - \frac{2}{P} \lambda_k (f(\mathbf{x}^k) - f(\mathbf{z})) \\
&\quad + 4\lambda_k^2 \sum_{\ell=1}^P w_\ell \sum_{s=2}^{m(\ell)} C_{i_s^\ell} \sum_{r=1}^{s-1} (C_{i_r^\ell} + \nu_{k+1}) \\
&\quad + 2 \left(\sum_{\ell=1}^P w_\ell m(\ell) \right) \lambda_k \nu_{k+1} \|\mathbf{x}^k - \mathbf{z}\| \\
&\quad + \lambda_k^2 \sum_{\ell=1}^P w_\ell m(\ell) \sum_{s=1}^{m(\ell)} (C_{i_s^\ell} + \nu_{k+1})^2,
\end{aligned}$$

holds with probability 1. □

3.3.3 Convergence results

We now present convergence results to the Algorithm 9 using diminishing step-sizes.

Proposition 3.4. Let $\{\lambda_k\}$ be a step-size sequence such that

$$\lambda_k \rightarrow 0^+ \quad \text{and} \quad \sum_{k=0}^{\infty} \lambda_k = \infty. \quad (3.16)$$

If Assumptions 1 and 2 hold and $\nu_k \rightarrow 0$, then we have that, with probability 1,

$$\liminf_{k \rightarrow \infty} f(\mathbf{x}^k) \leq f^*.$$

Proof. Suppose, for contradiction, that the proposition is false. In such a case, with positive probability, there will exist $\delta > 0$, $\mathbf{x}_\delta \in \mathbf{X}$ and $k_0 > 0$ such that $f(\mathbf{x}^k) - f(\mathbf{x}_\delta) \geq \delta$ for all $k \geq k_0$. Lemma 3.3 ensures that for all $\mathbf{z} \in \mathbf{X}$ we have that

$$\begin{aligned} \mathbb{E}[\|\mathbf{x}^{k+1} - \mathbf{z}\|^2 | G_k] &\leq \|\mathbf{x}^k - \mathbf{z}\|^2 - \frac{2}{P} \lambda_k (f(\mathbf{x}^k) - f(\mathbf{z})) \\ &\quad + 4\lambda_k^2 \sum_{\ell=1}^P w_\ell \sum_{s=2}^{m(\ell)} C_{i_s^\ell} \sum_{r=1}^{s-1} (C_{i_r^\ell} + \nu_{k+1}) \\ &\quad + 2C_\ell \lambda_k \nu_{k+1} \|\mathbf{x}^k - \mathbf{z}\| \\ &\quad + \lambda_k^2 \sum_{\ell=1}^P w_\ell m(\ell) \sum_{s=1}^{m(\ell)} (C_{i_s^\ell} + \nu_{k+1})^2, \end{aligned}$$

holds with probability 1. Taking $\mathbf{z} = \mathbf{x}_\delta$ we have that, for all $k \geq k_0$

$$\begin{aligned} \mathbb{E}[\|\mathbf{x}^{k+1} - \mathbf{x}_\delta\|^2 | G_k] &\leq \|\mathbf{x}^k - \mathbf{x}_\delta\|^2 - \frac{2}{P} \lambda_k \delta \\ &\quad + 4\lambda_k^2 \sum_{\ell=1}^P w_\ell \sum_{s=2}^{m(\ell)} C_{i_s^\ell} \sum_{r=1}^{s-1} (C_{i_r^\ell} + \nu_{k+1}) \\ &\quad + 2C_\ell \lambda_k \nu_{k+1} \|\mathbf{x}^k - \mathbf{x}_\delta\| \\ &\quad + \lambda_k^2 \sum_{\ell=1}^P w_\ell m(\ell) \sum_{s=1}^{m(\ell)} (C_{i_s^\ell} + \nu_{k+1})^2, \end{aligned} \quad (3.17)$$

holds with positive probability. Denoting by

$$\rho_k = \lambda_k \left[4 \sum_{\ell=1}^P w_\ell \sum_{s=2}^{m(\ell)} C_{i_s^\ell} \sum_{r=1}^{s-1} (C_{i_r^\ell} + \nu_{k+1}) + \sum_{\ell=1}^P w_\ell m(\ell) \sum_{s=1}^{m(\ell)} (C_{i_s^\ell} + \nu_{k+1})^2 \right]$$

and

$$\mu_k = 2\|\mathbf{x}^k - \mathbf{x}_\delta\| C_\ell \nu_{k+1},$$

these sequences have limits equal to zero (notice that the sequence $\|\mathbf{x}^k - \mathbf{x}_\delta\|$ is bounded due to Assumption 1 and (3.1)-(i)). Thus, there is $k_1 > k_0$ such that for all $k \geq k_1$ we have

$$\rho_k < \frac{\delta}{2P} \quad \text{and} \quad \mu_k < \frac{\delta}{2P}.$$

Now, (3.17) gives, with positive probability,

$$\mathbb{E}[\|\mathbf{x}^{k+1} - \mathbf{x}_\delta\|^2 | G_k] \leq \|\mathbf{x}^k - \mathbf{x}_\delta\|^2 - \lambda_k \frac{\delta}{P}.$$

Since $G_0 \subset G_1 \subset \dots \subset G_{k-1} \subset G_k$, it follows from the law of iterated expectations that

$$\mathbb{E}[\|\mathbf{x}^{k+1} - \mathbf{x}_\delta\|^2] = \mathbb{E}[\mathbb{E}[\|\mathbf{x}^{k+1} - \mathbf{x}_\delta\|^2 | G_0]] \leq \|\mathbf{x}^0 - \mathbf{x}_\delta\|^2 - \frac{\delta}{P} \sum_{r=0}^k \lambda_r.$$

The contradiction follows by taking the limit in both sides of the previous inequality, since $\sum_{r=0}^{\infty} \lambda_r = \infty$. $\square$

A mild assumption can improve the previous result.

Proposition 3.5. If in addition to the conditions established in Proposition 3.4, we have that $[f(\mathcal{P}_{\mathbf{X}}(\mathbf{x}^k)) - f(\mathbf{x}^k)]_+ \rightarrow 0$ holds with probability 1, then

$$\liminf_{k \rightarrow \infty} f(\mathbf{x}^k) = f^*,$$

with probability 1.

Proof. We can note that

$$\begin{aligned} f(\mathbf{x}^k) &= f(\mathcal{P}_{\mathbf{X}}(\mathbf{x}^k)) - [f(\mathcal{P}_{\mathbf{X}}(\mathbf{x}^k)) - f(\mathbf{x}^k)] \\ &\geq f(\mathcal{P}_{\mathbf{X}}(\mathbf{x}^k)) - [f(\mathcal{P}_{\mathbf{X}}(\mathbf{x}^k)) - f(\mathbf{x}^k)]_+ \\ &\geq f^* - [f(\mathcal{P}_{\mathbf{X}}(\mathbf{x}^k)) - f(\mathbf{x}^k)]_+ \end{aligned}$$

Since the right hand side of the previous inequality tends to f^* with probability 1, it is clear that $\lim_{k \rightarrow \infty} \inf f(\mathbf{x}^k) \geq f^*$ with probability 1. Together with Proposition 3.4, this provides the desired result. $\square$

Since Algorithm 9 does not use the Euclidean projection onto $\mathbf{X}$ to maintain the feasibility of the solutions, we need to show that the sequence $\{\mathbf{x}^k\}$ converges (in some sense) to the feasible set $\mathbf{X}$.

Proposition 3.6. Suppose that the Assumptions 1-3 hold. If $\{\nu_k\}$ is bounded, then $\mathbb{E}[d_{\mathbf{X}}^2(\mathbf{x}^k)] \rightarrow 0$.

Proof. Define

$$\mathbf{y}^k = \sum_{\ell=1}^P w_{\ell} \mathbf{x}_{\ell}^k, \quad \hat{d} = \sup_{k \geq 0} \{d_{\mathbf{X}}(\mathbf{x}^k)\}, \quad \text{and} \quad \nu = \sup_{k \geq 0} \{\nu_k\}.$$

The triangle inequality, Assumption 3 and Lemma 3.2 provide

$$\begin{aligned} \|\mathbf{y}^k - \mathbf{x}^k\|^2 &= \left\| \sum_{\ell=1}^P w_{\ell} [\mathbf{x}^k - \lambda_k \sum_{s=1}^{m(\ell)} (\mathbf{g}_{i_s^{\ell}}^k + \mathbf{e}_{i_s^{\ell}}^{k+1})] - \mathbf{x}^k \right\|^2 \\ &= \lambda_k^2 \left\| \sum_{\ell=1}^P w_{\ell} \sum_{s=1}^{m(\ell)} (\mathbf{g}_{i_s^{\ell}}^k + \mathbf{e}_{i_s^{\ell}}^{k+1}) \right\|^2 \\ &\leq \lambda_k^2 \left(\sum_{\ell=1}^P \sum_{s=1}^{m(\ell)} w_{\ell} (C_{i_s^{\ell}} + \|\mathbf{e}_{i_s^{\ell}}^{k+1}\|) \right)^2 \\ &\leq \lambda_k^2 m \sum_{\ell=1}^P \sum_{s=1}^{m(\ell)} [w_{\ell} (C_{i_s^{\ell}} + \|\mathbf{e}_{i_s^{\ell}}^{k+1}\|)]^2 \end{aligned}$$

Denoting $\nu = \sup_{k \geq 0} \{\nu_k\}$ and $\gamma = m \sum_{\ell=1}^P \sum_{s=1}^{m(\ell)} [w_{\ell} (C_{i_s^{\ell}} + \nu)]^2$, we have that

$$\mathbb{E}[\|\mathbf{y}^k - \mathbf{x}^k\|^2 | G_k] \leq \lambda_k^2 \gamma \quad (3.18)$$

holds with probability 1. In particular, by Jensen inequality, we have that

$$\mathbb{E}[\|\mathbf{y}^k - \mathbf{x}^k\| | G_k] \leq \lambda_k \sqrt{\gamma} \quad (3.19)$$

holds with probability 1. Let $\delta > 0$, we split our analysis in two cases.

- (1) $0 \leq \alpha = d_{\mathbf{X}}(\mathbf{y}^k) < \sqrt{\delta}$: in this case, Assumption 1 ensures that there is $\epsilon_{\alpha} > 0$ such that $\|\mathcal{H}(\mathbf{y}^k) - \mathbf{x}\|^2 \leq \|\mathbf{y}^k - \mathbf{x}\|^2 - \epsilon_{\alpha}$ for all $\mathbf{x} \in \mathbf{X}$. Taking $\mathbf{x} = \mathcal{P}_{\mathbf{X}}(\mathbf{y}^k)$ we have

$$\begin{aligned} d_{\mathbf{X}}^2(\mathbf{x}^{k+1}) &\leq \|\mathbf{x}^{k+1} - \mathcal{P}_{\mathbf{X}}(\mathbf{y}^k)\|^2 \\ &\leq \alpha^2 - \epsilon_{\alpha} \\ &< \delta - \epsilon_{\alpha}, \end{aligned}$$

and thus, $\mathbb{E}[d_{\mathbf{X}}^2(\mathbf{x}^{k+1})] < \delta - \epsilon_{\alpha}$.

- (2) $d_{\mathbf{X}}(\mathbf{y}^k) \geq \sqrt{\delta}$: notice that

$$\begin{aligned} d_{\mathbf{X}}^2(\mathbf{y}^k) &= \|\mathbf{y}^k - \mathcal{P}_{\mathbf{X}}(\mathbf{y}^k)\|^2 \\ &\leq \|\mathbf{y}^k - \mathcal{P}_{\mathbf{X}}(\mathbf{x}^k)\|^2 \\ &\leq (\|\mathbf{y}^k - \mathbf{x}^k\| + d_{\mathbf{X}}(\mathbf{x}^k))^2 \\ &\leq d_{\mathbf{X}}^2(\mathbf{x}^k) + 2\|\mathbf{y}^k - \mathbf{x}^k\|\hat{d} + \|\mathbf{y}^k - \mathbf{x}^k\|^2. \end{aligned}$$

Taking the conditional expectation with respect to G_k , we have, due to (3.18) and (3.19) that

$$\begin{aligned}\mathbb{E}[d_{\mathbf{X}}^2(\mathbf{y}^k) | G_k] &\leq d_{\mathbf{X}}^2(\mathbf{x}^k) + 2\lambda_k\sqrt{\gamma}\hat{d} + \lambda_k^2\gamma, \\ &\leq d_{\mathbf{X}}^2(\mathbf{x}^k) + M\lambda_k,\end{aligned}\tag{3.20}$$

holds with probability 1, where $M = 2\sqrt{\gamma}\hat{d} + \gamma \sup_{k \geq 0} \{\lambda_k\}$. Assumption 1 guarantees that there is $\epsilon_{\sqrt{\delta}} > 0$, such that $\|\mathcal{H}(\mathbf{y}^k) - \mathbf{x}\|^2 \leq \|\mathbf{y}^k - \mathbf{x}\|^2 - \epsilon_{\sqrt{\delta}}$ for all $\mathbf{x} \in \mathbf{X}$. Taking $\mathbf{x} = \mathcal{P}_{\mathbf{X}}(\mathbf{y}^k)$ we have

$$\begin{aligned}d_{\mathbf{X}}^2(\mathbf{x}^{k+1}) &\leq \|\mathbf{x}^{k+1} - \mathcal{P}_{\mathbf{X}}(\mathbf{y}^k)\|^2 \\ &\leq \|\mathbf{y}^k - \mathcal{P}_{\mathbf{X}}(\mathbf{y}^k)\|^2 - \epsilon_{\sqrt{\delta}} \\ &= d_{\mathbf{X}}^2(\mathbf{y}^k) - \epsilon_{\sqrt{\delta}}\end{aligned}\tag{3.21}$$

We can take $k_1 > 0$ such that $k > k_1$ implies that $M\lambda_k < \epsilon_{\sqrt{\delta}}/2$. Thus, applying (3.20), we can rewrite (3.21) as

$$\mathbb{E}[d_{\mathbf{X}}^2(\mathbf{x}^{k+1}) | G_k] < d_{\mathbf{X}}^2(\mathbf{x}^k) + \frac{\epsilon_{\sqrt{\delta}}}{2} - \epsilon_{\sqrt{\delta}} = d_{\mathbf{X}}^2(\mathbf{x}^k) - \frac{\epsilon_{\sqrt{\delta}}}{2},$$

that holds with probability 1. Hence, taking expectation in the above inequality, we have

$$\mathbb{E}[d_{\mathbf{X}}^2(\mathbf{x}^{k+1})] < \mathbb{E}[d_{\mathbf{X}}^2(\mathbf{x}^k)] - \frac{\epsilon_{\sqrt{\delta}}}{2}.$$

Therefore, the results obtained in the two cases above imply that, for any $\delta > 0$, there exists k_1 such that $\mathbb{E}[d_{\mathbf{X}}^2(\mathbf{x}^k)] < \delta$, which proves the result. $\square$

3.4 Numerical experiments

In this section we test and compare our method with a well-known stochastic subgradient method in a simulated numerical experiment.

In order to perform the tests under more general conditions, we do not require here that $\nu_k \rightarrow 0$, so that we weaken the conditions of Proposition 3.4. Instead, we take random variables such that the sequence $\{\mathbb{E}[\|\epsilon_i^{k+1}\|^2 | G_k]\}$ is bounded. Moreover, as it is not always trivial to verify that $[f(\mathcal{P}(\mathbf{x}^k)) - f(\mathbf{x}^k)]_+ \rightarrow 0$ with probability 1, as prescribed by Proposition 3.5, we simply skip checking this condition. This constitutes a more general setting that does not allow to conclude on convergence from our theoretical results. Even though, the numerical results given in Section 3.4.2.2 are satisfactory.

The settings of our experiment are inspired on a real world problem with many applications, as we present next.

3.4.1 Motivation

The *Fair Rate Allocation Problem* (FRAP) is a well-known problem that consists in allocating resources (rates) in a network that can be described by a directed graph $\mathcal{G}(\mathcal{N}, \mathcal{L})$, where $\mathcal{N}$ represents a set of nodes and $\mathcal{L}$ denotes a set of directed links (arcs, communications) of $\mathcal{G}$. Usually, an user (represented by a node in $\mathcal{N}$) expects to receive a resource from any other source node in $\mathcal{N}$. This transmission occurs by means of *flows*, that are characterized by a source node $b(i)$, a destination node (user) $e(i)$ and a subset of links in $\mathcal{L}$ used to compose the flow route. The *rate vector* $\mathbf{x}$ contains the allocated rate of some resource for each flow $i = 1, \dots, n$.

There are different approaches to deal with FRAP. Some authors investigate how to obtain a “fair” rate vector that provides resources for all users, without any of them being harmed and respecting some capacity constraints of the links. The notion of fairness characterizes how competing users should share the bottleneck resources (BERTSEKAS; GALLAGER, 1987; JAFFE, 1981; MO; WALRAND, 2000). Another approach considers the tradeoff between fairness and *efficiency*. On one hand, the fairness concept tries to establish whether the network provides a fair share of system resources. On the other hand, these models try to maximize an *utility function*, that is used as a metric to evaluate how efficiently the informations are exchanged in the network. In general, these utility functions are concave and twice differentiable over the capacity region of the network (see, e.g., (ERYILMAZ; SRIKANT, 2006)). However, in (ZUKERMAN *et al.*, 2008) an utility function that is non-smooth and non-concave is proposed and a degree of fairness is imposed explicitly by requiring an assumption called (α, β) -*fairness* to bound the flows rates x_i , for all $i = 1, \dots, n$. In (ZUKERMAN *et al.*, 2008) it is stated that bounded utility functions generally provide a better reflex of the operator’s true profit from the allocation process and these types of utility functions are more realistic, for example, in an inefficient market scenario. Furthermore, the (α, β) -fairness concept is used together with capacity constraints to compose a non-empty, closed, bounded and convex feasible set $\mathbf{X}$.

Such problems can also be stochastic. For example, consider modeling an intruder detection situation by penalizing the objective function in the flows that have some degree (possibility) of intrusion. In these cases, we may consider the following maximization problem:

$$\begin{aligned} \text{Maximize} \quad & \mathbb{E}[U(\mathbf{x}, \boldsymbol{\theta})] \\ \text{s.t.} \quad & \mathbf{x} \in \mathbf{X} \subset \mathbb{R}^n, \end{aligned} \tag{3.22}$$

where $\boldsymbol{\theta}$ is a random vector. From the ideas in (ZUKERMAN *et al.*, 2008), the feasible

set $\mathbf{X}$ could be visualized of the following manner: $\mathbf{x} \in \mathbf{X}$ if and only if

$$A\mathbf{x} \leq \mathbf{C}, \quad A \in \{0,1\}^{r \times n}, \quad (3.23)$$

$$\alpha[\boldsymbol{\psi}]_i \leq x_i \leq \beta[\boldsymbol{\psi}]_i, \quad i = 1, \dots, n, \quad (3.24)$$

where A contains the coefficients of the linear capacity constraints, r denotes the number of constraints (this number must coincide with the quantity of directed links of the network) and the vector $\mathbf{C}$ describes the capacities $C_1, \dots, C_r$ of each link. Therefore, constraints described in (3.23) indicate that the total amount of flow rates, related to the flows that goes through a link j , can not be greater than C_j , for all $j = 1, \dots, r$. The bound constraints in (3.24) reflect a relaxation of the fairness concept, where $\boldsymbol{\psi}$ is the *max-min fair vector* that is obtained from the network topology data (A and $\mathbf{C}$) and the values α and β represent the parameters of the relaxation. There are several techniques for calculating the vector $\boldsymbol{\psi}$ (see, e.g., (BERTSEKAS; GALLAGER, 1987; JAFFE, 1981; DANNA; MANDAL; SINGH, 2012)).

3.4.2 The numerical experiment

We test our method in a constrained convex stochastic optimization problem that is inspired by the problem we mentioned in Section 3.4.1. For comparison purposes, we built several instances of the problem, each of them containing randomly generated constraints.

3.4.2.1 Experimental setup

a) *The optimization problem:* We consider the following objective function

$$f(\mathbf{x}) = \sum_{i=1}^n \mathbb{E}[F_i(\mathbf{x}, \boldsymbol{\theta})], \quad (3.25)$$

with

$$F_i(\mathbf{x}, \boldsymbol{\theta}) := \min_{q=1, \dots, Q} \{a_q \ln(b_q \mathbf{e}_i^T \mathbf{x} + c_q) - d_q \mathbf{e}_i^T \boldsymbol{\theta} \mathbf{e}_i^T \mathbf{x}\}, \quad (3.26)$$

where $\boldsymbol{\theta}$ is a random vector such that each element $[\boldsymbol{\theta}]_i$, $i = 1, \dots, n$, has a uniform probability distribution between zero and one, i.e., $[\boldsymbol{\theta}]_i \sim \mathcal{U}(0,1)$ for all $i = 1, \dots, n$. Further, $\mathbf{e}_i$ is the n -vector such that $[\mathbf{e}_i]_j = 1$ if $j = i$ and $[\mathbf{e}_i]_j = 0$ otherwise. The numbers a_q, b_q, c_q and d_q are positive for all $q = 1, \dots, Q$, what, together with the definition of $\mathbf{X}$ that will be provided below, make the functions F_i in (3.26) be well-defined. Since the pointwise minimum of concave functions is concave, then $F_i(\cdot, \boldsymbol{\theta})$ is concave on $\mathbf{X}$, for all $\boldsymbol{\theta} \in \mathbb{R}^n$. In this experiment, we consider the following minimization problem:

$$\begin{aligned} &\text{Minimize} && -f(\mathbf{x}) \\ &\text{s.t.} && \mathbf{x} \in \mathbf{X} \subset \mathbb{R}^n, \end{aligned} \quad (3.27)$$

where $\mathbf{X}$ is non-empty, convex and compact.

In comparison to (3.1) with conditions (i)-(v), model (3.25)-(3.27) suggests constant weights $w_\ell = 1/P$ for all $\ell = 1, \dots, P$ to satisfy conditions (iv) and (v). To define the sets S_ℓ , we ordered the indices $\{1, \dots, n\}$ randomly and then distributed in P equally sized sets (or as close to it as possible) aiming at satisfying condition (iii). Furthermore, we choose $Q = 4$ and a_q, b_q, c_q, d_q as in the Table 2.

q	1	2	3	4
a_q	4.0	6.0	5.0	3.5
b_q	1.0	1.2	0.9	1.5
c_q	1.0	1.2	1.0	1.1
d_q	1.0	0.5	0.7	0.9

Table 2 – Parameters of the objective function used in tests for SAISSA.

b) *Feasible set $\mathbf{X}$ and instances*: The feasible set $\mathbf{X}$ we propose is obtained from inequalities such as (3.23) and (3.24). Then, let us now explain how we choose the matrix $A_{r \times n}$, $\mathbf{C}$, $\boldsymbol{\psi}$ (that depends on A and $\mathbf{C}$), α and β . Denote n_{inst} as the number of a particular instance of the problem. We choose $n_{\text{inst}} = 1, \dots, 1000$. Each instance has an influence on the dimension of the problem:

$$n(n_{\text{inst}}) = 500 + 5(n_{\text{inst}} - 1). \quad (3.28)$$

Using (3.28), we define for each $n_{\text{inst}} = 1, \dots, 1000$

$$r(n_{\text{inst}}) = \left\lceil \frac{\left\lceil \sqrt{n(n_{\text{inst}})}(X + 1) \right\rceil \left(\left\lceil \sqrt{n(n_{\text{inst}})}(X + 1) \right\rceil - 1 \right)}{10} \right\rceil,$$

where $X \sim \mathcal{U}(0,1)$. Let $\mathbf{a}_j$ be the j -th row of A . Define

$$\kappa_j(n_{\text{inst}}) = \left\lceil \frac{n(n_{\text{inst}})}{r(n_{\text{inst}})} \right\rceil + Y, \quad j = 1, \dots, r,$$

where Y is a random variable with discrete uniform distribution between zero and $3 \left\lceil \frac{n(n_{\text{inst}})}{r(n_{\text{inst}})} \right\rceil$, i.e., $Y \sim \mathcal{U} \left\{ 0, 3 \left\lceil \frac{n(n_{\text{inst}})}{r(n_{\text{inst}})} \right\rceil \right\}$. We define $\kappa_j(n_{\text{inst}})$ as the minimal number of entries $[\mathbf{a}_j]_i = 1$ for each $j = 1, \dots, r(n_{\text{inst}})$. Next, we need to determine, for each $j = 1, \dots, r(n_{\text{inst}})$, the $\kappa_j(n_{\text{inst}})$ indices $i \in \{1, \dots, n(n_{\text{inst}})\}$ such that $[\mathbf{a}_j]_i = 1$. These selections were taken randomly, i.e., we make $\kappa_j(n_{\text{inst}})$ entries $[\mathbf{a}_j]_i = 1$ and $n(n_{\text{inst}}) - \kappa_j(n_{\text{inst}})$ entries $[\mathbf{a}_j]_i = 0$ for each $j = 1, \dots, r(n_{\text{inst}})$ and the positions $i \in \{1, \dots, n(n_{\text{inst}})\}$ in which $[\mathbf{a}_j]_i = 1$ are obtained randomly. We also require that for each $j = 1, \dots, r(n_{\text{inst}})$, there is at least one index $i \in \{1, \dots, n(n_{\text{inst}})\}$ such that $[\mathbf{a}_j]_i = 1$, which is done by picking an index i at random and setting $[\mathbf{a}_j]_i = 1$ when necessary. This completes the definition of matrix A . Regarding the vector $\mathbf{C}$, we define $C_j \sim \mathcal{U}\{1, 10\}$ for all $j = 1, \dots, r(n_{\text{inst}})$.

In each instance n_{inst} of our experiments, the vector $\boldsymbol{\psi}$ was computed by the *max-min fair algorithm* described in (BERTSEKAS; GALLAGER, 1987), p.451-452. Furthermore, for each n_{inst} , we set $\alpha \sim \mathcal{U}(0, 0.9)$ and $\beta \sim \mathcal{U}(1.1, 32)$.

c) *Stochastic subgradients*: In order to obtain a stochastic subgradient, we define for each $q = 1, \dots, Q$ and $i = 1, \dots, n$

$$h_{q,i}(\mathbf{x}, \boldsymbol{\theta}) = -a_q \ln(b_q \mathbf{e}_i^T \mathbf{x} + c_q) + d_q \mathbf{e}_i^T \boldsymbol{\theta} \mathbf{e}_i^T \mathbf{x}.$$

The $h_{q,i}$ functions are differentiable with respect to $\mathbf{x}$ and

$$\nabla_{\mathbf{x}} h_{q,i}(\mathbf{x}, \boldsymbol{\theta}) = \left(-\frac{a_q b_q}{b_q \mathbf{e}_i^T \mathbf{x} + c_q} + d_q \mathbf{e}_i^T \boldsymbol{\theta} \right) \mathbf{e}_i.$$

Corollary 4.3.2 in (HIRIART-URRUTY; LEMARÉCHAL, 1993c) states that

$$\partial_{\mathbf{x}}[-F_i(\mathbf{x}, \boldsymbol{\theta})] = \text{conv}\{\cup \nabla_{\mathbf{x}} h_{q,i}(\mathbf{x}, \boldsymbol{\theta}) \mid q \in I_{q,i}(\mathbf{x}, \boldsymbol{\theta})\},$$

where $I_{q,i}(\mathbf{x}, \boldsymbol{\theta}) = \{q \mid h_{q,i}(\mathbf{x}, \boldsymbol{\theta}) = -F_i(\mathbf{x}, \boldsymbol{\theta})\}$. For each $i = 1, \dots, n$, vectors $\nabla_{\mathbf{x}} h_{q,i}(\mathbf{x}, \boldsymbol{\theta})$ have the same direction than $\mathbf{e}_i$ for all $q = 1, \dots, Q$. Hence, $\partial_{\mathbf{x}}[-F_i(\mathbf{x}, \boldsymbol{\theta})]$ is a line segment joining the outermost two points $\nabla_{\mathbf{x}} h_{q,i}(\mathbf{x}, \boldsymbol{\theta})$, so that $q \in I_{q,i}(\mathbf{x}, \boldsymbol{\theta})$. Thus, we take

$$\tilde{\mathbf{g}}_i := \nabla_{\mathbf{x}} h_{q,i}(\mathbf{x}, \boldsymbol{\theta}) \in \partial_{\mathbf{x}}[-F_i(\mathbf{x}, \boldsymbol{\theta})],$$

by randomly choosing $q \in I_{q,i}(\mathbf{x}, \boldsymbol{\theta})$. Making $f_i(\mathbf{x}) := \mathbb{E}[-F_i(\mathbf{x}, \boldsymbol{\theta})]$ and assuming that $\mathbf{g}_i := \mathbb{E}[\tilde{\mathbf{g}}_i] \in \partial f_i(\mathbf{x}) = \mathbb{E}[\partial_{\mathbf{x}}[-F_i(\mathbf{x}, \boldsymbol{\theta})]]$ (cf. Proposition 2.2 in (BERTSEKAS, 1973)), we can define $\boldsymbol{\epsilon}_i = \tilde{\mathbf{g}}_i - \mathbf{g}_i$ and thus the subgradients $\tilde{\mathbf{g}}_i$ have the same format as those used in the Algorithm 9.

d) *The algorithm*: We use the expression SAISSA- P to emphasize that we run SAISSA with P strings, making $\Delta_1 = S_1, \dots, \Delta_P = S_P$. The step-sizes used in SAISSA- P are computed using the following procedure:

$$\lambda_k = \frac{\lambda}{(k+1)^{\gamma_P}}, \quad (3.29)$$

$$\lambda = \mu_P P \phi \left(\frac{\left| -\sum_{i=1}^{n(n_{\text{inst}})} F_i(\mathbf{x}^0, \boldsymbol{\theta}) \right|}{\|\tilde{\mathbf{g}}^0\|^2} \right), \quad (3.30)$$

where $\tilde{\mathbf{g}}^0 \in \partial_{\mathbf{x}}[-\sum_{i=1}^{n(n_{\text{inst}})} F_i(\mathbf{x}^0, \boldsymbol{\theta})]$ and $\phi(\sigma(\boldsymbol{\theta}))$ computes the empirical mean of the random variable $\sigma(\boldsymbol{\theta})$, that is, $\phi(\sigma(\boldsymbol{\theta})) := \frac{1}{n_{\text{samples}}} \sum_{t=1}^{n_{\text{samples}}} \sigma(\boldsymbol{\theta}_t)$, where each $\boldsymbol{\theta}_t$ is a sample of the random variable $\boldsymbol{\theta}$. Notice that $\mathbf{x}^0$ is a deterministic vector and in our tests it was determined by equation

$$\mathbf{x}^0 = \mathcal{P}_{\mathbf{X}}(0.5(\alpha \boldsymbol{\psi} + \beta \boldsymbol{\psi})). \quad (3.31)$$

Parameters μ_P and γ_P were determined through the scheme that follows. Let $\mathbf{x}_{(P,\mu,\gamma)}^{\bar{k}}$ be the output obtained after a fixed number $\bar{k}$ of iterations by SAISSA- P using parameters (μ, γ) . Define

$$\bar{f}(\hat{\mathbf{x}}, \boldsymbol{\theta}) = \frac{1}{N} \sum_{t=1}^N \sum_{i=1}^{n(\text{inst})} -F_i(\hat{\mathbf{x}}, \boldsymbol{\theta}_t), \quad (3.32)$$

where $\hat{\mathbf{x}}$ is the approximate solution generated by the algorithm in question (in this case SAISSA- P). Then (μ_P, γ_P) was given by

$$(\mu_P, \gamma_P) := \underset{(\mu, \gamma) \in \{0.1, 0.2, \dots, 1.0\} \times \{0.5, 0.6, \dots, 0.9\}}{\operatorname{argmin}} \bar{f}(\mathbf{x}_{(P,\mu,\gamma)}^{\bar{k}}, \boldsymbol{\theta}^{\bar{k}}). \quad (3.33)$$

We set $\bar{k} = 50$ and, in order to avoid a high computational cost, we consider a small number of samples to determine $\phi(\sigma(\boldsymbol{\theta}))$ and $\bar{f}(\mathbf{x}_{(P,\mu,\gamma)}^{\bar{k}}, \boldsymbol{\theta}^{\bar{k}})$: we use $N = n_{\text{samples}} = 100$. We also perform a simple pre-processing to tune the number of strings P used in SAISSA- P . Basically, we run SAISSA- P for $P = 1, \dots, 10$ using $\hat{k} = 75$ iterations and the step-size policy defined in (3.29), (3.30) and (3.33). In this sense we define,

$$\hat{P} = \underset{P \in \{1, 2, \dots, 10\}}{\operatorname{argmin}} \bar{f}(\mathbf{x}_P^{\hat{k}}, \boldsymbol{\theta}^{\hat{k}}),$$

where $\mathbf{x}_P^{\hat{k}}$ is the output obtained by SAISSA- P . In this case, to compute $\bar{f}(\mathbf{x}_P^{\hat{k}}, \boldsymbol{\theta}^{\hat{k}})$, we use $N = 500$.

It remains to define the operator $\mathcal{H}$ used in equation (3.8) of the Algorithm 9. For that, we define for all $j = 1, \dots, r$ the functions

$$c_j(\mathbf{x}) = \mathbf{a}_j^T \mathbf{x} - C_j, \quad \text{with} \quad \nabla c_j(\mathbf{x}) = \mathbf{a}_j \neq \mathbf{0}.$$

The main component to define $\mathcal{H}$ is derived of subgradient method proposed by Polyak (POLYAK, 1969; SHOR, 1985), which generates a sequence that, from any $\mathbf{x}^0 \in \mathbb{R}^n$, converges to a point at the sublevel set of a given convex function defined on $\mathbb{R}^n$. We define such component for each $j = 1, \dots, r(n_{\text{inst}})$ as a function $\mathcal{S}_{c_j} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ given by

$$\mathcal{S}_{c_j}(\mathbf{x}) := \mathbf{x} - \frac{[c_j(\mathbf{x})]_+}{\|\mathbf{a}_j\|^2} \mathbf{a}_j, \quad (3.34)$$

where $[x]_+ = \max\{x, 0\}$ with $x \in \mathbb{R}$. Our operator $\mathcal{H} : \mathbb{R}^n \rightarrow \mathbf{Y}$ is defined as a sequential application of $\mathcal{S}_{c_j}$:

$$\mathcal{H} := \mathcal{P}_{\mathbf{Y}} \circ \mathcal{S}_{c_r} \circ \dots \circ \mathcal{S}_{c_2} \circ \mathcal{S}_{c_1}, \quad (3.35)$$

where $\mathbf{Y} = B(\alpha, \beta, \boldsymbol{\psi})$ is the set of vectors that satisfy the bound constraints defined in (3.24). We need to check that $\mathcal{H}$ defined in (3.35) satisfies Assumption 1. If $\mathbf{z} \in \mathbf{X}$ then it is clear that $\mathcal{H}(\mathbf{z}) = \mathbf{z}$ because $c_j(\mathbf{z}) \leq 0$ for all j and $\mathcal{P}_{B(\alpha, \beta, \boldsymbol{\psi})}(\mathbf{z}) = \mathbf{z}$. It is easy to see that $\mathcal{P}_{B(\alpha, \beta, \boldsymbol{\psi})} = \mathcal{S}_{d_{B(\alpha, \beta, \boldsymbol{\psi})}}$ where $d_{B(\alpha, \beta, \boldsymbol{\psi})}(\mathbf{x}) = \|\mathbf{x} - \mathcal{P}_{B(\alpha, \beta, \boldsymbol{\psi})}(\mathbf{x})\|$ and in this way $\mathcal{H}$ can be rewritten as

$$\mathcal{H} := \mathcal{S}_{c_{r+1}} \circ \mathcal{S}_{c_r} \circ \dots \circ \mathcal{S}_{c_2} \circ \mathcal{S}_{c_1},$$

where $c_{r+1} := d_{B(\alpha, \beta, \psi)}$. Since the set $\{\mathbf{x} \in \mathbb{R}^n \mid c_{r+1}(\mathbf{x}) \leq 0\} = B(\alpha, \beta, \psi)$ is bounded and $\|\nabla c_j\| = \|\mathbf{a}_j\| \leq \sqrt{n}$ for all $j = 1, \dots, r$, then Proposition 3.4 in (NETO; PIERRO, 2009) ensures that $\mathcal{H}$ satisfies Assumption 1.

The two versions of the *robust mirror descent stochastic approximation algorithm* (NEMIROVSKI *et al.*, 2009), named as *Euclidean - Stochastic Approximation* (E-SA) and *Non-Euclidean - Stochastic Approximation* (N-SA) are prox-type algorithms that use stochastic subgradient directions to update each iteration. The main difference between them is in the distance generating function used to compose the prox-operator. For both E-SA and N-SA, the expected inaccuracy, in terms of $\bar{f}(\hat{\mathbf{x}}, \theta)$, of the approximate solution $\hat{\mathbf{x}}$ obtained after $k_{\max}$ iterations is $O(1/\sqrt{k_{\max}})$.

The pre-processing used to choose all parameters of E-SA and N-SA was made as in (NEMIROVSKI *et al.*, 2009, Section 4.1). Parameters used in the step-sizes rules were computed using NLOpt - a free/open-source library for nonlinear optimization - using the globally-convergent Method of Moving Asymptotes (MMA) for gradient-based local optimization (SVANBERG, 2002). The settings of our problem require that $\mathcal{P}_{\mathbf{x}}$ be computed in each iteration of E-SA and N-SA. In this sense, we use a state-of-the-art method to compute them aiming to make the tests least time-consuming and the algorithms most competitive as possible. To compute the projections $\mathcal{P}_{\mathbf{x}}$, we use the Gurobi optimizer mathematical programming solver ¹.

All described algorithms SAISSA- P , E-SA and N-SA were implemented in C++ and executed using resources of the computational cluster Euler ² of the Centre of Mathematics and Statistics Applied to Industry - CeMEAI.

3.4.2.2 Comparison of the methods

The obtained results were analyzed by means of *performance profiles* (DOLAN; MORÉ, 2002), using as a performance measure the values $\bar{f}(\hat{\mathbf{x}}, \theta)$, defined in (3.32). For that, we define the *performance ratio*

$$R_{n_{\text{inst}}}^{\text{sol}} = \frac{T_{n_{\text{inst}}}^{\text{sol}}}{\min_{\text{sol}} \{T_{n_{\text{inst}}}^{\text{sol}}\}}, \quad (3.36)$$

where $T_{n_{\text{inst}}}^{\text{sol}} = \bar{f}(\hat{\mathbf{x}}_{n_{\text{inst}}}^{\text{sol}}, \theta)$ and $\hat{\mathbf{x}}_{n_{\text{inst}}}^{\text{sol}}$ is the approximate solution obtained for the problem n_{inst} by solver $\text{sol} \in \{\text{SAISSA-}P\text{-}k_{\max}, \text{E-SA-}k_{\max}, \text{N-SA-}k_{\max}\}$, with $k_{\max} \in \{2000, 5000\}$. The empirical means $\bar{f}(\hat{\mathbf{x}}_{n_{\text{inst}}}^{\text{sol}}, \theta)$ are taken over a large ($N = 10^4$) sample set. It is important to notice the possible ranges over which $R_{n_{\text{inst}}}^{\text{sol}}$ is defined in (3.36). Our tests show that the values of $T_{n_{\text{inst}}}^{\text{sol}}$ are negative for all n_{inst} and sol (see (3.26) and (3.27)). In

¹ www.gurobi.com

² www.cemeai.icmc.usp.br/Euler

order to maintain the conventional format of the performance profiles, we apply the following transformation to the data: given $T_{n_{\text{inst}}}^{\text{sol}}$, we use

$$\hat{T}_{n_{\text{inst}}}^{\text{sol}} = \ln(T_{n_{\text{inst}}}^{\text{sol}} + 1.1 - \min_{\text{sol}}\{T_{n_{\text{inst}}}^{\text{sol}}\}),$$

and

$$\hat{R}_{n_{\text{inst}}}^{\text{sol}} = \frac{\hat{T}_{n_{\text{inst}}}^{\text{sol}}}{\min_{\text{sol}}\{\hat{T}_{n_{\text{inst}}}^{\text{sol}}\}}.$$

The performance profile of the solver sol is defined as

$$\rho_{\text{sol}}(\tau) = \frac{1}{1000} \text{size}\{n_{\text{inst}} = 1, \dots, 1000 \mid \hat{R}_{n_{\text{inst}}}^{\text{sol}} \leq \tau\}, \quad \tau \in \mathbb{R}.$$

Note that $\rho_{\text{sol}}(\tau)$ represents the probability that solver sol has a performance ratio $R_{n_{\text{inst}}}^{\text{sol}}$ within a factor τ of the best possible ratio. The function ρ_{sol} is the estimated (cumulative) distribution function for the performance ratio. Notice also that, when we are interested in the number of wins of a solver, we need to compare the values of $\rho_{\text{sol}}(1)$ for all of the solvers.

Figure 19 – Performance profiles of the tested methods by using $k_{\text{max}} = 2000$.

Figure 19 and Figure 20 show the performance profiles of $\text{SAISSA-}\hat{P}$, E-SA and N-SA over the set of 1000 instances of the problem discussed in Section 3.4.2.1, by using $k_{\text{max}} = 2000$ and $k_{\text{max}} = 5000$, respectively. We can note that the performance of the methods in both cases is very similar. When $k_{\text{max}} = 2000$, $\text{SAISSA-}\hat{P}$ has the most

Figure 20 – Performance profiles of the tested methods by using $k_{\max} = 5000$.

Figure 21 – Computational time of the tested methods by using $k_{\max} = 5000$.

wins with probability about 0.68 of being the winner on a given problem, whereas the same probability for E-SA is about 0.32. The probability of N-SA winning in some instance is almost zero. Moreover, N-SA can be used if we are interested in solutions with a factor $\tau \geq 65$. When $k_{\max} = 5000$, Figure 20 shows that the probability of being the winner on a given problem is about 0.72 for SAISSA- $\hat{P}$ and 0.28 for E-SA.

SAISSA- $\hat{P}$ is executed in an inferior computational time than E-SA and N-SA for all $n_{\text{inst}} = 1, \dots, 1000$, even taking into account the computational time to compute $\hat{P}$. Figure 21 shows the computational time in logarithmic scale for the methods with $k_{\max} = 5000$. For SAISSA- $\hat{P}$, we consider the pre-processing time to compute $\hat{P}$ and we do not consider, already in the processing step, neither the computational time to compute $(\mu_{\hat{P}}, \gamma_{\hat{P}})$ in (3.33) nor the time to obtain the step-size policy and parameters for E-SA and N-SA.

3.5 Final comments

We have presented a new method to solve non-smooth convex stochastic optimization problems, based on stochastic incremental subgradients and string-averaging techniques, which has shown to be promising in large scale problems and with a large number of constraints, that is, in problems such that the calculation of the Euclidean projection onto the feasible set is not simple. As in the deterministic approach analyzed in (OLIVEIRA; HELOU; COSTA, 2016), our method here stands out because it can perform a certain number of iterations in a smaller time range when compared to other classical methods (sequential/cyclic, E-SA or N-SA in these cases) showing itself to be competitive compared to the general stochastic subgradient methods. Future research can be carried out by applying the string-averaging techniques discussed in this paper to Markov randomized incremental subgradient methods (such as the one explored in (RAM; NEDIĆ; VEERAVALLI, 2009)), applied to large scale constrained stochastic optimization problems.

Acknowledgements

Rafael M. de Oliveira was funded by FAPESP Grant 2015/10171-2. Elias S. Helou Neto was partially funded by FAPESP Grants 2013/16508-3 and 2013/07375-0 and CNPq Grant 311476/2014-7. Eduardo F. Costa was funded by FAPESP under Grant 13/19380-8 and CNPq under Grant 311290/2013-2. Research developed using the computational resources of the Centre of Mathematics and Statistics Applied to Industry (CeMEAI) funded by FAPESP.

CONCLUSION

We have presented new methods to deal with deterministic/stochastic convex optimization problems. We have shown by means of numerical experiments how these methods can be promising, once that they benefit from parallel processing. In both deterministic and stochastic cases, we provide some convergence results under reasonable assumptions. However, more research can be carry out, for example, to determine bounds on the error $f(\mathbf{x}^k) - f^*$ for this class of methods.

Future research will be carried out by the same group involved in this thesis, focusing on incremental subgradient methods and string-averaging algorithms that use time-homogeneous Markov chains to choose the index i to be used to update each iteration. Basically, we will analyze iterations of the type:

$$\mathbf{x}^{k+1} = M_k(\mathbf{x}^k, \mathbf{s}(k+1)), \quad \mathbf{s}(k+1)^T = [s_1(k+1) \dots s_N(k+1)], \quad (4.1)$$

where $s_1(k+1), \dots, s_N(k+1)$ are time-homogeneous Markov chains with states in $S_1, \dots, S_N$, respectively and $M_k : \mathbb{R}^n \times (S_1 \times \dots \times S_N) \rightarrow \mathbf{Y} \supset \mathbf{X}$ is given by

$$M_k(\mathbf{x}, \mathbf{i}) = \mathcal{H} \left(\sum_{\ell=1}^N w_\ell (\mathbf{x} - \lambda_k \tilde{\mathbf{g}}_{i_\ell}^k) \right) \quad (4.2)$$

where $\tilde{\mathbf{g}}_{i_\ell}^k$ is a stochastic subgradient of f_{i_ℓ} at $\mathbf{x}$ in the instant k , $\mathbf{i}^T = [i_1 \dots i_N]$ and $\mathcal{H} : \mathbb{R}^n \rightarrow \mathbf{Y}$ is defined as in Assumption 1, Chapter 3.

Equations (4.1) and (4.2) show that this method fits as a dynamic system with jump in the parameters, following the state of the augmented Markov chain $\mathbf{s}(k)$. Therefore, we can expect that the convergence analysis of this method (and consequently of several incremental subgradient methods that can be seen as a particular case of (4.1)-(4.2)) can also be approached using the available theory for systems with jumps in the parameters, since there are a plethora of techniques for jump linear systems (COSTA; FRAGOSO; MARQUES, 2005; DRAGAN; MOROZAN; STOICA, 2013) and

also some for nonlinear systems, such as Lyapunov-Krasovskii (FEI; GAO; SHI, 2009). However, even if it is not possible, we can investigate convergence directly, based on the dynamic of the state $\mathbf{x}^{k+1}$, which has been successful for the class of problems in question.

BIBLIOGRAPHY

AHN, S.; FESSLER, J. A. Globally convergent image reconstruction for emission tomography using relaxed ordered subsets algorithms. **Medical Imaging, IEEE Transactions on**, IEEE, v. 22, n. 5, p. 613–626, 2003. Citations on pages 46 e 48.

ANTHOINE, S.; AUJOL, J.-F.; BOURSIER, Y.; MELOT, C. Some proximal methods for Poisson intensity CBCT and PET. **Inverse Problems and Imaging**, v. 6, n. 4, p. 565–598, 2012. Citation on page 49.

BAUSCHKE, H. H.; BORWEIN, J. M. On projection algorithms for solving convex feasibility problems. **SIAM Review**, Society for Industrial and Applied Mathematics, v. 38, n. 3, p. 367–426, 1996. Citation on page 48.

BAUSCHKE, H. H.; COMBETTES, P. L.; KRUK, S. G. Extrapolation algorithm for affine-convex feasibility problems. **Numerical Algorithms**, Springer, v. 41, n. 3, p. 239–274, 2006. Citations on pages 36, 52 e 53.

BECK, A.; TEOULLE, M. Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems. **IEEE Transactions on Image Processing**, IEEE, v. 18, n. 11, p. 2419–2434, 2009. Citation on page 47.

_____. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. **SIAM Journal on Imaging Sciences**, SIAM, v. 2, n. 1, p. 183–202, 2009. Citations on pages 29 e 47.

_____. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. **SIAM Journal on Imaging Sciences**, v. 2, n. 1, p. 183–202, 2009. Available: <<http://dx.doi.org/10.1137/080716542>>. Citation on page 77.

BERTSEKAS, D. P. Stochastic optimization problems with nondifferentiable cost functionals. **Journal of Optimization Theory and Applications**, v. 12, n. 2, p. 218–231, 1973. ISSN 1573-2878. Citations on pages 77, 78 e 95.

_____. A new class of incremental gradient methods for least squares problems. **SIAM Journal on Optimization**, SIAM, v. 7, n. 4, p. 913–926, 1997. Citations on pages 22, 46 e 77.

_____. **Nonlinear programming**. Belmont, MA: Athena scientific, 1999. Citations on pages 23 e 49.

_____. Incremental proximal methods for large scale convex optimization. **Mathematical Programming**, v. 129, n. 2, p. 129–163, 2011. ISSN 1436-4646. Available: <<http://dx.doi.org/10.1007/s10107-011-0472-0>>. Citations on pages 37 e 38.

_____. **Convex optimization algorithms**. [S.l.]: Athena Scientific Belmont, 2015. Citations on pages 23, 24 e 26.

BERTSEKAS, D. P.; GALLAGER, R. **Data Networks**. Englewood Cliffs, NJ: Prentice-Hall, 1987. Citations on pages 92, 93 e 95.

BERTSEKAS, D. P.; TSITSIKLIS, J. N. **Neuro-Dynamic Programming**. Belmont, MA: Athena Scientific, 1996. Citation on page 22.

_____. Gradient convergence in gradient methods with errors. **SIAM Journal on Optimization**, SIAM, v. 10, n. 3, p. 627–642, 2000. Citations on pages 46 e 77.

BLATT, D.; HERO, A. O.; GAUCHMAN, H. A convergent incremental gradient method with a constant step size. **SIAM Journal on Optimization**, SIAM, v. 18, n. 1, p. 29–51, 2007. Citations on pages 22, 46 e 77.

BLONDEL, V. D.; HENDRICKX, J. M.; OLSHEVSKY, A.; TSITSIKLIS, J. N. Convergence in multiagent coordination, consensus, and flocking. In: **Proceedings of the 44th IEEE Conference on Decision and Control**. [S.l.: s.n.], 2005. p. 2996–3000. ISSN 0191-2216. Citation on page 39.

BREDIES, K. A forward-backward splitting algorithm for the minimization of non-smooth convex functionals in Banach space. **Inverse Problems**, v. 25, n. 1, p. 015005, 2009. Available: <<http://stacks.iop.org/0266-5611/25/i=1/a=015005>>. Citations on pages 47 e 77.

BROWNE, J.; PIERRO, A. R. D. A row-action alternative to the EM algorithm for maximizing likelihoods in emission tomography. **Medical Imaging, IEEE Transactions on**, IEEE, v. 15, n. 5, p. 687–699, 1996. Citations on pages 22, 46, 48 e 77.

CENSOR, Y.; ELFVING, T.; HERMAN, G. T. Averaging strings of sequential iterations for convex feasibility problems. **Inherently parallel algorithms in feasibility and optimization and their applications**, Amsterdam, North-Holland, v. 8, p. 101–113, 2001. Citations on pages 43, 47, 51, 55, 78 e 79.

CENSOR, Y.; SEGAL, A. On the string averaging method for sparse common fixed-point problems. **International Transactions in Operational Research**, Blackwell Publishing Ltd, v. 16, n. 4, p. 481–494, 2009. ISSN 1475-3995. Available: <<http://dx.doi.org/10.1111/j.1475-3995.2008.00684.x>>. Citation on page 78.

CENSOR, Y.; TOM, E. Convergence of string-averaging projection schemes for inconsistent convex feasibility problems. **Optimization Methods & Software**, v. 18, p. 543–554, 2003. Citations on pages 43, 47 e 78.

CENSOR, Y.; ZASLAVSKI, A. J. Convergence and perturbation resilience of dynamic string-averaging projection methods. **Computational Optimization and Applications**, Springer US, v. 54, n. 1, p. 65–76, 2013. Citations on pages 43, 55 e 78.

_____. String-averaging projected subgradient methods for constrained minimization. **Optimization Methods and Software**, Taylor & Francis, v. 29, n. 3, p. 658–670, 2014. Citations on pages 43, 55 e 78.

CHEN, G. H.-G.; ROCKAFELLAR, R. T. Convergence rates in forward-backward splitting. **SIAM Journal on Optimization**, v. 7, n. 2, p. 421–444, 1997. Citations on pages 47 e 77.

CHOUZENOUX, E.; JEZIERSKA, A.; PESQUET, J.-C.; TALBOT, H. A majorize-minimize subspace approach for $\ell_2 - \ell_0$ image regularization. **SIAM Journal on Imaging Sciences**, v. 6, n. 1, p. 563–591, 2013. Citation on page 49.

CHOUZENOUX, E.; PESQUET, J. C.; TALBOT, H.; JEZIERSKA, A. A memory gradient algorithm for $\ell_2 - \ell_0$ regularization with applications to image restoration. In: **2011 18th IEEE International Conference on Image Processing**. [S.l.: s.n.], 2011. p. 2717–2720. Citation on page 49.

CHOUZENOUX, E.; ZOLYNIK, F.; GOUILLART, E.; TALBOT, H. A majorize-minimize memory gradient algorithm applied to X-ray tomography. In: **2013 IEEE International Conference on Image Processing**. [S.l.: s.n.], 2013. p. 1011–1015. Citation on page 49.

COMBETTES, P. L. Inconsistent signal feasibility problems: Least-squares solutions in a product space. **Signal Processing, IEEE Transactions on**, IEEE, v. 42, n. 11, p. 2955–2966, 1994. Citation on page 48.

_____. Convex set theoretic image recovery by extrapolated iterations of parallel subgradient projections. **Image Processing, IEEE Transactions on**, IEEE, v. 6, n. 4, p. 493–506, 1997. Citations on pages 36, 48, 52 e 53.

COMBETTES, P. L.; PESQUET, J.-C. A proximal decomposition method for solving convex variational inverse problems. **Inverse Problems**, v. 24, n. 6, p. 065014, 2008. Available: <<http://stacks.iop.org/0266-5611/24/i=6/a=065014>>. Citation on page 47.

_____. Fixed-point algorithms for inverse problems in science and engineering. In: _____. New York, NY: Springer New York, 2011. chap. Proximal Splitting Methods in Signal Processing, p. 185–212. ISBN 978-1-4419-9569-8. Available: <http://dx.doi.org/10.1007/978-1-4419-9569-8_10>. Citations on pages 29, 47 e 76.

_____. Stochastic approximations and perturbations in forward-backward splitting for monotone operators. **Pure and Applied Functional Analysis**, Yokohama Publishers, v. 1, n. 1, p. 13–37, 2016. Citations on pages 47 e 77.

COSTA, O. L. V.; FRAGOSO, M. D.; MARQUES, R. P. **Discrete-Time Markovian Jump Linear Systems**. New York: Springer-Verlag, 2005. Citation on page 101.

DANNA, E.; MANDAL, S.; SINGH, A. A practical algorithm for balancing the max-min fairness and throughput objectives in traffic engineering. In: **2012 Proceedings IEEE INFOCOM**. [S.l.: s.n.], 2012. p. 846–854. ISSN 0743-166X. Citation on page 93.

DEM'YANOV, V. F.; VASIL'EV, L. V. **Nondifferentiable optimization**. [S.l.]: Springer, 1985. Citations on pages 46, 49 e 76.

DEWARAJA, Y. K.; KORAL, K. F.; FESSLER, J. A. Regularized reconstruction in quantitative SPECT using CT side information from hybrid imaging. **Physics in Medicine and Biology**, v. 55, n. 9, p. 2523, 2010. Available: <<http://stacks.iop.org/0031-9155/55/i=9/a=007>>. Citation on page 49.

DOLAN, E. D.; MORÉ, J. J. Benchmarking optimization software with performance profiles. **Mathematical Programming**, v. 91, n. 2, p. 201–213, 2002. ISSN 1436-4646. Available: <<http://dx.doi.org/10.1007/s101070100263>>. Citation on page 97.

DRAGAN, V.; MOROZAN, T.; STOICA, A. M. **Mathematical Methods in Robust Control of Linear Stochastic Systems**. 2nd. ed. [S.l.]: Springer, 2013. ISBN 9781461486633. Citation on page 101.

ERMOLIEV, Y. Stochastic quasigradient methods and their application to system optimization. **Stochastics**, v. 9, n. 1-2, p. 1-36, 1983. Citation on page 77.

ERYILMAZ, A.; SRIKANT, R. Joint congestion control, routing, and mac for stability and fairness in wireless networks. **IEEE Journal on Selected Areas in Communications**, v. 24, n. 8, p. 1514-1524, 2006. ISSN 0733-8716. Citation on page 92.

FEI, Z.; GAO, H.; SHI, P. New results on stabilization of markovian jump systems with time delay. **Automatica**, v. 45, n. 10, p. 2300 - 2306, 2009. ISSN 0005-1098. Available: <<http://www.sciencedirect.com/science/article/pii/S0005109809003021>>. Citation on page 102.

GAIVORONSKII, A. A. Nonstationary stochastic programming problems. **Cybernetics**, v. 14, n. 4, p. 575-579, 1978. ISSN 1573-8337. Available: <<http://dx.doi.org/10.1007/BF01069841>>. Citation on page 77.

HARMANY, Z. T.; MARCIA, R. F.; WILLETT, R. M. This is SPIRAL-TAP: Sparse Poisson Intensity Reconstruction Algorithms - theory and practice. **IEEE Transactions on Image Processing**, v. 21, n. 3, p. 1084-1096, 2012. Citation on page 49.

HELOU, E. S.; CENSOR, Y.; CHEN, T.-B.; CHERN, I.-L.; De Pierro, Á. R.; JIANG, M.; LU, H. H.-S. String-averaging expectation-maximization for maximum likelihood estimation in emission tomography. **Inverse Problems**, IOP Publishing, v. 30, n. 5, p. 055003, 2014. Available: <<http://stacks.iop.org/0266-5611/30/i=5/a=055003>>. Citations on pages 43 e 48.

_____. String-averaging expectation-maximization for maximum likelihood estimation in emission tomography. **Inverse Problems**, IOP Publishing, v. 30, n. 5, p. 055003, 2014. Available: <<http://stacks.iop.org/0266-5611/30/i=5/a=055003>>. Citation on page 78.

HERMAN, G. T. **Fundamentals of computerized tomography: image reconstruction from projections**. [S.l.]: Springer-Verlag London, 2009. (Advances in Computer Vision and Pattern Recognition). Citation on page 60.

HIRIART-URRUTY, J.-B.; LEMARÉCHAL, C. **Convex Analysis and Minimization Algorithms**. Berlin: Springer-Verlag, 1993. (A Series of Comprehensive Studies in Mathematics, v. 305). Citations on pages 23 e 30.

_____. **Convex Analysis and Minimization Algorithms I and II**. Berlin: Springer-Verlag, 1993. (A Series of Comprehensive Studies in Mathematics). Citations on pages 47, 49 e 64.

_____. **Convex Analysis and Minimization Algorithms I. Fundamentals**. Berlin: Springer-Verlag, 1993. (A Series of Comprehensive Studies in Mathematics, v. 305). Citation on page 95.

HOFFMANN, A. The distance to the intersection of two convex sets expressed by the distances to each of them. **Mathematische Nachrichten**, Wiley, v. 157, n. 1, p. 81-98, 1992. Available: <<http://onlinelibrary.wiley.com/doi/10.1002/mana.19921570108/abstract>>. Citation on page 54.

HUDSON, H. M.; LARKIN, R. S. Accelerated image reconstruction using ordered subsets of projection data. **Medical Imaging, IEEE Transactions on**, IEEE, v. 13, n. 4, p. 601–609, 1994. Citations on pages 46 e 48.

JADBABAIE, A.; LIN, J.; MORSE, A. S. Coordination of groups of mobile autonomous agents using nearest neighbor rules. **IEEE Transactions on Automatic Control**, v. 48, n. 6, p. 988–1001, 2003. ISSN 0018-9286. Citation on page 39.

JAFFE, J. Bottleneck flow control. **IEEE Transactions on Communications**, v. 29, n. 7, p. 954–962, 1981. ISSN 0090-6778. Citations on pages 92 e 93.

JAKOVETIĆ, D.; XAVIER, J.; MOURA, J. M. F. Fast distributed gradient methods. **IEEE Transactions on Automatic Control**, v. 59, n. 5, p. 1131–1146, 2014. ISSN 0018-9286. Citation on page 41.

JOHANSSON, B.; CARRETTI, C. M.; JOHANSSON, M. On distributed optimization using peer-to-peer communications in wireless sensor networks. In: **2008 5th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks**. [S.l.: s.n.], 2008. p. 497–505. ISSN 2155-5486. Citation on page 38.

JOHANSSON, B.; RABI, M.; JOHANSSON, M. A simple peer-to-peer algorithm for distributed optimization in sensor networks. In: **2007 46th IEEE Conference on Decision and Control**. [S.l.: s.n.], 2007. p. 4705–4710. ISSN 0191-2216. Citation on page 38.

_____. A randomized incremental subgradient method for distributed optimization in networked systems. **SIAM Journal on Optimization**, v. 20, n. 3, p. 1157–1170, 2010. Citations on pages 37, 38 e 47.

KAK, A. C.; SLANEY, M. **Principles of Computerized Tomographic Imaging**. [S.l.]: IEEE press, 1988. Citation on page 62.

KIBARDIN, V. M. Decomposition into functions in the minimization problem. **Automatika i Telemekhanika**, Russian Academy of Sciences, Branch of Power Industry, Machine Building, Mechanics and Control Processes, n. 9, p. 66–79, 1979. Available: <<http://mi.mathnet.ru/eng/at9530>>. Citations on pages 33, 49 e 77.

LOBEL, I.; OZDAGLAR, A. Convergence analysis of distributed subgradient methods over random networks. In: **46th Annual Allerton Conference on Communication, Control, and Computing**. [S.l.: s.n.], 2008. p. 353–360. Citation on page 41.

MARTÍNEZ, J. M.; PILOTTA, E. A. Inexact-restoration algorithm for constrained optimization. **Journal of Optimization Theory and Applications**, v. 104, n. 1, p. 135–163, 2000. ISSN 1573-2878. Citation on page 77.

_____. Inexact restoration methods for nonlinear programming: advances and perspectives. In: _____. **Optimization and control with applications**. Boston, MA: Springer US, 2005. p. 271–291. ISBN 978-0-387-24255-2. Citation on page 77.

MO, J.; WALRAND, J. Fair end-to-end window-based congestion control. **IEEE/ACM Transactions on Networking**, IEEE Press, Piscataway, NJ, USA, v. 8, n. 5, p. 556–567, 2000. ISSN 1063-6692. Citation on page 92.

NATTERER, F. **The Mathematics of Computerized Tomography**. [S.l.]: Wiley, 1986. Citations on pages 48 e 60.

NATTERER, F.; WÜBBELING, F. **Mathematical methods in image reconstruction**. [S.l.]: SIAM, 2001. Citations on pages 48 e 60.

NEDIĆ, A.; BERTSEKAS, D. P. Convergence rate of incremental subgradient algorithms. In: _____. **Stochastic Optimization: Algorithms and Applications**. Boston, MA: Springer US, 2001. p. 223–264. ISBN 978-1-4757-6594-6. Available: <http://dx.doi.org/10.1007/978-1-4757-6594-6_11>. Citations on pages 34 e 35.

_____. Incremental subgradient methods for nondifferentiable optimization. **SIAM Journal on Optimization**, SIAM, v. 12, n. 1, p. 109–138, 2001. Citations on pages 22, 26, 33, 34, 35, 37, 45, 46, 47, 50 e 77.

NEDIĆ, A.; LEE, S. On stochastic subgradient mirror-descent algorithm with weighted averaging. **SIAM Journal on Optimization**, v. 24, n. 1, p. 84–107, 2014. Citations on pages 32 e 77.

NEDIĆ, A.; OZDAGLAR, A. Distributed subgradient methods for multi-agent optimization. **IEEE Transactions on Automatic Control**, v. 54, n. 1, p. 48–61, 2009. ISSN 0018-9286. Citations on pages 39, 40 e 41.

NEDIĆ, A.; OZDAGLAR, A. Convergence rate for consensus with delays. **Journal of Global Optimization**, v. 47, n. 3, p. 437–456, 2010. ISSN 1573-2916. Citation on page 40.

NEDIĆ, A.; OZDAGLAR, A.; PARRILO, P. A. Constrained consensus and optimization in multi-agent networks. **IEEE Transactions on Automatic Control**, v. 55, n. 4, p. 922–938, 2010. ISSN 0018-9286. Citation on page 40.

NEMIROVSKI, A.; JUDITSKY, A.; LAN, G.; SHAPIRO, A. Robust stochastic approximation approach to stochastic programming. **SIAM Journal on Optimization**, v. 19, n. 4, p. 1574–1609, 2009. Citations on pages 32, 33, 75, 77 e 97.

NEMIROVSKI, A.; YUDIN, D. B. **Problem Complexity and Method Efficiency in Optimization**. New York: John Wiley, 1983. (Wiley-Intersci. Ser. Discrete Math, v. 15). Citations on pages 26, 29, 31, 32 e 76.

NESTEROV, Y. **Introductory lectures on convex optimization: A Basic Course**. Dordrecht: Kluwer Academic Publishers, 2004. Citations on pages 26, 28, 32 e 76.

_____. Smooth minimization of non-smooth functions. **Mathematical Programming**, v. 103, n. 1, p. 127–152, 2005. ISSN 1436-4646. Citation on page 76.

_____. Primal-dual subgradient methods for convex problems. **Mathematical Programming**, v. 120, n. 1, p. 221–259, 2009. ISSN 1436-4646. Citations on pages 26, 27 e 76.

NETO, E. S. H.; PIERRO, A. R. D. Convergence results for scaled gradient algorithms in positron emission tomography. **Inverse problems**, IOP Publishing, v. 21, p. 1905, 2005. Citations on pages 22 e 48.

NETO, E. S. H.; PIERRO, Á. R. D. Incremental subgradients for constrained convex optimization: a unified framework and new methods. **SIAM Journal on Optimization**, v. 20, n. 3, p. 1547–1572, 2009. Citations on pages 22, 35, 36, 45, 46, 47, 48, 50, 51, 53, 59, 65, 77, 78, 80 e 97.

NEUMAIER, A. OSGA: a fast subgradient algorithm with optimal complexity. **Mathematical Programming**, v. 158, n. 1, p. 1–21, 2016. ISSN 1436-4646. Citations on pages 26, 30, 31 e 76.

OLIVEIRA, R. M.; HELOU, E. S.; COSTA, E. F. String-averaging incremental subgradients for constrained convex optimization with applications to reconstruction of tomographic images. **Inverse Problems**, v. 32, n. 11, p. 115014, 2016. Available: <<http://stacks.iop.org/0266-5611/32/i=11/a=115014>>. Citations on pages 21, 22, 78, 80 e 100.

PENFOLD, S. N.; SCHULTE, R. W.; CENSOR, Y.; BASHKIROV, V.; MCALLISTER, S.; SCHUBERT, K. E.; ROSENFELD, A. B. Block-iterative and string-averaging projection algorithms in proton computed tomography image reconstruction. **Biomedical Mathematics: Promising Directions in Imaging, Therapy Planning and Inverse Problems**, Medical Physics Publishing, Madison, WI, USA, p. 347–367, 2010. Citations on pages 43, 47 e 78.

PFLUG, G. C. **Optimization of stochastic models: The interface between simulation and optimization**. [S.l.]: Springer US, 1996. (The Springer International Series in Engineering and Computer Science, v. 373). Citation on page 77.

PIERRO, A. R. D.; YAMAGISHI, M. E. B. Fast EM-like methods for maximum “a posteriori” estimates in emission tomography. **Medical Imaging, IEEE Transactions on**, IEEE, v. 20, n. 4, p. 280–288, 2001. Citations on pages 46 e 48.

POLYAK, B. T. Minimization of unsmooth functionals. **USSR Computational Mathematics and Mathematical Physics**, v. 9, n. 3, p. 14–29, 1969. ISSN 0041-5553. Available: <<http://www.sciencedirect.com/science/article/pii/0041555369900615>>. Citations on pages 80 e 96.

_____. **Introduction to Optimization**. [S.l.: s.n.], 1987. Citations on pages 23, 25, 46, 49 e 76.

RABBAT, M.; NOWAK, R. Distributed optimization in sensor networks. In: **Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks**. New York, NY, USA: ACM, 2004. (IPSN '04), p. 20–27. ISBN 1-58113-846-6. Available: <<http://doi.acm.org/10.1145/984622.984626>>. Citation on page 38.

RAGUET, H.; FADILI, J.; PEYRÉ, G. A generalized forward-backward splitting. **SIAM Journal on Imaging Sciences**, v. 6, n. 3, p. 1199–1226, 2013. Citations on pages 47 e 77.

RAM, S. S.; NEDIĆ, A.; VEERAVALLI, V. V. Incremental stochastic subgradient algorithms for convex optimization. **SIAM Journal on Optimization**, SIAM, v. 20, n. 2, p. 691–717, 2009. Citations on pages 36, 37, 38, 41, 46, 47, 71, 77, 78, 79 e 100.

_____. Distributed stochastic subgradient projection algorithms for convex optimization. **Journal of optimization theory and applications**, Springer, v. 147, n. 3, p. 516–545, 2010. Citations on pages 41, 42, 43, 46, 47, 71 e 77.

ROBBINS, H.; MONRO, S. A stochastic approximation method. **The Annals of Mathematical Statistics**, Institute of Mathematical Statistics, v. 22, n. 3, p. 400–407, 1951. ISSN 00034851. Available: <<http://www.jstor.org/stable/2236626>>. Citation on page 77.

RUSZCZYŃSKI, A.; SYSKI, W. A method of aggregate stochastic subgradients with on-line stepsize rules for convex stochastic programming problems. In: _____. **Stochastic Programming 84 Part II**. Springer Berlin Heidelberg, 1986. (Math. Prog. Stud, v. 28), p. 113–131. ISBN 978-3-642-00927-3. Available: <<http://dx.doi.org/10.1007/BFb0121128>>. Citation on page 77.

SHEPP, L. A.; VARDI, Y. Maximum likelihood reconstruction for emission tomography. **Medical Imaging, IEEE Transactions on**, IEEE, v. 1, n. 2, p. 113–122, 1982. Citation on page 48.

SHOR, N. Z. **Minimization methods for non-differentiable functions**. [S.l.]: Springer-Verlag New York, Inc., 1985. Citations on pages 23, 46, 49, 76, 80 e 96.

SOLODOV, M. V. Incremental gradient algorithms with stepsizes bounded away from zero. **Computational Optimization and Applications**, Springer, v. 11, n. 1, p. 23–35, 1998. Citations on pages 46 e 77.

SOLODOV, M. V.; ZAVRIEV, S. K. Error stability properties of generalized gradient-type algorithms. **Journal of Optimization Theory and Applications**, v. 98, n. 3, p. 663–680, 1998. Citations on pages 33, 45, 46, 50 e 77.

STRASSEN, V. The existence of probability measures with given marginals. **The Annals of Mathematical Statistics**, Institute of Mathematical Statistics, v. 36, n. 2, p. 423–439, 1965. ISSN 00034851. Available: <<http://www.jstor.org/stable/2238148>>. Citation on page 78.

SVANBERG, K. A class of globally convergent optimization methods based on conservative convex separable approximations. **SIAM Journal on Optimization**, v. 12, n. 2, p. 555–573, 2002. Available: <<http://dx.doi.org/10.1137/S1052623499362822>>. Citation on page 97.

TANAKA, E.; KUDO, H. Subset-dependent relaxation in block-iterative algorithms for image reconstruction in emission tomography. **Physics in medicine and biology**, IOP Publishing, v. 48, n. 10, p. 1405, 2003. Citations on pages 46, 48 e 77.

TOURI, B.; NEDIĆ, A.; RAM, S. S. Asynchronous stochastic convex optimization over random networks: Error bounds. In: **2010 Information Theory and Applications Workshop (ITA)**. [S.l.: s.n.], 2010. p. 1–10. Citation on page 77.

TSENG, P. An incremental gradient (-projection) method with momentum term and adaptive stepsize rule. **SIAM Journal on Optimization**, SIAM, v. 8, n. 2, p. 506–531, 1998. Citations on pages 46 e 77.

VARDI, Y.; SHEPP, L.; KAUFMAN, L. A statistical model for positron emission tomography. **Journal of the American Statistical Association**, Taylor & Francis Group, v. 80, n. 389, p. 8–20, 1985. Citation on page 48.

YAMADA, I.; OGURA, N. Adaptive projected subgradient method for asymptotic minimization of sequence of nonnegative convex functions. **Numerical Functional Analysis and Optimization**, Taylor & Francis, v. 25, n. 7, p. 593–617, 2005. Citations on pages [36](#), [52](#) e [53](#).

_____. Hybrid steepest descent method for variational inequality problem over the fixed point set of certain quasi-nonexpansive mappings. **Numerical Functional Analysis and Optimization**, Taylor & Francis, v. 25, n. 7, p. 619–655, 2005. Citations on pages [36](#), [52](#) e [53](#).

YOUSEFIAN, F.; NEDIĆ; SHANBHAG, U. V. Convex nondifferentiable stochastic optimization: A local randomized smoothing technique. In: **Proceedings of the 2010 American Control Conference**. [S.l.: s.n.], 2010. p. 4875–4880. ISSN 0743-1619. Citation on page [77](#).

YOUSEFIAN, F.; NEDIĆ, A.; SHANBHAG, U. V. On stochastic gradient and subgradient methods with adaptive steplength sequences. **Automatica**, v. 48, n. 1, p. 56–67, 2012. ISSN 0005-1098. Citation on page [77](#).

ZUKERMAN, M.; MAMMADOV, M.; TAN, L.; OUYEYSI, I.; ANDREW, L. L. To be fair or efficient or a bit of both. **Computers & Operations Research**, v. 35, n. 12, p. 3787–3806, 2008. ISSN 0305-0548. Citation on page [92](#).