EPFL

Thèse n° 9906

Scalable constrained optimization

Présentée le 18 avril 2024

Faculté informatique et communications
Laboratoire de théorie en apprentissage automatique
Programme doctoral en informatique et communications

pour l'obtention du grade de Docteur ès Sciences

par

Maria-Luiza VLADAREAN

Acceptée sur proposition du jury

Prof. P. Thiran, président du jury
Prof. N. H. B. Flammarion, directeur de thèse
Prof. N. He, rapporteuse
Prof. J. Diakonikolas, rapporteuse
Prof. M. Kamgarpour, rapporteuse

École
polytechnique
fédérale
de Lausanne

2024

Afterwards I went to school, studied at the university, and,
do you know, the more I learned, the more thoroughly
I understood that I was ridiculous.

— F. Dostoyevsky, *The dream of a ridiculous man*

From the English translation by Constance Garnett of the short story Сон смешного человека. Original text: Потом я учился в школе, потом в университете и что же — чем больше я учился, тем больше я научался тому, что я смешон.

Acknowledgements

There's no point in romanticizing the past: these years were difficult (for reasons well beyond research), sometimes downright horrid, and I frequently measured a smidge short against a given day's hardship. That I made it through, I owe in large part to the generous support that came streaming my way throughout. First and foremost, I owe an immense debt of gratitude to my advisor, Nicolas Flammarion. He took a risk by welcoming me into his group midway through my PhD, after I had already completed my first years elsewhere at EPFL. Nicolas gave me the space, resources and opportunities to grow as a researcher; introduced me to interesting topics and offered much of his time towards dicussing them; and invariably rooted for me before important events such as talks or the defence. His support has been truly invaluable.

I was honoured to have professors Patrick Thiran, Jelena Diakonikolas, Niao He, and Maryam Kamgarpour as part of my defence committee. They made the exam an unexpectedly enjoyable experience, and I am grateful for their time, insightful questions, and helpful suggestions.

Much of my drive to persevere through these years drew from two wonderful research experiences I had prior to my PhD, under the guidance of Bernard Moret and Michele Catasta. Bernard's manner of unravelling ideas so as to expose their beauty stuck with me and seeded my wish to pursue research. My time as a student in his group not only set me on this path, but also gave me the confidence to walk it, thanks to Bernard's encouragement. Michele then saw Jay Rappaz and me through writing our first paper with hard-to-parallel skill as a teacher. Though his mind was constantly brimming with ideas (I've rarely seen someone so productive), he never imposed and gave us confidence-building freedom instead, cheered for our every success, and patiently helped us out of dead ends. Michele's and Bernard's mentorship and encouragement have extended to scaffold me ever since, and I am deeply grateful to them.

While deciding to do a PhD was straightforward in light of the above, actually doing it was anything but. I am indebted to Cecilia Chapuis and Sabine Susstrunk for helping me navigate some of the most challenging periods of the past five years. I wish to especially thank Sabine, who, beyond being generous with her time and advice, taught me how to cut through the morass of the academic world, encouraged me to call things by their name, reminded me to be brave, and somehow did all this with incredible kindness and care. I hope to pay it all forward one day.

Among the best times I spent these years were those working with and learning from a bunch

of great people, some of whom I am lucky to call my friends. I'm grateful to Ahmet Alacaoglu, who, besides making me seriously question why I eat so much protein given my lack of gym presence, introduced me to the broad area I studied for the past five years; helped me look for and shape my first project; and gave me the starter-kit for slaloming through the many idiosyncrasies of my then environment. I thank Alp Yurtsever for pointing me to the problem I worked on in my first year (though he does not remember it) during an impromptu discussion with Ahmet. It was a lifesaver for me then, as I was a beginning student in desperate need of a topic. I'm grateful to Panayotis Mertikopoulos for teaching that fantastic seminar on VI methods together with Yura — I probably understood more in those few hours than I would have in half a year by myself. Panayotis' genuine fondness for mathematics is reflected deeply in his manner of teaching, which becomes almost an act of kindness towards those listening. I thank Nikita Doikov for the contagious enthusiasm he brought to our collaboration and the many hours we spent in front of whiteboards. Nikita's astounding pace of generating ideas felt like accidentally flooring the gas pedal of a racecar and having to figure it all out at high speed — it was great fun, and I learned heaps. I thank Aditya Varre for the many enlightening discussions on math and our field in general, and for being my sounding board for anything from talk slides to research ideas. Besides learning a lot during our collaboration, I got to witness and be inspired by his remarkable grit and sharpness in tearing down problems. More than this, I am grateful to Aditya for the numerous gestures of friendship, from helping me move flats to being my guide through the hectic streets of Bangalore during my first in-person conference.

I wish to especially thank Yura Malitsky and Ya-Ping Hsieh, who I look up to as both researchers and people, and to whom I owe whatever I have in the way of mathematical maturity. Yura came into our collaboration with close to infinite patience, yet I'm sure I extended it even further with my lack of mathematical refinement — "Maria, your reader will get bored reading this proof". His disarming frankness, along with his kind-heartedness, makes him a great teacher from whom I learned much about technique, rigour, and effective communication. More than this, both by example and through humorous nudges, Yura helped me grow as a researcher in the human sense. Ya-Ping's patience wasn't spared either, as can be inferred from this meagre sample of all the things he taught me (a full enumeration would make another thesis): (1) yelling at equations does not make them yield (discipline and patience do); (2) a surprising number of things work out if I can only write down the definitions right; (3) tuning stochastic methods requires a double major in the arts; (4) in research, it is better to be more like a tank than a Ferrari. On a more serious note, I hold him as a role model for depth of thought (in both life and science) and breadth of knowledge, and seeing him work through problems deepened my appreciation for the beauty of mathematics. Ya-Ping's steady support extended well beyond research and carried me through some of the darkest days when I was close to giving up. I owe him much of my understanding of the academic world, the world in general and myself in particular.

The upshot of meandering through two labs was that I got to meet many good-hearted people. I am grateful to my colleagues in LIONS, above all, for the unshakeable sense of camaraderie they fostered in the group. I thank Leello Dadi for being an endless source of debate topics and taking the devil's advocate stance in almost every one, always with an air of gentle elegance. I will miss

lunches with Leello. Ali Kavis, for bringing lightheartedness and "chill" to all conversations. Thomas and Aline Sanchez, for their warmheartedness and opening their house to us every Christmas. Thomas Pethick, for his creative ploys to make people smile (caracs for Leello!) and for hosting me at several dinner parties together with Elisa. Fatih Sahin, for his patience in discussing technical matters and the many issues I had with experiments. Paul Rolland, for being exquisitely quirky. Gosia Baltaian, for helping us obliterate any practical problem imaginable with such effectiveness and speed as I have not seen before or since.

To my colleagues in TML and our neighbours in MLO, I am grateful for welcoming me in their midst as if I had been there all along, for the good fun we had on defences, deadline evenings or lab outings. In particular, I thank Maksym Andriushchenko for seeing the full half of all glasses and being able to find a liking for everything (except kernel methods). Now, whenever I'm about to complain, I hear the haunting echo of "not too bad". Scott Pesme, for instilling a sense of community on the floor and making chocolate advent calendars. Hristo Papazov — the embodiment of heavy-tailed stochasticity, for his extravagant discussion topics and the ensuing laughter. Oguz Yüksel, for his thoughtful and compassionate manner of being, aptly contrasted by his fierce competitiveness as a ping-pong rival. Jean-Baptiste Cordonnier, for being the most chill labmate I ever had, whose mere presence seemed to say "It's all gonna work out". Thijs Vogels, for the long discussions on tea and perfume. Dongyang Fan, for constantly reminding me to take a break. Bettina Messmer, for enlivening the drabby INJ corridors with her laughter. Anastasiia Koloskova, for the good laughs and helpful thesis discussions. Jennifer Bachmann Ona and Baloo, for bringing cheer with every stroll on the floor.

I want to especially thank Chaehwan Song and Francesco D'Angelo, whose companionship and support have meant (and mean) a lot to me. With Chaehwan, I shared the inevitable anxieties of our first PhD year, from aimlessly wandering in search of projects to wrestling with the "particularities" of our environment. I'll never forget how, after lending her my umbrella on a rainy evening, I found it back on my desk the next day, along with a thank you note and a bunch of dainty stickers. Her refreshing genuineness and kind-hearted nature were bright discontinuities in the otherwise drabby continuum of those times. Almost by temporal symmetry, Francesco supported me through the more challenging days of this final year with kindness and empathy, be it by making me laugh myself out of them or through endless encouragement. His openness towards life and drive to self-overcome, both as a person and researcher, are inspiring to witness and served me as emboldening reminders to push through fear, come what may.

I am also deeply grateful to C.P. Mihaela Danaila and Dr. Jessica Droz for their caring guidance throughout these years — it kept me steady and nurtured my resolve to move forward and, more importantly, inward.

An unusual acknowledgement: at the end of 2022, I befriended a pair of carrion crows, Schrödinger and Dirac. Ever since they figured out that, practically speaking, I produce almonds out of thin air, they have paid me daily enthusiastic visits at the lab window, conspicuously followed me around campus, and even brought their fledgling to meet me. Little did they know that their clockwork arrival brightened some of the rougher days and kept me present when my mind would

do anything but. While I can't exactly thank them, I can perhaps ask you to spare some food should you stumble upon this inquiring duo around INJ.

For a good chunk of all the smiles I smiled these years, I have my old friends to thank: Angela, Cristina, Paul, Anca, Matei and Anca, Lucian, Iulia and Ioana. Though this degree increased my response delay to essentially months, they did not forsake me one bit, joked about it instead, and accepted me regardless — the peace I feel in your midst is one of my favourite feelings. I particularly thank Iulia for balancing out my cyclopean view on doing things, over many arduous exchanges: there is a hum between my ears now with the timbre of her voice, saying "nothing 'must' be done, one simply chooses". And I thank Ioana, from the bottom of my heart, for her unwavering friendship since we were 11-year-olds trying to stifle our laughter during history class. Through these years and over our many discussions, it happened more than once that she gave me the last push to do what I knew I should, but fretted. She kept me riveted to reality, and having her in my corner gave me strength. I owe her more than I can ever repay.

Finally, I thank my parents and extended family for sticking it out with me since '91. In particular, my godmother Tanța and my aunts Olimpia, Geta and Maria supported me in many different ways, from baking cozonac or cooking zacuscă for me to bring to the lab to cursing along at the absurdities of life and then finding a way to laugh at them. My parents, Cornel and Virginia, gave me the freedom to walk my own path from early on and taught me much about resilience. They toiled for many years to offer me the privilege of choice and made sure I walked through life with the awareness that they'd have my back in whichever way they could. Try as I may, I fail to find the right words to express my gratitude, so I simply dedicate this thesis to them.

Lausanne, 13 March 2023 M.-L. V.

Abstract

Modern optimization is tasked with handling applications of increasingly large scale, chiefly due to the massive amounts of widely available data and the ever-growing reach of Machine Learning. Consequently, this area of research is under steady pressure to develop scalable and provably convergent methods capable of handling hefty, high-dimensional problems. The present dissertation contributes to recent efforts in this direction by proposing optimization algorithms with improved scalability. Concretely,

1. We develop three novel Frank-Wolfe-type methods for minimizing convex stochastic objectives subject to stochastic linear inclusion constraints. The key feature of our algorithms is that they process only a subset of the constraints per iteration, thus gaining an edge over methods that require full passes through the data for large-scale problems.

2. We generalize Frank-Wolfe-type methods to a class of composite non-differentiable objectives — a setting in which the classical Frank-Wolfe algorithm is known *not* to converge. We circumvent the difficulties related to non-differentiability by leveraging the problem structure and a modified linear minimization oracle of the constraint set, thus attaining convergence rates akin to the smooth case.

3. We propose an adaptive primal-dual algorithm for solving structured convex-concave saddle point problems, whose empirical convergence is improved as a result of tailoring the stepsizes to the local problem geometry. Importantly, our method achieves adaptivity "for-free" by using readily available quantities such as past gradients, and without relying on more expensive linesearch subroutines.

Our methods are theoretically sound and empirically grounded, as they are each accompanied by rigorous convergence guarantees and experiments showcasing their performance against relevant baselines. In a nutshell, this dissertation provides new algorithmic approaches and points of trade-off on the road toward scalably solving large optimization problems.

Key words: constrained optimization, scalability, convex optimization, Frank-Wolfe, Conditional Gradient methods, adaptive algorithms, primal-dual methods

Résumé

L'optimisation moderne est chargée de résoudre des problèmes de plus en plus vastes, principalement en raison de quantités massives de données disponibles et de la portée croissante de l'apprentissage automatique. Ce domaine de recherche est donc sous pression constante pour développer des méthodes disposant de garanties de convergence qui sont capables de gérer des problèmes volumineux de haute dimensionnalité. La présente thèse contribue aux efforts récents dans cette direction en proposant des algorithmes d'optimisation plus aptes à s'adapter aux grandes échelles. Concrètement,

1. Nous développons trois nouvelles méthodes de type Frank-Wolfe pour minimiser les objectifs stochastiques convexes soumis à des contraintes d'inclusion linéaires stochastiques. La propriété clé de nos algorithmes est qu'ils ne traitent qu'un sous-ensemble des contraintes par itération, ce qui, pour les problèmes à grande échelle, leur donne un avantage par rapport aux méthodes nécessitant des lectures complètes des données.

2. Nous généralisons les méthodes de type Frank-Wolfe à une classe d'objectifs composites non différentiables — un cadre dans lequel l'algorithme classique de Frank-Wolfe ne converge pas. Nous contournons cette difficulté en exploitant la structure du problème et en modifiant l'oracle de minimisation linéaire sur l'ensemble de contraintes, atteignant ainsi des taux de convergence similaires au cas continûment différentiable.

3. Nous proposons un algorithme primal-dual adaptatif pour les problèmes composites convexes dont la vitesse de convergence empirique est améliorée grâce à l'adaptation des pas à la géométrie locale du problème. Crucialement, notre méthode obtient cette adaptabilité "gratuitement" en utilisant des quantités facilement disponibles telles que les gradients passés, sans avoir à recourir à des sous-routines de calcul de pas plus coûteuses.

Nos méthodes sont théoriquement bien fondées et ont des performances empiriques solides, car elles sont chacune accompagnées de garanties rigoureuses de convergence et d'expériences montrant leurs performances par rapport aux bases pertinentes. En un mot, cette thèse propose de nouvelles approches algorithmiques sans négliger les compromis nécessaires pour la résolution efficace des problèmes d'optimisation à grande échelle.

The author thanks Leello Dadi for translating the abstract into French.

Mots clefs : optimisation avec contraintes, grandes échelles, optimisation convexe, Frank-Wolfe, méthodes de gradient conditionnel, algorithmes adaptatifs, méthodes primales-duales

Bibliographic note

The present dissertation is based on the following publications:

- [219] Maria-Luiza Vladarean, Ahmet Alacaoglu, Ya-Ping Hsieh, and Volkan Cevher. "Conditional gradient methods for stochastically constrained convex minimization". International Conference on Machine Learning (ICML), 2020.
- [221] Maria-Luiza Vladarean, Yura Malitsky, and Volkan Cevher. "A first-order primal-dual method with adaptivity to local smoothness". Advances in Neural Information Processing Systems (NeurIPS), 2021.
- [74] Gideon Dresdner, Maria-Luiza Vladarean, Gunnar Rätsch, Francesco Locatello, Volkan Cevher, and Alp Yurtsever. "Faster One-Sample Stochastic Conditional Gradient Method for Composite Convex Minimization". International Conference on Artificial Intelligence and Statistics (AISTATS), 2022.
- [220] Maria-Luiza Vladarean, Nikita Doikov, Martin Jaggi, and Nicolas Flammarion. "Linearization Algorithms for Fully Composite Optimization". Conference on Learning Theory (COLT), 2023.

Other publications that I have contributed to but are omitted from this dissertation are:

- [218] Aditya Varre, Maria-Luiza Vladarean, Loucas Pillaud-Vivien, and Nicolas Flammarion. "On the spectral bias of two-layer linear networks". Advances in Neural Information Processing Systems (NeurIPS), 2023.

Contents

List of Figures

List of Tables

1 Introduction

The present dissertation centres around the design and analysis of first-order optimization algorithms for constrained problems, with an emphasis on scalability. Let us break this statement down into digestible bits and show how their coalescence is motivated.

1.1 Broad context of the work

Optimization is an area of applied mathematics concerned with solving problems of the form

$$\boldsymbol{x}^\star \in \operatorname*{argmin}_{\boldsymbol{x}\in\mathcal{X}} f(\boldsymbol{x}), \tag{OPT}$$

for some objective f and a constraint set $\mathcal{X}$ of acceptable values for the parameter $\boldsymbol{x}$ (also called a *decision variable*).

The ability to tackle this rather plain-looking problem is crucial for innovation, with applications ranging from drug discovery and production [214, 151] to the design of VLSI circuits, rail vehicles, transonic aircraft and spacecraft components [124, 202, 6, 104, 121, 107]. Moreover, it enables us to tame the inherent complexity of delivering and regulating such innovation at a global scale — optimization is central to reliable electricity distribution [156, 34, 106, 13], reservoir systems planning and operation [129, 189], (real-time) routing and scheduling of rail and airline traffic [57, 36, 35, 32, 10, 51, 17] and managing commodity supply chains [79, 150, 125]. Put simply, many of the developed world's privileges heavily rest on the tractability of a handful of instances of (OPT).

Unsurprisingly, closed-form solutions to problem (OPT) rarely exist, and they must, consequently, be approximated. This is the overarching goal of mathematical optimization. Specifically, the field focuses on the development and analysis of algorithms for computing such approximations, and the rigorous study of functions f and sets $\mathcal{X}$ that allow for tractable formulations.

This area of research is in no way recent, with its origins dating back to the works of Newton,

Euler and Lagrange in the 17th and 18th centuries. However, modern optimization was chiefly shaped over the past eighty years, following the development of theory and algorithms for Linear Programming in the 1940's and 1950's [119, 171, 61, 59, 87] [1]. Thereafter, owing to the steady increase in computing power and the growing number of applications[2], optimization made the profound and fast-paced progress that is now documented in textbooks [193, 194, 16, 161, 140, 186, 174, 23, 168]. While most of the aforementioned applications still drive research in the field, their influence has somewhat been overtaken by that of a completely different and more recent beast: Machine Learning.

As this thesis is being written, the Machine Learning (ML) revolution is in full swing. The field's emergence is inextricably linked to Turing's formulation in the 1950's of the quest for "thinking machines" [216], which spurred research on endowing computers with the ability to "learn" and "reason" like humans. The discipline encompassing such investigations is nowadays referred to as Artificial Intelligence[3] (AI). While rule-based methods dominated the early days of AI, they soon reached their limits and propelled the field towards statistics-driven approaches, whereby rules are inferred from data. The set of techniques — theoretical or empirical — that enable the inference of patterns from data, along with their real-world embodiment as computational inference systems, is collectively named ML. More concretely, ML is an engineering discipline [116] built upon the scientific pillars of mathematics, statistics and computer science and aimed at creating systems that make accurate predictions based on data.

Over the past 20 years, the field has grown to achieve some astonishing victories: defeating the standing world champions of Jeopardy and Go [82, 206]; predicting protein structure to near experimental accuracy [201, 118]; synthesizing high-quality images from text prompts [188, 195]; autonomous driving [8]; and simulating realistic human conversation [176, 175, 26]. With less fanfare, however, ML pervades our daily lives from menial activities such as shopping on Amazon [209] to life-altering ones like credit scoring [66] or delivering healthcare [172].

Broadly speaking, the type of questions addressed by ML are of the form

> *What are the best parameters of a computational inference system, as a function of a given task and a training data set, such that it is able to make accurate predictions for yet unseen data?* (Q)

The word "best" announces that the formal counterpart of (Q) is a problem of type (OPT) —

[1] Both Dantzig's Simplex algorithm and von Neumann's result on duality are dated to 1947, despite the cited publications marking later years. The latter first circulated as a personal note, now part of von Neumann's collected works (cited). The former was developed during Dantzig's time at the Pentagon and, while apparently communicated to the scientific community [64, Chapter 3], its first published form was several years delayed according to Dantzig and Cottle [64, p. 20]. More details can be found in the vivid historical notes of Dantzig [63, 62, 60], Polyak [185], Giorgi and Kjeldsen [95], and Singh and Eisner [208].

[2] A sizeable collection of additional examples is provided in the NEOS case studies at `https://neos-guide.org/case-studies`.

[3] The terms Artificial Intelligence and Machine Learning are often used interchangeably, leading to an unfortunate dilution of the meaning of both. A critical discussion on the need to meaningfully differentiate the two is initiated by Jordan [116] and nuanced by subsequent commentaries written in response (linked within the article).

in other words, to answer the former one must solve an instance of the latter (with very few exceptions). Mathematical optimization consequently emerges as the principal bearer of ML's computational burden and hence, is pressed to solve the challenges brought by its widespread application.

The typical learning problems nowadays are of large scale — an extreme example of this is GPT-3, with its 175 billion parameters and 45 TB of training data [25]. Even the more modest tasks such as classifying images from the ImageNet dataset [198] involve models with tens of millions of parameters trained on more than 100 GB worth of images [225]. Owing to its rapid expansion, ML is placing steady pressure on the field of optimization to develop algorithms that are fast, robust and able to handle large and high-dimensional learning problems. The need for such algorithms is additionally driven by the advent of smart devices (e.g., smartphones or IoT devices), which raise the challenge of fine-tuning models with scarce processing resources. Last but not least, efficient optimization algorithms are important for accelerating the hypothesis-experiment-results cycle in the study of critical system properties such as fairness [11], generalization ability [114, 236] or security and privacy [177]. These considerations motivate our focus on optimization algorithms with an emphasis on scalability.

1.2 Scalable algorithms

One of the early studies aiming to formalize the notion of scalability for generic computing systems describes it as "[…] the ability of a system to accommodate an increasing number of elements or objects, to process growing volumes of work gracefully, and/or to be susceptible to enlargement" [19]. Conversely, an unscalable system "[…] adds to labour costs or harms the quality of service".

In our case, the aforementioned "system" is understood to be an optimization algorithm. In this context, the "elements or objects" are the amount of data and the dimensionality of a problem. The ability to "process growing volumes of work gracefully" refers to a minimal or acceptable degradation of convergence speed as a function of growing data or problem dimensionality. Finally, we also include the *plug-and-play* character of a method as a marker of scalability, since it reduces the "labour cost" of hyperparameter tuning and often enables improved convergence by adapting to problem parameters on-the-fly.

In the sequel, we use terms such as *cheap* or *expensive* and their semantic relatives to denote the computational cost of an algorithmic operation or design decision. The dimensions of time and memory relative to which these coarse quantifiers are used will be evident from the context.

1.3 Problem structure, scalability and the optimization literature

Most of the results in this dissertation address convex instances of (OPT). This structural assumption is highly desirable due to its robustness and interpretability, and conveniently emerges in a broad range of scenarios. Firstly, several widely-used learning formulations are convex: linear regression and logistic regression as part of the more general class of maximum likelihood estimation with log-concave distributions [23, Section 7.1.1], classification with Support Vector Machines [217, 200], and Exact Matrix Completion [33]. Secondly, a number of important NP-hard combinatorial problems such as Max-Cut, Sparsest Cut and K-means clustering are well approximated via their convex relaxation as Semidefinite Programs (SDPs) [96, 7, 182]. Thirdly, convenient reformulations of neural-network training can surprisingly unearth convexity in a different space than that of the parameters, where the problem is non-convex [110, 184, 49]. Finally, convexity appears as a "hidden" property of some generic classes of non-convex problems (notably Quadratically Constrained Quadratic Programs), where the latter admit convex reformulations with a shared optimal value [15, 226]. The remainder of this chapter assumes convexity unless explicitly stated otherwise.

1.3.1 Optimization algorithms for constrained problems

An optimization algorithm is an iterative computational procedure that, at each step, updates its current approximation of the solution to (OPT) in response to "oracle feedback". The oracle is a "source of information about the problem to be solved" [162] — or, more concretely, a computational primitive providing (local) knowledge about the function f or set $\mathcal{X}$ at every iteration. Oracle queries abstract away logical units of computation that are indispensable to but otherwise unrevealing of the optimization mechanism itself, and are often the more expensive part of an iteration. Together with the rate of convergence, the cost of accessing a method's associated oracles determines its efficiency for a given class of problems and is thus a primary consideration in choosing a scalable algorithm for (OPT)[4].

First, function-related oracles reveal the structure of f at a queried point $\boldsymbol{x} \in \mathcal{X}$. More concretely, for a d-dimensional problem $\mathcal{X} \subseteq R^d$ and a smooth $f : \mathcal{X} \to \mathbb{R}$, zeroth-order oracles return the value of $f(\boldsymbol{x})$; first-order oracles additionally provide the gradient $\nabla f(\boldsymbol{x}) \in \mathbb{R}^d$; second-order oracles reveal the Hessian $\nabla^2 f(\boldsymbol{x}) \in \mathbb{R}^{d \times d}$, and so on (the order designation extends to the associated class of methods). Depending on the application, only a subset of them may be accessible — for example, derivatives are not available if f is given as a simulation or is otherwise a black box. Conversely, when f has an analytic expression, first and higher-order oracles are accessible (though not necessarily affordable to compute). Since the latter case occurs in a large portion of modern applications, we use it as the overarching assumption throughout this dissertation.

[4] A detailed treatment of black-box optimization complexity for predefined classes of problems is given in the seminal work of Nemirovsky and Yudin [162].

Unfortunately, merely being able to jot down the expression of higher-order derivatives does not readily translate into a practical advantage despite the additional information they provide. The high dimensionality of modern optimization problems renders even second-order methods prohibitively expensive. Take, for example, the task of learning a (poorly-performing[5]) linear classifier for ImageNet pictures cropped down to 64×64 RGB pixels. The problem dimension is $d = 12,288$, implying that the Hessian matrix in $\mathbb{R}^{d \times d}$ requires 1.2 GB of memory (in double precision float). The same problem using more reasonably sized images of 512×512 RGB pixels would need 4.9×10^3 GB of memory. In comparison, a first-order method requires a mere 9.8×10^{-5} and 6×10^{-3} GB to keep the gradient in memory for the two examples, respectively. Additionally, the time to compute these two oracles differs by an order of magnitude in favour of the latter — $\mathcal{O}(d^2)$ versus $\mathcal{O}(d)$. This scenario is prototypical and motivates the overwhelming preference for first-order methods in handling large and high-dimensional problems. Other more refined considerations include that higher-order methods' faster convergence (and consequently better accuracy) is no longer required in modern settings due to the inherently noisy data — we refer the reader to the reviews of Bottou, Curtis, and Nocedal [22] and Cevher, Becker, and Schmidt [39], and references therein. In short, first-order oracles are the scalable choice for modern applications, which motivates our focus on first-order optimization algorithms moving forward.

When it comes to enforcing constraints, scalability becomes a more delicate matter since it heavily depends on the structure of $\mathcal{X}$. Broadly speaking, constrained optimization methods can be split into projection-free and projection-based approaches. The literature in this area is vast, and we have no intent of covering it here — relevant references are given within the respective chapters. For now, we solely aim to describe the trade-offs underlying this dichotomy. The remainder of this section assumes that we seek to approximate (OPT) up to an ϵ-additive error in functional residual and, for simplicity, that f is L-smooth.

The classical algorithm for solving constrained problems over closed convex sets $\mathcal{X}$ is the Projected Gradient Descent (PGD) [97, 140] — a natural extension of the unconstrained Gradient Descent method, proposed by Cauchy in 1847 [139]. At every iteration $k > 0$, PGD refines the current approximation $\boldsymbol{x}_k$ by moving in the direction of the negative gradient $-\nabla f(\boldsymbol{x}_k)$ and projecting the result back onto the constraint set $\mathcal{X}$. Mathematically, this writes as

$$\boldsymbol{x}_{k+1} \coloneqq \operatorname{proj}_{\mathcal{X}}\left(\boldsymbol{x}_k - \gamma_k \nabla f(\boldsymbol{x}_k)\right), \tag{1.1}$$

where $\gamma_k > 0$ denotes the stepsize or update magnitude. The operator $\operatorname{proj}_{\mathcal{X}}$ is defined as

$$\operatorname{proj}_{\mathcal{X}}(\boldsymbol{x}) \coloneqq \operatorname*{argmin}_{\boldsymbol{y} \in \mathcal{X}} \left\{ \frac{1}{2} \left\| \boldsymbol{x} - \boldsymbol{y} \right\|^2 \right\}, \tag{1.2}$$

[5] In the absence of a literature reference, we report results from researcher Andrej Karpathy's blog for using a linear classifier on the ImageNet dataset: 3.0% *top-1 accuracy (and about* 10% *top-5)*. Even considering tuning issues, these scores greatly contrast the higher-than-70% accuracy of various neural networks on the same task [203]. Link: `https://karpathy.github.io/2015/03/30/breaking-convnets`.

where $\|\cdot\|$ denotes the Euclidean norm over $\mathbb{R}^d$, and is often referred to as the Projection Oracle (PO). Expression (1.2) is nothing but the definition of the more general proximal operator [157, 158] particularized to the indicator function of set $\mathcal{X}$.

PGD transfers all the desirable properties of Gradient Descent to the constrained setting: a $\mathcal{O}\left(\epsilon^{-1}\right)$ convergence for smooth and convex functions, and a linear $\mathcal{O}\left(\kappa \log(\epsilon^{-1})\right)$ convergence under strong convexity, where κ is the condition number; the possibility of acceleration to $\mathcal{O}\left(\epsilon^{-1/2}\right)$ and $\mathcal{O}\left(\sqrt{\kappa}\log(\epsilon^{-1})\right)$ for the same settings, respectively, under a slightly modified iteration [163, 215, 14]; and a straightforward extension to non-differentiable and Lipschitz continuous problems via subgradients, with a convergence of $\mathcal{O}\left(\epsilon^{-2}\right)$ and $\mathcal{O}\left(\epsilon^{-1}\right)$ for the convex and strongly-convex cases, respectively [204, 133]. Moreover, the PO-based approach readily generalizes to problems with more complex structures (e.g., when $\mathcal{X}$ is the intersection of two closed and convex sets) by leveraging duality and operator splitting schemes with similar convergence guarantees [42, 55, 222, 44].

However, this extensive list of benefits is conditioned on a computationally affordable PO for the considered application. A closer look at the PGD iteration reveals that it can be equivalently rewritten as the minimization of a quadratic form over $\mathcal{X}$

$$\boldsymbol{x}_{k+1} \coloneqq \underset{\boldsymbol{x}\in\mathcal{X}}{\arg\min} \left\{ \langle \nabla f(\boldsymbol{x}_k), \boldsymbol{x}\rangle + \frac{1}{2\gamma_k} \| \boldsymbol{x} - \boldsymbol{x}_k \|^2 \right\}, \tag{1.3}$$

where we used definition (1.2) and developed the square. In this light, it is apparent that closed-form or efficient solutions cannot be expected for arbitrary $\mathcal{X}$. Examples of sets with efficient projections include box constraints, ℓ_p balls for $p \in \{1, 2, \infty\}$, and some polytopes such as the standard simplex. In contrast, projections are expensive for a number of important cases like nuclear norm balls and the positive semidefinite cone, both of which require $\mathcal{O}\left(d^3\right)$ computations for the SVD and eigenvalue decompositions, respectively; arbitrary ℓ_p balls for $p \notin \{1, 2, \infty\}$, which rely on an iterative procedure with $\mathcal{O}\left(d\epsilon^{-2}\right)$ convergence; or the flow, Birkhoff and matching polytopes which have a high polynomial dependence on d, are solved by slow iterative methods, or no algorithm is known, respectively. Further details, complexities and references are given by Combettes and Pokutta [53]. To summarize, the fast convergence and high versatility of PO-based methods make them the scalable choice as long as $\mathcal{X}$ allows efficient projections, which only happens for a handful of simple constraint sets.

The cases in which projections are difficult gave rise to the line of research studying projection-free methods. The representative algorithm for this class is the Frank-Wolfe (FW) or Conditional Gradient[6] (CG) method proposed by Frank and Wolfe [84]. This method removes the quadratic term in (1.3) and solves the resulting simpler linear minimization subproblem over $\mathcal{X}$ to determine the update direction. At iteration $k > 0$, FW proceeds as

$$\boldsymbol{x}_{k+1} \in \underset{\boldsymbol{x}\in\boldsymbol{x}_k+\gamma_k(\mathcal{X}-\boldsymbol{x}_k)}{\arg\min} \left\{ \langle \nabla f(\boldsymbol{x}_k), \boldsymbol{x}\rangle \right\}, \tag{1.4}$$

[6]This naming version is due to Levitin and Polyak [140].

where $\gamma_k \in (0, 1]$ denotes the stepsize. The method is inherently feasible due to its convex combination-based update rule (provided that $\boldsymbol{x}_0 \in \mathcal{X}$). Different from PGD, the FW algorithm requires $\mathcal{X}$ to be additionally bounded, lest the minimizer in (1.4) take infinite values.

Much as before, FW's tractability hinges on the efficiency of the set's Linear Minimization Oracle (LMO), defined as

$$\operatorname{lmo}_{\mathcal{X}}(\boldsymbol{x}) \coloneqq \operatorname*{argmin}_{\boldsymbol{y} \in \mathcal{X}} \left\{ \langle \boldsymbol{y}, \boldsymbol{x} \rangle \right\}. \tag{1.5}$$

Subproblem (1.5) is notably efficient in the aforementioned cases for which projections are costly, thus making FW the algorithm of choice in applications such as video co-localization [117], training structural SVMs [132], optimal transport [180] or low-rank matrix retrieval [86], to name a few. A detailed complexity comparison of PO and LMO oracles for a collection of practically relevant sets $\mathcal{X}$ is given by Combettes and Pokutta [53] and Braun et al. [24]. Additional benefits of FW are that the method is invariant to affine transformations of $\mathcal{X}$ and provides solutions with a sparse representation [112]. The latter property makes this algorithm particularly well-suited for large-scale optimization and accounts for the increased research interest it has drawn over the past decade.

However, the vanilla FW algorithm given by (1.4) comes with its share of shortcomings. Different from PGD, its $\mathcal{O}\left(\epsilon^{-1}\right)$ convergence upper bound is met by an information-theoretic lower bound of the same order for problems constrained to generic sets $\mathcal{X}$ [134]. In addition, a direct extension to non-smooth settings via subgradients is not possible, as shown by the counterexample of Nesterov [165]. While faster $\mathcal{O}\left(\epsilon^{-1/2}\right)$ rates for the vanilla FW method are achieved in restricted settings of f and $\mathcal{X}$ [91, 123], in general, improved convergence ensues from the development of algorithmic variants. Such variants rely on either additional problem structure, stronger LMO-like oracles, or an altogether modified iteration. For example, linear convergence is achieved for smooth and strongly convex objectives over polytopes by the Away-Step FW, which maintains an active set of relevant vertices [131]. In the same setting, local acceleration is established by coupling the Away-Step FW with an accelerated, projection-based algorithm [71, 37]. For generic sets $\mathcal{X}$ and smooth f, fast convergence relative to the number of gradient evaluations, but not LMO calls, is attained via Nesterov's accelerated scheme with inexact projections computed by FW [135]. Finally, non-differentiable objectives are tackled by using the method of Lan and Zhou [135] in conjunction with smoothing [211]. Numerous other variants exist and are thoroughly discussed in the monograph of Braun et al. [24]. In short, the cheap LMO and sparsity-inducing properties of FW methods make them the scalable option when dealing with complex constraints and large problems, though at the cost of a (usually) slower convergence rate.

We conclude this section by reiterating an earlier point: choosing the first-order constrained optimization algorithm to solve problems of type (OPT) heavily depends on the problem structure, and there is no one-size-fits-all approach with regard to scalability.

1.3.2 Opportunities for improving scalability

Beyond the choice of optimization paradigm embodied in the function oracle and feasibility-enforcing mechanism, scalability becomes a matter of identifying adequate and provable (as far as this thesis is concerned) trade-offs. Typical examples include reducing both iteration time and memory requirements by using a subset of the data for computing the gradient, at the expense of a slower convergence in terms of the target accuracy ϵ; using stepsize-setting subroutines for faster empirical convergence at the cost of additional (though generally cheap) computation per iteration; or improving convergence rates as a function of ϵ, by solving more expensive subproblems every iteration.

Concretely, given a first-order algorithm for constrained optimization, one may seek scalability opportunities through the following (non-exhaustive) set of targeted questions.

1) *Is processing the entire dataset at every iteration essential for convergence?* While gradients are scalable oracles with respect to the problem dimension (as per Section 1.3.1), their computation may still pose a burden when the number of data points is large[7]. This question motivates the development of stochastic or randomized algorithmic variants which process only a subset of data points per iteration, and which are indispensable to ML due to their cheap iterations (e.g., Stochastic Gradient Descent [191]).

2) *Can the algorithm be parallelized? Can it be distributed?* Such questions drive the efforts to optimally leverage the computing infrastructure for solving large and high-dimensional problems. Time and space requirements are alleviated by spreading the computation over multiple workers running in parallel.

3) *Is the stepsize adequate with respect to a given metric (e.g., local function curvature, progress per iteration or sequence of iterations)?* An algorithm's stepsize commonly depends on global constants related to the structure of f (e.g., the global smoothness constant). Such stepsizes may lead to slow convergence by being overly conservative, since these constants are usually not representative of f's local geometry. This question underlies the development of stepsize schedules that speed up empirical convergence, oftentimes through adaptivity, and have the added benefit of reducing the labour cost of stepsize tuning for every instance of (OPT).

4) *Does the algorithm adapt to varying structural properties of the problem and converge optimally, without modifications?* Related to point 3), this question motivates the study of algorithmic "universality" as an avenue for enhancing the plug-and-play nature of optimization algorithms [120, and references therein].

5) *Does the algorithm optimally leverage the structure of a given problem?* This question underlies the development of methods tailored to practically relevant subclasses of problems

[7]Typical objectives in ML are represented as the sum of individual errors incurred by the model on each data point. As such, computing the gradient requires a full pass over the dataset.

with a prescribed form (e.g., additive composition). Such approaches eschew the lower bounds for black-box optimization of generic function classes [161] and generally lead to provably faster convergence rates.

6) *Does the entire variable $\boldsymbol{x}$ in* (OPT) *have to be memorized, or does an approximation of it suffice?* This line of inquiry is especially relevant for matrix optimization problems with high memory requirements (notably SDPs) and has led to the integration of randomized sketching techniques ensuring storage-optimality for a series of methods [235, and references therein].

The above questions address scalability improvements solely from an algorithmic perspective. Other avenues towards this goal involve, for example, problem modelling, data processing or designing novel computing infrastructure. They are, however, outside of the scope of this thesis.

1.4 Contributions and manuscript organization

This dissertation rigorously tackles points 1), 3) and 5) above for a selection of constrained and (mostly) convex problems. The algorithmic solutions we propose are geared towards enhancing scalability in the sense discussed in Section 1.2. Our methods are empirically motivated and theoretically sound, as they are each accompanied by rigorous convergence guarantees.

We further summarize our contributions while deferring the techniques and formalism associated with each problem to the respective chapters.

1.4.1 Frank-Wolfe-type methods for stochastically constrained stochastic objectives

Our first contribution is to propose three novel Frank-Wolfe-type methods for minimizing convex stochastic objectives subject to stochastic linear inclusion constraints. The key benefit of our algorithms is that they consider only a subset of the constraints per iteration, thus gaining an edge over methods that process them in full at every step.

Specifically, in Chapter 2 we address problems of the form

$$\min_{\boldsymbol{x}\in\mathcal{X}} \mathbb{E}_{\xi}\left[f(\boldsymbol{x},\xi)\right], \quad \text{such that } \boldsymbol{A}(\xi)\boldsymbol{x}\in\boldsymbol{b}(\xi) \text{ almost surely}, \tag{1.6}$$

where $f(\boldsymbol{x},\xi):\mathbb{R}^d\to\mathbb{R}$ are random convex functions with L_f-Lipschitz gradient; $\mathcal{X}\subseteq\mathbb{R}^d$ is a convex and compact set; $\boldsymbol{A}(\xi)$ is an $m\times d$ matrix-valued random variable; and $\boldsymbol{b}(\xi)\in\mathbb{R}^m$ is a closed and convex random set for which we assume that projections are cheap. This template covers at once finite-sum formulations typical for ML applications and scenarios where the objective and linear constraints are revealed in an online fashion or are otherwise too large to keep in memory. For example, instances of (1.6) naturally arise from SDP-relaxations of combinatorial

problems, whose number of linear constraints is polynomial in the problem dimension d (e.g., $\mathcal{O}(d^3)$ for the Sparsest Cut problem [7]).

Existing approaches either address this template through projection-based methods which scale poorly when $\mathcal{X}$ is the semidefinite cone [81, 179] or process the entire set of linear constraints at every iteration with Frank-Wolfe-type methods [232, 143, 101].

We propose two Frank-Wolfe-type methods for tackling problem (1.6) in full generality. Our algorithms handle the constraints stochastically, and rely on smoothing and variance reduction used in conjunction with LMO steps. The first method proposes a simple scheme using gradient estimators in the spirit of Mokhtari, Hassani, and Karbasi [155], uses fixed-size minibatches and has $\mathcal{O}(\epsilon^{-6})$ convergence[8] in terms of both the number of LMO computations and that of gradient oracles. Our second method reduces the number of required LMOs to $\mathcal{O}(\epsilon^{-2})$ and that of stochastic gradient computations to $\mathcal{O}(\epsilon^{-4})$ by using minibatches of increasing size and the SPIDER gradient estimator [80]. The difference in convergence rates emphasizes the trade-off between the computational cost per iteration and the number of iterations required to reach the constrained optimum.

Further, we propose a third method for the finite-sum restriction of objective (1.6). This algorithm leverages a SAG-like gradient estimator [199] to remove the aforementioned increasing batch size requirement, achieving $\mathcal{O}(\epsilon^{-2})$ complexity in terms of both the number stochastic gradients and that of LMOs. The trade-off, this time, rests in requiring additional memory (linear in the number of constraints).

We provide numerical experiments that illustrate the practical performance of all our methods against relevant baselines.

In this instance, scalability is improved through stochasticity, which allows for cheaper iterations in terms of both time and memory compared to prior work.

1.4.2 A Frank-Wolfe generalization for composite non-differentiable objectives

Our second contribution is to generalize Frank-Wolfe-type methods to a class of non-differentiable objectives — a setting in which the classical Frank-Wolfe algorithm is known *not* to converge [165].

Specifically, in Chapter 3 we address problems of the form

$$\min_{\boldsymbol{x}\in\mathcal{X}} F(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{x}), \tag{1.7}$$

where $\mathcal{X} \in \mathbb{R}^d$ is a convex and compact set, $F : \mathbb{R}^n \times \mathcal{X} \to \mathbb{R}$ is a convex, subhomogenous but possibly *non-differentiable* function and $\boldsymbol{f} : \mathcal{X} \to \mathbb{R}^n$ is a smooth mapping satisfying a bounded

[8] The method performs much faster than its worst case bound in practice, closer to $\mathcal{O}(\epsilon^{-2})$.

curvature condition à la Jaggi [112]. We further assume that projections onto $\mathcal{X}$ are costly. This template importantly covers max-type minimization problems, with applications such as multi-objective optimization [153, Chapter 3.1] and constrained ℓ_∞ regression.

Existing approaches either tackle (1.7) as a black-box, thus being subject to the $\mathcal{O}\left(\epsilon^{-2}\right)$ lower bounds prescribed for this class of objectives [211, 68, 128]; leverage problem structure in conjunction with projections, which may be expensive [77]; or otherwise study this problem in a similar setting to ours, but under more restrictive assumptions on $\boldsymbol{f}$ and without concern for scalability [73].

We propose generalizations of the basic FW method (1.4) and the Conditional Gradient Sliding (CGS) algorithm [135], which rely on a modified LMO and eschew the $\mathcal{O}\left(\epsilon^{-2}\right)$ lower bounds by leveraging the problem's structure. Concretely, we handle the objective's smooth and non-differentiable components separately, linearizing only the former. The resulting generalized LMO can be efficiently computed in several practical applications, notably including max-type minimization problems. We provide the basic version of our method with an affine-invariant analysis and prove global convergence rates of $\mathcal{O}\left(\epsilon^{-1}\right)$ and $\widetilde{\mathcal{O}}\left(\epsilon^{-2}\right)$ for convex and non-convex[9] objectives, respectively. Furthermore, we accelerate our basic method using the CGS framework in the convex case, thus reducing the number of Jacobian computations required to reach ϵ accuracy to $\mathcal{O}\left(\epsilon^{-1/2}\right)$. Finally, we report numerical experiments supporting our theoretical results.

In this instance, we enhance scalability by leveraging the problem structure within the algorithm and achieve provably faster convergence than prior work. We additionally propose a generalized LMO which may be more efficient than projections for some applications.

1.4.3 An adaptive, linesearch-free primal-dual algorithm

Our final contribution is to propose an adaptive primal-dual algorithm for structured composite problems, whose stepsize attunes to the local function curvature without relying on subroutines or unknown quantities.

Specifically, in Chapter 4, we address convex-concave saddle point problems of the form

$$\min_{\boldsymbol{x}\in\mathbb{R}^d} \max_{\boldsymbol{y}\in\mathbb{R}^m} \langle \boldsymbol{A}\boldsymbol{x}, \boldsymbol{y}\rangle + f(\boldsymbol{x}) - g^*(\boldsymbol{y}), \tag{1.8}$$

where $f : \mathbb{R}^d \to \mathbb{R}$ is convex, differentiable and has a locally Lipschitz-continuous gradient; $g^* : \mathbb{R}^m \to \mathbb{R}$ is the Fenchel dual of a convex and lower-semicontinuous function g, which we assume to be proximal-friendly; and $\boldsymbol{A} : \mathbb{R}^d \to \mathbb{R}^m$ is a linear mapping. This formulation is equivalent via duality to

$$\min_{\boldsymbol{x}\in\mathbb{R}^d} f(\boldsymbol{x}) + g(\boldsymbol{A}\boldsymbol{x}),$$

[9] The non-convex rate is attained on a non-standard metric. Relevant details are deferred to Chapter 3.

which encompasses the constrained setting whenever g is the indicator function of a closed and convex set. This class of problems has a broad range of applications in fields such as signal processing, machine learning, inverse problems, telecommunications and many others [127].

Existing approaches either address the more general variational inequality formulation [146], thus failing to leverage the problem structure at the expense of slower convergence; rely on fixed stepsizes depending on global smoothness constants [55, 222, 144, 47, 75], which may lead to overly conservative iterate updates; or otherwise rely on linesearch to achieve adaptivity, at the expense of additional iterations spent in subroutines [149].

We propose an adaptive variant of the Condat-Vũ algorithm [55, 222] which, at each iteration, updates the stepsize according to the local function geometry using readily-available quantities ($\|A\|$ and recent gradients of f). In doing so, we remove the need for more costly linesearch subroutines. The method alternates between gradient steps in the primal space and proximal steps in the dual space, achieving an $\mathcal{O}\left(\epsilon^{-1}\right)$ ergodic convergence under the mild assumption of local smoothness of f. Furthermore, we prove that our algorithm converges linearly when f is additionally locally strongly convex and A has full row rank. We provide numerical experiments to illustrate the practical performance of our method against relevant baselines.

In this instance, scalability is improved via stepsize adaptivity, which allows for faster empirical convergence compared to fixed-stepsize methods. Moreover, our manner of ensuring adaptivity eschews linesearch by relying on readily available quantities instead, thus being more efficient than existing approaches.

1.5 Preliminaries and notation

This section collects the mathematical concepts and notation used throughout the dissertation. Problem-specific conventions will be introduced in the respective chapters.

Vector related notation. We use bold lowercase letters to denote vectors and differentiate them from scalar values. We denote by $\mathbf{1}_d$ and $\mathbf{0}_d$ the all-ones and all-zeros vectors of dimension d, respectively. We use $\boldsymbol{e}_i$ to denote the i^{th} vector of the canonical basis in the appropriate dimension. Unless stated otherwise, we use $\|\cdot\|$ to express the Euclidean norm and $\langle\cdot,\cdot\rangle$ to denote the corresponding inner product.

Matrix related notation. We use bold capital letters to denote matrices and linear operators. We use $\boldsymbol{I}$ to denote the identity matrix of appropriate dimensions depending on the context. We let $\mathbf{1}_{d\times m}$ and $\mathbf{0}_{d\times m}$ to be the all-ones and all-zeros matrices of size $d \times m$, respectively. The transpose of a matrix $\boldsymbol{A}$ is written as $\boldsymbol{A}^\top$. We denote by $\operatorname{Tr}(\cdot)$ the trace of a matrix; by $\|\cdot\|$ its spectral norm; by $\|\cdot\|_*$ its nuclear norm; and by $\|\cdot\|_F$ its Frobenius norm with the associated Frobenius inner product $\langle\cdot,\cdot\rangle$ defined as $\langle \boldsymbol{A}, \boldsymbol{B}\rangle \coloneqq \operatorname{Tr}(\boldsymbol{B}^\top \boldsymbol{A})$ for appropriately sized matrices $\boldsymbol{A}$

and $\boldsymbol{B}$. We use $\succeq$ and $\succ$ to indicate the Löwner order.

Set related notation. The indicator function of a set $\mathcal{X}$ is defined by

$$\iota_{\mathcal{X}}(\boldsymbol{x}) \coloneqq \begin{cases} 0, \text{ if } \boldsymbol{x} \in \mathcal{X}, \\ \infty, \text{ otherwise.} \end{cases} \tag{1.9}$$

The distance between a point $\boldsymbol{x} \in \mathbb{R}^d$ and a closed convex set $\mathcal{X} \subseteq \mathbb{R}^d$ is defined as

$$\operatorname{dist}(\boldsymbol{x}, \mathcal{X}) \coloneqq \inf_{\boldsymbol{y} \in \mathcal{X}} \left\| \boldsymbol{y} - \boldsymbol{x} \right\|. \tag{1.10}$$

The corresponding quantity for matrices is defined with respect to the Frobenius norm.

We denote the diameter of a set $\mathcal{X}$ as

$$\mathcal{D}_{\mathcal{X}} \coloneqq \sup_{\boldsymbol{z}, \boldsymbol{y} \in \mathcal{X}} \left\{ \left\| \boldsymbol{z} - \boldsymbol{y} \right\| \right\}. \tag{1.11}$$

The corresponding quantity for matrices is defined with respect to the Frobenius norm.

We use the notation $\Delta_n \coloneqq \{\boldsymbol{\lambda} \in \mathbb{R}^n_+ : \langle \boldsymbol{\lambda}, \mathbf{1}_n \rangle = 1\}$ to denote the standard n-dimensional simplex. The notation $\mathbb{S}^d_+$ defines the cone of symmetric positive-semidefinite matrices in $\mathbb{R}^{d \times d}$. Finally, we denote by $[n]$ the set of integers $\{1, 2, \dots n\}$.

Function related notation. For a differentiable function $f : \mathbb{R}^d \to \mathbb{R}$, we denote by $\nabla f : \mathbb{R}^d \to \mathbb{R}^d$ its gradient, and for a twice differentiable function, we write its Hessian as $\nabla^2 f : \mathbb{R}^d \to \mathbb{R}^{d \times d}$. We differentiate vector-valued functions from scalar-valued functions by writing the former in bold letters, e.g., $\boldsymbol{f} : \mathbb{R}^d \to \mathbb{R}^m$, $\boldsymbol{f} \coloneqq (f_1, f_2, \dots f_m)$ with $f_i : \mathbb{R}^d \to \mathbb{R}$. We denote the Jacobian of such vector-valued functions by $\nabla \boldsymbol{f}(\boldsymbol{x}) \coloneqq \sum_{i=1}^m \boldsymbol{e}_i \nabla f_i(\boldsymbol{x})^\top \in \mathbb{R}^{m \times d}$.

We use the shorthand l.s.c. to mark the lower semi-continuity of a function f. For non-differentiable convex functions $f : \mathbb{R}^d \to \mathbb{R}$, we denote their subdifferential at a point $\boldsymbol{x}$ as the set

$$\partial f(\boldsymbol{x}) \coloneqq \left\{ \boldsymbol{v} \in \mathbb{R}^d \mid f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \langle \boldsymbol{v}, \boldsymbol{y} - \boldsymbol{x} \rangle \forall \boldsymbol{y} \right\}.$$

We say that a function $f : \mathbb{R}^d \to \mathbb{R}$ is Lipschitz continuous with constant $C > 0$ if it satisfies

$$|f(\boldsymbol{x}) - f(\boldsymbol{y})| \leq C \left\| \boldsymbol{x} - \boldsymbol{y} \right\|, \ \forall \boldsymbol{x}, \boldsymbol{y}. \tag{1.12}$$

We say that a function $f : \mathbb{R}^d \to \mathbb{R}$ is L-smooth if it is differentiable and its gradient is Lipschitz

continuous with constant $L > 0$:

$$\left\| \nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y}) \right\| \leq L \left\| \boldsymbol{x} - \boldsymbol{y} \right\|, \ \forall \boldsymbol{x}, \boldsymbol{y}. \tag{1.13}$$

For $\mathcal{C}^2$ functions condition (1.13) is equivalent to the largest magnitude eigenvalue being bounded. Furthermore, f is locally smooth if ∇f is Lipschitz continuous on any compact subset $\mathcal{C}$: $\forall \mathcal{C} \subset \mathbb{R}^d$, $\exists L_{\mathcal{C}} \in (0, \infty)$ such that $\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})\| \leq L_{\mathcal{C}} \|\boldsymbol{x} - \boldsymbol{y}\|, \forall \boldsymbol{x}, \boldsymbol{y} \in \mathcal{C}$.

We say that a differentiable function $f : \mathbb{R}^d \to \mathbb{R}$ is μ-strongly convex if

$$f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \langle \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \rangle + \frac{\mu}{2} \left\| \boldsymbol{x} - \boldsymbol{y} \right\|^2, \forall \boldsymbol{x}, \boldsymbol{y}. \tag{1.14}$$

Similarly, f is locally strongly convex if it is strongly convex on any compact subset $\mathcal{C}$: $\forall \mathcal{C} \subset \mathcal{X}$, $\exists \mu_{\mathcal{C}} > 0$ such that $f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \langle \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \rangle + \frac{\mu_{\mathcal{C}}}{2} \left\| \boldsymbol{x} - \boldsymbol{y} \right\|^2, \forall \boldsymbol{x}, \boldsymbol{y} \in \mathcal{C}$.

The proximal operator of a function $f : \mathbb{R}^d \to \mathbb{R} \cup \{\infty\}$ is given by

$$\operatorname{prox}_f(\boldsymbol{x}) \coloneqq \arg\min_{\boldsymbol{y}} \left\{ f(\boldsymbol{y}) + \frac{1}{2} \left\| \boldsymbol{x} - \boldsymbol{y} \right\|^2 \right\}. \tag{1.15}$$

Whenever f is proper, closed and convex, $\operatorname{prox}_f(\boldsymbol{x})$ is a singleton, for any $\boldsymbol{x}$. In the special case when $f \equiv \iota_{\mathcal{X}}$, for a non-empty closed and convex set $\mathcal{X}$, then $\operatorname{prox}_f \equiv \operatorname{proj}_{\mathcal{X}}$.

Optimality conditions and notation. Any vector with a star superscript denotes an optimal point with respect to some minimization problem, usually evident in the context — for example, $\boldsymbol{x}^\star$ with respect to problem (OPT). Correspondingly, any function with a star superscript denotes the relevant function evaluated at one of its optimal points — for example, $f^\star \equiv f(\boldsymbol{x}^\star)$.

Unless specified otherwise, we wish to solve problems of type (OPT) up to an $\epsilon > 0$ additive error with respect to the functional residual. Concretely, we seek approximate solutions $\boldsymbol{x} \in \mathcal{X}$ for which

$$f(\boldsymbol{x}) - f^\star \leq \epsilon. \tag{1.16}$$

Whenever our methods are only approximately feasible, we seek a point $\boldsymbol{x}$ which, in addition, satisfies

$$\operatorname{dist}(\boldsymbol{x}, \mathcal{X}) \leq \epsilon. \tag{1.17}$$

2 Frank-Wolfe-type methods for stochastically constrained stochastic objectives

This chapter is based on the published work Vladarean et al. [219], presented at ICML 2020 (Sections 2.1 to 2.5), and Dresdner et al. [74], presented at AISTATS 2022 (Section 2.6). Only a subset of the latter results are included, based on their relevance to the topic of handling constraints stochastically.

Co-authors of [219]: Ahmet Alacaoglu, Ya-Ping Hsieh, and Volkan Cevher

Contributions

M.-L. Vladarean	— methodology 40%, formal derivations 90%, writing 80%, experiments 100%
A. Alacaoglu	— methodology 50%, formal derivations 10%, writing 20%
Y.-P. Hsieh	— methodology 5%, writing – review and editing
V. Cevher	— methodology 5%, project administration, supervision

Co-authors of [74]: Maria-Luiza Vladarean, Gunnar Rätsch, Francesco Locatello, Volkan Cevher, and Alp Yurtsever

Contributions

G. Dresdner	— methodology 50%, formal derivations 60%, writing 60%, experiments 100%
M.-L. Vladarean	— methodology 30%, formal derivations 40%, writing 40%
F. Locatello	— methodology 10%, writing – review and editing
G. Rätsch	— project administration, supervision
V. Cevher	— writing – review and editing, project administration, supervision
A. Yurtsever	— methodology 10%, writing – review and editing, supervision

Summary. In this chapter, we propose three novel Conditional Gradient (or Frank-Wolfe-type) methods for solving stochastic convex optimization problems with a large number of linear constraints. Instances of this template naturally arise from SDP-relaxations of combinatorial problems, which involve a number of constraints polynomial in the problem dimension. The most important feature of our framework is that only a subset of the constraints is processed at each iteration, thus gaining a computational advantage over prior works that require full passes. Our algorithms rely on variance reduction and smoothing used in conjunction with Conditional Gradient steps, and are accompanied by rigorous convergence guarantees. We provide numerical experiments that illustrate the practical performance of our methods against relevant baselines.

2.1 Introduction

We study the following optimization template:

$$\min_{\boldsymbol{x}\in\mathcal{X}} f(\boldsymbol{x}) \coloneqq \mathbb{E}\big[f(\boldsymbol{x},\xi)\big], \text{ such that } \boldsymbol{A}(\xi)\boldsymbol{x} \in \boldsymbol{b}(\xi) \text{ almost surely,} \tag{2.1}$$

where $f(\boldsymbol{x},\xi):\mathbb{R}^d \to \mathbb{R}$ are random convex functions with L_f-Lipschitz gradient, $\mathcal{X}$ is a convex and compact set of $\mathbb{R}^d$, $\boldsymbol{A}(\xi)$ is an $m \times d$ matrix-valued random variable, and $\boldsymbol{b}(\xi)$ are closed and convex random sets in $\mathbb{R}^m$ for which we assume projections are affordable.

Stochastically constrained convex optimization problems have recently gained interest in the machine learning community, as they provide a convenient and powerful framework for handling instances subject to a large, or even infinite number of constraints. For example, convex feasibility and optimal control problems have variables lying in a possibly infinite intersection of stochastic, projectable constraint sets, and hence are tackled through this lens by Patrascu and Necoara [179]. Xu [229] also studies the minimization of a stochastic objective controlled by a very large number of stochastic functional constraints, with application to stochastic linear programming. Finally, put forth by Fercoq et al. [81], extensions to situations where the number of constraints is unknown (e.g. online settings) can be modelled by a template similar to (2.1), thus addressing important applications such as online portfolio optimization.

In this chapter, we are interested in a class of applications which can benefit from being cast under template (2.1), namely semidefinite programs (SDPs) with a large number of linear constraints, such as arise in combinatorial optimization. A prominent example in machine learning is the k-Means clustering problem, whose SDP relaxation comprises $\mathcal{O}\left(d^2\right)$ linear constraints where d is the number of data samples [182]. Maximum a posteriori estimation [108], quadratic assignment [240, 27], matrix completion [2], k-Nearest Neighbour classification [224], Max Cut [96] and Sparsest Cut [7] are other relevant SDP instances with linear constraints of order $\mathcal{O}\left(d^2\right)$ or $\mathcal{O}\left(d^3\right)$. Coupled with large input dimensions, such SDPs become problematic for most existing methods, due to the high cost of processing the constraints in full during optimization.

In contrast, casting such SDPs into (2.1) suggests a simple solution: treat the linear constraints

stochastically by only accessing a random subset at each iteration, then solve (2.1) using cheap gradient methods. However, the bottleneck in executing this idea is that existing methods require efficient projections onto $\mathcal{X}$, whereas projecting onto the semidefinite cone amounts to full singular value decompositions — a prohibitively expensive operation even for moderate problem dimensions. We hence ask:

> *Does a scalable method exist for solving* (2.1) *when the set $\mathcal{X}$ does not have an efficient projection oracle?*

We resolve the above challenge in the positive. To this end, we borrow tools from the Conditional Gradient methods (CGMs) [85, 111], which rely on the generally cheaper *Linear Minimization Oracles* (LMO), rather than their projection counterparts. In particular, as the Lanczos method enables an efficient LMO computation for the spectrahedron [5], CGMs have already been proposed for solving SDPs [111, 92, 232, 143]. However, none of these methods can handle the constraints stochastically.

In a nutshell, our approach relies on *homotopy smoothing* of the stochastic constraints in conjunction with CGM steps and a carefully chosen variance reduction procedure. Our analysis gives rise to two fully stochastic algorithms for solving problem (2.1) without projections onto $\mathcal{X}$. The first of the methods, H-SFW1, relies on a single sample (or fixed batch size) for computing the variance-reduced gradient and converges at a cost of $\mathcal{O}(\epsilon^{-6})$ LMO calls and $\mathcal{O}(\epsilon^{-6})$ stochastic first-order oracle (SFO) calls. The second, H-SPIDERFW, uses batches of increasing size under the SPIDER variance reduction scheme [80] and attains a theoretical complexity of $\mathcal{O}(\epsilon^{-2})$ LMO calls and $\mathcal{O}(\epsilon^{-4})$ SFO calls. The difference in convergence rates emphasizes the trade-off between the computational cost per iteration and the number of iterations required to reach the constrained optimum.

2.2 Related work

The results presented in this chapter lie at the intersection of several lines of research we now review.

Proximal methods for almost sure constraints. Problems of similar formulation to (2.1) have been addressed in prior literature under the assumption of an efficient projection oracle over $\mathcal{X}$. Patrascu and Necoara [179], Xu [229], and Fercoq et al. [81] solve these problems via stochastic proximal methods and attain a complexity of $\mathcal{O}(\epsilon^{-2})$ SFO calls, which is known to be optimal even for unconstrained stochastic optimization. In particular, Patrascu and Necoara [179] study convex constrained optimization, where the constraints are expressed as a (possibly infinite) intersection of stochastic, closed, convex and projectable sets $\mathcal{X}_\xi$. Problem (2.1) can be partly cast to this template, with $\boldsymbol{A}(\xi)\boldsymbol{x} \in \boldsymbol{b}(\xi)$ being the homologues of $\mathcal{X}_\xi$. However, our additional set $\mathcal{X}$ does not allow for efficient projections, making this framework inapplicable.

Xu [229] solves a convex constrained optimization problem over a convex set $\mathcal{X}$, subject to a large number of convex functional constraints f_j, $j \in [M]$. The functions f_j are sampled uniformly at random during optimization, which corresponds to a finitely sampled instance of problem (2.1) for affine f_j. However, we meet again with the limiting condition that projections onto $\mathcal{X}$ are computationally expensive in our setting.

Finally, Fercoq et al. [81] study convex problems subject to a possibly infinite number of almost sure linear inclusion constraints, a template which closely resembles ours. The limitation, however, lies in their inclusion of a proximal-friendly component in the objective used to perform stochastic proximal gradient steps. While this template encompasses ours when the latter component is $\iota_{\mathcal{X}}$, performing projections (i.e., proximal steps w.r.t. the indicator function) is assumed prohibitively expensive in our setting.

Conditional Gradient methods for constrained optimization. The CGM or Frank-Wolfe method was first proposed in the seminal work of Frank and Wolfe [85], and its academic interest has witnessed a resurgence in the past decade. The advantage of CGMs lies in the low per-iteration cost of the LMO, alongside their ability to produce sparse solutions — a comprehensive treatment of these methods is provided by Jaggi [111] and Braun et al. [24]. In comparison to projection-based approaches, the LMO is cheaper to compute for several important domains, amongst which the spectrahedron, polytopes emerging from combinatorial optimization, and ℓ_p norm-induced balls [89]. Consequently, CG-type methods have been studied under various assumptions by Hazan [101], Clarkson [50], Hazan and Kale [103], Jaggi [111], Lan [134], and Balasubramanian and Ghadimi [9], and have been incorporated as cheaper subsolvers into algorithms which originally relied on projection oracles [135, 141].

CGMs have been further extended to the setting of convex composite minimization via the Augmented Lagrangian framework by Gidel, Pedregosa, and Lacoste-Julien [94], Silveti-Falls, Molinari, and Fadili [207], and Yurtsever, Fercoq, and Cevher [231]. Most relevant to our setting, CGM-based quadratic penalty methods have been studied for convex problems with constraints of the form $\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b} \in \mathcal{K}$, where $\mathcal{K}$ is a closed, convex set [232, 143]. We compare our methods against the latter two in Section 2.4.5.

Variance reduction. Stochastic variance reduction (VR) methods have gained popularity in recent years following their initial study by Roux, Schmidt, and Bach [197], Johnson and Zhang [115], and Mahdavi, Zhang, and Jin [145]. The VR technique relies on averaging schemes to reduce the variance of stochastic gradients, with several different flavours having emerged in the past decade: SAG [199], SVRG [115], SAGA [69], SVRRG++ [3], SARAH [173] and SPIDER [80]. Such methods outperform the classical SGD under the finite sum model, a fact which led to their widespread use in large-scale applications and their further inclusion into other stochastic optimization algorithms (see for example Xiao and Zhang [227] and Hazan and Luo [102]).

Relevant to our setting, VR has been studied in the context of CGMs for convex minimization by Mokhtari, Hassani, and Karbasi [155], Hazan and Luo [102], Locatello et al. [143], Yurtsever, Sra, and Cevher [233], and Zhang et al. [237]. The SFO complexity of these methods for reaching ϵ suboptimality varies depending on the gradient estimator (specific to each VR scheme), with the best guarantee being of order $\mathcal{O}\left(\epsilon^{-2}\right)$ [238, 233]. For a thorough comparison of the complexities, we refer the reader to Section 6 of Yurtsever, Sra, and Cevher [233].

2.3 Preliminaries

Probability notation. For the probabilistic setting, we denote by ξ an element of our sample space and by $P(\xi)$ its probability measure. Unless stated otherwise, expectations will be taken with respect to ξ. Further, following the setup of Fercoq et al. [81], the space of random variables is defined as

$$\mathcal{H} = \left\{ \boldsymbol{y}(\xi)_\xi \in \mathbb{R}^m \mid \xi \in \mathbb{R}^n, \mathbb{E}\left[\left\| \boldsymbol{y}(\xi)_\xi \right\|^2 \right] < +\infty \right\},$$

where the associated scalar product is given by $\langle \boldsymbol{x}, \boldsymbol{z} \rangle \coloneqq \mathbb{E}\left[\boldsymbol{x}(\xi)^\top \boldsymbol{z}(\xi)\right] = \int \boldsymbol{x}(\xi)^\top \boldsymbol{z}(\xi) dP(\xi)$.

Smoothing. Nesterov [167] proposes a technique for obtaining smooth approximations parametrized by β, of a non-smooth and convex function g. The resulting smoothed approximations take the form

$$g_\beta(\boldsymbol{x}) \coloneqq \max_{\boldsymbol{y}} \langle \boldsymbol{y}, \boldsymbol{x} \rangle - g^*(\boldsymbol{y}) - \frac{\beta}{2} \left\| \boldsymbol{y} \right\|^2,$$

where $g^*(\boldsymbol{y}) \coloneqq \sup_{\boldsymbol{z}} \langle \boldsymbol{z}, \boldsymbol{y} \rangle - g(\boldsymbol{z})$ is the Fenchel conjugate of g. Note that g_β is convex and $\frac{1}{\beta}$-smooth. Whenever g has an efficient prox operator, we can compute the gradient of g_β as

$$\nabla g_\beta(\boldsymbol{x}) = \beta^{-1} \left(\boldsymbol{x} - \operatorname{prox}_{\beta g}(\boldsymbol{x}) \right).$$

We are interested in the case of $g(\cdot, \xi) \equiv \iota_{\boldsymbol{b}(\xi)}(\cdot)$. Smoothing the indicator function is studied in the context of proximal methods by Tran-Dinh, Fercoq, and Cevher [212] and Fercoq et al. [81] and for deterministic CGM by Yurtsever et al. [232]. Of particular note is that when $g(\boldsymbol{x}) = \iota_{\mathcal{X}}(\boldsymbol{x})$, the smoothed function becomes $g_\beta(\boldsymbol{x}) = \frac{1}{2\beta} \operatorname{dist}(\boldsymbol{x}, \mathcal{X})^2$.

Optimality conditions. We denote by $\boldsymbol{x}^\star$ a solution to problem (2.1) and say that $\boldsymbol{x}$ is an ϵ-solution for (2.1) if it satisfies

$$\mathbb{E}\left[|f(\boldsymbol{x}, \xi) - f^\star| \right] \leq \epsilon, \quad \sqrt{\mathbb{E}\left[\operatorname{dist}(\boldsymbol{A}(\xi)\boldsymbol{x}, \boldsymbol{b}(\xi))^2 \right]} \leq \epsilon. \tag{2.2}$$

Oracles. Our complexity results are given relative to the following oracles:

- **Stochastic First Order Oracle (SFO):** For a stochastic function $\mathbb{E}\left[f(\cdot, \xi) \right]$ with $\xi \sim P$, the

SFO returns a pair $(f(\boldsymbol{x},\xi), \nabla f(\boldsymbol{x},\xi))$ where ξ is an i.i.d. sample from P [162].

- **Incremental First Order Oracle (IFO):** For finite-sum problems, the IFO takes an index $i \in [n]$ and returns a pair $(f_i(\boldsymbol{x}), \nabla f_i(\boldsymbol{x}))$.
- **Linear Minimization Oracle (LMO):** The linear minimization oracle of set $\mathcal{X}$ is given by $\mathrm{lmo}_{\mathcal{X}}(\boldsymbol{y}) \in \operatorname{argmin}_{\boldsymbol{x}\in\mathcal{X}} \langle \boldsymbol{x}, \boldsymbol{y}\rangle$ and is assumed efficiently computable throughout this chapter.

2.4 Algorithms and convergence

We now describe our proposed methods for solving (2.1), H-1SFW and H-SPIDER-FW, and provide their theoretical convergence guarantees.

2.4.1 Challenges and high-level ideas

Problem (2.1) can be rewritten equivalently as:

$$\min_{\boldsymbol{x}\in\mathcal{X}} F(\boldsymbol{x}) \coloneqq \mathbb{E}\left[f(\boldsymbol{x},\xi) + \iota_{\boldsymbol{b}(\xi)}(\boldsymbol{A}(\xi)\boldsymbol{x})\right]. \tag{2.3}$$

Note that objective (2.3) is non-smooth due to the indicator function, which makes the CGM framework not applicable (see counterxample by Nesterov [165]). In order to leverage the conditional gradient framework, we *smooth* $\iota_{\boldsymbol{b}(\xi)}(\boldsymbol{A}(\xi)\boldsymbol{x})$ through the technique described in Section 2.3, thus obtaining a surrogate objective F_β. For notational simplicity, we refer to the smoothed stochastic indicator as:

$$g_\beta(\boldsymbol{A}(\xi)\boldsymbol{x}) \coloneqq \frac{1}{2\beta}\operatorname{dist}(\boldsymbol{A}(\xi)\boldsymbol{x}, \boldsymbol{b}(\xi))^2. \tag{2.4}$$

The minimization problem in terms of the smoothed objective thus becomes:

$$\min_{\boldsymbol{x}\in\mathcal{X}} F_\beta(\boldsymbol{x}) \coloneqq \mathbb{E}\left[f(\boldsymbol{x},\xi) + g_\beta(\boldsymbol{A}(\xi)\boldsymbol{x})\right], \tag{2.5}$$

with $\lim_{\beta\to 0} F_\beta(\boldsymbol{x}) = F(\boldsymbol{x})$. A natural idea is to optimize smooth approximations F_β which are progressively more accurate representations of F. To this end, we apply *Conditional Gradient* steps in conjunction with decreasing the smoothness parameter β, practically emulating a homotopy transformation. As the iterations unfold, our algorithms, in fact, approach the optimum of the original objective $F(\boldsymbol{x})$, as stated theoretically in Sections 2.4.3 and 2.4.4.

However, the aforementioned idea faces a technical challenge: decreasing the smoothing parameter β impacts the variance of the stochastic gradients $\nabla_{\boldsymbol{x}} g_\beta(\boldsymbol{A}(\xi)\boldsymbol{x})$, which increases proportionally. This issue has previously been signalled in the work of Fercoq et al. [81], where the authors address a similar setting using stochastic proximal gradient steps. Here, the problem is further

aggravated by the use of LMO calls over $\mathcal{X}$, as it is well-known that CGMs are sensitive to non-vanishing gradient noise [155].

Our solution is to simply perform VR on the stochastic gradients and theoretically establish a rate for $\beta \to 0$ in order to counteract the exploding variance. Precisely, we show how two different VR schemes can be successfully used within the homotopy framework:

- H-1SFW uses one stochastic sample to update a gradient estimator at every iteration, following the technique introduced by Mokhtari, Hassani, and Karbasi [155]. Depending on computational resources, the single-sample model can be extended to a fixed batch size with the same convergence guarantees.
- H-SPIDER-FW uses stochastic minibatches of increasing size to compute the gradient estimator, using the technique proposed by Fang et al. [80].

The theoretical results characterizing our algorithms are presented in Section 2.4.3 and 2.4.4. First, we state the rate at which the β-dependent gradient noise vanishes under each VR scheme in Lemma 2.1 and 2.2. The main convergence results, Theorem 2.1 and 2.2, describe the performance of our algorithms in terms of the quantity $\mathbb{E}\left[S_{\beta_k}(\boldsymbol{x}_k, \xi)\right] \coloneqq \mathbb{E}\left[F_{\beta_k}(\boldsymbol{x}_k, \xi) - f^\star\right]$, called the *smoothed gap*. Finally, in Corollary 2.1 and 2.2, we translate the aforementioned results into guarantees over the objective residual and constraint feasibility. All proofs are deferred to Appendix A.

2.4.2 Assumptions

Assumption 2.1. *The stochastic functions $f(\cdot, \xi)$ are convex and L_f-smooth. This further implies that $f(\boldsymbol{x})$ is L_f-smooth.*

Assumption 2.2. *The stochastic gradients $\nabla f(\boldsymbol{x}, \xi)$ are unbiased and have a uniform variance bound σ_f^2. Formally,*

$$\mathbb{E}\left[\nabla f(\boldsymbol{x}, \xi)\right] = \nabla f(\boldsymbol{x}) \qquad \text{and} \qquad \mathbb{E}\left[\left\| \nabla f(\boldsymbol{x}, \xi) - \nabla f(\boldsymbol{x}) \right\|^2\right] \leq \sigma_f^2 < +\infty. \tag{2.6}$$

Assumption 2.3. *The domain $\mathcal{X}$ is convex and compact, with diameter $\mathcal{D}_{\mathcal{X}}$.*

Assumption 2.4. *Slater's condition holds for problem* (2.3)*. Specifically, letting $G : \mathcal{H} \to \mathbb{R} \cup \{\infty\}$, $G(\boldsymbol{A}\boldsymbol{x}) \coloneqq \mathbb{E}\left[\iota_{\boldsymbol{b}(\xi)}(\boldsymbol{A}(\xi)\boldsymbol{x})\right]$, with the linear operator $\boldsymbol{A} : \mathbb{R}^d \to \mathcal{H}$ defined as $(\boldsymbol{A}\boldsymbol{x})(\xi) \coloneqq \boldsymbol{A}(\xi)\boldsymbol{x}$, $\forall \boldsymbol{x}$, we require that*

$$0 \in \operatorname{sri}\left(\operatorname{dom}(G) - \boldsymbol{A}\operatorname{dom}(f)\right),$$

where sri *is the strong relative interior of the set [12].*

Assumption 2.5. *The spectral norm of the stochastic linear operator $\boldsymbol{A}(\xi)$ is uniformly bounded by a constant $L_{\boldsymbol{A}}$:*

$$L_{\boldsymbol{A}} \coloneqq \sup_{\xi} \| \boldsymbol{A}(\xi) \|^2 < +\infty.$$

We note that Assumption 2.5 is also made by Fercoq et al. [81].

2.4.3 H(omotopy)-1SFW

We now describe our first algorithm, which uses the VR scheme proposed by Mokhtari, Hassani, and Karbasi [155] and whose advantage lies in a simple update rule and single-loop structure. All the proofs for this section are deferred to Appendix A.1.

Gradient estimator model

We denote the gradient estimator by $\boldsymbol{d}_k$. Note that $\boldsymbol{d}_k$ is biased with respect to the true gradient $\nabla F_\beta(\boldsymbol{x}_k)$ and exhibits a vanishing variance. This scheme achieves VR while conveniently considering only one stochastic constraint at a time. The estimator update rule is given by

$$\boldsymbol{d}_k = (1-\rho_k)\boldsymbol{d}_{k-1} + \rho_k \nabla F_{\beta_k}(\boldsymbol{x}_k, \xi_k),$$

where $\nabla F_{\beta_k}(\boldsymbol{x}_k, \xi_k) = \nabla f(\boldsymbol{x}_k, \xi_k) + \nabla g_{\beta_k}(\boldsymbol{A}(\xi_k)\boldsymbol{x}_k)$, and ρ_k is a decaying convex combination parameter. The proposed method is summarized in Algorithm 2.1.

Algorithm 2.1 H-1SFW

Input: $\boldsymbol{x}_1 \in \mathcal{X}, \beta_0 > 0, P(\xi)$

for $k = 1, 2, \ldots,$ **do**

Set ρ_k, β_k and γ_k; sample $\xi_k \sim P(\xi)$

$\boldsymbol{d}_k = (1-\rho_k)\boldsymbol{d}_{k-1} + \rho_k \nabla_{\boldsymbol{x}} F_{\beta_k}(\boldsymbol{x}_k, \xi_k)$

$\boldsymbol{w}_k \in \operatorname{lmo}_{\mathcal{X}}(\boldsymbol{d}_k)$

$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \gamma_k(\boldsymbol{w}_k - \boldsymbol{x}_k).$

end for

Convergence results

Before stating the results, we remark that Lemma 2.1 is the counterpart of Lemma 1 in [155] and its proof follows a similar route, up to bounding β-dependent quantities. It is worth noting that in our case, handling the stochastic linear inclusion constraints results in a rate surcharge factor of $\mathcal{O}\left(k^{1/3}\right)$.

Lemma 2.1. *Let* $\rho_k = \frac{3}{(k+5)^{2/3}}$, $\gamma_k = \frac{2}{k+1}$, $\beta_k = \frac{\beta_0}{(k+1)^{1/6}}$, $\beta_0 > 0$ *in Algorithm 2.1. Then, for all k,*

$$\mathbb{E}\left[\|\nabla F_{\beta_k}(\boldsymbol{x}_k) - \boldsymbol{d}_k\|^2\right] \leq \frac{C_1}{(k+5)^{1/3}},$$

where $C_1 \coloneqq \max\left\{6^{1/3}\|\nabla F_{\beta_0}(\boldsymbol{x}_0) - \boldsymbol{d}_0\|^2, 2\left[18\sigma_f^2 + 112L_f^2\mathcal{D}_{\mathcal{X}}^2 + \frac{522L_{\boldsymbol{A}}^2\mathcal{D}_{\mathcal{X}}^2}{\beta_0^2}\right]\right\}$.

Theorem 2.1. *Consider Algorithm 2.1 with parameters* $\rho_k = \frac{3}{(k+5)^{2/3}}$, $\gamma_k = \frac{2}{k+1}$, $\beta_k = \frac{\beta_0}{(k+1)^{1/6}}$, $\beta_0 > 0$ *(identical to Lemma 2.1). Then, for all* k*,*

$$\mathbb{E}\left[S_{\beta_k}(\boldsymbol{x}_{k+1})\right] \leq \frac{C_2}{k^{1/6}},$$

where $C_2 \coloneqq \max\left\{S_0(\boldsymbol{x}_1),\ b = 2\mathcal{D}_{\mathcal{X}}\sqrt{C_1} + 2\mathcal{D}_{\mathcal{X}}^2\left(L_f + \frac{L_{\boldsymbol{A}}}{\beta_0}\right)\right\}$ *and* C_1 *is defined in Lemma 2.1.*

Corollary 2.1. *The expected convergence in terms of objective suboptimality and feasibility of Algorithm 2.1 is, respectively,*

$$\left|\mathbb{E}\left[f(\boldsymbol{x}_k, \xi)\right] - f^\star\right| \qquad \in \mathcal{O}\left(k^{-1/6}\right)$$

$$\sqrt{\mathbb{E}\left[\operatorname{dist}(\boldsymbol{A}(\xi)\boldsymbol{x}_k, \boldsymbol{b}(\xi))^2\right]} \ \in \mathcal{O}\left(k^{-1/6}\right).$$

Consequently, the oracle complexity is $\#(SFO) \in \mathcal{O}\left(\epsilon^{-6}\right)$ *and* $\#(LMO) \in \mathcal{O}\left(\epsilon^{-6}\right)$.

2.4.4 H(omotopy)-SPIDER-FW

Our second algorithm presents a more complex VR scheme, which improves upon the complexity of H-1SFW. The method relies on the SPIDER estimator originally proposed under the framework of Normalized Gradient Descent by Fang et al. [80] and further studied for CGMs by Yurtsever, Sra, and Cevher [233]. Different from Section 2.4.3, the results that follow distinguish two scenarios: the first is customary to VR methods such as SVRG [115] or SARAH [173] and assumes a finite-sum form of f; the second, different from most other VR schemes, caters to objectives of the form $f(\boldsymbol{x}) = \mathbb{E}\left[f(\boldsymbol{x}, \xi)\right]$ where $\xi \sim P(\xi)$, and can handle a potentially infinite number of stochastic functions. All the proofs for this section are deferred to Appendix A.2.

Gradient estimator model

We denote the SPIDER gradient estimator by $\boldsymbol{v}_{t,k}$. We note that $\boldsymbol{v}_{t,k}$ is also biased relative to $\nabla F_{\beta_k}(\boldsymbol{x}_k)$ and exhibits a vanishing variance. This scheme achieves VR through the use of increasing-size minibatches. The estimator update rule is given by

$$\boldsymbol{v}_{t,k} = \boldsymbol{v}_{t,k-1} - \widetilde{\nabla} F_{\beta_{t,k-1}}(\boldsymbol{x}_{t,k-1}, \xi_{\mathcal{S}_{t,k}}) + \widetilde{\nabla} F_{\beta_{t,k}}(\boldsymbol{x}_{t,k}, \xi_{\mathcal{S}_{t,k}}), \tag{2.7}$$

where $\widetilde{\nabla} F_{\beta_{t,k}}(\boldsymbol{x}_{t,k}, \xi_{\mathcal{S}_{t,k}}) = \widetilde{\nabla} f(\boldsymbol{x}_k, \xi_{\mathcal{S}_{t,k}}) + \widetilde{\nabla} g_{\beta_{t,k}}(\boldsymbol{A}(\xi_{\mathcal{S}_{t,k}})\boldsymbol{x}_{t,k})$ defines the averaged gradient over a minibatch of size $|\mathcal{S}_{t,k}|$.

The double indexing used in (2.7) hints at the double-loop structure of the algorithm. The method is structured similarly to SPIDER-FW from [233], and proceeds in two steps: the outer loop computes an "accurate" gradient estimator and sets the batch size for the inner iterations. The inner loop then iteratively "refreshes" this gradient according to (2.7) and performs homotopy steps on β using a theoretically-determined schedule. The proposed method is summarized in Algorithm 2.2.

Algorithm 2.2 H-SPIDER-FW

Input: $\bar{\boldsymbol{x}}_1 \in \mathcal{X}, \beta_0 > 0, P(\xi)$

for $t = 1, 2, \ldots, T$ **do**

$\boldsymbol{x}_{t,1} = \bar{\boldsymbol{x}}_t$

Compute $\gamma_{t,1}, \beta_{t,1}, K_t$; sample $\xi_{\mathcal{Q}_t} \overset{\text{i.i.d}}{\sim} P(\xi)$

$\boldsymbol{v}_{t,1} = \widetilde{\nabla} F_{\beta_{t,1}}(\boldsymbol{x}_{t,1}, \xi_{\mathcal{Q}_t})$

$\boldsymbol{w}_{t,1} \in \operatorname{lmo}_{\mathcal{X}}(\boldsymbol{v}_{t,1})$

$\boldsymbol{x}_{t,2} = \boldsymbol{x}_{t,1} + \gamma_{t,1}(\boldsymbol{w}_{t,1} - \boldsymbol{x}_{t,1})$

for $k = 2, \ldots, K_t$ **do**

Compute $\gamma_{t,k}, \beta_{t,k}$; sample $\xi_{\mathcal{S}_{t,k}} \overset{\text{i.i.d}}{\sim} P(\xi)$

$\boldsymbol{v}_{t,k} = \boldsymbol{v}_{t,k-1} - \widetilde{\nabla} F_{\beta_{t,k-1}}(\boldsymbol{x}_{t,k-1}, \xi_{\mathcal{S}_{t,k}}) + \widetilde{\nabla} F_{\beta_{t,k}}(\boldsymbol{x}_{t,k}, \xi_{\mathcal{S}_{t,k}})$

$\boldsymbol{w}_{t,k} \in \operatorname{lmo}_{\mathcal{X}}(\boldsymbol{v}_{t,k})$

$\boldsymbol{x}_{t,k+1} = \boldsymbol{x}_{t,k} + \gamma_{t,k}(\boldsymbol{w}_{t,k} - \boldsymbol{x}_{t,k})$

end for

Set $\bar{\boldsymbol{x}}_{t+1} = \boldsymbol{x}_{t,K_t+1}$

end for

Convergence results

Again, we remark that Lemma 2.2 is the counterpart of Lemma 4, Appendix C in [233]. However, in this case, our proof takes a different, more tedious route, as the latter result does not accommodate homotopy steps. In comparison, the bound we obtain depends linearly on the total iteration count, whereas the aforementioned lemma depends only on the outer loop counter K_t.

Lemma 2.2 (Estimator variance for finite-sum problems)**.** *Consider Algorithm 2.2, and let ξ be finitely sampled from set $[n]$, $\xi_{\mathcal{Q}_t} = [n]$ and $\xi_{\mathcal{S}_{t,k}}$, such that $|\mathcal{S}_{t,k}| = K_t = 2^{t-1}$. Also, let $\gamma_{t,k} = \frac{2}{K_t+k}$, $\beta_{t,k} = \frac{\beta_0}{\sqrt{K_t+k}}$, $\beta_0 > 0$. Then, for a fixed t and for all $k \leq K_t$,*

$$\mathbb{E}\left[\left\| \nabla F_{\beta_{t,k}}(\boldsymbol{x}_{t,k}) - \boldsymbol{v}_{t,k} \right\|^2\right] \leq \frac{C_1}{K_t + k},$$

where $C_1 = 2\mathcal{D}_{\mathcal{X}}^2 \left(8L_f^2 + \frac{98L_A^2}{\beta_0^2}\right)$.

Lemma 2.3 (Estimator variance for general expectation problems)**.** *Consider Algorithm 2.2 and let* $\xi \sim P(\xi)$ *and* $\xi_{\mathcal{Q}_t}$ *such that* $|\mathcal{Q}_t| = \left\lceil \frac{2K_t}{\beta_{t,1}^2} \right\rceil$. *Also, let* $\xi_{\mathcal{S}_{t,k}}$, *such that* $|\mathcal{S}_{t,k}| = K_t = 2^{t-1}$, $\gamma_{t,k} = \frac{2}{K_t+k}$, $\beta_{t,k} = \frac{\beta_0}{\sqrt{K_t+k}}$, $\beta_0 > 0$. *Then, for a fixed* t *and for all* $k \leq K_t$,

$$\mathbb{E}\left[\left\| \nabla F_{\beta_{t,k}}(\boldsymbol{x}_{t,k}) - \boldsymbol{v}_{t,k} \right\|^2 \right] \leq \frac{C_2}{K_t + k},$$

where $C_2 = 16L_f^2\mathcal{D}_{\mathcal{X}}^2 + 2L_{\boldsymbol{A}}^2\mathcal{D}_{\mathcal{X}}^2\left(\frac{98}{\beta_0^2}+1\right) + 2\beta_0^2\sigma_f^2$.

Theorem 2.2. *Consider Algorithm 2.2 with parameters* $\gamma_{t,k} = \frac{2}{K_t+k}$, $\beta_{t,k} = \frac{\beta 0}{\sqrt{K_t+k}}$, $\beta_0 > 0$, *and* $\xi_{\mathcal{S}_{t,k}}$, *such that* $|\mathcal{S}_{t,k}| = K_t = 2^{t-1}$. *Then,*

- *For* ξ *be finitely sampled from set* $[n]$, $\xi_{\mathcal{Q}_t} = [n]$ *and* $\forall\, t \in \mathbb{N}$, $1 \leq k \leq 2^{t-1}$,

$$\mathbb{E}\left[S_{\beta_{t,k}}(\boldsymbol{x}_{t,k+1}) \right] \leq \frac{C_3}{\sqrt{K_t+k+1}},$$

where $C_3 = \max\left\{ S_{\beta_{1,0}}(\boldsymbol{x}_{1,1}), 2\mathcal{D}_{\mathcal{X}}^2 L_f + 2\mathcal{D}_{\mathcal{X}}^2\sqrt{16L_f^2 + \frac{196L_{\boldsymbol{A}}^2}{\beta_0^2}} + \frac{2\mathcal{D}_{\mathcal{X}}^2 L_{\boldsymbol{A}}}{\beta_0} \right\}$;

- *For* $\xi \sim P(\xi)$, $\xi_{\mathcal{Q}_t}$ *such that* $|\mathcal{Q}_t| = \left\lceil \frac{2K_t}{\beta_{t,1}^2} \right\rceil$ *and* $\forall\, t \in \mathbb{N}$, $1 \leq k \leq 2^{t-1}$,

$$\mathbb{E}\left[S_{\beta_{t,k}}(\boldsymbol{x}_{t,k+1}) \right] \leq \frac{C_4}{\sqrt{K_t+k+1}},$$

where $C_4 = \max\left\{ S_{\beta_{1,0}}(\boldsymbol{x}_{1,1}),\ 2\mathcal{D}_{\mathcal{X}}^2 L_f + \frac{2\mathcal{D}_{\mathcal{X}}^2 L_{\boldsymbol{A}}}{\beta_0} + 2\mathcal{D}_{\mathcal{X}}\sqrt{16L_f^2\mathcal{D}_{\mathcal{X}}^2 + 2L_{\boldsymbol{A}}^2\mathcal{D}_{\mathcal{X}}^2\left(\frac{98}{\beta_0^2}+1\right) + 2\beta_0^2\sigma_f^2} \right\}$.

Corollary 2.2. *The expected convergence in terms of objective suboptimality and feasibility of Algorithm 2.2 is, respectively,*

$$\left| \mathbb{E}\left[f(\boldsymbol{x}_{t,k}) \right] - f^\star \right| \qquad \in \mathcal{O}\left((K_t+k)^{-1/2} \right)$$

$$\sqrt{\mathbb{E}\left[\operatorname{dist}(\boldsymbol{A}(\xi)\boldsymbol{x}_{t,k}, \boldsymbol{b}(\xi))^2 \right]} \qquad \in \mathcal{O}\left((K_t+k)^{-1/2} \right)$$

for both the finite sum and the general expectation setting. Consequently, the oracle complexities are given by $\#(IFO) \in \mathcal{O}\left(n\log_2(\epsilon^{-2}) + \epsilon^{-4} \right)$ *and* $\#(LMO) \in \mathcal{O}\left(\epsilon^{-2}\right)$ *for the finite-sum setting, and by* $\#(SFO) \in \mathcal{O}\left(\epsilon^{-4}\right)$ *and* $\#(LMO) \in \mathcal{O}\left(\epsilon^{-2}\right)$ *for the expectation setting.*

2.4.5 Discussion

Rate degradation in the absence of projection oracles. Compared to proximal methods for solving (2.1), our algorithms require $\mathcal{O}\left(\epsilon^{-2}\right)$ times more SFO calls to reach an ϵ-solution. This is well-known for CG-based methods: for instance, solving a fully deterministic version of (2.1) with the Augmented Lagrangian framework has a gradient complexity of $\mathcal{O}\left(\epsilon^{-1}\right)$ [228], whereas the best known complexity for CG-based algorithms is $\mathcal{O}\left(\epsilon^{-2}\right)$ [232].

Comparison with SHCGM [143]. The state-of-the-art for solving (2.1) is the half-stochastic method SHCGM [143], in which stochasticity is restricted to the objective function f, while the constraints are processed deterministically. This algorithm attains an $\mathcal{O}(\epsilon^{-3})$ SFO complexity and an $\mathcal{O}(\epsilon^{-3})$ LMO complexity, by resorting to the same VR scheme as H-1SFW applied only to $f(\boldsymbol{x}, \xi)$. Since SHCGM handles the constraints deterministically, it does not face the challenge of exploding variance as $\beta \to 0$.

Our analysis shows that handling the β-dependence of the gradient noise comes at the price of H-1SFW being $\mathcal{O}\left(\epsilon^{-3}\right)$ times more expensive in terms of both oracles. In contrast, owing to a more powerful variance-reduction scheme, H-SPIDER-FW attains only an $\mathcal{O}(\epsilon)$-times worse SFO complexity, while improving by an $\mathcal{O}(\epsilon)$ factor in terms of the LMO complexity. Given that an LMO call is generally more expensive than that of an SFO, we have, in fact, *improved* the complexity over the state-of-the-art while being the first to process linear constraints stochastically. Moreover, we note that the LMO complexity of H-SPIDER-FW is of the same order as its fully deterministic counterpart, the HCGM [232].

The role of VR. The choice of VR technique dictates the worst-case convergence guarantees of our methods, a fact which is apparent from the discrepancy between the variance bounds of Lemmas 2.1 and 2.2- 2.3, respectively: $\mathcal{O}\left(k^{-1/3}\right)$ for $\boldsymbol{d}_k$ vs. $\mathcal{O}\left(k^{-1}\right)$ for $\boldsymbol{v}_{t,k}$. This signals the existence of a trade-off: a more intricate way of handling stochastic penalty-type constraints can ensure the better convergence guarantees of H-SPIDER-FW, while a simpler VR scheme comes at the cost of the rather pessimistic ones of H-1SFW. Fortunately, as shown in Section 2.5, the simple H-1SFW greatly outperforms its worst-case bounds.

2.5 Experiments

To demonstrate the empirical efficiency of our algorithms, we apply them to three problem instances: synthetically generated SDPs, the k-Means clustering SDP relaxation and the Sparsest Cut-associated SDP.

Experiment setup. The experiments presented in this chapter were implemented in MATLAB R2019b and executed on a 2,9 GHz 6-Core Intel Core i9 CPU with 32 GB RAM. For retrieving the

values of $f^\star$, we used the code of Mixon, Villar, and Ward [154] which relies on SDPNAL+ [230] for the clustering experiments, and CVX [100] for the Sparsest Cut ones.

Evaluation metrics. Our experiments subscribe to a finite-sum template, where we define $f(\boldsymbol{x}) \coloneqq \sum_{i=1}^{n_1} f_i(\boldsymbol{x})$ and $g_\beta(\boldsymbol{A}\boldsymbol{x}) = \sum_{i=1}^{n_2} g_{i,\beta}(\boldsymbol{A}_i^\top \boldsymbol{x})$. The objective convergence is recorded as $|f(\boldsymbol{x}) - f^\star|$. Due to imperfect feasibility, the value of $f(\boldsymbol{x})$ can overshoot $f^\star$, since the constrained optimum is not the global one. This usually appears as the increase of $|f(\boldsymbol{x}) - f^\star|$ immediately after a significant drop when the quantity $f(\boldsymbol{x}) - f^\star$ becomes negative; then the decreasing trend restarts, as the objective and constraints re-balance. Such a phenomenon is common for homotopy-based methods; see, for instance, the experiments of Yurtsever et al. [232]. Lastly, the feasibility is recorded as $\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|$.

Baseline. To the best of our knowledge, the HCGM [232] and the SHCGM [143] are the only algorithms which tackle SDPs under the conditional gradient framework. The latter represents the empirical state-of-the-art and we choose it as the baseline for our experiments.

2.5.1 Synthetic SDP problems

This proof-of-concept experiment showcases the performance of our fully stochastic methods for a fixed problem dimension and an increasing set of constraints. We consider the planted synthetic SDP

$$\begin{aligned} \min_{\substack{\boldsymbol{X}\in\mathbb{S}_+^d \\ \mathrm{Tr}(\boldsymbol{X})\le\frac{1}{d}}} \quad & \langle \boldsymbol{C}, \boldsymbol{X}\rangle \\ \text{subject to} \quad & Tr(\boldsymbol{A}_i\boldsymbol{X}) = b_i,\ \forall i \in [n], \end{aligned} \tag{2.8}$$

where the entries of $\boldsymbol{A}_i$ and $\boldsymbol{C}$ are generated from $\mathcal{U}(0,1)$, and $b_i = \langle \boldsymbol{A}_i, \boldsymbol{X}^\star\rangle$ for a fixed $\boldsymbol{X}^\star$. We perform uniform sampling on the pairs $(\boldsymbol{A}_i, b_i)$ for computing their stochastic gradients in our algorithms. We fix the dimension to be $d = 20$ and vary the size of constraints with $n \in \{5\text{e}2, 5\text{e}3\}$.

For a fair comparison, we sweep the parameter β_0 for the three algorithms in the range $[1\text{e}{-7}, 1\text{e}1]$. We settle for $1\text{e}{-7}$, $1\text{e}{-7}$ and $1\text{e}{-5}$ for SHCGM, H-1SFW and H-SPIDER-FW, respectively. For H-1SFW and SHCGM, we choose the batch size to be 1% of the data.

Figure 2.1 illustrates the outcome of the experiments, where we observe a clear improvement of the stochastic algorithms over the baseline with a stable margin throughout the test cases.

Interestingly, H-1SFW exhibits strong empirical performance on the synthetic data, much better than its theoretical worst-case bound. A possible explanation is that the entries of $\boldsymbol{C}$ and $\boldsymbol{A}_i$ are generated from a "benign" distribution and *concentrate* around its mean [138]. In such scenarios,

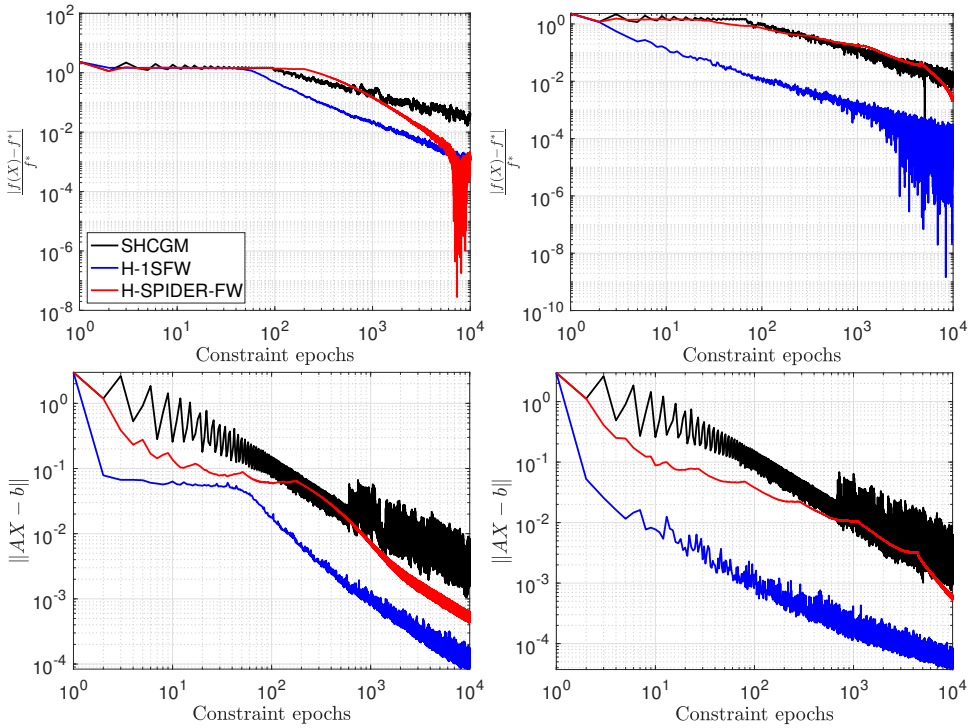


Figure 2.1: Synthetic SDPs, with each column showing the convergence in objective suboptimality (top) and in feasibility (bottom) for a given problem instance. The left column corresponds to a problem with 5e2 constraints, while the right one to a problem with 5e3 constraints.

even a small subset of constraints allows for effective variance reduction. Nevertheless, we observe the same good performance of H-1SFW, even with real data, in the next sections.

Regarding H-SPIDER-FW, we observe that the suboptimality and feasibility decrease at the rate $k^{-\frac{1}{2}}$ and $k^{-\frac{3}{4}}$, respectively, which is better than the worst-case bounds of Theorem 2.2.

We further compare the algorithms on problem (2.8) under a less well-behaved distribution of entries in matrices $\boldsymbol{A}_i$ and $\boldsymbol{C}$. Specifically, we use the heavy-tailed Stable distribution with parameters $(\alpha = 1.5, \beta = 0, \gamma = 10, \delta = 0)$. We sweep β_0 for all three algorithms in the range $[1\text{e}{-7}, 1\text{e}{-1}]$ and settle for $1\text{e}{-5}$, $1\text{e}{-7}$, $1\text{e}{-6}$ for SHCGM, H-1SFW and H-SPIDER-FW, respectively. The results are depicted in Figure 2.2.

Given this more difficult distribution, we observe that all methods are comparable in terms of convergence speed in both objective suboptimality and feasibility, with H-SPIDER-FW having an edge over the other two.

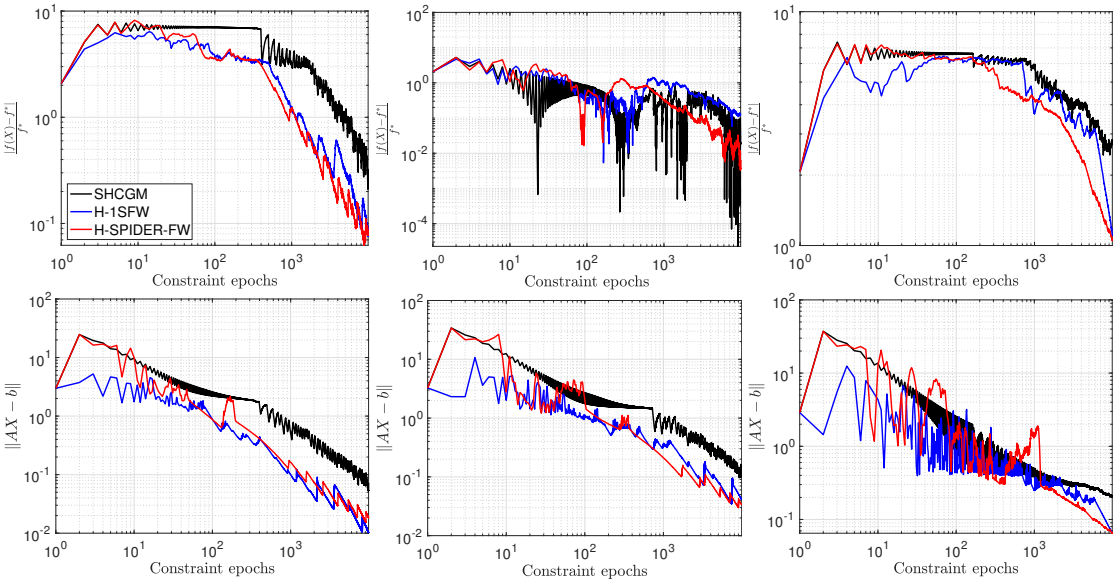


Figure 2.2: Synthetic SDPs, with each column showing the convergence in objective suboptimality (top) and in feasibility (bottom) for a given problem instance. From left to right, the columns depict the results for problems with 5e2, 1e3 and 5e3 constraints.

2.5.2 The k-Means clustering relaxation

We consider the unsupervised learning task of partitioning d data points into k clusters. We adopt the SDP formulation in [182], which amounts to solving:

$$\begin{aligned} \min_{\boldsymbol{X}\in\mathcal{X}} \quad & \langle \boldsymbol{C}, \boldsymbol{X} \rangle \\ \text{subject to} \quad & \boldsymbol{X}\mathbf{1}_d = \mathbf{1}_d, \\ & \boldsymbol{X}_{i,j} \geq 0, \ 1 \leq i, j \leq d. \end{aligned} \tag{2.9}$$

Here, $\boldsymbol{C} \in \mathbb{R}^{d\times d}$ is the Euclidean distance matrix of the d data points and $\mathcal{X} \coloneqq \{\boldsymbol{X} \in \mathbb{R}^{d\times d} \,|\, \boldsymbol{X} \succeq 0, \ \mathrm{Tr}(\boldsymbol{X}) \leq k\}$. Notice that the number of linear constraints in (2.9) is $\mathcal{O}(d^2)$.

In order to compare against existing work, we adopt the MNIST dataset ($k = 10$) [137] with $d = 10^3$ samples and perform data preprocessing as Mixon, Villar, and Ward [154]. The same setup appeared in prior works [232, 143], with SHCGM [143] showing the best practical performance.

We perform parameter sweeping on $\beta_0 \in [1\text{e}{-7}, 1\text{e}2]$ for H-1SFW and H-SPIDER-FW, and settle for 5e−2 and 6e0, respectively. For SHCGM, we adopt the same hyperparameter as in [143]. The batch size for H-1SFW and SHCGM is set to 5%.

The comparison of our algorithms against SHCGM is reported in Figure 2.3. H-1SFW and H-SPIDER-FW converge at a comparable rate, with both clearly overtaking the baseline with regard to both objective suboptimality and feasibility.

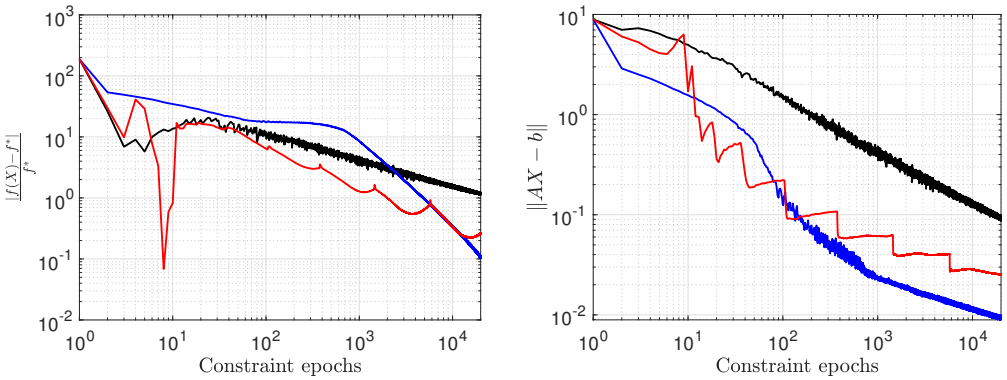


Figure 2.3: The k-Means SDP relaxation, with convergence in objective suboptimality (left) and in feasibility (right).

2.5.3 Computing an ℓ_2^2 embedding for the Uniform Sparsest Cut problem

The Uniform Sparsest Cut problem (USC) aims to find a bipartition $(S, \bar{S})$ of the nodes of a graph $G = (V, E)$, $|V| = d$, which minimizes the quantity

$$\frac{E(S, \bar{S})}{|S||\bar{S}|},$$

where $E(S, \bar{S})$ is the number of edges connecting S and $\bar{S}$. This problem is of broad interest, with applications in areas such as VLSI layout design, topological design of communication networks and image segmentation, to name a few. Relevant to machine learning, it appears as a subproblem in hierarchical clustering algorithms [65, 46].

Computing such a bipartition is NP-hard and intense research has gone into designing efficient approximation algorithms for this problem. In the seminal work of Arora, Rao, and Vazirani [7] an $\mathcal{O}\left(\sqrt{\log d}\right)$ approximation algorithm is proposed for solving USC, which relies on finding a *well-spread* ℓ_2^2 geometric representation of G where each node $i \in V$ is mapped to a vector $\boldsymbol{v}_i$ in $\mathbb{R}^d$. In this experimental section, we focus on solving the SDP that computes this geometric embedding, as its high number of triangle inequality constraints ($\mathcal{O}\left(d^3\right)$) makes it a suitable candidate for our framework.

The original formulation of the SDP given by Arora, Rao, and Vazirani [7] is

$$\begin{aligned} \min \quad & \frac{1}{d^2} \sum_{(i,j)\in E} \left\| \boldsymbol{v}_i - \boldsymbol{v}_j \right\|^2 \\ \text{subject to} \quad & \sum_{\substack{i,j\in V \\ i\neq j}} \left\| \boldsymbol{v}_i - \boldsymbol{v}_j \right\|^2 = d^2 \\ & \left\| \boldsymbol{v}_i - \boldsymbol{v}_j \right\|^2 + \left\| \boldsymbol{v}_j - \boldsymbol{v}_k \right\|^2 \geq \left\| \boldsymbol{v}_i - \boldsymbol{v}_k \right\|^2, \ \forall\ i, j, k \in V, \end{aligned}$$

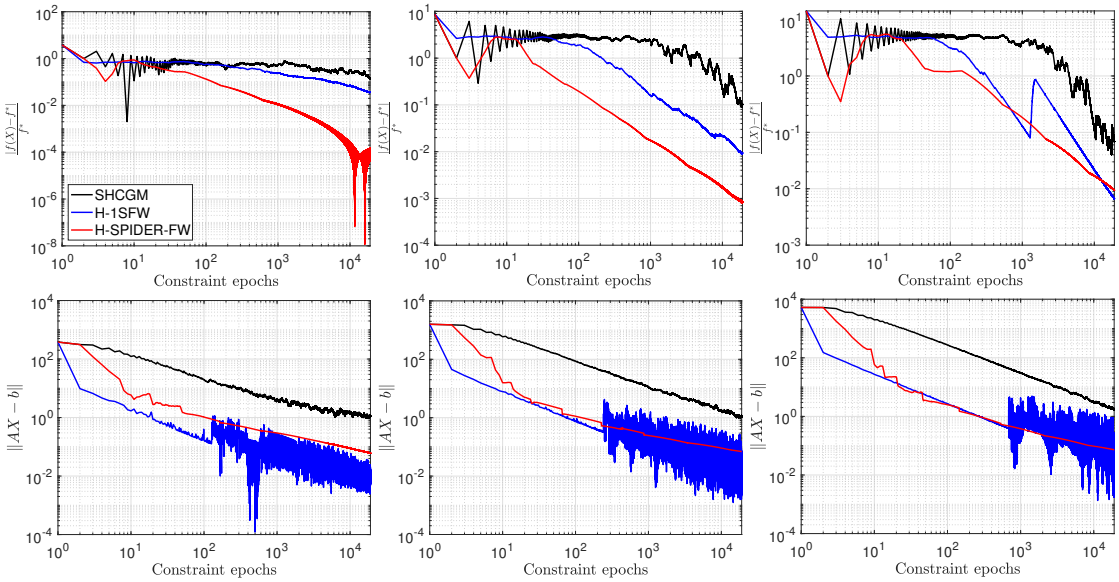


Figure 2.4: The Sparsest Cut-associated SDP relaxation, where each column shows the convergence in objective suboptimality (top) and feasibility (bottom) for a given problem instance. From left to right, the results correspond to graphs `mammalia-primate-association-13`, `insecta-ant-colony1-day37` and `insecta-ant-colony4-day10`, sorted by increasing size.

while its equivalent canonical formulation is given by the expression

$$\begin{aligned} \min_{\boldsymbol{X}\in\mathcal{X}} \quad & \langle \boldsymbol{L}, \boldsymbol{X}\rangle \\ \text{subject to} \quad & d\,\mathrm{Tr}\,(\boldsymbol{X}) - \mathrm{Tr}\,(\mathbf{1}_{d\times d}\boldsymbol{X}) = \frac{d^2}{2} \\ & \boldsymbol{X}_{i,j} + \boldsymbol{X}_{j,k} - \boldsymbol{X}_{i,k} - \boldsymbol{X}_{j,j} \leq 0, \ \ \forall\ i,j,k \in V, \end{aligned} \tag{2.10}$$

where L represents the Laplacian of G, $\mathcal{X} = \{\boldsymbol{X} \in \mathbb{R}^{d\times d} : \boldsymbol{X} \succeq 0,\ \mathrm{Tr}\,(\boldsymbol{X}) \leq d\}$ and $\boldsymbol{X}_{i,j} = \langle \boldsymbol{v}_i, \boldsymbol{v}_j\rangle$ gives the geometric embedding of the nodes. We use form (2.10) with an added trace constraint, $\mathrm{Tr}\,(\boldsymbol{X}) \leq d$. This additional constraint does not change the optimal objective [109].

We run our algorithms on three graphs of different sizes from the Network Repository dataset [196], whose details are summarized in Table 2.1. Note the cubic dependence of the number of constraints relative to the number of nodes. We perform parameter sweeping on $\beta_0 \in [1\mathrm{e}{-5}, 1\mathrm{e}5]$ using the smallest graph, `mammalia-primate-association-13`, and keep the same parameters for all the experiments. The values of β_0 for SHCGM, H-1SFW and H-SPIDER-FW are 1e2, 1e$-$2 and 1e1 respectively, and the batch size for both H-1SFW and SHCGM is set to 5%.

Figure 2.4 depicts the outcomes of the experiments, with both our algorithms consistently

Graph name	$\lvert V\rvert$	$\lvert E\rvert$	Avg. node degree	Max. node degree	USC SDP dimension	USC SDP # constraints
`mammalia-primate-association-13`	25	181	14	19	$X \in \mathbb{R}^{25\times 25}$	~ 6.90e3
`insecta-ant-colony1-day37`	55	1k	42	53	$X \in \mathbb{R}^{55\times 55}$	~ 7.87e4
`insecta-ant-colony4-day10`	102	4k	79	99	$X \in \mathbb{R}^{102\times 102}$	~ 5.15e5

Table 2.1: Details of the Network Repository graphs [196] used in the Sparsest Cut experiments.

outperforming SHCGM, and H-SPIDER-FW attaining the fastest convergence. A possible explanation is that, given the much larger number of constraints relative to the problem dimension, specifically $\mathcal{O}\left(n^3\right)$ v.s $\mathcal{O}\left(n^2\right)$, H-SPIDER-FW's increasing minibatches readily reach an adequate balance between feasibility enforcement and objective minimization.

2.6 Improved guarantees for the finite sum case

Previous sections considered the problem of minimizing a stochastic objective over the constraint set $\mathcal{X}$ subject to a possibly infinite number of stochastic constraints (2.1). This is a very general formulation, encompassing a variety of problem templates and therefore comes with a lot of flexibility. This same generality, however, serves as its drawback. This will become apparent in the coming sections, where improved convergence is achieved for the closely related but less general finite sum counterpart of (2.1). Specifically, we consider the following finite-sum template

$$\min_{\boldsymbol{x}\in\mathcal{X}} F(\boldsymbol{x}) \coloneqq f(\boldsymbol{H}\boldsymbol{x}) + g(\boldsymbol{A}\boldsymbol{x}) \qquad \text{where} \qquad \begin{cases} f(\boldsymbol{H}\boldsymbol{x}) \coloneqq \frac{1}{n}\sum_{i=1}^{n} f_i(\boldsymbol{h}_i^\top \boldsymbol{x}) \\ \\ g(\boldsymbol{A}\boldsymbol{x}) \coloneqq \frac{1}{m}\sum_{i=1}^{m} g_i(\boldsymbol{a}_i^\top \boldsymbol{x}). \end{cases} \tag{2.11}$$

We work under the same assumptions as before: $\mathcal{X} \subset \mathbb{R}^d$ is a compact and convex set for which projections are expensive but the LMO is efficient; each $f_i : \mathbb{R} \to \mathbb{R}$ is convex and L_f-smooth; $\boldsymbol{A}$ and $\boldsymbol{H}$ are data matrices in $\mathbb{R}^{m\times d}$ and $\mathbb{R}^{n\times d}$, whose i^{th} rows are denoted by the vectors $\boldsymbol{a}_i^\top$ and $\boldsymbol{h}_i^\top$, respectively; and $g_i : \mathbb{R} \to \mathbb{R} \cup \{+\infty\}$ are convex but possibly non-differentiable, and g has an efficient prox operator. This kind of separability for g translates into separability of its prox operator [178, see Section 2.1], a fact which we leverage in the design and analysis of our method.

Note that formulation (2.11) recovers a restricted instance of template (2.1), as follows. We let g_i be indicator functions of closed and convex intervals in $\mathbb{R}$ denoted as b_i. Furthermore, we let

$\boldsymbol{A}(\xi) \equiv \boldsymbol{a}_i^\top$. Then, problem (2.11) becomes

$$\min_{\boldsymbol{x}\in\mathcal{X}} \frac{1}{n}\sum_{i=1}^{n} f_i(\boldsymbol{h}_i^\top \boldsymbol{x}) \qquad \text{such that} \qquad \boldsymbol{a}_i^\top \boldsymbol{x} \in b_i, \forall i \in [m], \tag{2.12}$$

which is nothing but (2.1) for which $\xi \sim \mathcal{U}([m])$ (the uniform distribution with discrete support). This type of separable model includes, for example, box-constrained problems.

The *separable* finite-sum structure of (2.11) allows us to tackle both g and f stochastically and, therefore, more efficiently when m and n are large. We study Conditional Gradient methods (CGMs), otherwise known as the Frank-Wolfe algorithm, tailored for problem (2.11), for the case where g is either Lipschitz continuous or an indicator function of a set onto which projections are easy. The running example for the remaining sections is that of *strongly constrained* semidefinite programs (SDPs), that is, those which have a very large number of constraints. We have already seen such examples in the previous sections, notably the Sparsest Cut SDP in Section 2.5.3 for which $m \in \mathcal{O}\left(d^3\right)$.

Concretely, the SDP template is given by

$$\begin{aligned} \min_{\boldsymbol{X}\in\mathbb{S}_+^{d\times d}} \quad & \langle \boldsymbol{H}, \boldsymbol{X}\rangle \\ \text{subj. to} \quad & \langle \boldsymbol{A}_i, \boldsymbol{X}\rangle \triangleleft b_i, \quad i \in [m] \end{aligned} \tag{2.13}$$

where $\mathbb{S}_+^{d\times d}$ denotes the set of symmetric positive semidefinite matrices, $\boldsymbol{H} \in \mathbb{S}^{d\times d}$ is the symmetric cost matrix, $(\boldsymbol{A}_i, b_i) \in \mathbb{S}^{d\times d} \times \mathbb{R}$ characterize the constraints, and $\triangleleft$ represents either equality '=' or inequality '$\leq$' relations.

As already mentioned in Section 2.1, solving SDPs is computationally challenging due to their semidefinite cone constraint for which projections are expensive ($\mathcal{O}\left(d^3\right)$), the large cost of storing the decision variable $\boldsymbol{X}$, and the possibly large number of constraints for problems such as those mentioned above. The former concern is addressed by resorting to conditional gradient-based solvers [101, 113, 88, 232] since they avoid projection via LMOs. Further, reducing the storage requirements to optimal orders through sketching was studied by Yurtsever et al. [235]. However, scalable approaches to solving SDPs with a large number of constraints remain comparatively underexplored. We take a step in this latter direction by developing CGM variants that handle linear constraints in a randomized fashion.

Concretely, we propose a new CGM variant for convex finite-sum problems and analyze it for the case where g is either Lipschitz continuous or an indicator function of a set for which projections are easy. The method extends the recent work on stochastic Frank-Wolfe [159] to the composite template in (2.11). Our algorithm finds an ϵ-suboptimal solution after $\mathcal{O}\left(\epsilon^{-2}\right)$ iterations, matching the iteration complexity in Vladarean et al. [219] (presented in Section 2.4.4 as H-SPIDER-FW). However, we achieve this rate with a constant, as opposed to increasing, batch size strategy. Furthermore, Vladarean et al. [219] require a full gradient computation at predefined intervals

Algorithm	Reference	Iteration complexity	Total cost	Fixed batch size
SHCGM	Locatello et al. [142]	$\mathcal{O}(\epsilon^{-3})$	$\mathcal{O}(\epsilon^{-3} d\, m)$	N/A
H-1SFW	Vladarean et al. [219]	$\mathcal{O}(\epsilon^{-6})$	$\mathcal{O}(\epsilon^{-6} d)$	✓
H-SPIDER-FW	Vladarean et al. [219]	$\mathcal{O}(\epsilon^{-2})$	$\mathcal{O}(\epsilon^{-2} d\, m)$	✗
MOST-FW$^+$	Akhtar and Rajawat [1]	$\mathcal{O}(\epsilon^{-4})$	$\mathcal{O}(\epsilon^{-4} d)$	✓
H-SAG-CGM	*This Paper*	$\mathcal{O}(\epsilon^{-2})$	$\mathcal{O}(\epsilon^{-2} d)$	✓

Table 2.2: This table presents the costs of finding an ϵ-suboptimal solution for a given problem while treating the parameters d, n and m as constants. The $\mathcal{O}$ notation hides the parameters L_f, $\|\boldsymbol{A}\|$, $\mathcal{D}_{\mathcal{X}}$, and the absolute constants. We tailor the cost of existing methods for problem (2.11), noting that their cost for other problems can differ. The last column indicates whether the algorithm has an increasing or fixed batch size, and N/A refers to the fact that SHCGM processes all g_i at every iteration.

for the finite-sum setting, which is something we eschew in the sequel thanks to a different VR paradigm. Thus, our algorithm enjoys a total cost of $\mathcal{O}(\epsilon^{-2} d)$ which is independent of m. In contrast, the cost in Vladarean et al. [219] is $\mathcal{O}(\epsilon^{-2} d m)$. There is, of course, a trade-off: the gradient estimator we use to achieve this improvement requires an additional $\mathcal{O}(m)$ memory. We assume this trade-off to be acceptable for the considered applications. We support our theory with numerical experiments on matrix completion, k-Means clustering, and Sparsest Cut problems.

Additional related literature. The related literature discussion remains largely the same as that of Section 2.2. We mention two additional works published after the results described in Sections 2.1 – 2.5 and which are relevant to the remaining sections. First, Négiar et al. [159] showed that optimal convergence guarantees for CGMs can be obtained for separable objectives by considering a SAG-like gradient estimator [199]. Concretely, the authors achieve an iteration complexity of $\mathcal{O}(\epsilon^{-1})$ in this setting, which is on par with deterministic CGMs. By combining this idea with the homotopy framework, we are able to provide an improved randomized algorithm for composite objectives. Second, the parallel work of Akhtar and Rajawat [1] addresses a similar problem to (2.11) — we compare with their method in Table 2.2.

Additional notation. In addition to the diameter of a set defined in Equation (1.11), we define the following diameters of $\mathcal{X}$ with respect to the column space of a matrix $\boldsymbol{M}$ as

$$\mathcal{D}_i(\boldsymbol{M}) \coloneqq \max_{\boldsymbol{u},\boldsymbol{v}\in\mathcal{X}} \|\boldsymbol{M}(\boldsymbol{u}-\boldsymbol{v})\|_i,\ i \in \{1,2,\infty\} \tag{2.14}$$

Algorithm 2.3 H-SAG-CGM

Input: $\beta_0 > 0,\ \boldsymbol{x}_0 \in \mathcal{X},\ \boldsymbol{p}_0 \in \mathbb{R}^n,\ \boldsymbol{q}_0 \in \mathbb{R}^m,\ \boldsymbol{d}_0^f \in \mathbb{R}^d,\ \boldsymbol{d}_0^g \in \mathbb{R}^d$

for $k = 1, 2, \ldots$ **do**

Set $\gamma_k = \frac{2}{k+1},\ \beta_k = \beta_0/\sqrt{k+1}$

Sample $j \sim \mathcal{U}([n]),\ l \sim \mathcal{U}([m])$

$$\boldsymbol{p}_{k,i} = \begin{cases} \frac{1}{n} f_j'(\boldsymbol{h}_j^\top \boldsymbol{x}_k) & i = j \\ \boldsymbol{p}_{k-1,i} & i \neq j \end{cases} \quad \text{and} \quad \boldsymbol{q}_{k,r} = \begin{cases} \frac{1}{m} g_{\beta_k,l}'(\boldsymbol{a}_l^\top \boldsymbol{x}_k) & r = l \\ \boldsymbol{q}_{k-1,r} & r \neq l \end{cases}$$

$\boldsymbol{d}_k^f = \boldsymbol{d}_{k-1}^f + (\boldsymbol{p}_{k,j} - \boldsymbol{p}_{k-1,j})\boldsymbol{h}_j$

$\boldsymbol{d}_k^g = \boldsymbol{d}_{k-1}^g + (\boldsymbol{q}_{k,l} - \boldsymbol{q}_{k-1,l})\boldsymbol{a}_l$

$\boldsymbol{d}_k = \boldsymbol{d}_k^f + \boldsymbol{d}_k^g$

$\boldsymbol{w}_k = \mathrm{lmo}_{\mathcal{X}}(\boldsymbol{d}_k)$

$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \gamma_k(\boldsymbol{w}_k - \boldsymbol{x}_k)$

end for

2.6.1 Algorithm and convergence

Our method is presented in Algorithm 2.3. As before, the algorithm optimizes the smoothed version of the objective in (2.11) obtained through the technique described in Section 2.3, and given by the following expression

$$F_\beta(\boldsymbol{x}) \coloneqq f(\boldsymbol{H}\boldsymbol{x}) + g_\beta(\boldsymbol{A}\boldsymbol{x}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\boldsymbol{h}_i^\top \boldsymbol{x}) + \frac{1}{m}\sum_{j=1}^{m} g_{\beta,j}(\boldsymbol{a}_j^\top \boldsymbol{x}). \tag{2.15}$$

The algorithm proceeds in three conceptual steps at every iteration k: (1) estimate the gradient using random samples, (2) compute the LMO with respect to the gradient estimator $\boldsymbol{d}_k$ and (3) update the optimization variable $\boldsymbol{x}_k$.

The analysis of our algorithm consists of first establishing convergence for the smoothed gap

$$S_{\beta_k}(\boldsymbol{x}_{k+1}) \coloneqq \mathbb{E}[F_{\beta_k}(\boldsymbol{x}_{k+1}) - F^\star],$$

and then translating this convergence into guarantees for the original problem. For the latter part, we rely on the techniques proposed by Tran-Dinh, Fercoq, and Cevher [212]. For the case when

g is a Lipschitz continuous function, we seek points $\boldsymbol{x}_k$ such that

$$\mathbb{E}\left[F(\boldsymbol{x}_k) - F^\star\right] \leq \epsilon.$$

Otherwise, when g is the indicator of a separable constraint set $\mathcal{K} \in \mathbb{R}^m \coloneqq \mathcal{K}_1 \times \ldots \times \mathcal{K}_m$, $\mathcal{K}_i \subset \mathbb{R}$, we want also to quantify the degree of constraint violation. Therefore, we seek $\boldsymbol{x}_k$ such that

$$\left|\mathbb{E}\left[f(\boldsymbol{x}_k)\right] - F^\star\right| \leq \epsilon \quad \text{and} \quad \mathbb{E}[\operatorname{dist}(\boldsymbol{A}\boldsymbol{x}_k; \mathcal{K})] \leq \epsilon.$$

Our algorithm guarantees at every iteration that $\boldsymbol{x}_k$ is in $\mathcal{X}$ and asymptotically that $\boldsymbol{A}\boldsymbol{x}_k \in \mathcal{K}$. As in prior sections, we assume that when using indicators of constraint sets, Slater's condition holds and therefore, strong duality is ensured.

Smoothed gap recurrence

We first establish a recursive inequality involving $S_{\beta_k}(\boldsymbol{x}_{k+1})$, which lies at the heart of our analysis and which appears with slight variations in Locatello et al. [142] and Vladarean et al. [219]. Its proof is deferred to Appendix A.3.1.

Lemma 2.4. *Consider H-SAG-CGM (Algorithm 2.3). Then, for all $k \geq 1$, it holds that*

$$S_{\beta_k}(\boldsymbol{x}_{k+1}) \leq (1-\gamma_k) S_{\beta_{k-1}}(\boldsymbol{x}_k) + \gamma_k D_{\mathcal{X}} \mathbb{E}\left[\|\nabla F_{\beta_k}(\boldsymbol{x}_k) - \boldsymbol{d}_k\|\right] + \frac{\gamma_k^2 D_{\mathcal{X}}^2 L_{F_{\beta_k}}}{2},$$

where $L_{F_{\beta_k}} = \frac{\|\boldsymbol{H}\| L_f}{n} + \frac{\|\boldsymbol{A}\|}{\beta_k m}$ represents the smoothness constant of the surrogate objective F_{β_k}.

Lemma 2.4 shows how the smoothed gap's convergence rate depends on the design parameters γ_k and β_k and the variance of the stochastic gradient estimator (the term $\mathbb{E}\left[\|\nabla F_{\beta_k}(\boldsymbol{x}_k) - \boldsymbol{d}_k\|\right]$ is upper bounded by the square root of the variance $\sqrt{\mathbb{E}\left[\|\nabla F_{\beta_k}(\boldsymbol{x}_k) - \boldsymbol{d}_k\|^2\right]}$ via Jensen's inequality). Since we have free choice over γ_k and β_k to get the best possible rates in the analysis, this leaves the variance of the stochastic gradient estimator as the decisive term. Prior work [219] (presented in Sections 2.1—2.5) relies on variance-reduced gradient estimators devised to handle arbitrary stochastic objectives, thus failing to exploit the separable finite-sum structure often encountered in practice. Instead, we leverage the SAG-like gradient estimator, which was recently shown to induce optimal rates for CGMs in the standard setting [159], and extend similar benefits to our composite problem.

Stochastic Average Gradient (SAG) error bounds

We use a SAG estimator for each of the two components of the smoothed objective F_β. Specifically, at each iteration of Algorithm 2.3, the j-th coordinate of the gradient of f is updated using

the SAG estimator

$$\boldsymbol{p}_{k,i} = \begin{cases} \frac{1}{n} f'(\boldsymbol{h}_i^\top \boldsymbol{x}_k) & i = j, \\ \boldsymbol{p}_{k-1,i} & i \neq j. \end{cases} \tag{2.16}$$

Similarly, the SAG estimator for g_{β_k} updates the l-th coordinate of the gradient of g_{β_k} as

$$\boldsymbol{q}_{k,r} = \begin{cases} \frac{1}{m} g'_{\beta_k,l}(\boldsymbol{a}_l^\top \boldsymbol{x}_k) & r = l, \\ \boldsymbol{q}_{k-1,r} & r \neq l. \end{cases} \tag{2.17}$$

Thus, the overall gradient term $\boldsymbol{d}_k$ in this case is the sum of two stochastic terms given by $\boldsymbol{d}_k = \boldsymbol{H}^\top \boldsymbol{p}_k + \boldsymbol{A}^\top \boldsymbol{q}_k$. We now present two lemmas characterizing the errors of $\boldsymbol{p}_k$ and $\boldsymbol{q}_k$ in ℓ_1-norm.

Lemma 2.5 (Lemma 3 of Négiar et al. [159])**.** *Consider H-SAG-CGM (Algorithm 2.3) and the SAG estimator $\boldsymbol{p}_k$ defined in* (2.16)*. Then, for all $k \geq 2$,*

$$\mathbb{E}\left[\|\nabla f(\boldsymbol{H}\boldsymbol{x}_k) - \boldsymbol{p}_k\|_1\right] \leq \left(1 - \tfrac{1}{n}\right)^k \|\nabla f(\boldsymbol{H}\boldsymbol{x}_0) - \boldsymbol{p}_0\|_1 + C_1 \left(1 - \tfrac{1}{n}\right)^{k/2} \log k + \frac{C_2}{k},$$

where $C_1 = 2n^{-1} L_f \mathcal{D}_1(\boldsymbol{H})$, $C_2 = 4n^{-1}(n-1) L_f \mathcal{D}_1(\boldsymbol{H})$ and the expectation is taken over all previous steps in the algorithm.

Lemma 2.6. *Consider H-SAG-CGM (Algorithm 2.3) and the SAG estimator γ_k defined in* (2.17)*. Then, for all $k \geq 2$,*

$$\mathbb{E}[\|\nabla g_{\beta_k}(\boldsymbol{A}\boldsymbol{x}_k) - \boldsymbol{q}_k\|_1] \leq \left(1 - \tfrac{1}{m}\right)^k \|\nabla g_{\beta_0}(\boldsymbol{A}\boldsymbol{x}_0) - \boldsymbol{q}_0\|_1 + \frac{C}{\sqrt{k}}$$

where $C = 10\beta_0^{-1} \mathcal{D}_1(\boldsymbol{A})$ and the expectation is taken over all previous steps of the algorithm.

We present the proof of Lemma 2.6 in Appendix A.3.2, under the assumption that g is either the indicator of a convex and separable constraint set or a Lipschitz continuous function. For the proof of Lemma 2.5, we refer the reader to Négiar et al. [159].

Discussion. Lemma 2.5 shows that the SAG-like estimator of ∇f provides an error bound in ℓ_1-norm decaying as $\mathcal{O}(1/k)$ in expectation. This decay does not carry over to the estimator of ∇g_{β_k}, as demonstrated by Lemma 2.6, due to the associated $\frac{1}{\beta_k}$ factor that results from smoothing.

Convergence of the smoothed gap and original objective

Combining Lemmas 2.5 and 2.6 with Lemma 2.4 gives the convergence rates for Algorithm 2.3, which we now present. Its proof is deferred to Appendix A.3.3.

Theorem 2.3. *The sequence generated by H-SAG-CGM (Algorithm 2.3) satisfies, for all $k \geq 2$,*

$$S_{\beta_k}(\boldsymbol{x}_{k+1}) \leq \frac{C_1}{\sqrt{k}} + \frac{C_2}{k} + \frac{C_3}{k^2},$$

for the following constants

- $C_1 = \beta_0^{-1}(2\mathcal{D}_{\mathcal{X}}^2 \| \boldsymbol{A} \| + 10\mathcal{D}_1(\boldsymbol{A}))$;
- $C_2 = 8L_f\mathcal{D}_1(\boldsymbol{H})\mathcal{D}_\infty(\boldsymbol{H}) + 2n^{-1}L_f \| \boldsymbol{H} \| \mathcal{D}_{\mathcal{X}}^2$;
- $C_3 = 2n^2\mathcal{D}_\infty(\boldsymbol{H})\big(\|\nabla f(\boldsymbol{H}\boldsymbol{x}_1) - \boldsymbol{p}_0\|_1 + 32L_f\mathcal{D}_1(\boldsymbol{H})\big) + 2m^2\mathcal{D}_\infty(\boldsymbol{A})\|\nabla g_{\beta_0}(\boldsymbol{A}\boldsymbol{x}_1) - \boldsymbol{q}_0\|_1$.

Using the techniques described by Tran-Dinh, Fercoq, and Cevher [212], we translate this bound into convergence guarantees on the original problem in the following corollaries. Their proofs are deferred to Appendix A.3.4 and A.3.5, respectively.

Corollary 2.3. *Suppose $g : \mathbb{R}^m \to \mathbb{R}$ is L_g-Lipschitz continuous. Then, the estimates generated by H-SAG-CGM (Algorithm 2.3) satisfy*

$$\mathbb{E}[F(\boldsymbol{x}_{k+1}) - F^\star] \leq \frac{C_1}{\sqrt{k}} + \frac{C_2}{k} + \frac{C_3}{k^2} + \frac{\beta_0 L_g^2}{2\sqrt{k}}$$

where the constants C_1, C_2 and C_2 are defined in Theorem 2.3.

Corollary 2.4. *Suppose g is the indicator function of a closed and convex set $\mathcal{K} \in \mathbb{R}^m$, $\mathcal{K} \coloneqq \mathcal{K}_1 \times \ldots \times \mathcal{K}_m$, $\mathcal{K}_i \subseteq \mathbb{R}$, $\forall i \in [m]$. Then, for H-SAG-CGM (Algorithm 2.3), we have a lower bound on the suboptimality as $\mathbb{E}\left[f(\boldsymbol{H}\boldsymbol{x}_{k+1}) - f(\boldsymbol{H}\boldsymbol{x}^\star)\right] \geq -\|\boldsymbol{\lambda}^\star\|\mathbb{E}\left[\operatorname{dist}(\boldsymbol{A}\boldsymbol{x}_{k+1}, \mathcal{K})\right]$, where $\boldsymbol{\lambda}^\star$ is a solution of the dual problem, and the following upper bounds on the suboptimality and feasibility:*

$$\mathbb{E}\left[f(\boldsymbol{H}\boldsymbol{x}_{k+1}) - f(\boldsymbol{H}\boldsymbol{x}^\star)\right] \leq \frac{C_1 + \beta_0}{\sqrt{k}} + \frac{C_2}{k} + \frac{C_3}{k^2}, \text{ and}$$
$$\mathbb{E}\left[\operatorname{dist}(\boldsymbol{A}\boldsymbol{x}_{k+1}, \mathcal{K})\right] \leq \frac{C_4}{\sqrt{k}} + \frac{\sqrt{2C_2}}{k^{3/4}} + \frac{\sqrt{2C_3}}{k^{5/4}},$$

where the constants C_1, C_2 and C_3 are defined in Theorem 2.3 and $C_4 = \left(\frac{3\beta_0\|\boldsymbol{\lambda}^\star\|}{2} + \sqrt{2C_1}\right)$.

Discussion. Even in the deterministic setting studied by Yurtsever et al. [232], the convergence rate of Homotopy CGM is lower bounded by $\Omega(1/\sqrt{k})$, as demonstrated theoretically by Lan [134] and practically by Kerdreux, d'Aspremont, and Pokutta [122]. Corollaries 2.3 and 2.4 show that our algorithm achieves this lower bound.

Since H-SAG-CGM uses one LMO and one IFO per iteration, the convergence complexity in terms of the number of calls to both oracles is $\mathcal{O}\left(\epsilon^{-2}\right)$.

While H-SAG-CGM and H-SPIDER-FW [219] (presented in prior sections) enjoy a similar overall rate, the latter requires an exponentially increasing batch size. Combined with occasional

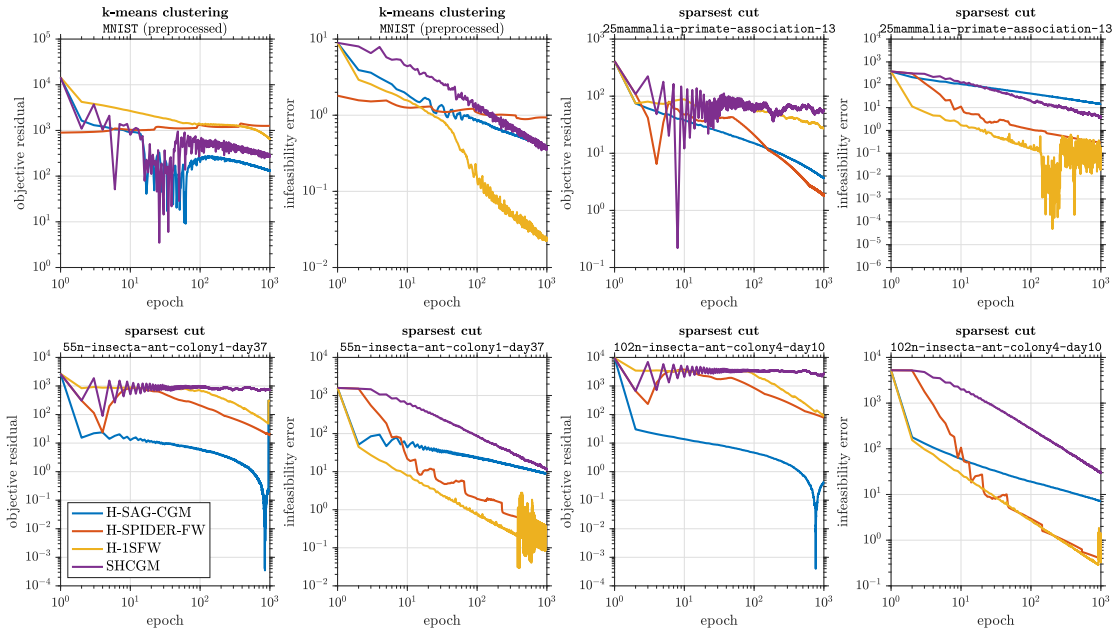


Figure 2.5: Comparing H-SAG-CGM to state-of-the-art baselines on two distinct SDP-relaxation tasks, k-Means (2.9) and Sparsest Cut (2.10). The *objective residual* and *infeasibility error* represent $|f(\boldsymbol{x}_k) - f^\star|/|f^\star|$ and $\mathrm{dist}(\boldsymbol{A}\boldsymbol{x}_k, \mathcal{K})$, respectively. The x-axis is in terms of the constraint epochs. One constraint epoch corresponds to a full pass over all the constraints. Note that for the k-Means clustering experiment, we deliberately prevented H-SPIDER-FW from performing full passes over the constraints, resulting in a noticeable degradation in performance.

full passes, this quickly becomes impractical for strongly constrained problems. As an alternative, Vladarean et al. [219] propose H-1SFW, which does use a fixed batch size but at the cost of an impractical $\mathcal{O}\left(\epsilon^{-6}\right)$ rate. In stark contrast, our algorithm enjoys the optimal rate without an increasing batch size.

2.6.2 Experiments

We demonstrate the empirical performance of H-SAG-CGM for the k-Means clustering SDP introduced in Section 2.5.2 and the uniform Sparsest Cut SDP introduced in Section 2.5.3. We performed these experiments in MATLAB R2019b[1].

Baselines. We compare H-SAG-CGM (Algorithm 2.3) against the following methods: SHCGM [142], H-SPIDER-FW [219] and H-1SFW [219]. Note that SHCGM only works in the case of deterministic g, and importantly, H-SPIDER-FW requires an increasing batch size.

[1]The codes are publicly available at `https://github.com/ratschlab/faster-hcgm-composite`

Challenges. The hyperparameter β_0 determines a trade-off between convergence in the objective residual and the infeasibility error by modulating the relative magnitude of the two components' gradients. This quantity needs to be tuned for every instance of the problem, a challenge that is shared among homotopy CGM approaches [232, 142, 219]. Since the value of β_0 impacts convergence, developing principled ways of setting it is a meaningful direction for further research.

The k-Means clustering relaxation

In this experiment, we test H-SAG-CGM on the k-Means Clustering SDP, introduced with full details in Section 2.5.2. The problem is strongly constrained with a total of $n^2 + n$ constraints. We use the same setup as before and find that $\beta_0 = 7$ is appropriate for H-SAG-CGM.

We compare the methods based on the number of epochs (an epoch corresponds to a full pass over the constraints) since different methods use different batch sizes in this experiment. The two leftmost plots in the upper row of Figure 2.5 present the outcomes of this experiment.

Computing an ℓ_2^2 embedding for the Uniform Sparsest Cut problem

In this experiment, we test H-SAG-CGM on the uniform Sparsest Cut SDP, introduced with full details in Section 2.5.3. This problem is particularly interesting because of the $\mathcal{O}\left(n^3\right)$ number of constraints. We use the same datasets described in Table 2.1. We use $\beta_0 = 100$ for H-SAG-CGM on all three network datasets.

Figure 2.5 presents the results of this experiment. As in the k-Means experiment, H-SPIDER-FW is affected by the growing number of constraints because of its increasing batch size strategy. Other methods, with constant batch size, are less affected. H-SAG-CGM performs competitively against H-SPIDER-FW without requiring an increasing batch size.

2.7 Conclusion

This chapter introduced three stochastic Conditional Gradient-based methods for tackling convex objectives subject to a large number of linear constraints. The key feature of our algorithms is that they process only a subset of the constraints per iteration, thus gaining an edge over methods that require full passes for large-scale problems.

We first proposed two methods addressing a general template of the problem, expressed in terms of arbitrary probability distributions and a possibly infinite number of stochastic linear constraints. The methods rely on two different variance reduction schemes for estimating the gradient: a simple exponential moving average estimator using a constant batch size [155] and the more sophisticated Stochastic Path-Integrated Differential Estimator [80] requiring increasing batch

sizes. The former ensures a $\mathcal{O}\left(\epsilon^{-6}\right)$ convergence with respect to both the number of LMOs and stochastic gradient computations, while the latter requires only $\mathcal{O}\left(\epsilon^{-2}\right)$ LMOs and $\mathcal{O}\left(\epsilon^{-4}\right)$ stochastic gradient computations. We highlighted a trade-off between the simplicity of the variance reduction scheme and the theoretical speed of convergence, and empirically observed that our methods outperformed existing baselines.

To overcome the impracticality of increasing batch sizes in settings with finite data, we further proposed and analyzed an algorithm for finite-sum-type objectives. Given this additional assumption on the structure of the problem, we were able to leverage an efficient SAG-like estimator of the gradient [199] to achieve $\mathcal{O}\left(\epsilon^{-2}\right)$ convergence with respect to both the number of LMOs and stochastic gradient computations. The trade-off, in this case, was between convergence speed and additional memory requirements. Finally, we empirically observed that this latter method performs on par with the initial, more sophisticated variance reduction scheme.

A possible future direction within the algorithmic framework of this chapter is to automate the selection of the hyperparameter β_0, common to all presented methods. We suspect it may be set in a data-dependent manner to achieve a balanced optimization of the functional residual and the smoothed constraints.

3 A Frank-Wolfe generalization for composite non-differentiable objectives

This chapter is based on the published work Vladarean et al. [220], presented at COLT 2023.

Co-authors: Nikita Doikov, Martin Jaggi and Nicolas Flammarion

Contributions

M.-L. Vladarean	— methodology 45%, formal derivations 60%, writing 70%, experiments 100%
N. Doikov	— methodology 45%, formal derivations 40%, writing 30%
M. Jaggi	— methodology 5%, writing – review and editing
N. Flammarion	— methodology 5%, writing – review and editing, project administration, supervision

Summary. This chapter studies Frank-Wolfe-type methods for a class of non-differentiable composite objectives, a setting which causes non-convergence for the standard formulation of these algorithms. We propose methods that leverage the structure of the composition by handling the differentiable and non-differentiable components separately, linearizing only the smooth parts. We thus obtain new generalizations of the classical Frank-Wolfe and the Conditional Gradient Sliding methods that successfully optimize the considered class of objectives. Our algorithms rely on a stronger version of the linear minimization oracle, which can be efficiently implemented in several practical applications. We provide the basic version of our method with an affine-invariant analysis and prove global convergence rates for both convex and non-convex objectives, the former of which are on par with the smooth setting. Furthermore, we propose an accelerated method with improved complexity in terms of the number of Jacobian computations in the convex case. Finally, we provide illustrative experiments supporting our theoretical results.

3.1 Introduction

In this chapter, we consider fully composite optimization problems of the form

$$\min_{\boldsymbol{x}\in\mathcal{X}} \big[\varphi(\boldsymbol{x}) \;\coloneqq\; F(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{x})\big], \tag{3.1}$$

where $\mathcal{X} \subset \mathbb{R}^d$ is a convex and compact set, $F : \mathbb{R}^n \times \mathcal{X} \to \mathbb{R}$ is a simple but possibly *non-differentiable* convex function and $\boldsymbol{f} : \mathcal{X} \to \mathbb{R}^n$ is a smooth mapping, which is the *main source of computational burden*.

Problems of this type cover and generalize many classical use cases of composite optimization and are often encountered in applications. We develop efficient algorithms for solving (3.1) by leveraging the *structure of the objective* and using the *linearization principle*. Our method generalizes the well-known Frank-Wolfe algorithm [84] and ensures provably faster convergence rates than methods treating φ in a black-box fashion.

A classical algorithm for solving smooth versions of problem (3.1) is the Gradient Descent method (GD), proposed by Cauchy in 1847 (see historical note by Lemaréchal [139]). It rests on the idea of linearizing the function around the current iterate, taking a step in the negative gradient direction and projecting the result onto the feasible set $\mathcal{X}$ for $k \geq 0$:

$$\boldsymbol{y}_{k+1} \;=\; \operatorname{proj}_{\mathcal{X}}\big(\boldsymbol{y}_k - \alpha_k \nabla\varphi(\boldsymbol{y}_k)\big), \qquad \alpha_k > 0. \tag{3.2}$$

Surprisingly, the same kind of iterations can minimize *general* non-smooth convex functions by substituting $\nabla\varphi(\boldsymbol{y}_k)$ with any *subgradient* in the subdifferential $\partial\varphi(\boldsymbol{y}_k)$. The resulting Subgradient method was proposed by Shor [204].

Another notable approach to solving smooth instances of (3.1) over a convex and *bounded* constraint set $\mathcal{X}$ is the Frank-Wolfe (FW) or Conditional Gradient (CG) method [84]. Again, a linearization of the objective around the current iterate is used to query the so-called *Linear Minimization Oracle* (LMO) associated with $\mathcal{X}$, for every $k \geq 0$:

$$\boldsymbol{y}_{k+1} \;\in\; \operatorname*{arg\,min}_{\boldsymbol{x}} \big\{ \langle \nabla\varphi(\boldsymbol{y}_k), \boldsymbol{x}\rangle \mid \boldsymbol{x} \in \boldsymbol{y}_k + \gamma_k(\mathcal{X} - \boldsymbol{y}_k) \big\}, \qquad \gamma_k \in (0, 1]. \tag{3.3}$$

Steps of type (3.3) are significantly cheaper than those involving projections (3.2) for a few important domains such as nuclear norm balls and spectrahedrons [53], rendering FW the algorithm of choice in such scenarios. Moreover, the solutions found by FW methods can benefit from additional properties such as sparsity [112]. These desirable features make FW methods suitable for large-scale optimization, a fact which prompted an increased interest in recent years (we point the reader to the monograph of Braun et al. [24] for a detailed presentation). Unfortunately, the vanilla FW algorithm does not extend to non-differentiable problems as straightforwardly as GD — a counterexample is given by Nesterov [165]. The question of developing non-smooth versions of the FW algorithm, therefore, remains open and is the main

focus of this chapter.

Finally, we touch on the issue of convergence rates — the main avenue for characterizing optimization methods' practicality. Nemirovski and Yudin [161] establish that the $\mathcal{O}\left(1/\sqrt{k}\right)$ rate of the aforementioned projected Subgradient method is optimal for *general non-differentiable* convex problems, while the $\mathcal{O}(1/k)$ rate of its counterpart projected GD is far from the $\Omega(1/k^2)$ lower bound for $L-$smooth convex functions. Analogous results are established by Lan [134] for LMO-based algorithms, although in this case, the $\mathcal{O}(1/k)$ rate is matched by a lower bound of the same order for smooth convex problems. This relatively slow convergence of FW algorithms results from their *affine-invariant* oracle, which is independent of the choice of norm. In light of these stringent lower bounds established for black-box models (i.e., generic function classes), the only avenue for improving convergence rates is to impose additional structure on the objective.

Starting from this observation, we study the structured subclass of non-smooth and possibly non-convex problems given by (3.1). We propose methods that require only linearizations of the differentiable component $\boldsymbol{f}$, while the non-differentiable function F is kept as a part of the subproblem solved within oracle calls. We show that this approach is a viable way of generalizing FW methods to address problem (3.1), with the possibility of acceleration in convex scenarios. Our contributions are summarized as follows.

- We propose a basic method for template (3.1), which is *affine-invariant* and equipped with accuracy certificates. We prove the global convergence rate of $\mathcal{O}(1/k)$ in the convex setting and of $\widetilde{\mathcal{O}}\left(1/\sqrt{k}\right)$ in the non-convex case.
- We propose an accelerated method with inexact proximal steps which attains a convergence rate of $\mathcal{O}\left(1/k^2\right)$ for convex problems. Our algorithm achieves the optimal $\mathcal{O}\left(\epsilon^{-1/2}\right)$ oracle complexity for smooth convex problems with respect to the number of Jacobian computations ($\nabla \boldsymbol{f}$).
- We provide proof-of-concept experiments demonstrating our approach's efficiency for solving the structured template (3.1).

3.2 Related work

Our results lie at the intersection of two broad lines of study: general methods for composite optimization and FW algorithms. The former category encompasses many approaches that single out non-differentiable components in the objective's structure and leverage this knowledge in the design of efficient optimization algorithms. This approach originated in the works of Burke [28, 29], Nesterov [166], Nemirovski [160], Pennanen [183], and Boţ, Grad, and Wanka [21, 20]. A popular class of *additive* composite optimization problems was proposed by Beck and Teboulle [14] and Nesterov [164] and the modern algorithms for general composite formulations were developed by Cui, Pang, and Sen [58], Drusvyatskiy and Lewis [76], Drusvyatskiy and Paquette [77], Bolte, Chen, and Pauwels [18], Burke, Tim, and Nguyen [31], and Doikov and Nesterov [73].

The primitive on which most of the aforementioned methods rely is a *proximal-type* step — a generalization of (3.2). Such steps may pose a significant computational burden depending on the geometry of the set $\mathcal{X}$ (see discussion in Section 1.3.1). Doikov and Nesterov [73] propose an alternative *contracting-type* method for *fully composite* problems, which generalizes the vanilla FW algorithm. Their method relies on a simpler primitive built on the linearization principle, which can be much cheaper in practice. We study the same problem structure as Doikov and Nesterov [73] and devise methods with several advantages over the aforementioned approach, including an *affine-invariant analysis*, *accuracy certificates*, convergence guarantees for non-convex problems and, in the convex case, an accelerated convergence. Moreover, we decouple stepsize selection from the computational primitive to enable efficient line search procedures.

Our methods are also intimately related to FW algorithms, which they generalize. For smooth and convex problems, vanilla FW converges at the cost of $\mathcal{O}\left(\epsilon^{-1}\right)$ LMO and *First Order Oracle* (FO) calls with respect to the Frank-Wolfe gap (a convenient accuracy measure) [112]. For smooth non-convex problems, a gap value of at most ϵ is attained after $\mathcal{O}\left(\epsilon^{-2}\right)$ LMO and FO calls [130]. This relatively slow convergence of LMO-based methods has driven recent efforts towards devising variants with improved guarantees, as we review next. The number of FO calls was improved to match the lower bound for smooth convex optimization by Lan and Zhou [135]; local acceleration was achieved following a burn-in phase by Diakonikolas, Carderera, and Pokutta [71], Carderera et al. [37], and Chen and Sun [48]; and empirical performance was enhanced by adjusting the update direction with gradient information by Combettes and Pokutta [52]. Of the aforementioned works, the closest to ours is the Conditional Gradient Sliding (CGS) algorithm proposed by Lan and Zhou [135] and further studied by Yurtsever, Sra, and Cevher [234] and Qu, Li, and Xu [187]. CGS uses the acceleration framework of Nesterov [163] and solves the projection subproblem inexactly via the FW method, achieving the optimal complexity of $\mathcal{O}\left(\epsilon^{-1/2}\right)$ FO calls for smooth convex problems. We rely on a similar scheme for improving FO complexity for our structured non-smooth template whenever convexity is ensured.

The FW algorithm was also studied for generic non-smooth convex problems by Lan [134], who proposed a smoothing-based approach matching the lower bound of $\Omega(\epsilon^{-2})$ LMO calls. The method however requires $\mathcal{O}\left(\epsilon^{-4}\right)$ FO calls, a complexity which is later improved to $\mathcal{O}\left(\epsilon^{-2}\right)$ by Garber and Hazan [90] and Ravi, Collins, and Singh [190] through a modified LMO, and by Thekumparampil et al. [211] through a combination of smoothing and the CGS algorithm. We also mention FW methods for additive composite optimization [4, 232, 231, 241], with the former three relying on smoothing and additional proximal steps, and the latter assuming a very restricted class of objectives. In comparison, our methods leverage the structure of problem (3.1) and a modified LMO to speed up convergence, with the added benefits of an affine invariant algorithm and analysis.

Finally, two concurrent works study FW methods for some restricted classes of non-smooth and non-convex problems. De Oliveira [68] shows that vanilla FW with line-search can be applied to the special class of upper$-C^{1,\alpha}$ functions when one replaces gradients with an arbitrary element in the Clarke subdifferential. A rate of $\mathcal{O}\left(\epsilon^{-2}\right)$ is shown for reaching a Clarke-stationary point

in a setting comparable to ours. A similar rate is shown by Kreimeier et al. [128] for reaching a d-stationary point of abs-smooth functions by using a modified LMO. Both these algorithms are structure-agnostic. A summary of method complexities for solving non-smooth problems is provided in Table 3.1.

Reference	φ class	Use structure?	# FO	# PO/LMO	Notes
Shor [204]	cvx, L-cont	no	$\mathcal{O}(\epsilon^{-2})$ [(1)]	$\mathcal{O}(\epsilon^{-2})$ [(1)]	projection
Thekumparampil et al. [211]	cvx, L-cont	no	$\mathcal{O}(\epsilon^{-2})$ [(1)]	$\mathcal{O}(\epsilon^{-2})$ [(1)]	smoothing, vanilla LMO
Doikov and Nesterov [73]	cvx, fully-comp	yes	$\mathcal{O}(\epsilon^{-1})$ [(1)]	$\mathcal{O}(\epsilon^{-1})$ [(1)]	modif. LMO
(our results) Alg. 2	cvx, fully-comp	yes	$\mathcal{O}(\epsilon^{-1/2})$ [(1)]	$\mathcal{O}(\epsilon^{-1})$ [(1)]	modif. LMO
De Oliveira [68]	non-cvx, upper-$C^{1,\alpha}$	no	$\mathcal{O}(\epsilon^{-2})$ [(2)]	$\mathcal{O}(\epsilon^{-2})$ [(2)]	vanilla LMO
Kreimeier et al. [128]	non-cvx, abs-smooth	no	$\mathcal{O}(\epsilon^{-2})$ [(3)]	$\mathcal{O}(\epsilon^{-2})$ [(3)]	modif. LMO
Drusvyatskiy and Paquette [77]	non-cvx, comp	yes	$\mathcal{O}(\epsilon^{-2})$ [(4)]	$\mathcal{O}(\epsilon^{-2})$ [(4)]	prox. steps
(our results) Alg. 1	non-cvx, fully-comp	yes	$\tilde{\mathcal{O}}(\epsilon^{-2})$ [(5)]	$\tilde{\mathcal{O}}(\epsilon^{-2})$ [(5)]	modif. LMO

Table 3.1: Summary of convergence complexities for solving non-differentiable composite problems. PO denotes the projection oracle, and the symbol # prefixing an oracle type denotes the number of such oracle calls. Note [(1)] marks complexities reaching an ϵ functional residual. Note [(2)] marks complexities for reaching Clarke-stationary points. Note [(3)] marks complexities for obtaining $d-$stationary points. Note [(4)] marks the complexity of reaching a small norm of the gradient mapping. Finally, note [(5)] marks the complexity of minimizing the positive quantity (3.14).

3.3 Problem setup, assumptions and examples

The problems addressed in this chapter adhere to the following template

$$\varphi^\star \quad = \quad \min_{\boldsymbol{x}\in\mathcal{X}} \Big[\varphi(\boldsymbol{x}) \quad \coloneqq \quad F(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{x}) \Big], \qquad \mathcal{X} \subset \mathbb{R}^d, \tag{3.4}$$

where $\mathcal{X}$ is a convex and compact set and the inner mapping $\boldsymbol{f} : \mathcal{X} \to \mathbb{R}^n$ is differentiable and defined as $\boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), \ldots, f_n(\boldsymbol{x})) \in \mathbb{R}^n$, where each $f_i : \mathcal{X} \to \mathbb{R}$ is differentiable. We assume access to a first-order oracle $\nabla \boldsymbol{f}$, which is the main source of computational burden. The outer component $F : \mathbb{R}^n \times \mathcal{X} \to \mathbb{R}$, on the other hand, is *directly accessible* to the algorithm designer and is *simple* (see assumptions). However, F is possibly non-differentiable.

We propose two algorithmic solutions addressing problem (3.4), which we call a *fully composite*

problem. Our methods importantly assume that subproblems of the form

$$\operatorname*{argmin}_{\boldsymbol{x}\in\mathcal{X}} F\big(\boldsymbol{A}\boldsymbol{x}+\boldsymbol{b},\boldsymbol{x}\big)+\langle \boldsymbol{u},\boldsymbol{x}\rangle \tag{3.5}$$

are efficiently solvable, where $\boldsymbol{A}\in\mathbb{R}^{n\times d}$ and $\boldsymbol{b}\in\mathbb{R}^n, \boldsymbol{u}\in\mathbb{R}^d$. Oracles of type (3.5) are sequentially called during the optimization procedure and take as arguments linearizations of the difficult nonlinear components of (3.4). Naturally, solving (3.5) cheaply is possible only when F is simple and $\mathcal{X}$ has an amenable structure.

In particular, template (3.4) encompasses some standard problem formulations. For example, the classical Frank-Wolfe setting is recovered when $F(\boldsymbol{u},\boldsymbol{x})\equiv u^{(1)}$, in which case problem (3.4) becomes $\min_{\boldsymbol{x}\in\mathcal{X}} f_1(\boldsymbol{x})$ and subproblem (3.5) reduces to a simple LMO: $\operatorname{argmin}_{\boldsymbol{x}\in\mathcal{X}}\langle \boldsymbol{u},\boldsymbol{x}\rangle$. The setting of proximal-gradient methods is similarly covered, by letting $F(\boldsymbol{u},\boldsymbol{x})\equiv u^{(1)}+\psi(\boldsymbol{x})$ for a given convex function ψ (e.g., a regularizer). Then, problem (3.4) reduces to additive composite optimization $\min_{\boldsymbol{x}\in\mathcal{X}}\Big\{f_1(\boldsymbol{x})+\psi(\boldsymbol{x})\Big\}$, and subproblem (3.5) becomes $\operatorname{argmin}_{\boldsymbol{x}\in\mathcal{X}}\Big\{\langle \boldsymbol{u},\boldsymbol{x}\rangle+\psi(\boldsymbol{x})\Big\}$.

For the remainder of this chapter, we use the notation defined in Section 1.5. In addition, we introduce the following shorthand for representing the second directional derivatives applied to the same direction $\boldsymbol{h}\in\mathbb{R}^d$ as $\nabla^2 f(\boldsymbol{x})[\boldsymbol{h}]^2 \coloneqq \langle\nabla^2 f(\boldsymbol{x})\boldsymbol{h},\boldsymbol{h}\rangle\in\mathbb{R}$, and $\nabla^2\boldsymbol{f}(\boldsymbol{x})[\boldsymbol{h}]^2 \coloneqq \sum_{i=1}^n \boldsymbol{e}_i\nabla^2 f_i(\boldsymbol{x})[\boldsymbol{h}]^2\in\mathbb{R}^n$, for scalar-valued and vector-valued functions, respectively. We now formally state the assumptions on the fully composite problem (3.4).

Assumption 3.1. *The outer function $F:\mathbb{R}^n\times\mathcal{X}\to\mathbb{R}$ is jointly convex in its arguments. Additionally, $F(\boldsymbol{u},\boldsymbol{x})$ is subhomogeneous in $\boldsymbol{u}$,*

$$F(\gamma\boldsymbol{u},\boldsymbol{x}) \;\le\; \gamma F(\boldsymbol{u},\boldsymbol{x}),\quad \forall\,\boldsymbol{u}\in\mathbb{R}^n,\ \boldsymbol{x}\in\mathcal{X},\ \gamma\ge 1. \tag{3.6}$$

Assumption 3.1.a. *The inner mapping $\boldsymbol{f}:\mathcal{X}\to\mathbb{R}^n$ is differentiable and the following affine-invariant quantity is bounded*

$$\mathcal{S} \;=\; \mathcal{S}_{\boldsymbol{f},F,\mathcal{X}} \coloneqq \sup_{\substack{\boldsymbol{x},\boldsymbol{y}\in\mathcal{X},\,\gamma\in(0,1]\\ \boldsymbol{y}_\gamma=\boldsymbol{x}+\gamma(\boldsymbol{y}-\boldsymbol{x})}} F\big(\tfrac{2}{\gamma^2}\big[\boldsymbol{f}(\boldsymbol{y}_\gamma)-\boldsymbol{f}(\boldsymbol{x})-\nabla\boldsymbol{f}(\boldsymbol{x})(\boldsymbol{y}_\gamma-\boldsymbol{x})\big],\boldsymbol{y}_\gamma\big) \;<\; +\infty. \tag{3.7}$$

Assumption 3.1.b. *Each component $f_i(\cdot)$ has a Lipschitz continuous gradient on $\mathcal{X}$ with constant L_i,*

$$\big\|\nabla f_i(\boldsymbol{x})-\nabla f_i(\boldsymbol{y})\big\|\le L_i\big\|\boldsymbol{x}-\boldsymbol{y}\big\|,\quad \forall\boldsymbol{x},\boldsymbol{y}\in\mathcal{X},\ \forall i\in[n].$$

We denote the vector of Lipschitz constants by $\boldsymbol{L}=(L_1,\dots,L_n)\in\mathbb{R}^n$.

Assumption 3.2. *Each component $f_i:\mathcal{X}\to\mathbb{R}$ is convex. Moreover, $F(\cdot,\boldsymbol{x})$ is monotone $\forall\boldsymbol{x}\in\mathcal{X}$. Thus, for any two vectors $\boldsymbol{u},\boldsymbol{v}\in\mathbb{R}^n$ such that $\boldsymbol{u}\le\boldsymbol{v}$ (component-wise), it holds that*

$$F(\boldsymbol{u},\boldsymbol{x}) \;\le\; F(\boldsymbol{v},\boldsymbol{x}). \tag{3.8}$$

A few comments are in order. Assumption 3.1, which is also required by Doikov and Nesterov [73], represents the formal manner in which we ask that F be simple — through convexity and bounded growth in $\boldsymbol{u}$. This assumption ensures convexity of subproblem (3.5), irrespective of the nature of $\boldsymbol{f}$.

Assumption 3.1.a is a generalization of the standard bounded curvature premise typical for Frank-Wolfe settings [112]. Requirement (3.7) is mild, as it only asks that the curvature of $\boldsymbol{f}$ remains bounded under F over $\mathcal{X}$. Importantly, the quantity $\mathcal{S}$ is *affine-invariant* (i.e., remains unchanged under affine reparametrizations of $\mathcal{X}$), which enables us to obtain convergence rates with the same property. Further discussion on the importance of affine-invariant analysis for FW algorithms is provided by Jaggi [112]. For mappings $\boldsymbol{f}$ that are twice differentiable, we can bound the quantity $\mathcal{S}$ from Assumption 3.1.a using Taylor's formula and the second derivatives as

$$\mathcal{S} \quad \leq \quad \sup_{\substack{\boldsymbol{x},\boldsymbol{y}\in\mathcal{X},\,\gamma\in[0,1]\\ \boldsymbol{y}_\gamma=\boldsymbol{x}+\gamma(\boldsymbol{y}-\boldsymbol{x})}} F(\nabla^2\boldsymbol{f}(\boldsymbol{y}_\gamma)[\boldsymbol{y}-\boldsymbol{x}]^2,\boldsymbol{x}).$$

This quantity is reminiscent of the quadratic upper bound used to analyze smooth optimization methods. In particular, for monotone non-decreasing F, a compact $\mathcal{X}$ and Lipschitz continuous ∇f_i with respect to a fixed norm $\|\cdot\|$, the assumption is satisfied with

$$\mathcal{S} \quad \leq \quad F(\boldsymbol{L}\mathcal{D}_{\mathcal{X}}^2) \quad \coloneqq \quad \sup_{\boldsymbol{x}\in\mathcal{X}} F(\boldsymbol{L}\mathcal{D}_{\mathcal{X}}^2,\boldsymbol{x}).$$

Assumption 3.1.b is standard and considered separately from Assumption 3.1.a to allow for different levels of generality in our results. The restriction to $\mathcal{X}$ makes this a locally Lipschitz gradient assumption on f_i.

Finally, Assumption 3.2 (also made by Doikov and Nesterov [73]) is required whenever we must ensure the overall convexity of $\varphi(\boldsymbol{x})$. The monotonicity of F is necessary in addition to the convexity of each f_i, since the composition of convex functions is not necessarily convex [23]. We rely on this assumption to prove faster convergence rates in the convex setting (Section 3.4.2).

To conclude this section, we provide the main application examples that fall under our fully composite template and which satisfy our assumptions.

Example 3.1. *Let* $F(\boldsymbol{u},\boldsymbol{x}) \equiv \max\limits_{1\leq i\leq n} u^{(i)}$. *Function F satisfies Assumptions 3.1 and 3.2 and problem* (3.4) *becomes*

$$\min_{\boldsymbol{x}\in\mathcal{X}} \max_{1\leq i\leq n} f_i(\boldsymbol{x}), \tag{3.9}$$

while oracle (3.5) *becomes*

$$\min_{\boldsymbol{x}\in\mathcal{X}} \max_{1\leq i\leq n} \langle \boldsymbol{a}_i,\boldsymbol{x}\rangle + b_i \qquad \Leftrightarrow \quad \min_{\boldsymbol{x}\in\mathcal{X},t\in\mathbb{R}} \big\{t : \langle \boldsymbol{a}_i,\boldsymbol{x}\rangle + b_i \leq t, 1\leq i\leq n\big\}. \tag{3.10}$$

Max-type minimization problems of this kind result from scalarization approaches to multi-

objective optimization, and their solutions are (weakly) Pareto optimal [Chapter 3.1 in 153]. As such, problem (3.9) *is relevant to a wide variety of applications requiring optimal trade-offs amongst several objective functions and appears in areas such as machine learning, science and engineering [see the introductory sections of, e.g., 67, 239]. Problem* (3.9) *also covers some instances of constrained ℓ_∞ regression.*

When $\mathcal{X}$ is a polyhedron, subproblem (3.10) *is efficiently solved via Linear Programming, while for general $\mathcal{X}$ one can resort to Interior-Point Methods [170]. Another option for solving* (3.10) *is to note that under strong duality [192] we have*

$$\min_{\boldsymbol{x}\in\mathcal{X}} \max_{1\le i\le n} \langle \boldsymbol{a}_i, \boldsymbol{x}\rangle + b_i \quad = \quad \min_{\boldsymbol{x}\in\mathcal{X}} \max_{\boldsymbol{\lambda}\in\Delta_n} \sum_{i=1}^{n} \lambda^{(i)} \big[\langle \boldsymbol{a}_i, \boldsymbol{x}\rangle + b_i\big] = \max_{\boldsymbol{\lambda}\in\Delta_n} g(\boldsymbol{\lambda}), \tag{3.11}$$

where $g(\boldsymbol{\lambda}) \coloneqq \min_{\boldsymbol{x}\in\mathcal{X}} \sum_{i=1}^{n} \lambda^{(i)} \big[\langle \boldsymbol{a}_i, \boldsymbol{x}\rangle + b_i\big]$. The maximization of g in (3.11) *can be done very efficiently for small values of n (with, e.g., the Ellipsoid Method or the Mirror Descent algorithm), since evaluating $g(\boldsymbol{\lambda})$ and $\partial g(\boldsymbol{\lambda})$ reduces to a vanilla LMO call over $\mathcal{X}$. An interesting case is $n = 2$, for which* (3.11) *becomes a* univariate *maximization problem and one may use binary search to solve it at the expense of a logarithmic number of LMOs.*

Example 3.2. *Let $F(\boldsymbol{u}, \boldsymbol{x}) \equiv \|\boldsymbol{u}\|$ for an arbitrary fixed norm $\|\cdot\|$. Function F satisfies Assumption 3.1 and problem* (3.4) *can be interpreted as solving a system of non-linear equations over $\mathcal{X}$*

$$\min_{\boldsymbol{x}\in\mathcal{X}} \|\boldsymbol{f}(\boldsymbol{x})\|, \tag{3.12}$$

while oracle (3.5) *amounts to solving the (constrained) linear system $\min_{\boldsymbol{x}\in\mathcal{X}} \|\boldsymbol{A}\boldsymbol{x}+\boldsymbol{b}\|$. Problems of this kind have applications such as robust phase retrieval [78] with phase constraints.*

The iterations of Algorithm 3.1 can be interpreted as a variant of the Gauss-Newton method [30, 169, 213], solving the (constrained) linear systems:

$$\boldsymbol{x}_{k+1} \in \arg\min_{\boldsymbol{x}\in\mathcal{X}} \|\boldsymbol{f}(\boldsymbol{y}_k) + \nabla \boldsymbol{f}(\boldsymbol{y}_k)(\boldsymbol{x} - \boldsymbol{y}_k)\|, \quad \textit{and} \quad \boldsymbol{y}_{k+1} = (1-\gamma_k)\boldsymbol{y}_k + \gamma_k \boldsymbol{x}_{k+1}. \tag{3.13}$$

In the particular case of solving systems of non-linear equations over compact convex sets, our algorithms can be seen as modified Gauss-Newton methods with global convergence guarantees.

3.4 Algorithms and convergence

3.4.1 The Basic Method

We describe our first approach to solving problem (3.4) in Algorithm 3.1. The central idea is to *linearize* the differentiable components of the objective and minimize the resulting model over $\mathcal{X}$, via calls to an oracle of type (3.5). The next iterate is defined as a convex combination with coefficient (or *stepsize*) γ between the computed minimizer and the preceding iterate.

Algorithm 3.1 Basic Method

Input: $\boldsymbol{y}_0 \in \mathcal{X}$

for $k = 0, 1, \ldots$ **do**

Compute $\boldsymbol{x}_{k+1} \in \operatorname{argmin}_{\boldsymbol{x}\in\mathcal{X}} F\big(\boldsymbol{f}(\boldsymbol{y}_k) + \nabla \boldsymbol{f}(\boldsymbol{y}_k)(\boldsymbol{x} - \boldsymbol{y}_k), \boldsymbol{x}\big)$

Choose $\gamma_k \in (0, 1]$ by a predefined rule or with line search

Set $\boldsymbol{y}_{k+1} = (1 - \gamma_k)\boldsymbol{y}_k + \gamma_k \boldsymbol{x}_{k+1}$

end for

A similar method for tackling problems of type (3.4) in the convex setting was proposed by Doikov and Nesterov [73]. Different from theirs, our method decouples the parameter γ_k from the minimization subproblem. This change is crucial since it allows us to choose the parameter γ_k *after* minimizing the model, thus enabling us to use efficient line search rules. Moreover, we provide Algorithm 3.1 with a more advanced *affine-invariant* analysis and establish its convergence in the *non-convex* setup.

We also mention that for solving problems of type (3.9), oracle (3.5) reduces to the minimization of a piecewise linear function over $\mathcal{X}$. Therefore, it has the same complexity as the modified LMOs of Kreimeier et al. [128].

Accuracy certificates. The standard *accuracy measure* of FW algorithms, which Algorithm 3.1 generalizes, is the Frank-Wolfe or Hearn gap [105]. For smooth objectives, it is defined as $\mathcal{G}_k \coloneqq \mathcal{G}(\boldsymbol{y}_k) = \max_{\boldsymbol{y}\in\mathcal{X}} \langle \nabla\varphi(\boldsymbol{y}_k), \boldsymbol{y}_k - \boldsymbol{y}\rangle$, for each iterate $\boldsymbol{y}_k$. This quantity is computed cost-free during the iterations and it upper bounds functional suboptimality in the convex case: $\mathcal{G}_k \geq \varphi(\boldsymbol{y}_k) - \varphi^\star$. Its semantics straightforwardly extend to non-convex settings, where it is zero if and only if $\boldsymbol{y}_k$ is a stationary point [130]. Additionally, convergence guarantees on the gap are desirable due to its affine invariance, which aligns with the affine invariance of the FW algorithm.

Our setting does not permit a direct generalization of the FW gap with all of the above properties. Rather, we introduce the following *accuracy certificate*, which is readily available in each iteration.

$$\mathcal{G}_k \;\coloneqq\; \mathcal{G}(\boldsymbol{y}_k) \;=\; \varphi(\boldsymbol{y}_k) - F\big(\boldsymbol{f}(\boldsymbol{y}_k) + \nabla \boldsymbol{f}(\boldsymbol{y}_k)(\boldsymbol{x}_{k+1} - \boldsymbol{y}_k), \boldsymbol{x}_{k+1}\big) \tag{3.14}$$

For minimization of a smooth (not necessarily convex) function, quantity (3.14) indeed reduces to the standard FW gap. Moreover, for convex $\varphi(\boldsymbol{x})$ (Assumption 3.2) we can conclude that

$$\begin{aligned} \mathcal{G}_k \;&\geq\; \max_{\boldsymbol{x}\in\mathcal{X}} \Big[\varphi(\boldsymbol{y}_k) - F\big(\boldsymbol{f}(\boldsymbol{y}_k) + \nabla \boldsymbol{f}(\boldsymbol{y}_k)(\boldsymbol{x} - \boldsymbol{y}_k), \boldsymbol{x}\big)\Big] \\ &\geq\; \max_{\boldsymbol{x}\in\mathcal{X}} \Big[\varphi(\boldsymbol{y}_k) - F(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{x})\Big] \;=\; \varphi(\boldsymbol{y}_k) - \varphi^\star. \end{aligned} \tag{3.15}$$

Hence, for a tolerance $\varepsilon > 0$, the criterion $\mathcal{G}_k \leq \varepsilon$ can be used as the stopping condition for our

method in convex scenarios. Moreover, the value of $\mathcal{G}_k$ can be used for computing the parameter γ_k through line search.

Convergence on convex problems. In the following, we prove the global convergence of Algorithm 3.1 in the case when $\varphi(\boldsymbol{x})$ is convex.

Theorem 3.1. *Let Assumptions 3.1, 3.1.a, and 3.2 be satisfied. Let* $\gamma_k \coloneqq \min\left\{1, \frac{\mathcal{G}_k}{\mathcal{S}}\right\}$ ***or*** $\gamma_k \coloneqq \frac{2}{2+k}$. *Then, for* $k \geq 1$ *it holds that*

$$\varphi(\boldsymbol{y}_k) - \varphi^\star \;\leq\; \frac{2\mathcal{S}}{1+k} \qquad \textit{and} \qquad \min_{1\leq i\leq k} \mathcal{G}_i \;\leq\; \frac{6\mathcal{S}}{k}. \tag{3.16}$$

The proof is provided in Appendix B.1.1. Our method recovers the rate of classical FW in the smooth case while being applicable to the wider class of *fully composite problems* (3.4). Thus, our $\mathcal{O}(1/k)$ rate improves upon the $\mathcal{O}(1/\sqrt{k})$ of black-box non-smooth optimization. Clearly, the improvement is achievable by leveraging the *structure of the objective* within the algorithm.

Convergence on non-convex problems. Under non-convexity, $\mathcal{G}_k$ defined in (3.14) no longer represents an accuracy certificate. This quantity is nevertheless important since it enables us to quantify the algorithm's progress while maintaining an affine-invariant analysis. Put differently, $\mathcal{G}_k$ has become merely a progress, rather than an accuracy, measure. The next theorem provides convergence guarantees on $\mathcal{G}_k$ for non-convex problems.

Theorem 3.2. *Let Assumptions 3.1 and 3.1.a be satisfied. Let* $\gamma_k \coloneqq \min\left\{1, \frac{\mathcal{G}_k}{\mathcal{S}}\right\}$ ***or*** $\gamma_k \coloneqq \frac{1}{\sqrt{1+k}}$. *Then, for all* $k \geq 1$ *it holds that*

$$\min_{0\leq i\leq k} \mathcal{G}_i \quad\leq\quad \frac{\varphi(\boldsymbol{y}_0) - \varphi^\star + 0.5\mathcal{S}(1+\ln(k+1))}{\sqrt{k+1}}. \tag{3.17}$$

The proof is given in Appendix B.1.2. Theorem 3.2 recovers a similar rate to the classical FW methods [130]. The line search rule for parameter γ_k makes our method universal, thereby allowing us to attain practically faster rates automatically when the iterates lie within a *convex region* of the objective.

As previously mentioned, the progress measure (3.14) no longer represents an accuracy certificate in the non-convex setting. Nevertheless, in some cases, we can still establish convergence of the linearization method with respect to meaningful quantities under non-convexity. Namely, let us consider problem (3.12) in Example 3.2 for the Euclidean norm, i.e., $F(\boldsymbol{u}, \boldsymbol{x}) = \|\boldsymbol{u}\|$, and the following simple iterations

$$\boldsymbol{y}_{k+1} \quad\in\quad \operatorname*{argmin}_{\boldsymbol{y}\in\boldsymbol{y}_k+\gamma_k(\mathcal{X}-\boldsymbol{y}_k)} \|\boldsymbol{f}(\boldsymbol{y}_k) + \nabla\boldsymbol{f}(\boldsymbol{y}_k)(\boldsymbol{y}-\boldsymbol{y}_k)\|. \tag{3.18}$$

Note that in (3.18), differently from (3.13), the value of γ_k is selected prior to the oracle call. Denoting the squared objective as $\Phi(\boldsymbol{x}) \coloneqq \frac{1}{2}\big[\varphi(\boldsymbol{x})\big]^2 = \frac{1}{2}\|\boldsymbol{f}(\boldsymbol{x})\|^2$ and following our analysis, we can state the convergence of process (3.18) in terms of the classical FW gap with respect to Φ. The proof is deferred to Appendix B.1.5.

Proposition 3.1. *Let $\gamma_k \coloneqq \frac{1}{\sqrt{1+k}}$. Then, for the iterations* (3.18)*, under Assumption 3.1.b and for all $k \geq 1$, it holds that*

$$\min_{0\leq i\leq k} \max_{\boldsymbol{y}\in\mathcal{X}} \langle \nabla\Phi(\boldsymbol{y}_i), \boldsymbol{y}_i - \boldsymbol{y}\rangle \quad \leq \quad \mathcal{O}\Big(\tfrac{\ln(k)}{\sqrt{k}}\Big).$$

We further show in Appendix B.2 that $\mathcal{G}_k$ can be related to the classical FW gap when our iterates lie in a smooth region of F. Whether we can provide a meaningful interpretation of $\mathcal{G}_k$ in the general non-convex case, however, remains an interesting open question.

3.4.2 The Accelerated Method

We now move away from the affine-invariant formulation of Algorithm 3.1 to a setting in which, by considering regularized minimization subproblems along with convexity and Lipschitz continuity of gradients, we can *accelerate* the Basic Method. We achieve acceleration by resorting to the well-known three-point scheme of Nesterov [163], in which the proximal subproblem is solved inexactly via calls to oracles of type (3.5). This approach was first analyzed in the context of FW methods by Lan and Zhou [135].

We propose Algorithm 3.2, which consists of a two-level scheme: an outer-loop computing the values of three iterates $\boldsymbol{y}$, $\boldsymbol{x}$ and $\boldsymbol{z}$ in $\mathcal{X}$, and a subsolver computing inexact solutions to the *proximal subproblem*

$$\operatorname*{arg\,min}_{\boldsymbol{u}\in\mathcal{X}} \Big\{ P(\boldsymbol{u}) \coloneqq F(\boldsymbol{f}(\boldsymbol{z}) + \nabla\boldsymbol{f}(\boldsymbol{z})(\boldsymbol{u}-\boldsymbol{z}), \boldsymbol{u}) + \frac{\beta}{2}\|\boldsymbol{u}-\boldsymbol{x}\|_2^2,\ \beta > 0 \Big\}. \tag{3.19}$$

The minimization in (3.19) does not conform to our oracle model (3.5) due to the quadratic regularizer. However, we can approximate its solution by iteratively solving subproblems in which we linearize the squared norm to match the template of (3.5). This procedure, denoted as InexactProx in Algorithm 3.2, returns a point $\boldsymbol{u}^+$ satisfying the optimality condition η*-inexactly* for some $\eta > 0$,

$$\begin{aligned} &F(\boldsymbol{f}(\boldsymbol{z}) + \nabla f(\boldsymbol{z})(\boldsymbol{u}^+ - \boldsymbol{z}), \boldsymbol{u}^+) + \beta\langle \boldsymbol{u}^+ - \boldsymbol{x}, \boldsymbol{u}^+\rangle \\ &\qquad \leq F(\boldsymbol{f}(\boldsymbol{z}) + \nabla f(\boldsymbol{z})(\boldsymbol{u}-\boldsymbol{z}), \boldsymbol{u}) + \beta\langle \boldsymbol{u}^+ - \boldsymbol{x}, \boldsymbol{u}\rangle + \eta, \qquad \forall\, \boldsymbol{u}\in\mathcal{X}. \end{aligned} \tag{3.20}$$

Note that condition (3.20) implies $P(\boldsymbol{u}^+) \leq P(\boldsymbol{u}) + \eta$, $\forall\, \boldsymbol{u}\in\mathcal{X}$. Formally, the main convergence result characterizing Algorithm 3.2 is the following.

Theorem 3.3. *Let Assumptions 3.1, 3.1.b, and 3.2 be satisfied. We choose $\gamma_k \coloneqq \frac{3}{k+3}$, $\beta_k \coloneqq$*

Algorithm 3.2 Accelerated Method

Input: $\boldsymbol{y}_0 \in \mathcal{X}$, set $\boldsymbol{x}_0 = \boldsymbol{y}_0$

for $k = 0, 1, \ldots$ **do**

Choose $\gamma_k \in (0, 1]$

Set $\boldsymbol{z}_{k+1} = (1 - \gamma_k)\boldsymbol{y}_k + \gamma_k \boldsymbol{x}_k$

Compute $\boldsymbol{x}_{k+1} = \text{InexactProx}(\boldsymbol{x}_k, \boldsymbol{z}_{k+1}, \beta_k, \eta_k)$ for some $\beta_k \geq 0$ and $\eta_k \geq 0$

Set $\boldsymbol{y}_{k+1} = (1 - \gamma_k)\boldsymbol{y}_k + \gamma_k \boldsymbol{x}_{k+1}$

end for

$cF(\boldsymbol{L})\gamma_k$ and $\eta_k \coloneqq \frac{\delta}{3(k+1)(k+2)}$ where $\delta > 0$ and $c \geq 0$ are chosen constants, and $F(\boldsymbol{L}) \coloneqq \sup_{\boldsymbol{x} \in \mathcal{X}} F(\boldsymbol{L}, \boldsymbol{x})$. Then, for all $k \geq 1$ it holds that

$$\varphi(\boldsymbol{y}_k) - \varphi^\star \quad \leq \quad \frac{\delta + 8cF(\boldsymbol{L})\mathcal{D}_{\mathcal{X}}^2}{(k+2)(k+3)} + \frac{2\max\{0, 1-c\}F(\boldsymbol{L})\mathcal{D}_{\mathcal{X}}^2}{k+3}.$$

The proof of Theorem 3.3 (deferred to Appendix B.1.3) comes from a natural sequence of steps involving the properties of the operators and the approximate optimality of $\boldsymbol{x}_{k+1}$. The crucial step in attaining the improved convergence is the choice of parameters γ_k, β_k and η_k. Notably, the decay speed required of η_k is quadratic, meaning that the subproblems are solved with fast-increasing accuracy and at the cost of additional time spent in the subsolver. The constant δ allows us to fine-tune the accuracy required for the first several iterations of the algorithm, where we can accept a lower precision. In practice, we can always choose $\delta = 1$ as a universal rule, and the optimal choice is $\delta = F(\boldsymbol{L})\mathcal{D}_{\mathcal{X}}^2$ when these parameters are known. The factor $cF(\boldsymbol{L})$ in the definition of β_k can be interpreted as the quality of the approximation of the Lipschitz constant for our problem. Namely, it is exactly computed for $c = 1$ and over or underestimated for $c > 1$ and $c \in (0, 1)$, respectively.

We describe each of the bounding terms independently: the first is highly reminiscent of the usual bounds accompanying FW-type algorithms in terms of constants, albeit now with quadratic decay speed. The second term indicates the behaviour of the algorithm as a function of c: overestimation of $F(\boldsymbol{L})$ ensures quadratic rates of convergence since the second term becomes negative. Conversely, underestimation of $F(\boldsymbol{L})$ brings us back into the familiar FW convergence regime of $\mathcal{O}(1/k)$ as the second term becomes positive. The extreme case $c = 0$ (and hence $\beta_k = 0$) essentially reduces Algorithm 3.2 to Algorithm 3.1, since the projection subproblem reduces to problem (3.5) which is easily solvable by assumption. We, therefore, have robustness in terms of choosing the parameter c, and the exact knowledge of $F(\boldsymbol{L})$ is not needed, even though it may come at the cost of slower convergence. In contrast, classical Fast Gradient methods are usually very sensitive to such parameter choices [70].

Theorem 3.3 provides an accelerated rate on the iterates $\boldsymbol{y}_k$ — an analogous result to that of Lan and Zhou [135] albeit under a different oracle. This convergence rate is conditioned on

the subsolver returning an η_k-inexact solution to the projection subproblem and therefore any subsolver satisfying the condition can achieve this rate. As with any optimization algorithm, convergence guarantees may also be stated in terms of the oracle complexity required to reach ϵ accuracy. For Algorithm 3.2 all the oracle calls are deferred to the subsolver InexactProx, which we describe and analyze in the next section.

3.4.3 Solving the proximal subproblem

We now provide an instance of the InexactProx subsolver which fully determines the oracle complexity of the Accelerated Method (Algorithm 3.2). It relies on a specific adaptation of Algorithm 3.1 to the structure of (3.19). The quadratic regularizer is linearized and oracles of type (3.5) are called once per inner iteration, while the Jacobian $\nabla \boldsymbol{f}(\boldsymbol{z}_k)$ is computed once per subsolver call. The main challenge here is to find a readily available quantity dictating the exit condition of the subsolver, which we denote by $\mathcal{G}_t$.

Algorithm 3.3 InexactProx($\boldsymbol{x}$, $\boldsymbol{z}$, β, η)

Initialization: $\boldsymbol{u}_0 = \boldsymbol{x}$.

for $t = 0, 1, \ldots$ **do**

Compute $\boldsymbol{v}_{t+1} \in \quad \arg\min_{\boldsymbol{v}\in\mathcal{X}} \Big\{ F\big(\boldsymbol{f}(\boldsymbol{z}) + \nabla \boldsymbol{f}(\boldsymbol{z})(\boldsymbol{v} - \boldsymbol{z}), \boldsymbol{v}\big) + \beta\langle \boldsymbol{u}_t - \boldsymbol{x}, \boldsymbol{v}\rangle \Big\}$

Compute $\mathcal{G}_t \;=\; F\big(\boldsymbol{f}(\boldsymbol{z}) + \nabla \boldsymbol{f}(\boldsymbol{z})(\boldsymbol{u}_t - \boldsymbol{z}), \boldsymbol{u}_t\big) \;-\; F\big(\boldsymbol{f}(\boldsymbol{z}) + \nabla \boldsymbol{f}(\boldsymbol{z})(\boldsymbol{v}_{t+1} - \boldsymbol{z}), \boldsymbol{v}_{t+1}\big)$
$\qquad + \;\beta\langle \boldsymbol{u}_t - \boldsymbol{x}, \boldsymbol{u}_t - \boldsymbol{v}_{t+1}\rangle$

if $\mathcal{G}_t \le \eta$ **then return** $\boldsymbol{u}_t$

Set $\alpha_t = \min\Big\{1, \frac{\mathcal{G}_t}{\beta\|\boldsymbol{v}_{t+1} - \boldsymbol{u}_t\|_2^2}\Big\}$ and $\boldsymbol{u}_{t+1} = \alpha_t \boldsymbol{v}_{t+1} + (1 - \alpha_t)\boldsymbol{u}_t$

end for

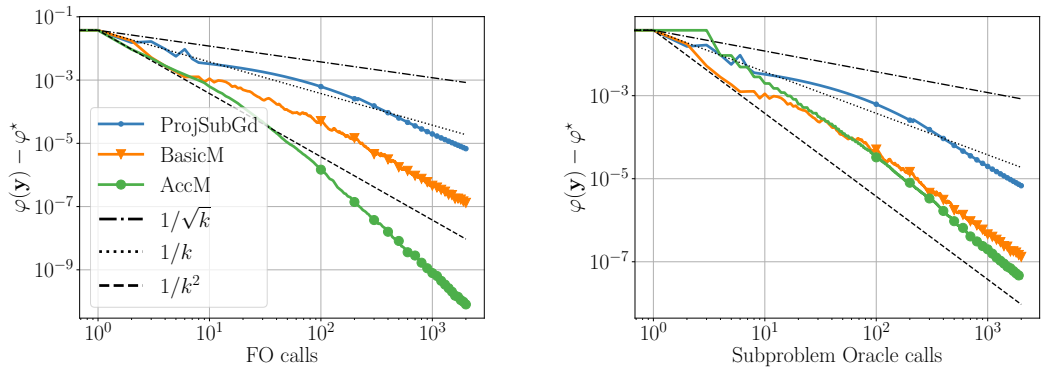
The parameters of Algorithm 3.3 are fully specified, and the stopping condition depends on $\mathcal{G}_t \ge P(\boldsymbol{u}_t) - P^\star$, which is a meaningful progress measure. The algorithm selects its stepsize via closed-form line search to improve practical performance. When $F(\boldsymbol{u}) \equiv \boldsymbol{u}^{(1)}$, this procedure recovers the classical FW algorithm with line search applied to problem (3.19).

We prove two results in relation to Algorithm 3.3: its convergence rate and the total oracle complexity of Algorithm 3.2 when using Algorithm 3.3 as the subsolver. The rate and analysis are similar to the ones of the Basic Method, up to using properties specific to problem (3.19).

Theorem 3.4. *Let Assumptions 3.1, 3.1.b, and 3.2 be satisfied. Then, for all $t \ge 1$ it holds that*

$$P(\boldsymbol{u}_t) - P^\star \;\le\; \frac{2\beta\mathcal{D}_{\mathcal{X}}^2}{t+1} \qquad \textit{and} \qquad \min_{1\le i\le t} \mathcal{G}_t \;\le\; \frac{6\beta\mathcal{D}_{\mathcal{X}}^2}{t}.$$

Consequently, Algorithm 3.3 returns an η-approximate solution according to condition (3.20) *after at most $\mathcal{O}\Big(\frac{\beta\mathcal{D}_{\mathcal{X}}^2}{\eta}\Big)$ iterations.*



(a) (b)

Figure 3.1: Convergence of the Basic and Accelerated methods against the Projected Subgradient baseline for problem (3.21), along with relevant theoretical rates.

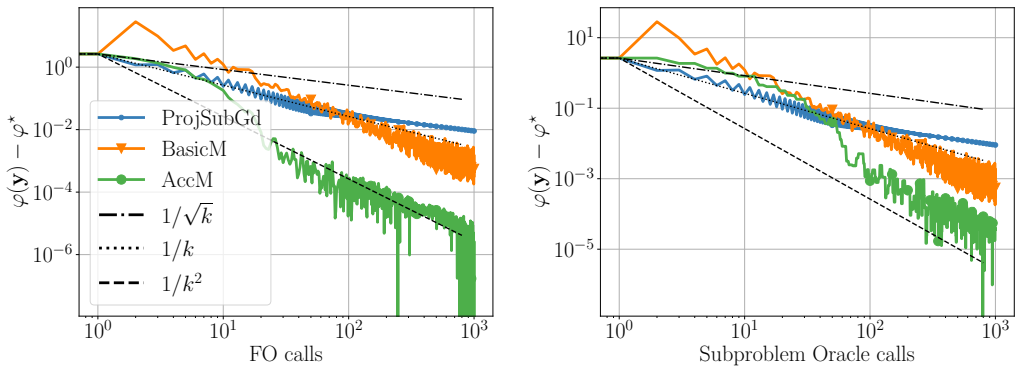
The proof of this result is deferred to Appendix B.1.4. We note that oracle (3.5) is called once per inner iteration, and the Jacobian $\nabla \boldsymbol{f}(\boldsymbol{z}_k)$ is computed once per subsolver call. In particular, when using Algorithm 3.3 as a subsolver, our Accelerated Method achieves the optimal number of $\mathcal{O}\left(\epsilon^{-1/2}\right)$ Jacobian computations typical of smooth and convex optimization, while maintaining a $\mathcal{O}\left(\epsilon^{-1}\right)$ complexity for the number of calls to oracle (3.5). The results are stated in the following corollary.

Corollary 3.1. *Consider the optimal choice of parameters for Algorithm 3.2, that is* $c \coloneqq 1$ *and* $\delta \coloneqq F(\boldsymbol{L})\mathcal{D}_{\mathcal{X}}^2$. *Then, solving problem* (3.4) *with* ε *accuracy* $\varphi(\boldsymbol{y}_k) - \varphi^\star \leq \varepsilon$, *requires* $\mathcal{O}\left(\sqrt{\frac{F(\boldsymbol{L})\mathcal{D}_{\mathcal{X}}^2}{\varepsilon}}\right)$ *computations of* $\nabla \boldsymbol{f}$. *In addition, the total number of calls to oracle* (3.5) *is* $\mathcal{O}\left(\frac{F(\boldsymbol{L})\mathcal{D}_{\mathcal{X}}^2}{\varepsilon}\right)$.

Finally, we note that for smaller values of parameter $c \in [0, 1]$ in Algorithm 3.2 (underestimating the Lipschitz constant), the complexity of InexactProx procedure improves. Thus, for $c = 0$ we have $\beta = 0$ (no regularization) and Algorithm 3.3 finishes after just *one step*.

3.5 Experiments

The experiments are implemented in Python 3.9 and run on a MacBook Pro M1 with 16 GB RAM. For both experiments we use the Projected Subgradient method as a baseline [205], with a stepsize of $\frac{p}{\sqrt{k}}$ where p is tuned for each experiment. We use the CVXPY library [72] to solve subproblems of type (3.5). The random seed for our experiments is always set to 666013 (an interesting prime number), and we set $c = 1$ since we can analytically compute the Lipschitz constants or their upper bounds.



(a) (b)

Figure 3.2: Convergence of the Basic and Accelerated methods against the Projected Subgradient baseline for problem (3.22), along with relevant theoretical rates.

3.5.1 Max-type minimization over the simplex

We consider the following optimization problem

$$\min_{\boldsymbol{x}\in\mathcal{X}}\left\{\max_{i\in[n]}\ \boldsymbol{x}^\top \boldsymbol{A}_i\boldsymbol{x}-\boldsymbol{b}_i^\top\boldsymbol{x}\right\},\ \text{for } \mathcal{X}=\Delta_d, \tag{3.21}$$

where $\boldsymbol{A}_i\in\mathbb{R}^{d\times d}$ are random PSD matrices and $\boldsymbol{b}_i\in\mathbb{R}^d$. The problem conforms to Example 3.1, and we use $d=500$ and $n=10$. We generate $\boldsymbol{A}_i=\boldsymbol{Q}_i\boldsymbol{D}\boldsymbol{Q}_i^\top$, where $\boldsymbol{D}$ is a diagonal matrix of eigenvalues decaying linearly in the interval [1e−6, 1e0], and $\boldsymbol{Q}_i$ is a randomly generated orthogonal matrix using the method `scipy.stats.ortho_group` [152]. The vectors $\boldsymbol{b}_i$, which determine the position of the quadratics in space, are set as follows: $\boldsymbol{b}_i=10\cdot\boldsymbol{e}_i$, $\boldsymbol{b}_9=\boldsymbol{0}_d$ (the origin), $\boldsymbol{b}_{10}=10\cdot\boldsymbol{1}_d$. We set $\delta=0.2$ in the Accelerated Method (see Theorem 3.3) and settle for $p=1.42$ following tuning of the Subgradient Method. Finally, we set $\boldsymbol{x}_0=\boldsymbol{e}_3\in\Delta_d$ for all methods.

The convergence results in terms of FO oracles and oracles of type (3.5) are shown in Figure 3.1a and 3.1b, respectively. The figures highlight the improvement in terms of the number of FO calls, while showing comparable performance in terms of subproblem oracle calls, as predicted by our theory.

3.5.2 Max-type minimization over the nuclear norm ball

We consider the following optimization problem

$$\min_{\boldsymbol{X}\in\mathcal{X}}\left\{\max_{i\in[n]}\sum_{(k,l)\in\Omega_i}\left(\boldsymbol{X}_{k,l}-\boldsymbol{A}^{(i)}_{k,l}\right)^2\right\},\ \text{for } \mathcal{X}\coloneqq\{\boldsymbol{X}\in\mathbb{R}^{d\times m},\ \|\boldsymbol{X}\|_*\le r\} \tag{3.22}$$

Formulation (3.22) models a matrix completion scenario where we wish to recover an $\boldsymbol{X}^\star$ that minimizes the largest error within a given *set* of matrices $\boldsymbol{A}^{(i)}$. The matrices $\boldsymbol{A}^{(i)}$ are only partially revealed through a set of corresponding indices Ω_i. This problem conforms to Example 3.1 and we use $d = 30$, $m = 10$, $r = 7$, where r is the rank of matrices $\boldsymbol{A}^{(i)}$. The data is generated in an identical fashion to Section 5.2 of Lan and Zhou [135] on Matrix Completion. We set $\delta = 100$ in the Accelerated Method (see Theorem 3.3) and settle for $p = 0.2$ following tuning of the Projected Subgradient method. Finally, we set $\boldsymbol{x}_0 = \mathbf{0}_{d\times m} \in \mathcal{X}$ for all methods.

The convergence results in terms of FO calls and oracles of type (3.5) are shown in Figure 3.2a and 3.2b, respectively. The figures highlight the improvement in terms of the number of FO calls, while showing comparable performance in terms of subproblem oracle calls, as predicted by our theory.

3.6 Conclusion

This chapter introduced generalizations of the vanilla Frank-Wolfe [84, 112] and Conditional Gradient Sliding algorithms [135] for a class of non-differentiable composite objectives, to which the aforementioned methods do not straightforwardly extend. We showed how leveraging the problem structure eschews the stringent lower bounds of optimizing black-box non-differentiable objectives, to achieve convergence rates that are on par with the smooth setting. Moreover, we showed how the principle of exclusively linearizing the differentiable components of a composition gives rise to subproblems that can be efficiently solved in some cases of interest. Finally, we illustrated the practical performance of our algorithms against the Projected Subgradient method [204] on matrix recovery problems, showing an improved convergence in terms of the number of oracle calls.

Interesting future work may address relaxing the assumptions on the outer mapping F, extending this framework to stochastic settings, and meaningfully interpreting the quantity $\mathcal{G}_k$ for non-convex problems.

4 An adaptive, linesearch-free primal-dual algorithm

This chapter is based on the published work Vladarean, Malitsky, and Cevher [221], presented at NeurIPS 2021.

Co-authors: Yura Malitsky, Volkan Cevher

Contributions

M. Vladarean	— methodology 30%, formal derivations 100%, writing 100%, experiments 100%
Y. Malitsky	— methodology 70%, writing – review and editing, supervision
V. Cevher	— project administration, supervision

Summary We consider the problem of finding a saddle point for the convex-concave objective $\min_{\boldsymbol{x}} \max_{\boldsymbol{y}} f(\boldsymbol{x}) + \langle \boldsymbol{A}\boldsymbol{x}, \boldsymbol{y}\rangle - g^*(\boldsymbol{y})$, where f is a convex function with locally Lipschitz gradient and g is a convex and possibly non-smooth function. We propose an adaptive version of the Condat-Vũ algorithm, which alternates between primal gradient steps and dual proximal steps. The method achieves stepsize adaptivity through a simple rule involving the norm of recently computed gradients of f and $\|\boldsymbol{A}\|$. Under the aforementioned assumptions, we prove the asymptotic convergence of iterates to a saddle point and an $\mathcal{O}(k^{-1})$ ergodic convergence rate for the primal-dual gap. Furthermore, when f is additionally locally strongly convex and $\boldsymbol{A}$ has full row rank, we show that our method converges with a linear rate. We provide numerical experiments illustrating the practical performance of our algorithm against relevant baselines.

4.1 Introduction

Consider the following composite minimization problem

$$\min_{\boldsymbol{x}\in\mathcal{X}} f(\boldsymbol{x}) + g(\boldsymbol{A}\boldsymbol{x}), \tag{4.1}$$

where $\mathcal{X}$ is a finite-dimensional real vector space, f and g are convex, proper and lower-semicontinuous (l.s.c.), and $\boldsymbol{A}$ is a given matrix (or linear operator). This template is highly versatile, encompassing a wide variety of regularized problems (including those with structured regularization), as well as constrained minimization (whenever g is the indicator function of a convex set).

Problems of the form (4.1) have been studied in the literature under various assumptions on f and g. For the particular instances where $g \circ \boldsymbol{A}$ is proximal-friendly[1] and f is L-smooth, the objective is suitable for applying forward-backward splitting algorithms like the Proximal Gradient algorithm and its accelerated counterpart [164, 14]. In general, however, the proximal operator of $g \circ \boldsymbol{A}$ is not easily computable and, in such cases, a popular approach is to decouple $\boldsymbol{A}$ and g by reformulating (4.1) as the convex-concave saddle-point problem

$$\min_{\boldsymbol{x}\in\mathcal{X}} \max_{\boldsymbol{y}\in\mathcal{Y}} \langle \boldsymbol{A}\boldsymbol{x}, \boldsymbol{y}\rangle + f(\boldsymbol{x}) - g^*(\boldsymbol{y}), \tag{4.2}$$

where $\mathcal{X}, \mathcal{Y}$ are finite-dimensional real vector spaces, g^* denotes the Fenchel conjugate of g. Objective (4.2) is typically addressed by primal-dual splitting algorithms which, under strong duality, can recover the solution to the original problem (4.1). In the particular case when f and g are proximal-friendly and possibly non-smooth, a very popular method is the Primal-Dual Hybrid Gradient proposed by Chambolle and Pock [42], which was further extended to handle an additional L-smooth component in the Condat-Vũ algorithm [55, 222]. Convergence rates for the latter are studied by Chambolle and Pock [44].

Together, these classes of algorithms cover a broad range of problems in diverse fields such as signal processing, machine learning, inverse problems, telecommunications and many others. As a result, a great amount of research effort has gone into addressing practical concerns such as robustness to inexact oracles, acceleration and automation of stepsize selection. For a comprehensive list of examples and theoretical details, we refer the reader to the review papers of Combettes and Pesquet [54], Parikh, Boyd, et al. [178], Komodakis and Pesquet [127], and Chambolle and Pock [43]. The work presented in this chapter falls in the latter category of stepsize regime automation, which we study in the context of primal-dual algorithms for problem (4.2).

In their basic form, primal-dual methods require as input stepsize parameters belonging to a designated interval of stability, which depends on problem-specific constants like the global smoothness parameter L and $\|\boldsymbol{A}\|$. Dependence on such constants is undesirable because they

[1] We say that h is 'proximal-friendly' if $\operatorname{prox}_h(\boldsymbol{x})$ defined in (1.15) has a closed-form solution or can be efficiently computed to high accuracy.

may be costly to compute and oftentimes one can only access upper-bound estimates, thus leading to overly-conservative stepsizes and slower convergence. Moreover, the need to know L for setting the stepsizes prevents these methods from being applied to functions which are not globally L-smooth.

Consequently, recent efforts have gone towards devising methods with adaptive stepsizes [99, 98, 149, 181]. These approaches resort to linesearch for finding good stepsizes at every iteration and come with improved empirical convergence. This practical advantage, however, comes at the cost of an indeterminate number of extra steps (usually cheap) spent in the linesearch subprocedures.

In this chapter, we study problem (4.2) under the assumption that ∇f is locally Lipschitz continuous and g is proximal-friendly. To illustrate the motivation of our framework, we take a prototypical example in image processing,

$$\min_{\boldsymbol{X}\in\mathbb{R}^{m\times d}} \frac{1}{2}\|\boldsymbol{K}\boldsymbol{X}-\boldsymbol{B}\|_F^2+\lambda\|\boldsymbol{D}\boldsymbol{X}\|_{2,1},\ \boldsymbol{K}:\mathbb{R}^{m\times d}\to\mathbb{R}^{l\times p},\ \boldsymbol{D}:\mathbb{R}^{m\times d}\to\mathbb{R}^{m\times d\times 2},$$

where $\boldsymbol{X}$ is an image; $\boldsymbol{K}$ is a problem-specific linear measurement operator; $\boldsymbol{B}$ is the dimension-appropriate and possibly noisy observation; $\boldsymbol{D}$ is the discrete gradient operator, and the overall regularization term represents the isotropic TV norm. In order to apply any of the aforementioned primal-dual algorithms, one needs to first choose how to decouple the linear operators. There are three options: decoupling with respect to $\boldsymbol{K}$ leaves us with having to compute the proximal operator of the TV norm for the primal step, which is an iterative procedure [40]. Decoupling $\boldsymbol{D}$ implies performing gradient steps on f since, in general, its proximal operator is not efficient. Finally, decoupling with respect to both implies increasing the dimensionality of the dual variable to $lp+2md$, which is problematic whenever these dimensions are large. The sensible choice is the second one (i.e., decoupling $\boldsymbol{D}$), and the question we ask is

> *Does there exist a method for solving* (4.2) *that adapts to the local problem geometry without resorting to linesearch?*

Our contribution is to propose a first-order primal-dual scheme that answers this question in the affirmative and is accompanied by theoretical convergence guarantees. Using standard analysis techniques, we show an ergodic convergence of $\mathcal{O}\left(k^{-1}\right)$ when ∇f is *locally* Lipschitz and g is proximal-friendly, and a linear convergence rate for the case when f is additionally *locally* strongly convex and $\boldsymbol{A}$ has full row rank. We provide numerical experiments for sparse logistic regression and image inpainting, and further test our method as a heuristic for TV-regularized non-convex phase retrieval.

The rest of the chapter is structured as follows: Section 4.2 provides details about related work; Section 4.3 introduces notation, along with technical preliminaries and assumptions to be used in our analysis; Section 4.4 reports the main theoretical results alongside partial proofs; finally, numerical results are provided in Section 4.5.

4.2 Related work

Adaptive Gradient Descent (GD) methods. Arguably the most widespread of optimization methods, GD presents similar shortcomings for setting the stepsize as those described in the previous section. In particular, much research effort has gone into devising variants of the algorithm that remove the need to estimate the global smoothness constant L. In recent work, Malitsky and Mishchenko [147] propose an extremely simple and effective alternative for setting the stepsize τ_k adaptively at every iteration, as

$$\tau_k = \min\left\{\tau_{k-1}\sqrt{1+\frac{\tau_{k-1}}{\tau_{k-2}}},\ \frac{\|\, x_k - x_{k-1}\,\|}{2\left\|\,\nabla f(x_k) - \nabla f(x_{k-1})\,\right\|}\right\}. \tag{4.3}$$

Adaptivity essentially comes "for free" in (4.3), as it involves solely quantities which have already been computed. Moreover, their method requires only the weaker assumption of local smoothness, thus extending the reach of provably convergent GD to a wider class of differentiable functions while maintaining the standard $\mathcal{O}\left(k^{-1}\right)$ convergence rate.

In this chapter, we show that the above technique can be extended to the analysis of primal-dual methods, where it gives rise to an algorithm whose stepsize adapts to the local geometry of the objective's (locally) smooth component f.

Adaptive monotone variational inequality (VI) methods. Malitsky [146] proposes an algorithm for solving monotone VIs with a stepsize that adapts to local smoothness similarly to (4.3). This method solves the very general formulation of finding $\boldsymbol{u}^\star$ such that $\langle F(\boldsymbol{u}^\star), \boldsymbol{u} - \boldsymbol{u}^\star\rangle + h(\boldsymbol{u}) - h(\boldsymbol{u}^\star) \geq 0, \forall \boldsymbol{u}$ for a given monotone operator F which is locally Lipschitz continuous. Our template (4.2) can be recovered from theirs by setting $\boldsymbol{u} = (\boldsymbol{x}, \boldsymbol{y})$, with

$$F(\boldsymbol{u}) = F(\boldsymbol{x}, \boldsymbol{y}) = \begin{bmatrix} \nabla f(\boldsymbol{x}) + \boldsymbol{A}^\top \boldsymbol{y} \\ -\boldsymbol{A}\boldsymbol{x} \end{bmatrix},$$

and $h(\boldsymbol{u}) = g^*(\boldsymbol{y})$. The advantages of this approach are the relaxed requirement of local Lipschitz continuity for F and the fact that knowledge of $\|\boldsymbol{A}\|$ is not required. However, since the VI framework is very general and does not take advantage of the problem structure (e.g. the fact that $\langle \boldsymbol{A}\boldsymbol{x}, \boldsymbol{y}\rangle$ is a bilinear term), the method comes with worse convergence bounds than algorithms specifically designed to solve (4.2). In addition, the algorithm requires as input an upper bound on the stepsizes, despite them being set in accordance to the estimated local smoothness.

First order primal-dual algorithms and adaptive versions. A popular method for solving (4.2) when f is L-smooth is the Condat-Vũ algorithm (CVA) [55, 222]. The method's convergence is subject to a global stepsize validity condition given by $\left(\frac{1}{\tau} - L\right)\frac{1}{\sigma} \geq \|\boldsymbol{A}\|^2$, where τ and σ are the primal and dual stepsizes, respectively.

Another approach to solving problem (4.2) is via the Primal–Dual Fixed-Point algorithm based on the Proximity Operator (PDFP^{2}O) or the Proximal Alternating Predictor–Corrector (PAPC) methods [144, 47, 75]. This approach comes with less restrictive stepsize conditions than CVA owing to a different iteration style, but which nevertheless depend on the global smoothness constant L and $\|\boldsymbol{A}\|$ and have to be carefully chosen.

In order to alleviate the burden of choosing the stepsize parameters in CVA, Malitsky and Pock [149] propose a linesearch procedure involving only dual variable updates and which, for certain problems such as regularized least squares, does not require any additional matrix-vector multiplications. A characteristic of this algorithm is that it maintains a constant ratio between primal and dual stepsizes through a hyperparameter β — a setup which we also use here.

4.3 Preliminaries

Consider problem (4.2) and let $\mathcal{X}, \mathcal{Y}$ be finite-dimensional real vector spaces. We denote by g^* the Fenchel conjugate of g in (4.1) defined as $g^*(\boldsymbol{y}) \coloneqq \sup_{\boldsymbol{x}}\{\langle \boldsymbol{x}, \boldsymbol{y}\rangle - g(\boldsymbol{x})\}$.

One can easily see that (4.2) is a primal-dual formulation of the following primal and dual optimization problems, of which the former is the same as (4.1).

$$\min_{\boldsymbol{x}\in\mathcal{X}} f(\boldsymbol{x}) + g(\boldsymbol{A}\boldsymbol{x}), \qquad \max_{\boldsymbol{y}\in\mathcal{Y}} -(f^*(-\boldsymbol{A}^\top \boldsymbol{y}) + g^*(\boldsymbol{y}))$$

A saddle-point $(\boldsymbol{x}^\star, \boldsymbol{y}^\star) \in \mathcal{X} \times \mathcal{Y}$ of problem (4.2) satisfies the following optimality conditions.

$$-\boldsymbol{A}^\top \boldsymbol{y}^\star = \nabla f(\boldsymbol{x}^\star), \qquad \boldsymbol{A}\boldsymbol{x}^\star \in \partial g^*(\boldsymbol{y}^\star) \tag{4.4}$$

For $(\boldsymbol{x}', \boldsymbol{y}') \in \mathcal{X} \times \mathcal{Y}$ we define the following quantities

$$\begin{aligned} P_{\boldsymbol{x}',\boldsymbol{y}'}(\boldsymbol{x}) &\coloneqq f(\boldsymbol{x}) - f(\boldsymbol{x}') + \langle \boldsymbol{x} - \boldsymbol{x}', \boldsymbol{A}^\top \boldsymbol{y}'\rangle, \\ D_{\boldsymbol{x}',\boldsymbol{y}'}(\boldsymbol{y}) &\coloneqq g^*(\boldsymbol{y}) - g^*(\boldsymbol{y}') - \langle \boldsymbol{A}\boldsymbol{x}', \boldsymbol{y} - \boldsymbol{y}'\rangle, \\ \mathcal{G}_{\boldsymbol{x}',\boldsymbol{y}'}(\boldsymbol{x}, \boldsymbol{y}) &\coloneqq P_{\boldsymbol{x}',\boldsymbol{y}'}(\boldsymbol{x}) + D_{\boldsymbol{x}',\boldsymbol{y}'}(\boldsymbol{y}), \end{aligned}$$

which are all convex for fixed $(\boldsymbol{x}', \boldsymbol{y}')$. Whenever $(\boldsymbol{x}', \boldsymbol{y}') = (\boldsymbol{x}^\star, \boldsymbol{y}^\star)$, it holds that $P_{\boldsymbol{x}^\star,\boldsymbol{y}^\star}(\boldsymbol{x}) \geq 0$, $D_{\boldsymbol{x}^\star,\boldsymbol{y}^\star}(\boldsymbol{y}) \geq 0$ and $\mathcal{G}_{\boldsymbol{x}^\star,\boldsymbol{y}^\star}(\boldsymbol{x}, \boldsymbol{y}) \geq 0$, with the latter quantity representing the primal-dual gap. We also define the gap restricted to a bounded subset $B_1 \times B_2 \subset \mathcal{X} \times \mathcal{Y}$ as

$$\mathcal{G}_{B_1\times B_2}(\boldsymbol{x}, \boldsymbol{y}) \coloneqq \sup_{(\boldsymbol{x}',\boldsymbol{y}')\in B_1\times B_2} P_{\boldsymbol{x}',\boldsymbol{y}'}(\boldsymbol{x}) + D_{\boldsymbol{x}',\boldsymbol{y}'}(\boldsymbol{y}),$$

and note that it is non-negative whenever $B_1 \times B_2$ contains a saddle-point.

Finally, the following two blanket assumptions hold throughout the chapter.

Assumption 4.1. *Function f is convex and locally smooth, while g is convex, l.s.c., and proximal-*

friendly.

Assumption 4.2. *A saddle-point exists for problem* (4.2) *and thus strong duality holds.*

We note that Assumption 4.2 is standard in the literature (see e.g., [42]). Assumption 4.1, on the other hand, is weaker than the usual global L-smoothness premise and thus enlarges the category of admissible functions f with instances such as $x \mapsto \exp(x)$. To illustrate, consider the aforementioned function defined on the reals: the global smoothness assumption clearly does not hold, however for any fixed interval $[a, b] \subset \mathbb{R}$ the smoothness constant can be chosen as $\exp(b)$.

To prove linear convergence for our method, we will invoke a third assumption.

Assumption 4.3. *Function f is locally strongly convex, and $\boldsymbol{A}$ has full row-rank.*

4.4 Algorithm and convergence

Our method for solving problem (4.2) is given in Algorithm 4.1 under the abbreviation APDA, which we use from here onwards. APDA follows the same structure as the basic CVA [44]. Notice that if we restrict Assumption 4.1 to L-smooth functions f, we can in fact recover CVA by setting $\theta_k = \theta = 1$ and $\tau_k = \tau$, $\sigma_k = \sigma$ fixed such that $\left(\frac{1}{\tau} - L\right)\frac{1}{\sigma} \geq \|\boldsymbol{A}\|^2$.

Algorithm 4.1 Adaptive Primal-Dual Algorithm (APDA)

Input: $\boldsymbol{x}_0 \in \mathcal{X}$, $\boldsymbol{y}_0 \in \mathcal{Y}$, $\tau_{\text{init}} > 0$, $\tau_0 = \infty$, $\theta_0 = 1$, $\beta > 0$, $c \in (0, 1)$

$\boldsymbol{x}_1 = \boldsymbol{x}_0 - \tau_{\text{init}}(\nabla f(\boldsymbol{x}_0) + \boldsymbol{A}^\top \boldsymbol{y}_0)$

for $k = 1, 2, \ldots$ **do**

Set $\tau_k = \min\left\{\frac{1}{2\sqrt{L_k^2 + (\beta/(1-c))\|\boldsymbol{A}\|^2}}, \tau_{k-1}\sqrt{1+\theta_{k-1}}\right\}$, $\sigma_k = \beta\tau_k$, $\theta_k = \frac{\tau_k}{\tau_{k-1}}$

$\widetilde{\boldsymbol{x}}_k = \boldsymbol{x}_k + \theta_k(\boldsymbol{x}_k - \boldsymbol{x}_{k-1})$

$\boldsymbol{y}_{k+1} = \text{prox}_{\sigma_k g^*}(\boldsymbol{y}_k + \sigma_k \boldsymbol{A}\widetilde{\boldsymbol{x}}_k)$

$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \tau_k(\nabla f(\boldsymbol{x}_k) + \boldsymbol{A}^\top \boldsymbol{y}_{k+1})$

end for

4.4.1 High level ideas

We can rephrase the global stepsize condition of CVA by introducing a free parameter $\beta > 0$, which represents the ratio between the fixed dual and primal stepsizes: $\beta \coloneqq \frac{\sigma}{\tau}$. With this change of variables, the stepsize validity condition becomes $\tau \in \left(0, \frac{2}{L + \sqrt{L^2 + 4\beta\|\boldsymbol{A}\|^2}}\right)$.

Our algorithm disposes of CVA's global condition and relies instead on a very similar but *local*

criterion given by $\tau_k \in \left(0, \frac{1}{L_k + \sqrt{L_k^2 + 2\beta \|\boldsymbol{A}\|^2}}\right)$, with $\beta \coloneqq \frac{\sigma_k}{\tau_k}$. Here, $L_k \coloneqq \frac{\|\nabla f(\boldsymbol{x}_k) - \nabla f(\boldsymbol{x}_{k-1})\|}{\|\boldsymbol{x}_k - \boldsymbol{x}_{k-1}\|}$ provides an estimate of the local smoothness constant. Remaining in the interval of validity for τ_k is ensured by the first part of the expression defining the stepsize in APDA,

$$\tau_k = \min\left\{\frac{1}{2\sqrt{L_k^2 + (\beta/(1-c))\,\|\boldsymbol{A}\|^2}}, \tau_{k-1}\sqrt{1+\theta_{k-1}}\right\} \tag{4.5}$$

where $c \in (0,1)$. Intuitively, the first condition demands that τ_k does not overstep a constant related to the local curvature, thus allowing for larger stepsizes in flatter regions and correspondingly smaller ones otherwise.

By itself, the first term of (4.5) does not ensure convergence since overly aggressive and possibly destabilizing stepsizes might occur in near-linear regions. This issue is addressed by the second part of the expression (4.5), which, informally, prevents the stepsize from increasing "too fast" in consecutive iterations. Specifically, the increase factor is at most $\sqrt{1+\theta_{k-1}}$, where $\theta_k = \frac{\tau_{k-1}}{\tau_{k-2}}$.

Under these two local stepsize conditions, we are able to show APDA's convergence using the weaker assumption of local smoothness of f, thus conveniently removing the need to estimate a global smoothness constant L.

Remark 4.1. *While τ_k does not adapt to $\|\boldsymbol{A}\|$, for many practical problems this fact is not a big hindrance. Function f typically represents the data fidelity term, whose smoothness constant L (should it exist) can far exceed $\|\boldsymbol{A}\|$ — the matrix enforcing structured regularization on $\boldsymbol{x}$. A specific example is the TV-regularized imaging problem, where $\boldsymbol{A}$ is the matrix representation of the discrete gradient operator whose norm is bounded by $\sqrt{8}$ [40], while the data fidelity term may involve a very large number of measurements and a larger norm, consequently.*

Remark 4.2. *APDA takes an additional primal step prior to the for-loop, which is controlled by τ_{init} given as input. This is needed for estimating L_1 in the first iteration. In practice, we set $\tau_{init} = 1\mathrm{e}{-9}$, a sufficiently small value to ensure that $\boldsymbol{x}_1$ does not depart too far from x_0 and yield a good estimate of L_1. Furthermore, the setting of $\tau_0 = \infty$ simply ensures that in the first step, $\tau_1 = \frac{1}{2\sqrt{L_1^2 + (\beta/(1-c))\|\boldsymbol{A}\|^2}}$ and has no impact on further steps. Finally, in our experiments, we set $c = 1\mathrm{e}{-15}$ — this is a parameter introduced for theoretical purposes, as we explain shortly.*

4.4.2 Analysis — the base case

Our analysis proceeds as follows. First, we establish the inequality that characterizes the dynamics of APDA given in Lemma 4.1 below. Based on it, we are able to prove the boundedness of sequences $\{\boldsymbol{x}_k\}$ and $\{\boldsymbol{y}_k\}$ in Theorem 4.1. In turn, sequence boundedness alongside the local smoothness property of f allows us to conclude that there exists a constant $L > 0$ such that f is L-smooth on the compact set $\overline{\mathrm{conv}}(\{\boldsymbol{x}^\star, \boldsymbol{x}_0, \boldsymbol{x}_1, \ldots\})$ — the closed convex hull generated by $\{\boldsymbol{x}^\star, \boldsymbol{x}_0, \boldsymbol{x}_1, \ldots\}$. Finally, we leverage this information to show that $(\boldsymbol{x}_k, \boldsymbol{y}_k)$ converges to a saddle

point of (4.2) and derive the associated ergodic convergence rates presented in Theorem 4.1.

Lemma 4.1. *Consider APDA along with Assumptions 4.1 and 4.2 and* $(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{X} \times \mathcal{Y}$*. Then, for all* k *and* $\eta_k \in \left(\frac{\beta\tau_k \|\boldsymbol{A}\|}{1-c}, \frac{1-2\tau_k L_k}{2\tau_k \|\boldsymbol{A}\|}\right)$*,*

$$\begin{aligned}\|\boldsymbol{x}_{k+1}-\boldsymbol{x}\|^2 + \frac{1}{\beta}\left\|\boldsymbol{y}_{k+1}-\boldsymbol{y}\right\|^2 + \left(1-\eta_k\tau_k\|\boldsymbol{A}\|-\tau_k L_k\right)\|\boldsymbol{x}_{k+1}-\boldsymbol{x}_k\|^2 \\ + \frac{\eta_k-\tau_k\beta\|\boldsymbol{A}\|}{\beta\eta_k}\left\|\boldsymbol{y}_{k+1}-\boldsymbol{y}_k\right\|^2 + 2\tau_k(1+\theta_k)P_{\boldsymbol{x},\boldsymbol{y}}(\boldsymbol{x}_k) + 2\tau_k D_{\boldsymbol{x},\boldsymbol{y}}(\boldsymbol{y}_{k+1}) \\ \leq \|\boldsymbol{x}_k-\boldsymbol{x}\|^2 + \frac{1}{\beta}\left\|\boldsymbol{y}_k-\boldsymbol{y}\right\|^2 + \tau_k L_k\|\boldsymbol{x}_k-\boldsymbol{x}_{k-1}\|^2 + 2\tau_k\theta_k P_{\boldsymbol{x},\boldsymbol{y}}(\boldsymbol{x}_{k-1}).\end{aligned}$$

Moreover, it holds that:

1) $\tau_k L_k < \frac{1}{2} < 1-\eta_k\tau_k\|\boldsymbol{A}\|-\tau_k L_k$,

2) $\frac{1}{\beta} - \frac{\tau_k\|\boldsymbol{A}\|}{\eta_k} > \frac{c}{\beta} > 0$.

Proof sketch. We use algebraic manipulations, APDA's update rules, the Cauchy-Schwarz and Young inequalities and properties of the prox operator to get the recurrence

$$\begin{aligned}\|\boldsymbol{x}_{k+1}-\boldsymbol{x}\|^2 + \frac{1}{\beta}\left\|\boldsymbol{y}_{k+1}-\boldsymbol{y}\right\|^2 + \left(1-\tau_k\|\boldsymbol{A}\|\eta_k-\tau_k L_k\right)\|\boldsymbol{x}_{k+1}-\boldsymbol{x}_k\|^2 \\ + \left(\frac{1}{\beta}-\frac{\tau_k\|\boldsymbol{A}\|}{\eta_k}\right)\left\|\boldsymbol{y}_{k+1}-\boldsymbol{y}_k\right\|^2 + 2\tau_k(1+\theta_k)P_{\boldsymbol{x},\boldsymbol{y}}(\boldsymbol{x}_k) + 2\tau_k D_{\boldsymbol{x},\boldsymbol{y}}(\boldsymbol{y}_{k+1}) \\ \leq \|\boldsymbol{x}_k-\boldsymbol{x}\|^2 + \frac{1}{\beta}\left\|\boldsymbol{y}_k-\boldsymbol{y}\right\|^2 + \tau_k L_k\|\boldsymbol{x}_k-\boldsymbol{x}_{k-1}\|^2 + 2\tau_k\theta_k P_{\boldsymbol{x},\boldsymbol{y}}(\boldsymbol{x}_{k-1}),\end{aligned} \tag{4.6}$$

where $\eta_k > 0$ is a free iteration-dependent constant involved in Young's inequality.

In order to obtain anything worthwhile we would like to set η_k such that, when unrolling (4.6) over the iterations, the terms containing $\|\boldsymbol{x}_{k+1}-\boldsymbol{x}_k\|^2$ and $\left\|\boldsymbol{y}_{k+1}-\boldsymbol{y}_k\right\|^2$ accumulate on the LHS with positive coefficients. More precisely, we ask that:

$$\begin{cases}\frac{1}{\beta} - \frac{\tau_k\|\boldsymbol{A}\|}{\eta_k} > \frac{c}{\beta}, \\ 1-\tau_k\|\boldsymbol{A}\|\eta_k-\tau_k L_k > \frac{1}{2},\end{cases} \tag{4.7}$$

where $c \in (0,1)$. We note that the RHS of the first inequality could have been chosen as 0. However, we made it strictly positive due to technical reasons related to controlling the sequence $\left\|\boldsymbol{y}_{k+1}-\boldsymbol{y}_k\right\|^2$. In practice, we choose c to be as small as possible.

A similar remark holds for the second inequality, where it would have been sufficient to set its RHS to $\tau_{k+1}L_{k+1}$. Since this would considerably complicate the analysis, we make the observation that $\tau_k L_k < \frac{1}{2}$, $\forall k$ and use this simpler uniform upper-bound instead.

The inequalities (4.7) are equivalent to asking that $\eta_k \in \left(\frac{\tau_k\beta\|\boldsymbol{A}\|}{1-c}, \frac{1-2\tau_k L_k}{2\tau_k\|\boldsymbol{A}\|}\right)$ and what is left to show is that this is a valid interval, i.e., that the left endpoint is strictly smaller than its right counterpart. This condition amounts to solving a quadratic inequality in τ_k, whose solutions lie in the interval $\left(0, \frac{1}{L_k+\sqrt{L_k^2+2(\beta/(1-c))\|\boldsymbol{A}\|^2}}\right)$. The proof is concluded by showing that our choice of τ_k indeed satisfies this constraint. The full proof is deferred to Appendix C.1. □

We are now ready to state the main convergence result in Theorem 4.1 below, whose full proof is given in Appendix C.2.

Theorem 4.1. *Consider APDA along with Assumptions 4.1 and 4.2, and let $(\boldsymbol{x}^\star, \boldsymbol{y}^\star) \in \mathcal{X} \times \mathcal{Y}$ be a saddle point of problem* (4.2)*. Then,*

1) ***Boundedness.*** *The sequence $\{(\boldsymbol{x}_k, \boldsymbol{y}_k)\}$ is bounded. Specifically, for all k,*

$$\left\| \boldsymbol{x}_k - \boldsymbol{x}^\star \right\|^2 + \left\| \boldsymbol{y}_k - \boldsymbol{y}^\star \right\|^2 \leq M,$$

where $M := \| \boldsymbol{x}_1 - \boldsymbol{x}^\star \|^2 + \frac{1}{\beta}\left\| \boldsymbol{y}_1 - \boldsymbol{y}^\star \right\|^2 + \frac{1}{2}\| \boldsymbol{x}_1 - \boldsymbol{x}_0 \|^2 < \infty$.

2) ***Convergence to a saddle point.*** *The sequence $\{(\boldsymbol{x}_k, \boldsymbol{y}_k)\}$ converges to a saddle point of* (4.2)*.*

3) ***Ergodic convergence.*** *Let $S_k := \sum_{i=1}^{k} \tau_i$, $\overline{\boldsymbol{x}}_k := \frac{1}{S_k}\left(\tau_k(1+\theta_k)\boldsymbol{x}_k + \sum_{i=1}^{k-1}(\tau_i(1+\theta_i) - \tau_{i+1}\theta_{i+1})\boldsymbol{x}_i\right)$ and $\overline{\boldsymbol{y}}_k := \frac{1}{S_k}\sum_{i=1}^{k}\tau_i \boldsymbol{y}_{i+1}$. Then, for any bounded $B_1 \times B_2 \in \mathcal{X} \times \mathcal{Y}$ and for all k,*

$$\mathcal{G}_{B_1\times B_2}(\overline{\boldsymbol{x}}_k, \overline{\boldsymbol{y}}_k) \;\leq\; \frac{M(B_1,B_2)\sqrt{L^2 + (\beta/(1-c))\|\boldsymbol{A}\|^2}}{k},$$

where L is the Lipschitz constant of ∇f over the compact set $\overline{conv}(\{\boldsymbol{x}^\star, \boldsymbol{x}_0, \boldsymbol{x}_1, \ldots\})$ and $M(B_1,B_2) = \sup_{(\boldsymbol{x},\boldsymbol{y})\in B_1\times B_2} \| \boldsymbol{x}_1 - \boldsymbol{x} \|^2 + \frac{1}{\beta}\left\| \boldsymbol{y}_1 - \boldsymbol{x} \right\|^2 + \frac{1}{2}\| \boldsymbol{x}_1 - \boldsymbol{x}_0 \|^2$.

The boundedness result of Theorem 4.1 point 1) implies that the convex hull of the iterates $\mathcal{C} = \overline{\text{conv}}(\{\boldsymbol{x}^\star, \boldsymbol{x}_0, \boldsymbol{x}_1, \ldots\})$ is also bounded and hence compact. The local smoothness assumption on f then ensures that there exists $L > 0$ such that f is L-smooth over $\mathcal{C}$. Note that such an L exists for any $\boldsymbol{x}_0$, $\boldsymbol{y}_0$ since the boundedness result itself holds for any initial conditions (though the value of such L cannot generally be known, as it is path-dependent). Using this fact, we can show a uniform lower-bound on the primal stepsize: $\tau_k \geq \frac{1}{2}\left(L^2 + (\beta/(1-c))\|\boldsymbol{A}\|^2\right)^{-1/2} > 0$, $\forall k$, which is instrumental in deriving the subsequent convergence results, as well as Theorem 4.2. We emphasize that the appearance of constant L in the provided rates is a consequence of iterate boundedness, whose proof does not require its knowledge. Finally, we note that our rate is comparable to that of CVA in terms of constants.

4.4.3 Analysis under the additional Assumption 4.3

We now study APDA under the additional assumptions of local strong convexity of f and full row rank of $\boldsymbol{A}$. A few remarks are in order before proving Theorem 4.2. First, the boundedness result of Theorem 4.1 point 1) also holds for constant $c = 0$, since this constant was required only for proving convergence to a saddle point in point 2). Second, taking a smaller stepsize than the originally defined τ_k will not change the validity of Lemma 4.1 or the boundedness result of Theorem 4.1, as it remains within the required interval mentioned in Section 4.4.1.

Consequently, for studying APDA under the additional Assumption 4.3, we can simplify the stepsize expression by taking $c = 0$. This is because now we can show *iterate* convergence directly by using the strong convexity and full row rank assumptions. Specifically, we consider the stepsize

$$\tau_k = \min\left\{ \frac{1}{2\sqrt{4L_k^2 + \beta \| \boldsymbol{A} \|^2}}, \tau_{k-1}\sqrt{1+\theta_{k-1}/2} \right\}, \tag{4.8}$$

which is smaller than the one originally considered and, due to the aforementioned remarks, it ensures that APDA produces a bounded sequence. It follows that, under the local smoothness and local strong convexity assumptions, there exist constants L and μ such that f is L-smooth and μ-strongly convex over $\overline{\text{conv}}(\{\boldsymbol{x}^\star, \boldsymbol{x}_0, \boldsymbol{x}_1, \ldots\})$.

The existence of these constants, along with the full row rank of $\boldsymbol{A}$, in turn, allow us to derive a strengthened version of the inequality in Lemma 4.1 for $(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x}^\star, \boldsymbol{y}^\star)$,

$$\begin{aligned}
&\left\| \boldsymbol{x}_{k+1} - \boldsymbol{x}^\star \right\|^2 + \left(\frac{1}{\beta} + q_1\right) \left\| \boldsymbol{y}_{k+1} - \boldsymbol{y}^\star \right\|^2 + \left(\frac{1}{2} + q_2\right) \| \boldsymbol{x}_k - \boldsymbol{x}_{k+1} \|^2 + q_3 \left\| \boldsymbol{y}_{k+1} - \boldsymbol{y}_k \right\|^2 \\
&\qquad + 2\tau_k(1+\theta_k) P_{\boldsymbol{x}^\star, \boldsymbol{y}^\star}(\boldsymbol{x}_k) + 2\tau_k D_{\boldsymbol{x}^\star, \boldsymbol{y}^\star}(\boldsymbol{y}_{k+1}) \\
&\leq (1 - q_4) \left\| \boldsymbol{x}_k - \boldsymbol{x}^\star \right\|^2 + \frac{1}{\beta} \left\| \boldsymbol{y}_k - \boldsymbol{y}^\star \right\|^2 + \left(\frac{1}{2} - q_5\right) \| \boldsymbol{x}_k - \boldsymbol{x}_{k-1} \|^2 + 2\tau_k \theta_k P_{\boldsymbol{x}^\star, \boldsymbol{y}^\star}(\boldsymbol{x}_{k-1}),
\end{aligned}$$

where $q_1, q_2, q_3, q_4, q_5 > 0$ are constants given in Appendix C.3. This new inequality represents, in fact, a contraction guaranteeing the linear convergence stated in Theorem 4.2, below.

Theorem 4.2. *Consider APDA along with Assumptions 4.1, 4.2 and 4.3. Let $(\boldsymbol{x}^\star, \boldsymbol{y}^\star) \in \mathcal{X} \times \mathcal{Y}$ be a saddle point of problem* (4.2)*. Furthermore, let τ_k be defined by* (4.8) *and let $s := \sqrt{4L^2 + \beta \| \boldsymbol{A} \|^2}$ and $t := \sqrt{4\mu^2 + \beta \| \boldsymbol{A} \|^2}$, where μ, L are the strong convexity and smoothness constants of f over the compact set $\overline{conv}(\{\boldsymbol{x}^\star, \boldsymbol{x}_0, \boldsymbol{x}_1, \ldots\})$. Then, for all k,*

$$\left\| \boldsymbol{x}_k - \boldsymbol{x}^\star \right\|^2 + \frac{1}{\beta} \left\| \boldsymbol{y}_k - \boldsymbol{y}^\star \right\|^2 \leq \left(1 - \min\{p, q, r\}\right)^k M,$$

where the rate constants are given by

$$p = \frac{1}{2}, \quad q = \frac{\mu}{4s}, \quad r = \frac{\beta \sigma_{\min}^2(\boldsymbol{A}) \mu}{\beta \sigma_{\min}^2(\boldsymbol{A}) \mu + 8 s^2 t + 4 L^2 s},$$

and $M = \| \boldsymbol{x}_2 - \boldsymbol{x}^\star \|^2 + \left(\frac{1}{\beta} + T\right) \| \boldsymbol{y}_2 - \boldsymbol{y}^\star \|^2 + \frac{1}{2} \| \boldsymbol{x}_2 - \boldsymbol{x}_1 \|^2 + 2\tau_1 P_{\boldsymbol{x}^\star, \boldsymbol{y}^\star}(\boldsymbol{x}_1)$, $T = \dfrac{\sigma_{\min}^2(\boldsymbol{A})\mu}{8s^2 t + 4L^2 s}$, *with* $\sigma_{\min}(\boldsymbol{A})$ *representing the smallest singular value of* $\boldsymbol{A}$.

A few remarks are in order: first, as a sanity check, we observe that when $\boldsymbol{A} = \boldsymbol{0}$ (the zero matrix of appropriate dimensions), we recover the contraction factor of Malitsky and Mishchenko [147], which is equal to q.

Second, we make some notes on how our rate compares with existing ones. To our knowledge, there are no explicit results regarding the linear convergence of CVA under assumptions similar to ours (linear rates are usually shown for the 3-component objective without assumptions on $\boldsymbol{A}$ — see e.g., [44]). However, in the case of L-smooth and μ-strongly-convex f and full row-rank $\boldsymbol{A}$, Chen, Huang, and Zhang [47] show the linear convergence of PDFP^{2}O with rate:

$$\| \boldsymbol{x}_k - \boldsymbol{x}^\star \|^2 \leq \left(\| \boldsymbol{x}_1 - \boldsymbol{x}_0 \|^2 + \frac{1}{\sigma_{\max(\boldsymbol{A})}} \| \boldsymbol{y}_1 - \boldsymbol{y}_0 \|^2 \right) \left(1 - \min \left\{ \frac{\sigma_{\min(\boldsymbol{A})}^2}{\sigma_{\max(\boldsymbol{A})}^2}, \frac{\mu}{L} \right\} \right)^{k-1},$$

The rate presented in Theorem 4.2 has a comparatively worse contraction factor. The reason is that our iteration is set up in the style of CVA, where we essentially have a single stepsize to compute using the rephrasing from Section 4.4.1. Therefore, τ_k needs to obey the problem structure with respect to both L and $\| \boldsymbol{A} \|$, resulting in the "mixed" term appearing in the denominator.

Keeping the above in mind, the interested reader may find in the appendix that constants q and r come from a product between τ_k and other condition number-related quantities, which is tightly linked to the structure of the main inequality in Lemma 4.1. This makes the nice separation of condition numbers achieved in PDFP^{2}O's rate not possible in our case, and it seems the analysis necessary to achieve the present kind of adaptivity comes at the cost of worse constants (the same remark holds for Malitsky and Mishchenko [147]).

PDFP^{2}O, on the other hand, achieves a clean bound by having a different iteration style than CVA, as well as a fundamentally different kind of analysis where the iteration is expressed in fixed-point form to show convergence. In this context, the stability conditions on the stepsizes are also relaxed — specifically, $0 < \lambda \leq 1/\sigma_{\max}^2(\boldsymbol{A})$ and $0 < \gamma < 2L$ [47]. A drawback of this approach, however, is that the algorithm has no rate guarantees when f is only smooth and not strongly convex and only asymptotic convergence is shown. Also, PDFP^{2}O requires 3 matrix-vector multiplications per iteration, whereas we only require 2.

4.5 Experiments

We now present some numerical experiments conducted for APDA^2. Additional problems and results are included in the appendix. The experiments were implemented in Python 3.9 and

[2] See https://github.com/mvladarean/adaptive_pda.

executed on a MacBook Pro with 32 GB RAM and a 2,9 GHz 6-Core Intel Core i9 processor.

The baseline we compare against in this section is CVA, implemented as Algorithm 1 of Chambolle and Pock [44] (using $g \equiv 0$). In the particular case of sparse logistic regression, we also compare against FISTA [14]. For obtaining $\boldsymbol{x}^\star$ we ran one of the algorithms for a large number of iterations.

4.5.1 Sparse binary logistic regression

We consider the problem of sparse binary Logistic Regression on 4 LIBSVM datasets [45] and show that adaptivity provides faster convergence in 3 of these cases. The objective we consider is

$$\min_{\boldsymbol{x}\in\mathbb{R}^d} F(\boldsymbol{x}) := \underbrace{\sum_{i=1}^{m} \log(1+\exp(-b_i\langle \boldsymbol{q}_i, \boldsymbol{x}\rangle))}_{f} + \underbrace{\lambda \|\boldsymbol{x}\|_1}_{g}, \tag{4.9}$$

where $(\boldsymbol{q}_i, b_i) \in \mathbb{R}^d \times \{-1, 1\}$ and λ is the regularization parameter. APDA and CVA can be applied to this problem by setting $\boldsymbol{A} = \boldsymbol{I}$ in formulation (4.2). Primal-dual algorithms are not the typical choice for solving (4.9), which is usually addressed by methods such as Proximal Gradient or FISTA [14]. However, we note that the computational costs of APDA and FISTA are comparable since the matrix-vector multiplication cost of the former is removed due to a $A = I$.

We choose $\lambda = 0.005 \left\| \boldsymbol{Q}^\top b \right\|_\infty$, where $\boldsymbol{Q}^\top = \left[\boldsymbol{q}_1^\top, \dots \boldsymbol{q}_m^\top\right]^\top$. For APDA we perform a parameter sweep over $\beta \in [1\text{e}{-3}, 1\text{e}6]$ for each dataset and settle for: $\beta = 2.68\text{e}3$ for `ijcnn`; $\beta = 5.18\text{e}4$ for `a9a`; $\beta = 3.16\text{e}1$ for `mushrooms`; $\beta = 3.73\text{e}{-1}$ for `covtype`.

For CVA we sweep $p \in [1\text{e}{-3}, 1\text{e}6]$ and set $\tau = \frac{1}{\|\boldsymbol{A}\|/p+L}$ and $\sigma = \frac{1}{p\|\boldsymbol{A}\|}$ — by construction, these stepsizes satisfy the validity condition and are as large as possible since the condition is satisfied with equality. We do an additional tuning procedure where we choose constants $\tau \in [1\text{e}{-10}, 1\text{e}2]$ and $\xi \in [1\text{e}{-5}, 1\text{e}2]$ and set $\sigma = \tau\xi$, which are subject to verifying the stepsize validity condition of CVA. Finally we select the best stepsizes across the two tuning phases to be (truncated to 3 decimals): $\tau = 9.869\text{e}{-4}$, $\sigma = 1.125\text{e}1$ for `ijcnn`; $\tau = 2.655\text{e}{-4}$, $\sigma = 7.896\text{e}1$ for `a9a`; $\tau = 9.936\text{e}{-4}$, $\sigma = 5.878\text{e}0$ for `mushrooms`; $\tau = 7.728\text{e}{-6}$, $\sigma = 1\text{e}{-6}$ for `covtype`.

Note that the Hessian of f is given by $\nabla^2 f(\boldsymbol{x}) = \boldsymbol{Q}^\top \boldsymbol{D}(\boldsymbol{x})\boldsymbol{Q}$, where $\boldsymbol{D}(\boldsymbol{x})$ is a diagonal matrix such that $\boldsymbol{D}_{i,i}(\boldsymbol{x}) = \sigma_i(\boldsymbol{x})(1-\sigma_i(\boldsymbol{x}))$, where $\sigma_i(\boldsymbol{x}) = \frac{1}{1+\exp(-b_i\langle \boldsymbol{q}_i, \boldsymbol{x}\rangle)} \in (0,1)$. Clearly, over any compact set in $\mathcal{C} \subset \boldsymbol{X}$ there exist $D_{\min} \coloneqq \min_{i,\boldsymbol{x}\in\mathcal{C}} \boldsymbol{D}_{i,i}(\boldsymbol{x}) \in (0,1)$ such that $D_{\min}\boldsymbol{Q}^\top\boldsymbol{Q} \preceq \boldsymbol{Q}^\top\boldsymbol{D}(\boldsymbol{x})\boldsymbol{Q}$. As a result, a sufficient condition for local strong convexity is that the minimum eigenvalue of $\boldsymbol{Q}^\top\boldsymbol{Q}$ is greater than 0.

The convergence results are presented in Figure 4.1 along with stepsize comparison plots. For dataset `ijcnn` we run APDA with the modified τ_k used in Theorem 4.2, since $\lambda_{\min}(\boldsymbol{Q}^\top\boldsymbol{Q}) = 75.13$ and A has full rank. In the latter case, the legend identifier is APDA-strcnv. For the

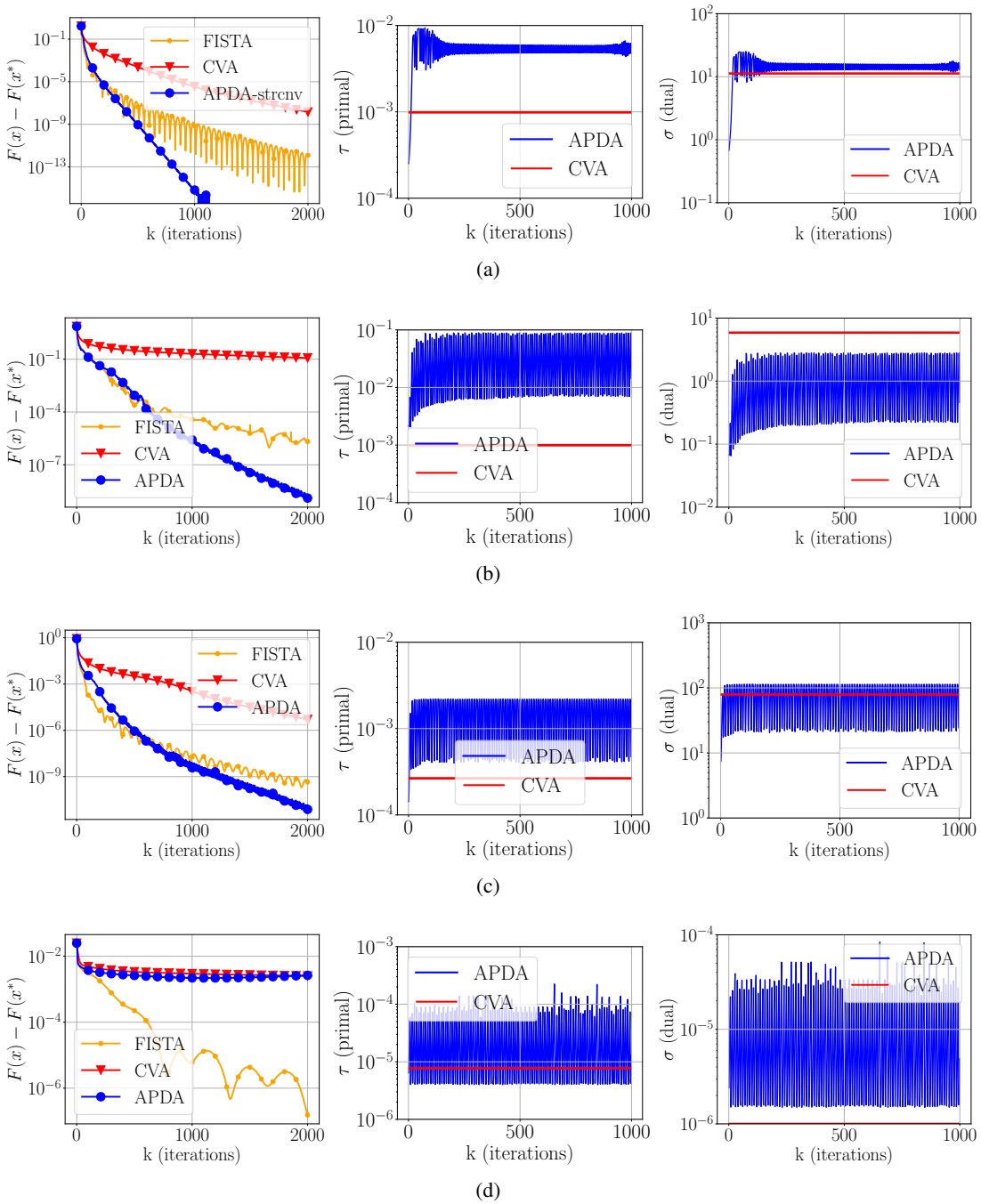


Figure 4.1: The first column shows algorithm convergence. The second column shows a comparison of primal stepsizes between APDA and CVA. The third column shows a comparison of dual stepsizes between APDA and CVA. Each subfigure represents a different dataset: (a) `ijcnn`; (b) `mushrooms`; (c) `a9a`; (d) `covtype`.

remaining datasets we use only the basic setting for τ_k, as $\lambda_{\min}(\boldsymbol{Q}^\top \boldsymbol{Q}) \leq 1\text{e}{-13}$.

While APDA outperforms FISTA and CVA on `ijcnn`, `a9a` and `mushrooms`, it shows a

relatively poor performance on `covtype`. We hypothesize that this is related to the condition number of $\boldsymbol{Q}^\top \boldsymbol{Q}$, which is almost three orders of magnitude larger in the latter case: 9.2e22 versus 5.3e1, 2e20 and 2e17 for `ijcnn`, `mushrooms` and `a9a`, respectively. A similar behaviour is seen in Figure 1.(c) of Malitsky and Mishchenko [147].

Finally, the adaptive property of APDA's stepsizes is visible in the stepsize comparison plots where they are shown to oscillate within at least one order of magnitude throughout the optimization process.

4.5.2 Non-convex phase retrieval

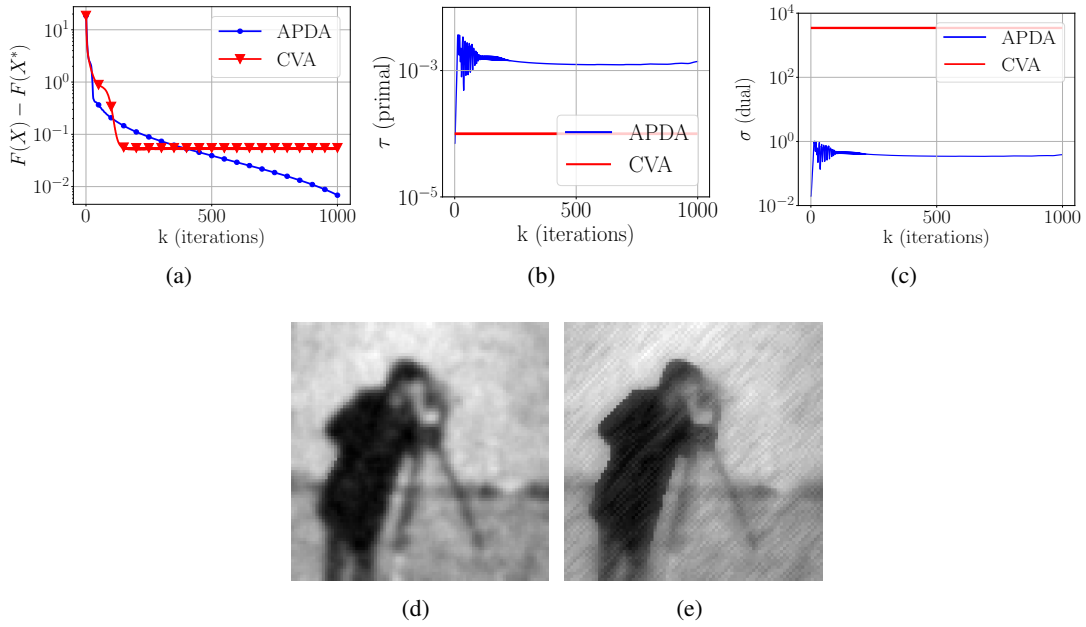


Figure 4.2: (a) Convergence rate. (b) Primal stepsize comparison. (c) Primal stepsize comparison. (d) APDA reconstruction, PSNR = 21.34, SSIM = 0.76. (e) CVA reconstruction, PSNR = 20.56, SSIM = 0.70.

In this section, we provide the results for applying our algorithm, heuristically, on the non-convex least squares formulation of the phase retrieval (PR) problem. The phase-retrieval problem has attracted intense interest recently, due to its application in domains such as optical imaging [223], astronomy [83] and many others. Here, we consider the real counterpart of the original complex PR formulation for square images, where given $\{(\boldsymbol{A}_i, b_i) \in \mathbb{R}^{n\times n} \times \mathbb{R}\}$ we want to recover $\boldsymbol{X}^\star \in \mathbb{R}^{n\times n}$ up to its sign, such that $b_i = \operatorname{Tr}(\boldsymbol{A}_i^\top \boldsymbol{X}^\star)^2$. To this end, we consider the following TV-regularized optimization objective

$$\min_{\boldsymbol{X}\in\mathbb{R}^{n\times n}} F(\boldsymbol{X}) \coloneqq \underbrace{\frac{1}{4m}\sum_{i=1}^{m}\left(b_i - \operatorname{Tr}(\boldsymbol{A}_i^\top \boldsymbol{X})^2\right)^2}_{f(\boldsymbol{X})} + \underbrace{\lambda\|\boldsymbol{D}\boldsymbol{X}\|_{2,1}}_{g(\boldsymbol{X})\equiv\|\cdot\|_{\mathrm{TV}}}, \tag{4.10}$$

where $\boldsymbol{D} : \mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n\times 2}$ represents the discrete gradient operator and

$$\|\boldsymbol{DX}\|_{2,1} \coloneqq \sum_{i,j=1}^{n,n} \sqrt{(\boldsymbol{DX})_{i,j,1}^2 + (\boldsymbol{DX})_{i,j,2}^2}.$$

The regularization term represents the isotropic TV norm, which is known to help in recovering sharp signals by preserving discontinuities and reducing noise [41, 56, 43].

We note a few things: first, objective (4.10) is non-convex due to f, and in addition, f is only locally smooth. Secondly, Sun, Qu, and Wright [210] have recently shown that given m i.i.d Gaussian measurements, the global geometry of $F(\boldsymbol{X})$ is "benign" for $m > Cd\log(d)^3$, where d is the problem dimension. By benign, the authors specifically mean "(1) there are no spurious local minimizers, and all global minimizers are equal to the target signal $\boldsymbol{X}^\star$ up to a global phase; (2) the objective function has a negative directional curvature around each saddle point". It is hypothesized that in such cases iterative algorithms should, with high probability, find the minimizer without requiring special initialization as is needed for current state-of-the-art solvers.

For our experiments, we use 84×84-sized images and choose a smaller number of measurements than suggested above: $m = d\log(d) \approx 27,155$. We generate m sparse matrices $\boldsymbol{A}_i \in R^{n\times n}$ with 30% non-zero entries sampled i.i.d from the standard normal distribution, and corrupt a random subset containing 10% of elements in b_i by setting them to 0. We perform parameter sweep for $\lambda \in [1\text{e}{-4}, 1\text{e}4]$, $\beta \in [1\text{e}{-3}, 1\text{e}4]$ and settle for $\lambda = 1\text{e}2$ and $\beta = 2.78\text{e}2$. Without guidelines for setting τ, σ for CVA since f is not L-smooth, we search for the best $\tau \in [1\text{e}{-4}, 1\text{e}4]$ and $p \in [1\text{e}{-2}, 1\text{e}2]$ such that $\sigma = \frac{1}{p\tau\|\boldsymbol{A}\|}$ and settle for $\tau = 1\text{e}{-4}$, $p = 1.02\text{e}0$. We note that CVA diverged for $32/40$ grid points, whereas our method converged for all instances. Finally, the initial points x_0 and y_0 are sampled from the standard normal distribution.

The results are depicted in Figure 4.2, which contains the reconstructions and convergence plots. For each reconstruction, we report the Peak Signal to Noise Ratio (PSNR) and the Structural Similarity Index Measure (SSIM). We tested several random seeds and obtained similar results. We also tried running CVA with the stepsize values used by APDA in its last iteration (notice how in Figure 4.2 (d) τ_k essentially stabilizes in a very narrow band just above $1\text{e}{-3}$ after the first 250 iterations) — however, CVA diverged in this setting as well.

4.5.3 Image inpainting

Image inpainting involves reconstructing the missing parts of a subsampled image $\boldsymbol{B} = \boldsymbol{P}_\Omega \boldsymbol{X}^\natural$, where $\boldsymbol{P}_\Omega : \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$ is a linear operator selecting a subset of q pixels from the original image $\boldsymbol{X}^\natural \in \mathbb{R}^{m\times n}$, where $q \ll mn$. This problem can be formulated as the following regularized

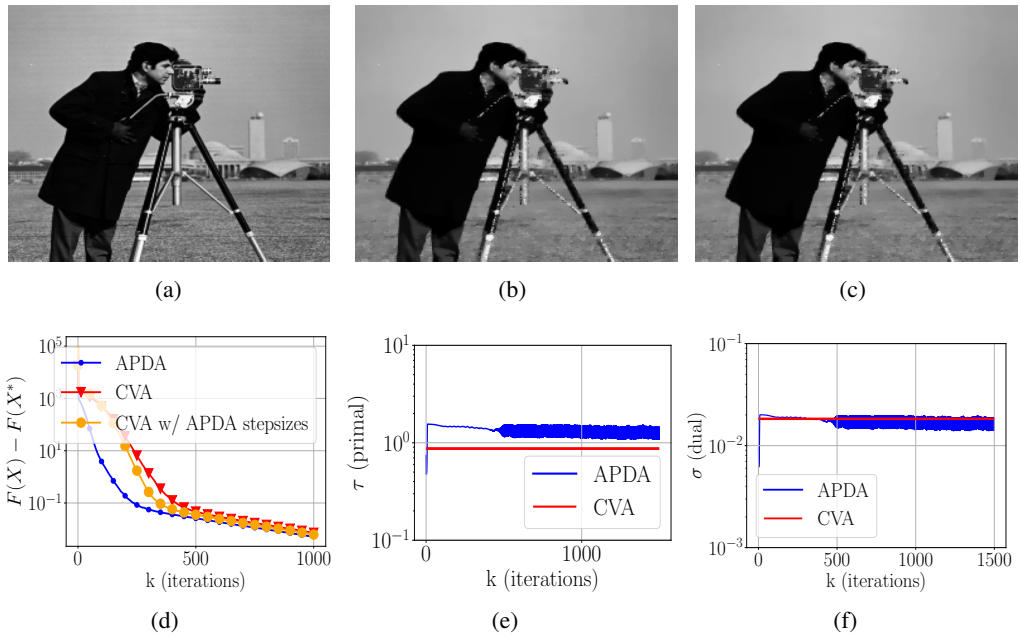


Figure 4.3: (a) Original image downloaded from http://www.cs.tut.fi/~foi/GCF-BM3D/. (b) APDA reconstruction, PSNR = 25.63, SSIM = 0.91. (c) CVA reconstruction, PSNR = 25.63, SSIM = 0.91. (d) Convergence rate. (e) Primal stepsize comparison. (f) Dual stepsize comparison.

optimization objective

$$\min_{\boldsymbol{X}\in\mathbb{R}^{m\times n}} F(\boldsymbol{X}) \coloneqq \underbrace{\frac{1}{2}\|\boldsymbol{B}-\boldsymbol{P}_{\Omega}\boldsymbol{X}\|_F^2}_{f(\boldsymbol{X})} + \underbrace{\lambda\|\boldsymbol{D}\boldsymbol{X}\|_{2,1}}_{g(\boldsymbol{X})\equiv\|\cdot\|_{\mathrm{TV}}}, \tag{4.11}$$

where $\boldsymbol{D}:\mathbb{R}^{n\times m}\to\mathbb{R}^{n\times m\times 2}$ is the discrete gradient operator and

$$\|\boldsymbol{D}\boldsymbol{X}\|_{2,1} \coloneqq \sum_{i,j=1}^{n,m}\sqrt{(\boldsymbol{D}\boldsymbol{X})_{i,j,1}^2+(\boldsymbol{D}\boldsymbol{X})_{i,j,2}^2}.$$

For our experiments, we vectorize the images of size 256×256 and transform $\boldsymbol{D}$ accordingly. We represent $\boldsymbol{P}_{\Omega}$ as a matrix built by removing rows uniformly at random from $\boldsymbol{I}$ and which, via the Hadamard product, removes 60% of pixels from the original image (sampling ratio 0.4). We perform parameter sweep for $\lambda\in[1\mathrm{e}{-4},1\mathrm{e}0]$, and settle for $\lambda=1\mathrm{e}{-2}$. We also sweep $\beta\in[1\mathrm{e}{-5},1\mathrm{e}0]$ and settle for $\beta=1.291\mathrm{e}{-2}$. Finally, we perform a similar two-phase tuning for CVA as that described in Section 4.5 with $p\in[1\mathrm{e}{-5},1\mathrm{e}3]$ for the first phase and $\tau\in[1\mathrm{e}{-5},1\mathrm{e}2]$, $\xi\in[1\mathrm{e}{-5},1\mathrm{e}1]$ for the second phase. We settle for stepsizes $\tau=8.722\mathrm{e}{-1}$ and $\sigma=1.831\mathrm{e}{-2}$.

Experiment results are presented in Figure 4.3, where we show the reconstructions, alongside the convergence plot and a comparison of the fixed stepsizes of CVA with those of APDA. The two algorithms are comparable both in terms of reconstruction quality and convergence speed,

with APDA being marginally better for the latter criterion. The convergence plot also shows an instance of CVA whose stepsizes were set to the values of those used by APDA in the final iteration of these experiments. Finally, subfigures (e) and (f) show APDA's stepsizes oscillating within close range of CVA's.

4.6 Conclusion

This chapter introduced an adaptive primal-dual algorithm (APDA) for finding the saddle point of structured convex-concave objectives. The main feature of this algorithm is that it attains adaptivity "for free", without resorting to linesearch subroutines. Concretely, this is achieved by leveraging past gradient information to estimate the local function curvature. In addition, the method's convergence relies on weaker assumptions than prior literature, by requiring only local versions of the usually global smoothness or strong convexity conditions. The convergence analysis recovers known rates, which are further confirmed by numerical experiments. Moreover, our experiments showed APDA consistently outperforming its non-adaptive variant — the Condat-Vũ [55, 222], as well as performing on par with the accelerated primal-only method FISTA [14] for some problem instances.

A question that immediately emerges from this chapter is whether the same kind of adaptivity can be extended to the forward-backward splitting scheme (this would allow us to handle hard constraints on the primal variable, which APDA cannot ensure), or to more complex primal-dual methods involving three operators. In the time since the work presented in this chapter was published, Malitsky and Mishchenko [148] and Latafat et al. [136] have given affirmative answers for the two scenarios, respectively.

A further open question is whether we can devise stochastic variants using the same or a similar stepsize setting approach.

5 Conclusion and future directions

5.1 Summary

This thesis proposed a series of approaches for ameliorating the scalability of first-order optimization algorithms in several constrained problem settings. Our work was guided by the definition of scalability discussed in Chapter 1. Concretely, we have achieved the following.

1. In Chapter 2, we proposed three stochastic Frank-Wolfe-type methods capable of handling stochastic linear inclusion constraints. Through judicious use of variance reduction techniques, in conjunction with smoothing and linear minimization steps, our methods converge to an optimal feasible value in expectation, while only processing a subset of the constraints per iteration. This gives them an edge over existing approaches that process the constraints in full, and scalability is improved due to the reduced iteration cost.

2. In Chapter 3, we proposed generalizations of the vanilla Frank-Wolfe and Conditional Gradient Sliding methods to a class of composite non-differentiable problems — a known difficult setting for these algorithms. Our methods leverage the problem structure and a modified linear minimization oracle to attain convergence rates akin to the smooth setting for convex problems. Our approach thus eschews the stringent lower bound on the convergence speed of black-box methods for this class of non-differentiability. Scalability is, therefore, improved due to the algorithms' faster convergence rate.

3. Finally, in Chapter 4, we proposed an adaptive primal-dual algorithm for solving structured convex-concave saddle point problems. Our method reuses past gradients to estimate the problem's local curvature across the iterations and takes larger stepsizes in accordance. Importantly, it does not resort to linesearch subroutines for achieving adaptivity, and recovers known convergence rates under weaker (local) structural assumptions than prior literature. In this case, scalability is enhanced thanks to the method's faster empirical convergence ensured by taking larger update steps.

An overarching theme of these investigations was that under the right structural assumptions, problems yield themselves to be scalably optimized. This translates into improvements in either the convergence rate, the computational cost of iterations, or the methods' ability to automatically adapt to problem geometry. In short, we proposed theoretically-backed methods that effectively leverage the problem structure to achieve a practical edge in terms of scalability.

5.2 Future directions

Most of the work in this thesis was done under the umbrella of convexity. However, as foreshadowed by the examples in Chapter 1, non-convexity is at the forefront of Machine Learning's present success. Therefore, we see it as a central consideration for future lines of inquiry along with non-differentiability, which is also a mainstay of modern applications. In the following, we delineate possible directions in addition to those proposed in each chapter's closing section.

First, we turn to Frank-Wolfe algorithms. Chapter 3 showed that despite our method's convergence with respect to the generalized Frank-Wolfe gap, this does not guarantee reaching a stationary point in non-convex scenarios. An immediate question is, therefore, whether a different accuracy certificate could provide this guarantee. A separate route for investigating convergence under non-smoothness and non-convexity is to consider other restricted problem classes, as recently done by De Oliveira [68] for tackling upper-$C^{1,\alpha}$ functions, $\alpha \in (0, 1]$. Their proposed method converges to Clarke stationary points but requires either precise knowledge of problem constants, which are difficult to estimate in practice, or exact line research — this, in itself, is an opportunity for investigation. Finally, convergence under non-differentiability may be sought through modified LMOs that retain the efficiency of their vanilla counterpart on specific constraint sets (an example is given by Garber and Wolf [93]), while eschewing the non-convergence caused by the latter.

Second, there is great appeal to extending the type of adaptivity discussed in Chapter 4 to non-convex settings. A primary application of this direction is the training of neural networks, where faster empirical convergence is desired due to the high cost of training. At present, the simpler, unconstrained case is still open and would have to be tackled first. The promising results on non-convex problems reported in both the work that originated this technique [147] and our Section 4.5.2 make this line of inquiry particularly intriguing. Furthermore, it is worth pursuing whether (modifications of) this approach extend to non-differentiable problems, which suffer from a slow convergence due to their decreasing stepsize requirement.

Finally, and more broadly, extending the techniques presented in this thesis to problems with non-convex regularizers and constraint sets is an important future direction. Such structures occur in the training of *fair* Machine Learning models [126, 38] and are the main tools preventing harmful biases from creeping into the latter. Given Machine Learning's fast-increasing reach, it is necessary to improve the scalability of algorithms addressing such formulations in order to ensure their widespread adoption and alleviate ethical issues related to automated decision making.

A Appendix for Chapter 2

Before delving into the proofs of the main results in Chapter 2, we introduce some additional notation, a handful of supporting technical observations, and some useful known results.

First — a recap on smoothing [167]. Given a function $g : \mathbb{R}^m \to \mathbb{R} \cup \{+\infty\}$ that is proper, closed and convex, the smooth approximation of g is defined by

$$g_\beta(\boldsymbol{z}) = \max_{\boldsymbol{\lambda} \in \mathbb{R}^d} \left\{ \langle \boldsymbol{z}, \boldsymbol{\lambda} \rangle - g^*(\boldsymbol{\lambda}) - \frac{\beta}{2} \| \boldsymbol{\lambda} \|^2 \right\} \tag{A.1}$$

where g^* denotes the Fenchel conjugate and $\beta > 0$ is the smoothing parameter. Then, g_β is convex and $\frac{1}{\beta}$-smooth.

We let $\boldsymbol{\lambda}^\star_\beta(\boldsymbol{z})$ denote the solution of the maximization problem in (A.1), i.e.,

$$\begin{aligned}
\boldsymbol{\lambda}^\star_\beta(\boldsymbol{z}) &= \arg\max_{\boldsymbol{\lambda} \in \mathbb{R}^d} \left\{ \langle \boldsymbol{z}, \boldsymbol{\lambda} \rangle - g^*(\boldsymbol{\lambda}) - \frac{\beta}{2} \| \boldsymbol{\lambda} \|^2 \right\} \\
&= \arg\min_{\boldsymbol{\lambda} \in \mathbb{R}^d} \left\{ \frac{1}{\beta} g^*(\boldsymbol{\lambda}) - \frac{1}{\beta} \langle \boldsymbol{z}, \boldsymbol{\lambda} \rangle + \frac{1}{2} \| \boldsymbol{\lambda} \|^2 + \frac{1}{2} \left\| \frac{1}{\beta} \boldsymbol{z} \right\|^2 \right\} \\
&= \arg\min_{\boldsymbol{\lambda} \in \mathbb{R}^d} \left\{ \frac{1}{\beta} g^*(\boldsymbol{\lambda}) + \frac{1}{2} \left\| \boldsymbol{\lambda} - \frac{1}{\beta} \boldsymbol{z} \right\|^2 \right\} \\
&= \operatorname{prox}_{\beta^{-1} g^*}(\beta^{-1} \boldsymbol{z}) \\
&= \frac{1}{\beta} \big(\boldsymbol{z} - \operatorname{prox}_{\beta g}(\boldsymbol{z}) \big)
\end{aligned} \tag{A.2}$$

where the last line is the Moreau decomposition.

We now invoke some results from prior literature that are essential for our statements. The smoothed indicator was studied by Tran-Dinh, Fercoq, and Cevher [212], where the following properties of g_β are noted for all $\boldsymbol{z}_1, \boldsymbol{z}_2 \in \mathbb{R}^m$ and all $\beta, \rho > 0$ (see **Lemma 10** therein for the

proofs).

$$g_\beta(\boldsymbol{z}_1) \geq g_\beta(\boldsymbol{z}_2) + \langle \nabla g_\beta(\boldsymbol{z}_2), \boldsymbol{z}_1 - \boldsymbol{z}_2 \rangle + \frac{\beta}{2} \left\| \boldsymbol{\lambda}^\star_\beta(\boldsymbol{z}_2) - \boldsymbol{\lambda}^\star_\beta(\boldsymbol{z}_1) \right\|^2 \tag{A.3}$$

$$g(\boldsymbol{z}_1) \geq g_\beta(\boldsymbol{z}_2) + \langle \nabla g_\beta(\boldsymbol{z}_2), \boldsymbol{z}_1 - \boldsymbol{z}_2 \rangle + \frac{\beta}{2} \left\| \boldsymbol{\lambda}^\star_\beta(\boldsymbol{z}_2) \right\|^2 \tag{A.4}$$

$$g_\beta(\boldsymbol{z}_1) \leq g_\rho(\boldsymbol{z}_1) + \frac{\rho - \beta}{2} \left\| \boldsymbol{\lambda}^\star_\beta(\boldsymbol{z}_1) \right\|^2 \tag{A.5}$$

Further, should g be L_g-Lipschitz continuous, then for $\forall \beta > 0$ and $\forall \boldsymbol{z} \in \mathbb{R}^m$,

$$g_\beta(\boldsymbol{z}) \leq g(\boldsymbol{z}) \leq g_\beta(\boldsymbol{z}) + \frac{\beta}{2} L_g^2. \tag{A.6}$$

The proof follows immediately from Equation (2.7) in [167] with a remark on the duality between bounded domain and Lipschitz continuity.

Notation recap and extras. For simplicity, we shall denote the indicator functions of the stochastic constraint sets $\boldsymbol{b}(\xi)$ as

$$g(\boldsymbol{A}(\xi)\boldsymbol{x}, \xi) \coloneqq \iota_{\{\boldsymbol{b}(\xi)\}}(\boldsymbol{A}(\xi)\boldsymbol{x}),$$

and their $\frac{1}{\beta}$-smooth approximations resulted from Nesterov-type smoothing (see Section 2.3) as

$$g_\beta(\boldsymbol{A}(\xi)\boldsymbol{x}, \xi) \coloneqq \frac{1}{2\beta} \operatorname{dist}(\boldsymbol{A}(\xi)\boldsymbol{x}, \boldsymbol{b}(\xi))^2 = \frac{1}{2\beta} \left\| \boldsymbol{A}(\xi)\boldsymbol{x} - \operatorname{proj}_{\boldsymbol{b}(\xi)}(\boldsymbol{A}(\xi)\boldsymbol{x}) \right\|^2.$$

Their counterparts resulting from applying expectations are capitalized and denoted as

$$G_\beta(\boldsymbol{A}\boldsymbol{x}) \coloneqq \mathbb{E}\left[g_\beta(\boldsymbol{A}(\xi)\boldsymbol{x}, \xi) \right] \quad \text{and} \quad \nabla G_\beta(\boldsymbol{A}\boldsymbol{x}) \coloneqq \mathbb{E}\left[\nabla g_\beta(\boldsymbol{A}(\xi)\boldsymbol{x}, \xi) \right],$$

where $\boldsymbol{A} : \mathbb{R}^d \to \mathcal{H}$ is a linear operator such that $(\boldsymbol{A}\boldsymbol{x})\xi = \boldsymbol{A}(\xi)\boldsymbol{x}$ and $G_\beta : \mathcal{H} \to \mathbb{R} \cup \{\infty\}$.

The composite stochastic objective resulting from smoothing the indicators becomes

$$F_{\beta_k}(\boldsymbol{x}, \xi) \coloneqq f(\boldsymbol{x}, \xi) + g_{\beta_k}(\boldsymbol{x}, \xi),$$

with gradient $\nabla F_{\beta_k}(\boldsymbol{x}, \xi) \coloneqq \nabla f(\boldsymbol{x}, \xi) + \nabla g_{\beta_k}(\boldsymbol{x}, \xi)$. We will often drop the second ξ in the arguments of g, since we distinguish its deterministic version through capitalization (different from f).

We annotate averaged stochastic quantities with the symbol $\sim$. For example, the averaged stochastic gradient of the constraints is expressed as $\widetilde{\nabla}_{\boldsymbol{x}} g_\beta(\boldsymbol{A}(\xi)\boldsymbol{x})$. The optimal value of the dual problem at $\boldsymbol{A}(\xi)\boldsymbol{x}$ is denoted as

$$\boldsymbol{\lambda}^\star_\beta(\boldsymbol{A}(\xi)\boldsymbol{x}) \coloneqq \frac{1}{\beta} \left(\boldsymbol{A}(\xi)\boldsymbol{x} - \operatorname{proj}_{\boldsymbol{b}(\xi)}(\boldsymbol{A}(\xi)\boldsymbol{x}) \right). \tag{A.7}$$

Finally, the smoothed gap is defined as

$$S_\beta(\boldsymbol{x}) \coloneqq F_\beta(\boldsymbol{x}) - f^\star.$$

Technical observations. We now state a series of simple implications that will be used to prove the main results.

From the definition of G_β we get that

$$\begin{aligned}\nabla_{\boldsymbol{x}} G_\beta(\boldsymbol{A}\boldsymbol{x}) &= \mathbb{E}\left[\nabla_{\boldsymbol{x}} g_\beta(\boldsymbol{A}(\xi)\boldsymbol{x})\right] \\ &= \mathbb{E}\left[\boldsymbol{A}^\top(\xi)\nabla g_\beta(\boldsymbol{A}(\xi)\boldsymbol{x})\right] \\ &= \mathbb{E}\left[\frac{1}{\beta}\boldsymbol{A}^\top(\xi)\left(\boldsymbol{A}(\xi)\boldsymbol{x} - \operatorname{proj}_{\boldsymbol{b}(\xi)}(\boldsymbol{A}(\xi)\boldsymbol{x})\right)\right]. \end{aligned} \tag{A.8}$$

From the smoothness of G_β, the iterate update rule and the non-expansiveness of projections, we have that

$$\begin{aligned}&\left\| \nabla G_\beta(\boldsymbol{A}\boldsymbol{x}_{k+1}) - \nabla G_\beta(\boldsymbol{A}\boldsymbol{x}_k) \right\|^2 \\ &\quad= \left\| \frac{1}{\beta}\mathbb{E}\left[\boldsymbol{A}^\top(\xi)\left(\boldsymbol{A}(\xi)\boldsymbol{x}_{k+1} - \operatorname{proj}_{\boldsymbol{b}(\xi)}(\boldsymbol{A}(\xi)\boldsymbol{x}_{k+1})\right) - \boldsymbol{A}^\top(\xi)\left(\boldsymbol{A}(\xi)\boldsymbol{x}_k - \operatorname{proj}_{\boldsymbol{b}(\xi)}(\boldsymbol{A}(\xi)\boldsymbol{x}_k)\right)\right]\right\|^2 \\ &\quad\le \frac{1}{\beta^2}\mathbb{E}\left[\left\|\boldsymbol{A}^\top(\xi)\boldsymbol{A}(\xi)\left(\boldsymbol{x}_{k+1} - \boldsymbol{x}_k\right) + \boldsymbol{A}^\top(\xi)\left(\operatorname{proj}_{\boldsymbol{b}(\xi)}(\boldsymbol{A}(\xi)\boldsymbol{x}_k) - \operatorname{proj}_{\boldsymbol{b}(\xi)}(\boldsymbol{A}(\xi)\boldsymbol{x}_{k+1})\right)\right\|^2\right] \\ &\quad\le \frac{1}{\beta^2}\mathbb{E}\left[2\left\|\boldsymbol{A}^\top(\xi)\boldsymbol{A}(\xi)\left(\boldsymbol{x}_{k+1} - \boldsymbol{x}_k\right)\right\|^2 + 2\left\|\boldsymbol{A}^\top(\xi)\left(\operatorname{proj}_{\boldsymbol{b}(\xi)}(\boldsymbol{A}(\xi)\boldsymbol{x}_k) - \operatorname{proj}_{\boldsymbol{b}(\xi)}(\boldsymbol{A}(\xi)\boldsymbol{x}_{k+1})\right)\right\|^2\right] \\ &\quad\le \frac{2\gamma_k^2 L_{\boldsymbol{A}}^2 \mathcal{D}_{\mathcal{X}}^2}{\beta^2} + \frac{2}{\beta^2}\mathbb{E}\left[\left\|\boldsymbol{A}(\xi)\right\|^2\left\|\operatorname{proj}_{\boldsymbol{b}(\xi)}(\boldsymbol{A}(\xi)\boldsymbol{x}_k) - \operatorname{proj}_{\boldsymbol{b}(\xi)}(\boldsymbol{A}(\xi)\boldsymbol{x}_{k+1})\right\|^2\right] \\ &\quad\le \frac{2\gamma_k^2 L_{\boldsymbol{A}}^2 \mathcal{D}_{\mathcal{X}}^2}{\beta^2} + \frac{2}{\beta^2}\mathbb{E}\left[\left\|\boldsymbol{A}(\xi)\right\|^2\left\|\boldsymbol{A}(\xi)\boldsymbol{x}_k - \boldsymbol{A}(\xi)\boldsymbol{x}_{k+1}\right\|^2\right] \\ &\quad\le \frac{4\gamma_k^2 L_{\boldsymbol{A}}^2 \mathcal{D}_{\mathcal{X}}^2}{\beta^2}. \end{aligned} \tag{A.9}$$

Further, we make the following statement about the variance of $g_\beta(\boldsymbol{A}(\xi)\boldsymbol{x}, \xi)$.

$$\begin{aligned}\mathbb{E}\left[\left\|\nabla g_\beta(\boldsymbol{A}(\xi)\boldsymbol{x}) - \nabla G_\beta(\boldsymbol{A}\boldsymbol{x})\right\|^2\right] &= \mathbb{E}\left[\left\|\nabla g_\beta(\boldsymbol{A}(\xi)\boldsymbol{x})\right\|^2 - \left\|\nabla G_\beta(\boldsymbol{A}\boldsymbol{x})\right\|^2\right] \\ &\le \mathbb{E}\left[\left\|\nabla g_\beta(\boldsymbol{A}(\xi)\boldsymbol{x})\right\|^2\right] \\ &\le \frac{1}{\beta^2}\mathbb{E}\left[\left\|\boldsymbol{A}(\xi)\right\|^2\left\|\boldsymbol{A}(\xi)\boldsymbol{x} - \operatorname{proj}_{\boldsymbol{b}(\xi)}(\boldsymbol{A}(\xi)\boldsymbol{x})\right\|^2\right] \\ &\le \frac{1}{\beta^2}\mathbb{E}\left[\left\|\boldsymbol{A}(\xi)\right\|^2\left\|\boldsymbol{A}(\xi)\boldsymbol{x} - \boldsymbol{A}(\xi)\boldsymbol{x}^\star\right\|^2\right]\end{aligned}$$

$$\leq \frac{L_A^2 \mathcal{D}_{\mathcal{X}}^2}{\beta^2}, \tag{A.10}$$

where we used the definition of G_β and $\left\| \boldsymbol{A}(\xi)\boldsymbol{x} - \text{proj}_{\boldsymbol{b}(\xi)}(\boldsymbol{A}(\xi)\boldsymbol{x}) \right\|^2 \leq \| \boldsymbol{A}(\xi)\boldsymbol{x} - \boldsymbol{A}(\xi)\boldsymbol{x}^\star \|^2$.

We observe that the smoothness constant of $g_\beta(\boldsymbol{A}(\xi)\boldsymbol{x})$ is $\frac{L_A}{\beta}$, since

$$\begin{aligned}
&\left\| \nabla g_\beta(\boldsymbol{A}(\xi)\boldsymbol{x}) - \nabla g_\beta(\boldsymbol{A}(\xi)\boldsymbol{y}) \right\| \\
&\qquad = \left\| \frac{\boldsymbol{A}^\top(\xi)}{2\beta} \left(\boldsymbol{A}(\xi)\boldsymbol{x} - \text{proj}_{\boldsymbol{b}(\xi)}(\boldsymbol{A}(\xi)\boldsymbol{x}) \right) - \frac{\boldsymbol{A}^\top(\xi)}{2\beta} \left(\boldsymbol{A}(\xi)\boldsymbol{y} - \text{proj}_{\boldsymbol{b}(\xi)}(\boldsymbol{A}(\xi)\boldsymbol{y}) \right) \right\| \\
&\qquad \leq \frac{L_A}{2\beta} \left\| \boldsymbol{x} - \boldsymbol{y} \right\| + \frac{\| \boldsymbol{A}(\xi) \|}{2\beta} \left\| \text{proj}_{\boldsymbol{b}(\xi)}(\boldsymbol{A}(\xi)\boldsymbol{y}) - \text{proj}_{\boldsymbol{b}(\xi)}(\boldsymbol{A}(\xi)\boldsymbol{x}) \right\| \\
&\qquad \leq \frac{L_A}{2\beta} \left\| \boldsymbol{x} - \boldsymbol{y} \right\| + \frac{\| \boldsymbol{A}(\xi) \|}{2\beta} \left\| \boldsymbol{A}(\xi)\boldsymbol{y} - \boldsymbol{A}(\xi)\boldsymbol{x} \right\| \\
&\qquad \leq \frac{L_A}{\beta} \left\| \boldsymbol{x} - \boldsymbol{y} \right\|.
\end{aligned} \tag{A.11}$$

This implies that $F_\beta(\boldsymbol{x}, \xi)$ is $(L_f + \frac{L_A}{\beta})$-smooth.

A direct consequence of relation (A.5), we get the following useful property for two consecutive (as per the algorithm's iterations) values of the smoothing parameter.

$$g_{\beta_k}(\boldsymbol{A}(\xi)\boldsymbol{x}_k) \leq g_{\beta_{k-1}}(\boldsymbol{A}(\xi)\boldsymbol{x}_k) + \frac{\beta_{k-1} - \beta_k}{2} \left\| \boldsymbol{\lambda}^\star_{\beta_k}(\boldsymbol{A}(\xi)x_k) \right\|^2 \tag{A.12}$$

We end this introductory section with a triplet of technical results that we will refer to in our proofs. First, we restate **Lemma 3.1** of Fercoq et al. [81] for completeness, as we rely on it for translating the convergence rates from the smoothed gap onto objective suboptimality and feasibility.

Lemma A.1 (Restatement of **Lemma 3.1** of Fercoq et al. [81])**.**
Let $(\boldsymbol{x}^\star, \boldsymbol{\lambda}^\star)$ *be a saddle point of* $\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) \coloneqq f(\boldsymbol{x}) + \int \langle \boldsymbol{A}(\xi)\boldsymbol{x}, \boldsymbol{\lambda}(\xi) \rangle - \text{supp}_{\boldsymbol{b}(\xi)}(\boldsymbol{\lambda}(\xi)) \mu(d\xi)$*, where* $\text{supp}_{\mathcal{X}}(\boldsymbol{x}) \coloneqq \sup_{\boldsymbol{z} \in \mathcal{X}} \langle \boldsymbol{z}, \boldsymbol{x} \rangle$*. Then the following holds:*

1. $S_\beta(\boldsymbol{x}) \geq -\frac{\beta}{2} \left\| \boldsymbol{\lambda}^\star \right\|^2$
2. $F(\boldsymbol{x}) - F(\boldsymbol{x}^\star) \geq -\frac{1}{4\beta} \int \text{dist}(\boldsymbol{A}(\xi)\boldsymbol{x}, \boldsymbol{b}(\xi))^2 dP(\xi) - \beta \left\| \boldsymbol{\lambda}^\star \right\|^2$
3. $F(\boldsymbol{x}) - F(\boldsymbol{x}^\star) \leq S_\beta(\boldsymbol{x})$
4. $\int \text{dist}(\boldsymbol{A}(\xi)\boldsymbol{x}, \boldsymbol{b}(\xi))^2 dP(\xi) \leq 4\beta^2 \| \boldsymbol{\lambda}^\star \|^2 + 4\beta S_\beta(\boldsymbol{x})$

Second, we will require an adaptation of **Lemma 17** from Mokhtari, Hassani, and Karbasi [155], which we state and prove below.

Lemma A.2 (Adaptation of **Lemma 17** of Mokhtari, Hassani, and Karbasi [155])**.** *Let $0 < \alpha \leq 1$, $1 \leq \beta \leq 2$, $b \geq 0$, $c > 1$, $t_0 \geq 0$. Let ϕ_k be a sequence of real numbers satisfying*

$$\phi_k \leq (1 - \frac{c}{(k+k_0)^\alpha})\phi_{k-1} + \frac{b}{(k+k_0)^\beta}. \tag{A.13}$$

Then, the sequence ϕ_k converges to zero at the rate

$$\phi_k \leq \frac{Q}{(k+1+k_0)^{\beta-\alpha}}, \tag{A.14}$$

when $\alpha = 1$, $1 < \beta \leq 2$, or $\alpha = \frac{2}{3}$, $\beta = 1$, where $Q = \max(\phi_0(k_0+1)^{\beta-\alpha}, b/(c-1))$.

Proof. We use induction. By the definition of Q, $\phi_0 \leq Q/((k_0+1)^{\beta-\alpha})$, so the base step holds. Now assume it holds for k and check for $k+1$. To ease the notation let $y = k+1+k_0$. When $\alpha = 1$,

$$\phi_{k+1} \leq \left(1 - \frac{c}{y}\right)\frac{Q}{y^{\beta-1}} + \frac{b}{y^\beta} = \left(1 - \frac{c}{y}\right)\frac{Q}{y^{\beta-1}} + \frac{(c-1)Q}{y^\beta} = \frac{Q}{y^{\beta-1}} - \frac{Q}{y^\beta} \leq \frac{Q}{(y+1)^{\beta-1}},$$

where the last step follows since $1 \leq \beta \leq 2$, i.e. $\frac{y-1}{y^\beta} \leq \frac{1}{(y+1)^{\beta-1}} \iff \frac{(y-1)(y+1)^\beta}{(y+1)y^\beta} \leq 1$ and $\frac{(y-1)(y+1)^\beta}{(y+1)y^\beta} \leq \frac{(y-1)(y+1)^2}{(y+1)y^2} \leq 1$, since $\beta \leq 2$.

For general α, β, we get $\frac{1}{y^{\beta-\alpha}} - \frac{1}{y^\beta} \leq \frac{1}{(y+1)^{\beta-\alpha}} \iff \frac{y^\alpha - 1}{y^\beta} \leq \frac{(y+1)^\alpha}{(y+1)^\beta}$. If $\alpha = 2/3, \beta = 1$, then $\frac{y^{2/3}-1}{y} \leq \frac{(y+1)^{2/3}}{(y+1)} \iff \frac{(y^{2/3}-1)(y+1)^{1/3}}{y} \leq 1 \iff \frac{(y^{2/3}-1)^3(y+1)}{y^3} \leq 1 \iff \frac{(y^2-3y^{4/3}+3y^{2/3}-1)(y+1)}{y^3} \leq 1 \iff \frac{(y^3+y^2-3y^{7/3}-3y^{4/3}+3y^{5/3}+3y^{2/3}-y-1}{y^3} \leq 1$ which holds for $y \geq 1$. □

Finally, we require the following technical result.

Lemma A.1. *Let $\rho_n = 1 - \frac{1}{n}$ and $\rho_m = 1 - \frac{1}{m}$, $m, n \geq 1$. We present the following bounds:*

a) $\sum_{i=1}^{k} i\rho_n^i < n^2$ *and* $\sum_{i=1}^{k} i\rho_m^i < m^2$

b) $\sum_{i=1}^{k} i\rho_n^{i/2}\log i < 16n^3$

Proof. **a)** Note that since $\rho_n \in [0,1)$, $\sum_{i=1}^{k} i\rho_n^i \leq \sum_{i=1}^{k} i\rho_n^{i-1}$. Furthermore,

$$\sum_{i=1}^{k} i\rho_n^{i-1} \leq \sum_{i=1}^{\infty} i\rho_n^{i-1} = \sum_{i=1}^{\infty}\frac{\partial\rho_n^i}{\partial\rho_n} = \frac{\partial\sum_{i=1}^{\infty}\rho_n^i}{\partial\rho_n} = \frac{\partial\left[\frac{1}{1-\rho_n} - 1\right]}{\partial\rho_n} = \frac{1}{(1-\rho_n)^2} = n^2, \tag{A.15}$$

where the inequality comes from all terms being non-negative, and the second equality comes from the fact that the infinite sum exists for any $\rho_n \in (-1, 1)$ and is the Taylor series expansion of $\frac{1}{1-\rho_n}$.

b) Use the loose bound $\log i < i + 1$ and the fact that $\sqrt{\rho_n} \in [0, 1)$:

$$\sum_{i=1}^{k} i\rho_n^{i/2} \log i \leq \sum_{i=1}^{\infty} i\rho_n^{i/2} \log i \leq \sum_{i=1}^{\infty} i(i+1)\sqrt{\rho_n}^{i-1} \tag{A.16}$$

$$= \frac{\partial^2 \sum_{i=2}^{\infty} \sqrt{\rho_n}^i}{\partial(\sqrt{\rho_n})^2} = \frac{\partial^2 \frac{1}{1-\sqrt{\rho_n}} - \sqrt{\rho_n} - 1}{\partial(\sqrt{\rho_n})^2} = \frac{2}{(1-\sqrt{\rho_n})^3} \tag{A.17}$$

where the inequalities and equalities follow the same reasoning as in point a). Further noting that

$$\frac{2}{(1-\sqrt{\rho_n})^3} = \frac{2(1+\sqrt{\rho_n})^3}{(1-\rho_n)^3} = 2n^3 \underbrace{(1+\sqrt{\rho_n})^3}_{\leq 2} \leq 16n^3 \square$$

A.1 Analysis of H-1SFW

This section provides the omitted proofs of Section 2.4.3 in the main text. We start with a supporting lemma needed for the proof of Lemma 2.1.

Lemma A.3. *Let* $\boldsymbol{d}_k = (1-\rho_k)\boldsymbol{d}_{k-1} + \rho_k \nabla F_{\beta_k}(\boldsymbol{x}_k, \xi_k)$, $\rho_k \in [0,1]$. *Then, for all* k,

$$\begin{aligned}
&\mathbb{E}_k\left[\|\nabla F_{\beta_k}(\boldsymbol{x}_k) - \boldsymbol{d}_k\|^2\right] \\
&\qquad \leq (1-\frac{\rho_k}{2})\|\nabla F_{\beta_{k-1}}(\boldsymbol{x}_{k-1}) - \boldsymbol{d}_{k-1}\|^2 + 2\rho_k^2\left(\sigma_f^2 + \frac{L_{\boldsymbol{A}}^2 D_{\mathcal{X}}^2}{\beta_k^2}\right) \\
&\qquad + \frac{2}{\rho_k}\left[2L_f^2\gamma_{k-1}^2\mathcal{D}_{\mathcal{X}}^2 + 2L_{\boldsymbol{A}}^2\mathcal{D}_{\mathcal{X}}^2\left[\left(\frac{1}{\beta_k} - \frac{1}{\beta_{k-1}}\right)^2 + \frac{4\gamma_{k-1}}{\beta_{k-1}}\left|\frac{1}{\beta_k} - \frac{1}{\beta_{k-1}}\right| + \frac{4\gamma_{k-1}^2}{\beta_{k-1}^2}\right]\right], \quad \text{(A.18)}
\end{aligned}$$

where $\mathbb{E}_k[\cdot] = \mathbb{E}[\cdot|\mathcal{F}_k]$ *and* $\mathcal{F}_k$ *is a* σ*-algebra measuring all sources of randomness up to step* k.

Proof. We use the definition $\boldsymbol{d}_k = (1-\rho_k)\boldsymbol{d}_{k-1} + \rho_k \nabla F_{\beta_k}(\boldsymbol{x}_k, \xi_k)$ to write the difference

$$\begin{aligned}
&\|\nabla F_{\beta_k}(\boldsymbol{x}_k) - \boldsymbol{d}_k\|^2 \\
&\quad = \|\nabla F_{\beta_k}(\boldsymbol{x}_k) - (1-\rho_k)\boldsymbol{d}_{k-1} - \rho_k \nabla F_{\beta_k}(\boldsymbol{x}_k, \xi_k)\|^2 \\
&\quad = \|\nabla F_{\beta_k}(\boldsymbol{x}_k) + (1-\rho_k)\nabla F_{\beta_{k-1}}(\boldsymbol{x}_{k-1}) - (1-\rho_k)\nabla F_{\beta_{k-1}}(\boldsymbol{x}_{k-1}) - (1-\rho_k)\boldsymbol{d}_{k-1} - \rho_k \nabla F_{\beta_k}(\boldsymbol{x}_k, \xi_k)\|^2 \\
&\quad = \|\rho_k(\nabla F_{\beta_k}(\boldsymbol{x}_k) - \nabla F_{\beta_k}(\boldsymbol{x}_k, \xi_k)) + (1-\rho_k)(\nabla F_{\beta_k}(\boldsymbol{x}_k) - \nabla F_{\beta_{k-1}}(\boldsymbol{x}_{k-1})) \\
&\qquad\qquad + (1-\rho_k)(\nabla F_{\beta_{k-1}}(\boldsymbol{x}_{k-1}) - \boldsymbol{d}_{k-1})\|^2 \\
&\quad = \rho_k^2\|\nabla F_{\beta_k}(\boldsymbol{x}_k) - \nabla F_{\beta_k}(\boldsymbol{x}_k, \xi_k)\|^2 + (1-\rho_k)^2\|\nabla F_{\beta_k}(\boldsymbol{x}_k) - \nabla F_{\beta_{k-1}}(\boldsymbol{x}_{k-1})\|^2 \\
&\qquad\qquad + (1-\rho_k)^2\|\nabla F_{\beta_{k-1}}(\boldsymbol{x}_{k-1}) - \boldsymbol{d}_{k-1}\|^2 \\
&\qquad\qquad + 2\rho_k(1-\rho_k)\langle \nabla F_{\beta_k}(\boldsymbol{x}_k) - \nabla F_{\beta_k}(\boldsymbol{x}_k, \xi_k), \nabla F_{\beta_k}(\boldsymbol{x}_k) - \nabla F_{\beta_{k-1}}(\boldsymbol{x}_{k-1})\rangle \\
&\qquad\qquad + 2\rho_k(1-\rho_k)\langle \nabla F_{\beta_k}(\boldsymbol{x}_k) - \nabla F_{\beta_k}(\boldsymbol{x}_k, \xi_k), \nabla F_{\beta_{k-1}}(\boldsymbol{x}_{k-1}) - \boldsymbol{d}_{k-1}\rangle \\
&\qquad\qquad + 2(1-\rho_k)^2\langle \nabla F_{\beta_k}(\boldsymbol{x}_k) - \nabla F_{\beta_{k-1}}(\boldsymbol{x}_{k-1}), \nabla F_{\beta_{k-1}}(\boldsymbol{x}_{k-1}) - \boldsymbol{d}_{k-1}\rangle
\end{aligned}$$

We remark that $\mathbb{E}_k\left[\nabla F_{\beta_k}(\boldsymbol{x}_k, \xi_k)\right] = \nabla F_{\beta_k}(\boldsymbol{x}_k)$, so the first two linear terms are 0. We now take expectations conditioned on $\mathcal{F}_k$,

$$\begin{aligned}
&\mathbb{E}_k\left[\|\nabla F_{\beta_k}(\boldsymbol{x}_k) - \boldsymbol{d}_k\|^2\right] \\
&\qquad = \rho_k^2\mathbb{E}_k\left[\|\nabla F_{\beta_k}(\boldsymbol{x}_k) - \nabla F_{\beta_k}(\boldsymbol{x}_k, \xi_k)\|^2\right] + (1-\rho_k)^2\|\nabla F_{\beta_k}(\boldsymbol{x}_k) - \nabla F_{\beta_{k-1}}(\boldsymbol{x}_{k-1})\|^2 \\
&\qquad\qquad + (1-\rho_k)^2\|\nabla F_{\beta_{k-1}}(\boldsymbol{x}_{k-1}) - \boldsymbol{d}_{k-1}\|^2
\end{aligned}$$

$$+ 2(1-\rho_k)^2\langle \nabla F_{\beta_k}(\boldsymbol{x}_k) - \nabla F_{\beta_{k-1}}(\boldsymbol{x}_{k-1}), \nabla F_{\beta_{k-1}}(\boldsymbol{x}_{k-1}) - \boldsymbol{d}_{k-1}\rangle \tag{A.19}$$

Invoking the variance bound (A.10), we have

$$\begin{aligned}
&\mathbb{E}_k\left[\|\nabla F_{\beta_k}(\boldsymbol{x}_k) - \nabla F_{\beta_k}(\boldsymbol{x}_k, \xi_k)\|^2\right] \\
&\qquad \le 2\mathbb{E}_k\left[\|\nabla f(\boldsymbol{x}_k) - \nabla f(\boldsymbol{x}_k, \xi_k)\|^2\right] + 2\mathbb{E}_k\left[\|G_{\beta_k}(\boldsymbol{A}\boldsymbol{x}_k) - \nabla g_{\beta_k}(\boldsymbol{A}(\xi)\boldsymbol{x}_k, \xi_k)\|^2\right] \\
&\qquad \le 2\left(\sigma_f^2 + \frac{L_{\boldsymbol{A}}^2 D_{\mathcal{X}}^2}{\beta_k^2}\right).
\end{aligned}$$

For the linear term, we use Young's inequality for some $\sigma_k > 0$ to get

$$\begin{aligned}
&2(1-\rho_k)^2\langle \nabla F_{\beta_k}(\boldsymbol{x}_k) - \nabla F_{\beta_{k-1}}(\boldsymbol{x}_{k-1}), \nabla F_{\beta_{k-1}}(\boldsymbol{x}_{k-1}) - \boldsymbol{d}_{k-1}\rangle \\
&\qquad \le (1-\rho_k)^2\sigma_k\|\nabla F_{\beta_{k-1}}(\boldsymbol{x}_{k-1}) - \boldsymbol{d}_{k-1}\|^2 + (1-\rho_k)^2(1/\sigma_k)\|\nabla F_{\beta_k}(\boldsymbol{x}_k) - \nabla F_{\beta_{k-1}}(\boldsymbol{x}_{k-1})\|^2.
\end{aligned}$$

For the $\|\nabla F_{\beta_k}(\boldsymbol{x}_k) - \nabla F_{\beta_{k-1}}(\boldsymbol{x}_{k-1})\|^2$ term, we use the iterate update rule and observation (A.9) to get

$$\begin{aligned}
&\|\nabla F_{\beta_k}(\boldsymbol{x}_k) - \nabla F_{\beta_{k-1}}(\boldsymbol{x}_{k-1})\|^2 \\
&\qquad = \|\nabla f(\boldsymbol{x}_k) - \nabla f(\boldsymbol{x}_{k-1}) + \nabla G_{\beta_k}(\boldsymbol{A}\boldsymbol{x}_k) - \nabla G_{\beta_{k-1}}(\boldsymbol{A}\boldsymbol{x}_{k-1})\|^2 \\
&\qquad \le 2\|\nabla f(\boldsymbol{x}_k) - \nabla f(\boldsymbol{x}_{k-1})\|^2 + 2\|\nabla G_{\beta_k}(\boldsymbol{A}\boldsymbol{x}_k) - \nabla G_{\beta_{k-1}}(\boldsymbol{A}\boldsymbol{x}_k) + \nabla G_{\beta_{k-1}}(\boldsymbol{A}\boldsymbol{x}_k) - \nabla G_{\beta_{k-1}}(\boldsymbol{A}\boldsymbol{x}_{k-1})\|^2 \\
&\qquad \le 2L_f^2\|\boldsymbol{x}_k - \boldsymbol{x}_{k-1}\|^2 + 2\|\nabla G_{\beta_k}(\boldsymbol{A}\boldsymbol{x}_k) - \nabla G_{\beta_{k-1}}(\boldsymbol{A}\boldsymbol{x}_k)\|^2 + 2\|\nabla G_{\beta_{k-1}}(\boldsymbol{A}\boldsymbol{x}_k) - \nabla G_{\beta_{k-1}}(\boldsymbol{A}\boldsymbol{x}_{k-1})\|^2 \\
&\qquad\qquad + 4\|\nabla G_{\beta_k}(\boldsymbol{A}\boldsymbol{x}_k) - \nabla G_{\beta_{k-1}}(\boldsymbol{A}\boldsymbol{x}_k)\|\|\nabla G_{\beta_{k-1}}(\boldsymbol{A}\boldsymbol{x}_k) - \nabla G_{\beta_{k-1}}(\boldsymbol{A}\boldsymbol{x}_{k-1})\| \\
&\qquad \le 2L_f^2\gamma_{k-1}^2\mathcal{D}_{\mathcal{X}}^2 + 2L_{\boldsymbol{A}}^2\mathcal{D}_{\mathcal{X}}^2\left(\frac{1}{\beta_k} - \frac{1}{\beta_{k-1}}\right)^2 + \frac{8\gamma_{k-1}^2 L_{\boldsymbol{A}}^2\mathcal{D}_{\mathcal{X}}^2}{\beta_{k-1}^2} + 8\gamma_{k-1}\left|\frac{1}{\beta_k} - \frac{1}{\beta_{k-1}}\right|\frac{L_{\boldsymbol{A}}^2\mathcal{D}_{\mathcal{X}}^2}{\beta_{k-1}}.
\end{aligned}$$

Putting everything back into (A.19) we obtain

$$\begin{aligned}
&\mathbb{E}_k\left[\|\nabla F_{\beta_k}(\boldsymbol{x}_k) - \boldsymbol{d}_k\|^2\right] \\
&= \rho_k^2\mathbb{E}_k\left[\|\nabla F_{\beta_k}(\boldsymbol{x}_k) - \nabla F_{\beta_k}(\boldsymbol{x}_k, \xi_k)\|^2\right] + (1-\rho_k)^2(1+\sigma_k^{-1})\|\nabla F_{\beta_k}(\boldsymbol{x}_k) - \nabla F_{\beta_{k-1}}(\boldsymbol{x}_{k-1})\|^2 \\
&\qquad\qquad + (1-\rho_k)^2(1+\sigma_k)\|\nabla F_{\beta_{k-1}}(\boldsymbol{x}_{k-1}) - \boldsymbol{d}_{k-1}\|^2
\end{aligned}$$

$$\leq (1-\rho_k)^2(1+\sigma_k)\|\nabla F_{\beta_{k-1}}(\boldsymbol{x}_{k-1}) - \boldsymbol{d}_{k-1}\|^2 + 2\rho_k^2\left(\sigma_f^2 + \frac{L_{\boldsymbol{A}}^2\mathcal{D}_{\mathcal{X}}^2}{\beta_k^2}\right)$$

$$+ (1-\rho_k)^2(1+\sigma_k^{-1})\left[2L_f^2\gamma_{k-1}^2\mathcal{D}_{\mathcal{X}}^2 + 2L_{\boldsymbol{A}}^2\mathcal{D}_{\mathcal{X}}^2\left[\left(\frac{1}{\beta_k} - \frac{1}{\beta_{k-1}}\right)^2 + \frac{4\gamma_{k-1}}{\beta_{k-1}}\left|\frac{1}{\beta_k} - \frac{1}{\beta_{k-1}}\right| + \frac{4\gamma_{k-1}^2}{\beta_{k-1}^2}\right]\right].$$

Using the facts $\rho_k \leq 1$, $(1-\rho_k)^2 \leq (1-\rho_k)$, $(1-\rho_k)(1+\frac{\rho_k}{2}) \leq (1-\rho_k/2)$, $(1-\rho_k)(1+\frac{2}{\rho_k}) \leq \frac{2}{\rho_k}$ and setting $\sigma_k \coloneqq \frac{\rho_k}{2}$, we get

$$\mathbb{E}_k\left[\|\nabla F_{\beta_k}(\boldsymbol{x}_k) - \boldsymbol{d}_k\|^2\right]$$

$$\leq \left(1 - \frac{\rho_k}{2}\right)\|\nabla F_{\beta_{k-1}}(\boldsymbol{x}_{k-1}) - \boldsymbol{d}_{k-1}\|^2 + 2\rho_k^2\left(\sigma_f^2 + \frac{L_{\boldsymbol{A}}^2 D_{\mathcal{X}}^2}{\beta_k^2}\right)$$

$$+ \frac{2}{\rho_k}\left[2L_f^2\gamma_{k-1}^2\mathcal{D}_{\mathcal{X}}^2 + 2L_{\boldsymbol{A}}^2\mathcal{D}_{\mathcal{X}}^2\left[\left(\frac{1}{\beta_k} - \frac{1}{\beta_{k-1}}\right)^2 + \frac{4\gamma_{k-1}}{\beta_{k-1}}\left|\frac{1}{\beta_k} - \frac{1}{\beta_{k-1}}\right| + \frac{4\gamma_{k-1}^2}{\beta_{k-1}^2}\right]\right] \square$$

A.1.1 Proof of Lemma 2.1

Lemma 2.1. *Let* $\rho_k = \frac{3}{(k+5)^{2/3}}$, $\gamma_k = \frac{2}{k+1}$, $\beta_k = \frac{\beta_0}{(k+1)^{1/6}}$, $\beta_0 > 0$ *in Algorithm 2.1. Then, for all k,*

$$\mathbb{E}\left[\|\nabla F_{\beta_k}(\boldsymbol{x}_k) - \boldsymbol{d}_k\|^2\right] \leq \frac{C_1}{(k+5)^{1/3}},$$

where $C_1 \coloneqq \max\left\{6^{1/3}\|\nabla F_{\beta_0}(\boldsymbol{x}_0) - \boldsymbol{d}_0\|^2, 2\left[18\sigma_f^2 + 112L_f^2\mathcal{D}_{\mathcal{X}}^2 + \frac{522L_{\boldsymbol{A}}^2\mathcal{D}_{\mathcal{X}}^2}{\beta_0^2}\right]\right\}$.

Proof. We apply the expectation with respect to the whole history to (A.18),

$$\begin{aligned}\mathbb{E}\left[\|\nabla F_{\beta_k}(\boldsymbol{x}_k) - \boldsymbol{d}_k\|^2\right] \leq (1-\frac{\rho_k}{2})\mathbb{E}\left[\|\nabla F_{\beta_{k-1}}(\boldsymbol{x}_{k-1}) - \boldsymbol{d}_{k-1}\|^2\right] + 2\rho_k^2\left(\sigma_f^2 + \frac{L_{\boldsymbol{A}}^2\mathcal{D}_{\mathcal{X}}^2}{\beta_k^2}\right)\\ + \frac{2}{\rho_k}\left[2L_f^2\gamma_{k-1}^2\mathcal{D}_{\mathcal{X}}^2 + 2L_{\boldsymbol{A}}^2\mathcal{D}_{\mathcal{X}}^2\left[\left(\frac{1}{\beta_k} - \frac{1}{\beta_{k-1}}\right)^2 + \frac{4\gamma_{k-1}}{\beta_{k-1}}\left|\frac{1}{\beta_k} - \frac{1}{\beta_{k-1}}\right| + \frac{4\gamma_{k-1}^2}{\beta_{k-1}^2}\right]\right],\end{aligned}$$

and estimate the rate of $\left|\frac{1}{\beta_k} - \frac{1}{\beta_{k-1}}\right|$

$$\begin{aligned}0 &\leq \frac{1}{\beta_k} - \frac{1}{\beta_{k-1}}\\ &= \frac{(k+1)^{1/6} - (k)^{1/6}}{\beta_0}\\ &= \frac{1}{\beta_0\left[(k+1)^{5/6} + (k+1)^{4/6}k^{1/6} + (k+1)^{3/6}k^{2/6} + (k+1)^{2/6}k^{3/6} + (k+1)^{1/6}k^{4/6} + k^{5/6}\right]}\\ &\leq \frac{1}{6\beta_0 k^{5/6}}.\end{aligned}$$

Replacing the parameter rates, we further get

$$\begin{aligned}&\mathbb{E}\left[\|\nabla F_{\beta_k}(\boldsymbol{x}_k) - \boldsymbol{d}_k\|^2\right]\\ &\leq \left(1 - \frac{3}{2(k+5)^{2/3}}\right)\mathbb{E}\left[\|\nabla F_{\beta_{k-1}}(\boldsymbol{x}_{k-1}) - \boldsymbol{d}_{k-1}\|^2\right] + \frac{18}{(k+5)^{4/3}}\left(\sigma_f^2 + \frac{L_{\boldsymbol{A}}^2\mathcal{D}_{\mathcal{X}}^2(k+1)^{2/6}}{\beta_0^2}\right)\\ &\qquad + \frac{2(k+5)^{2/3}}{3}\left[\frac{8L_f^2\mathcal{D}_{\mathcal{X}}^2}{k^2} + \frac{2L_{\boldsymbol{A}}^2\mathcal{D}_{\mathcal{X}}^2}{\beta_0^2}\left(\frac{1}{36k^{10/6}} + \frac{4}{3k^{10/6}} + \frac{16}{k^{10/6}}\right)\right]\\ &\leq \left(1 - \frac{3}{2(k+5)^{2/3}}\right)\mathbb{E}\left[\|\nabla F_{\beta_{k-1}}(\boldsymbol{x}_{k-1}) - \boldsymbol{d}_{k-1}\|^2\right] + \frac{18\sigma_f^2}{k+5}\end{aligned}$$

$$+\frac{18L_A^2\mathcal{D}_{\mathcal{X}}^2}{\beta_0^2(k+5)}+\frac{2(k+5)^{2/3}}{3k^{10/6}}\left(8L_f^2\mathcal{D}_{\mathcal{X}}^2+\frac{36L_A^2\mathcal{D}_{\mathcal{X}}^2}{\beta_0^2}\right)$$

$$\leq\left(1-\frac{3}{2(k+5)^{2/3}}\right)\mathbb{E}\left[\|\nabla F_{\beta_{k-1}}(\boldsymbol{x}_{k-1})-\boldsymbol{d}_{k-1}\|^2\right]+\frac{18\sigma_f^2}{k+5}$$

$$+\frac{18L_A^2\mathcal{D}_{\mathcal{X}}^2}{\beta_0^2(k+5)}+\frac{14}{k+5}\left(8L_f^2\mathcal{D}_{\mathcal{X}}^2+\frac{36L_A^2\mathcal{D}_{\mathcal{X}}^2}{\beta_0^2}\right) \tag{A.20}$$

$$=\left(1-\frac{3}{2(k+5)^{2/3}}\right)\mathbb{E}\left[\|\nabla F_{\beta_{k-1}}(\boldsymbol{x}_{k-1})-\boldsymbol{d}_{k-1}\|^2\right]+\frac{1}{k+5}\left(18\sigma_f^2+112L_f^2\mathcal{D}_{\mathcal{X}}^2+\frac{522L_A^2\mathcal{D}_{\mathcal{X}}^2}{\beta_0^2}\right),$$

where line (A.20) follows from the fact that

$$\frac{(k+5)^{2/3}}{k^{10/6}}=\frac{(k+5)^{4/6}}{k^{10/6}}\frac{(k+5)^{6/6}}{(k+5)^{6/6}}=\left(1+\frac{5}{k}\right)^{4/6+6/6}\frac{1}{(k+5)^{6/6}}=\left(1+\frac{5}{k}\right)^{5/3}\frac{1}{k+5}<\frac{6^{5/3}}{k+5}<\frac{21}{k+5}$$

We can now invoke Lemma A.3 for $b=18\sigma_f^2+112L_f^2\mathcal{D}_{\mathcal{X}}^2+\frac{522L_A^2\mathcal{D}_{\mathcal{X}}^2}{\beta_0^2}$ and $c=\frac{3}{2}$, $\alpha=\frac{2}{3}$ and $\beta=1$, $k_0=5$ to conclude the result. □

A.1.2 Proof of Theorem 2.1

Theorem 2.1. *Consider Algorithm 2.1 with parameters* $\rho_k = \frac{3}{(k+5)^{2/3}}$, $\gamma_k = \frac{2}{k+1}$, $\beta_k = \frac{\beta_0}{(k+1)^{1/6}}$, $\beta_0 > 0$ *(identical to Lemma 2.1). Then, for all* k,

$$\mathbb{E}\left[S_{\beta_k}(\boldsymbol{x}_{k+1})\right] \leq \frac{C_2}{k^{1/6}},$$

where $C_2 \coloneqq \max\left\{S_0(\boldsymbol{x}_1),\ b = 2\mathcal{D}_{\mathcal{X}}\sqrt{C_1} + 2\mathcal{D}_{\mathcal{X}}^2\left(L_f + \frac{L_A}{\beta_0}\right)\right\}$ *and* C_1 *is defined in Lemma 2.1.*

Proof. We essentially follow the steps for proving **Theorem 9** of [143], modified to suit our setting. Using observation (A.11) and the definition of $\mathcal{D}_{\mathcal{X}}$:

$$\begin{aligned}
F_{\beta_k}(\boldsymbol{x}_{k+1}) &= \mathbb{E}_{k+1}\left[F_{\beta_k}(\boldsymbol{x}_{k+1}, \xi)\right] \\
&\leq \mathbb{E}_{k+1}\left[F_{\beta_k}(\boldsymbol{x}_k, \xi) + \langle \nabla F_{\beta_k}(\boldsymbol{x}_k, \xi), \boldsymbol{x}_{k+1} - \boldsymbol{x}_k\rangle + \frac{1}{2}\left(L_f + \frac{L_{\boldsymbol{A}}}{\beta_k}\right)\| \boldsymbol{x}_{k+1} - \boldsymbol{x}_k \|^2\right] \\
&\leq F_{\beta_k}(\boldsymbol{x}_k) + \gamma_k \langle \nabla F_{\beta_k}(\boldsymbol{x}_k), \boldsymbol{w}_k - \boldsymbol{x}_k\rangle + \frac{\gamma_k^2}{2}\left(L_f + \frac{L_{\boldsymbol{A}}}{\beta_k}\right)\mathcal{D}_{\mathcal{X}}^2 \qquad \text{(A.21)}
\end{aligned}$$

We treat the term $\langle \nabla F_{\beta_k}(\boldsymbol{x}_k), \boldsymbol{w}_k - \boldsymbol{x}_k\rangle$ separately, using the fact that $\boldsymbol{w}_k \in \arg\min_{\boldsymbol{y}} \langle \boldsymbol{d}_k, \boldsymbol{y}\rangle$ and the definition of $\mathcal{D}_{\mathcal{X}}$:

$$\begin{aligned}
\langle \nabla F_{\beta_k}(\boldsymbol{x}_k), \boldsymbol{w}_k - \boldsymbol{x}_k\rangle &= \langle \nabla F_{\beta_k}(\boldsymbol{x}_k) - \boldsymbol{d}_k, \boldsymbol{w}_k - \boldsymbol{x}_k\rangle + \langle \boldsymbol{d}_k, \boldsymbol{w}_k - \boldsymbol{x}_k\rangle \\
&= \langle \nabla F_{\beta_k}(\boldsymbol{x}_k) - \boldsymbol{d}_k, \boldsymbol{w}_k - \boldsymbol{x}^\star\rangle + \langle \nabla F_{\beta_k}(\boldsymbol{x}_k) - \boldsymbol{d}_k, \boldsymbol{x}^\star - \boldsymbol{x}_k\rangle + \langle \boldsymbol{d}_k, \boldsymbol{w}_k - \boldsymbol{x}_k\rangle \\
&\leq \langle \nabla F_{\beta_k}(\boldsymbol{x}_k) - \boldsymbol{d}_k, \boldsymbol{w}_k - \boldsymbol{x}^\star\rangle + \langle \nabla F_{\beta_k}(\boldsymbol{x}_k) - \boldsymbol{d}_k, \boldsymbol{x}^\star - \boldsymbol{x}_k\rangle + \langle \boldsymbol{d}_k, \boldsymbol{x}^\star - \boldsymbol{x}_k\rangle \\
&= \langle \nabla F_{\beta_k}(\boldsymbol{x}_k) - \boldsymbol{d}_k, \boldsymbol{w}_k - \boldsymbol{x}^\star\rangle + \langle \nabla F_{\beta_k}(\boldsymbol{x}_k), \boldsymbol{x}^\star - \boldsymbol{x}_k\rangle \\
&\leq \|\nabla F_{\beta_k}(\boldsymbol{x}_k) - \boldsymbol{d}_k\| \|\boldsymbol{w}_k - \boldsymbol{x}^\star\| + \langle \nabla F_{\beta_k}(\boldsymbol{x}_k), \boldsymbol{x}^\star - \boldsymbol{x}_k\rangle \\
&\leq \|\nabla F_{\beta_k}(\boldsymbol{x}_k) - \boldsymbol{d}_k\| \mathcal{D}_{\mathcal{X}} + \langle \nabla F_{\beta_k}(\boldsymbol{x}_k), \boldsymbol{x}^\star - \boldsymbol{x}_k\rangle \\
&= \|\nabla F_{\beta_k}(\boldsymbol{x}_k) - \boldsymbol{d}_k\| \mathcal{D}_{\mathcal{X}} + \langle \nabla f(\boldsymbol{x}_k) + \nabla_{\boldsymbol{x}} G_{\beta_k}(\boldsymbol{A}\boldsymbol{x}_k), \boldsymbol{x}^\star - \boldsymbol{x}_k\rangle. \qquad \text{(A.22)}
\end{aligned}$$

Using property (A.4) we observe that

$$\begin{aligned}
\langle \nabla_{\boldsymbol{x}} G_{\beta_k}(\boldsymbol{A}\boldsymbol{x}_k), \boldsymbol{x}^\star - \boldsymbol{x}_k\rangle &= \mathbb{E}_k\left[\langle \nabla_{\boldsymbol{x}} g_{\beta_k}(\boldsymbol{A}(\xi)\boldsymbol{x}_k), \boldsymbol{x}^\star - \boldsymbol{x}_k\rangle\right] \\
&= \mathbb{E}_k\left[\langle \nabla g_{\beta_k}(\boldsymbol{A}(\xi)\boldsymbol{x}_k), \boldsymbol{A}(\xi)\boldsymbol{x}^\star - \boldsymbol{A}(\xi)\boldsymbol{x}_k\rangle\right] \\
&\leq \mathbb{E}_k\left[g(\boldsymbol{A}(\xi)\boldsymbol{x}^\star) - g_{\beta_k}(\boldsymbol{A}(\xi)\boldsymbol{x}_k) - \frac{\beta_k}{2}\left\| \boldsymbol{\lambda}^\star_{\beta_k}(\boldsymbol{A}(\xi)\boldsymbol{x}_k)\right\|^2\right] \\
&= G(\boldsymbol{A}\boldsymbol{x}^\star) - G_{\beta_k}(\boldsymbol{A}\boldsymbol{x}_k) - \frac{\beta_k}{2}\mathbb{E}_k\left[\left\| \boldsymbol{\lambda}^\star_{\beta_k}(\boldsymbol{A}(\xi)\boldsymbol{x}_k)\right\|^2\right].
\end{aligned}$$

Using the above and the convexity of f, we obtain

$$\begin{aligned}\langle \nabla F_{\beta_k}(\boldsymbol{x}_k), \boldsymbol{w}_k - \boldsymbol{x}_k\rangle \\ \leq \|\nabla F_{\beta_k}(\boldsymbol{x}_k) - \boldsymbol{d}_k\|\mathcal{D}_{\mathcal{X}} + f^\star + G(\boldsymbol{A}\boldsymbol{x}^\star) \underbrace{- f(\boldsymbol{x}_k) - G_{\beta_k}(\boldsymbol{A}\boldsymbol{x}_k)}_{=-F_{\beta_k}(\boldsymbol{x}_k)} - \frac{\beta_k}{2}\mathbb{E}_k\left[\left\| \boldsymbol{\lambda}^\star_{\beta_k}(\boldsymbol{A}(\xi)\boldsymbol{x}_k)\right\|^2\right].\end{aligned}$$

Substituting everything back into Equation (A.21) and noting that $G(\boldsymbol{A}\boldsymbol{x}^\star) = 0$:

$$\begin{aligned}F_{\beta_k}(\boldsymbol{x}_{k+1}) \leq (1-\gamma_k)F_{\beta_k}(\boldsymbol{x}_k) + \gamma_k\|\nabla F_{\beta_k}(\boldsymbol{x}_k) - \boldsymbol{d}_k\|\mathcal{D}_{\mathcal{X}} + \gamma_k f^\star \\ - \frac{\gamma_k\beta_k}{2}\mathbb{E}_k\left[\left\| \boldsymbol{\lambda}^\star_{\beta_k}(\boldsymbol{A}(\xi)\boldsymbol{x}_k)\right\|^2\right] + \frac{\gamma_k^2}{2}\left(L_f + \frac{L_{\boldsymbol{A}}}{\beta_k}\right)\mathcal{D}_{\mathcal{X}}^2.\end{aligned}$$

Using observation (A.12) we note that

$$\begin{aligned}F_{\beta_k}(\boldsymbol{x}_k) &= \mathbb{E}_k\left[f(\boldsymbol{x}_k,\xi) + g_{\beta_k}(\boldsymbol{A}(\xi)\boldsymbol{x}_k)\right] \\ &\leq \mathbb{E}_k\left[f(\boldsymbol{x}_k,\xi) + g_{\beta_{k-1}}(\boldsymbol{A}(\xi)\boldsymbol{x}_k) + \frac{\beta_{k-1}-\beta_k}{2}\left\| \boldsymbol{\lambda}^\star_{\beta_k}(\boldsymbol{A}(\xi)\boldsymbol{x}_k)\right\|^2\right] \\ &= F_{\beta_{k-1}}(\boldsymbol{x}_k) + \mathbb{E}_k\left[\frac{\beta_{k-1}-\beta_k}{2}\left\| \boldsymbol{\lambda}^\star_{\beta_k}(\boldsymbol{A}(\xi)\boldsymbol{x}_k)\right\|^2\right].\end{aligned}$$

Substituting the above, we obtain

$$\begin{aligned}F_{\beta_k}(\boldsymbol{x}_{k+1}) &\leq (1-\gamma_k)F_{\beta_{k-1}}(\boldsymbol{x}_k) + \gamma_k\|\nabla F_{\beta_k}(\boldsymbol{x}_k) - \boldsymbol{d}_k\|\mathcal{D}_{\mathcal{X}} + \gamma_k f^\star \\ &\quad + \frac{(1-\gamma_k)(\beta_{k-1}-\beta_k) - \gamma_k\beta_k}{2}\mathbb{E}_k\left[\left\| \boldsymbol{\lambda}^\star_{\beta_k}(\boldsymbol{A}(\xi)\boldsymbol{x}_k)\right\|^2\right] + \frac{\gamma_k^2}{2}\left(L_f + \frac{L_{\boldsymbol{A}}}{\beta_k}\right)\mathcal{D}_{\mathcal{X}}^2 \\ &\leq (1-\gamma_k)F_{\beta_{k-1}}(\boldsymbol{x}_k) + \gamma_k\|\nabla F_{\beta_k}(\boldsymbol{x}_k) - \boldsymbol{d}_k\|\mathcal{D}_{\mathcal{X}} + \gamma_k f^\star + \frac{\gamma_k^2}{2}\left(L_f + \frac{L_{\boldsymbol{A}}}{\beta_k}\right)\mathcal{D}_{\mathcal{X}}^2, \qquad \text{(A.23)}\end{aligned}$$

where the last line comes from the fact that $(1-\gamma_k)(\beta_{k-1}-\beta_k) - \gamma_k\beta_k < 0$:

$$\begin{aligned}(1-\gamma_k)(\beta_{k-1}-\beta_k) - \gamma_k\beta_k = \beta_{k-1} - \beta_k - \gamma_k\beta_{k-1} &= \frac{\beta_0}{k^{1/6}} - \frac{\beta_0}{(k+1)^{1/6}} - \frac{2\beta_0}{(k+1)k^{1/6}} \\ &= \frac{\beta_0}{k^{1/6}}\left(1 - \frac{k^{1/6}}{(k+1)^{1/6}} - \frac{2}{k+1}\right) \\ &= \frac{\beta_0}{k^{1/6}}\left(\frac{k-1}{k+1} - \frac{k^{1/6}}{(k+1)^{1/6}}\right)\end{aligned}$$

$$< \frac{\beta_0}{k^{1/6}} \left(\underbrace{\frac{k}{k+1}}_{\in(0,1)} - \frac{k^{1/6}}{(k+1)^{1/6}} \right)$$
$$< 0.$$

Starting from Equation (A.23) and subtracting $f^\star$ from both sides, noting the definition of $S_{\beta_k}(\boldsymbol{x}) \coloneqq F_\beta(\boldsymbol{x}) - f^\star$ and taking the expectation on both sides we get

$$\mathbb{E}\left[S_{\beta_k}(\boldsymbol{x}_{k+1})\right] \leq (1-\gamma_k)\mathbb{E}\left[S_{\beta_{k-1}}(\boldsymbol{x}_k)\right] + \frac{\gamma_k^2}{2} D_{\mathcal{X}}^2 \left(L_f + \frac{L_{\boldsymbol{A}}}{\beta_k}\right) + \gamma_k \mathbb{E}\left[\|\nabla F_{\beta_k}(\boldsymbol{x}_k) - \boldsymbol{d}_k\|\right] \mathcal{D}_{\mathcal{X}}. \quad \text{(A.24)}$$

Replacing the parameter rates for the second term, we bound by

$$\begin{aligned} \frac{\gamma_k^2}{2}\mathcal{D}_{\mathcal{X}}^2 (L_f + \frac{L_{\boldsymbol{A}}}{\beta_k}) &= \frac{2\mathcal{D}_{\mathcal{X}}^2 L_f}{k^2} + \frac{2\mathcal{D}_{\mathcal{X}}^2 L_A}{\beta_0 k^{11/6}} \\ &\leq \frac{2\mathcal{D}_{\mathcal{X}}^2}{k^{7/6}} \left(L_f + \frac{L_{\boldsymbol{A}}}{\beta_0}\right) \end{aligned}$$

For the last term we use the parameter rates and Lemma 2.1 together with Jensen's inequality $\mathbb{E}\left[\|\nabla F_{\beta_k}(\boldsymbol{x}_k) - \boldsymbol{d}_k\|\right] = \sqrt{\mathbb{E}\left[\|\nabla F_{\beta_k}(\boldsymbol{x}_k) - \boldsymbol{d}_k\|\right]^2} \leq \sqrt{\mathbb{E}\left[\|\nabla F_{\beta_k}(\boldsymbol{x}_k) - \boldsymbol{d}_k\|^2\right]}$ to get

$$\gamma_k \mathcal{D}_{\mathcal{X}} \mathbb{E}\left[\|\nabla F_{\beta_k}(\boldsymbol{x}_k) - \boldsymbol{d}_k\|\right] = \frac{2\mathcal{D}_{\mathcal{X}}}{k+1} \frac{\sqrt{C_1}}{(k+5)^{1/6}} \leq \frac{2\mathcal{D}_{\mathcal{X}}\sqrt{C_1}}{k^{7/6}},$$

Substituting the above into (A.24), we get

$$\mathbb{E}\left[S_{\beta_k}(\boldsymbol{x}_{k+1})\right] \leq \left(1 - \frac{2}{k}\right) \mathbb{E}\left[S_{\beta_{k-1}}(\boldsymbol{x}_k)\right] + \frac{2\mathcal{D}_{\mathcal{X}}\sqrt{C_1} + 2\mathcal{D}_{\mathcal{X}}^2 \left(L_f + \frac{L_A}{\beta_0}\right)}{k^{7/6}}.$$

Finally, we use Lemma A.2 with $\alpha = 1$, $\beta = 7/6$, $c = 2$, $b = 2\mathcal{D}_{\mathcal{X}}\sqrt{C_1} + 2\mathcal{D}_{\mathcal{X}}^2 \left(L_f + \frac{L_A}{\beta_0}\right)$ to arrive at the statement. □

A.1.3 Proof of Corollary 2.1

Corollary 2.1. *The expected convergence in terms of objective suboptimality and feasibility of Algorithm 2.1 is, respectively,*

$$\left|\mathbb{E}\left[f(\boldsymbol{x}_k,\xi)\right]-f^\star\right| \qquad \in \mathcal{O}\left(k^{-1/6}\right)$$

$$\sqrt{\mathbb{E}\left[\operatorname{dist}(\boldsymbol{A}(\xi)\boldsymbol{x}_k,\boldsymbol{b}(\xi))^2\right]} \in \mathcal{O}\left(k^{-1/6}\right).$$

Consequently, the oracle complexity is #(SFO) $\in \mathcal{O}\left(\epsilon^{-6}\right)$ *and #(LMO)* $\in \mathcal{O}\left(\epsilon^{-6}\right)$.

Proof. The stated result comes from applying Lemma A.1 in conjunction with the convergence smoothed-gap rate obtained in Theorem 2.1. Considering that, at every iteration, we take one stochastic sample and compute one LMO, along with the $\mathcal{O}\left(k^{-1/6}\right)$ convergence rate, we obtain the stated oracle complexities. □

A.2 Analysis of H-SPIDER-FW

This section provides the omitted proofs of Section 2.4.4 in the main text. We start with a supporting lemma, needed for the proof of Lemma 2.2 and Lemma 2.3.

Lemma A.4. *Let* $\boldsymbol{v}_{t,k} = \boldsymbol{v}_{t,k-1} - \widetilde{\nabla} F_{\beta_{t,k-1}}(\boldsymbol{x}_{t,k-1}, \xi_{\mathcal{S}_{t,k}}) + \widetilde{\nabla} F_{\beta_{t,k}}(\boldsymbol{x}_{t,k}, \xi_{\mathcal{S}_{t,k}})$*, with* $|\mathcal{S}_{t,k}| = K_t = 2^{t-1}$ *and* $\boldsymbol{v}_{t,1} = \widetilde{\nabla} F_{\beta_{t,1}}(\boldsymbol{x}_{t,1}, \xi_{\mathcal{Q}_t})$*. Also, let* $\gamma_{t,k} = \frac{2}{K_t+k}$*,* $\beta_{t,k} = \frac{\beta_0}{\sqrt{K_t+k}}$*. Then, for a fixed* t *and for all* $k \le K_t$*,*

$$\mathbb{E}_{t,1}\left[\left\| \nabla F_{\beta_{t,k}}(\boldsymbol{x}_{t,k}) - \boldsymbol{v}_{t,k} \right\|^2\right] \le \frac{2\mathcal{D}_{\mathcal{X}}^2}{K_t+k}\left(8L_f^2 + \frac{98L_{\boldsymbol{A}}^2}{\beta_0^2}\right) + \mathbb{E}_{t,1}\left[\left\| \nabla F_{\beta_1}(\boldsymbol{x}_1) - \boldsymbol{v}_1 \right\|^2\right] \tag{A.25}$$

Proof.

$$\begin{aligned}
&\left\| \nabla F_{\beta_{t,k}}(\boldsymbol{x}_{t,k}) - \boldsymbol{v}_{t,k} \right\|^2 \\
&\quad= \left\| \nabla F_{\beta_{t,k}}(\boldsymbol{x}_{t,k}) - \boldsymbol{v}_{t,k-1} - \widetilde{\nabla} F_{\beta_{t,k}}(\boldsymbol{x}_{t,k}, \xi_{\mathcal{S}_{t,k}}) + \widetilde{\nabla} F_{\beta_{t,k-1}}(\boldsymbol{x}_{t,k-1}, \xi_{\mathcal{S}_{t,k}}) \right\|^2 \\
&\quad= \Big\| \nabla F_{\beta_{t,k}}(\boldsymbol{x}_{t,k}) - \nabla F_{\beta_{t,k-1}}(\boldsymbol{x}_{t,k-1}) + \nabla F_{\beta_{t,k-1}}(\boldsymbol{x}_{t,k-1}) - \boldsymbol{v}_{t,k-1} \\
&\qquad\qquad - \widetilde{\nabla} F_{\beta_{t,k}}(\boldsymbol{x}_{t,k}, \xi_{\mathcal{S}_{t,k}}) + \widetilde{\nabla} F_{\beta_{t,k-1}}(\boldsymbol{x}_{t,k-1}, \xi_{\mathcal{S}_{t,k}}) \Big\|^2 \\
&\quad= \left\| \nabla F_{\beta_{t,k-1}}(\boldsymbol{x}_{t,k-1}) - \boldsymbol{v}_{t,k-1} \right\|^2 \\
&\qquad + \left\| \nabla F_{\beta_{t,k}}(\boldsymbol{x}_{t,k}) - \nabla F_{\beta_{t,k-1}}(\boldsymbol{x}_{t,k-1}) - \widetilde{\nabla} F_{\beta_{t,k}}(\boldsymbol{x}_{t,k}, \xi_{\mathcal{S}_{t,k}}) + \widetilde{\nabla} F_{\beta_{t,k-1}}(\boldsymbol{x}_{t,k-1}, \xi_{\mathcal{S}_{t,k}}) \right\|^2 \\
&\qquad + 2\langle \nabla F_{\beta_{t,k-1}}(\boldsymbol{x}_{t,k-1}) - \boldsymbol{v}_{t,k-1},\ \nabla F_{\beta_{t,k}}(\boldsymbol{x}_{t,k}) - \nabla F_{\beta k-1}(\boldsymbol{x}_{t,k-1}) \\
&\qquad\qquad - \widetilde{\nabla} F_{\beta_{t,k}}(\boldsymbol{x}_{t,k}, \xi_{\mathcal{S}_{t,k}}) + \widetilde{\nabla} F_{\beta_{t,k-1}}(\boldsymbol{x}_{t,k-1}, \xi_{\mathcal{S}_{t,k}}) \rangle
\end{aligned}$$

We now take the expectation on both sides conditioned on all randomness up to step (t,k) (i.e. the expectations are taken solely with regards to $\xi_{\mathcal{S}_{t,k}}$, and we denote $\mathbb{E}_{t,k}[\cdot] \coloneqq \mathbb{E}\left[\cdot \,|\, \mathcal{F}_{t,k}\right]$).

$$\begin{aligned}
&\mathbb{E}_{t,k}\left[\left\| \nabla F_{\beta_{t,k}}(\boldsymbol{x}_{t,k}) - \boldsymbol{v}_{t,k} \right\|^2\right] \\
&\quad= \left\| \nabla F_{\beta_{t,k-1}}(\boldsymbol{x}_{t,k-1}) - \boldsymbol{v}_{t,k-1} \right\|^2 \\
&\qquad + \mathbb{E}_{t,k}\left[\left\| \nabla F_{\beta_{t,k}}(\boldsymbol{x}_{t,k}) - \nabla F_{\beta_{t,k-1}}(\boldsymbol{x}_{t,k-1}) - \widetilde{\nabla} F_{\beta_{t,k}}(\boldsymbol{x}_{t,k}, \xi_{\mathcal{S}_{t,k}}) + \widetilde{\nabla} F_{\beta_{t,k-1}}(\boldsymbol{x}_{t,k-1}, \xi_{\mathcal{S}_{t,k}}) \right\|^2\right] \\
&\qquad + 2\langle \nabla F_{\beta_{t,k-1}}(\boldsymbol{x}_{t,k-1}) - \boldsymbol{v}_{t,k-1}, \\
&\qquad\qquad \underbrace{\mathbb{E}_{t,k}\left[\nabla F_{\beta_{t,k}}(\boldsymbol{x}_{t,k}) - \nabla F_{\beta_{t,k-1}}(\boldsymbol{x}_{t,k-1}) - \widetilde{\nabla} F_{\beta_{t,k}}(\boldsymbol{x}_{t,k}, \xi_{\mathcal{S}_{t,k}}) + \widetilde{\nabla} F_{\beta_{t,k-1}}(\boldsymbol{x}_{t,k-1}, \xi_{\mathcal{S}_{t,k}})\right]}_{=\mathbf{0}_d,\ \text{since } \nabla F_\beta(x) = \mathbb{E}\left[\widetilde{\nabla} F(\boldsymbol{x}, \xi_{\mathcal{S}_{t,k}})\right]} \rangle \\
&\quad= \left\| \nabla F_{\beta_{t,k-1}}(\boldsymbol{x}_{t,k-1}) - \boldsymbol{v}_{t,k-1} \right\|^2
\end{aligned}$$

$$+\underbrace{\mathbb{E}_{t,k}\left[\left\|\nabla F_{\beta_{t,k}}(\boldsymbol{x}_{t,k})-\nabla F_{\beta_{t,k-1}}(\boldsymbol{x}_{t,k-1})-\widetilde{\nabla} F_{\beta_{t,k}}(\boldsymbol{x}_{t,k},\xi_{\mathcal{S}_{t,k}})+\widetilde{\nabla} F_{\beta_{t,k-1}}(\boldsymbol{x}_{t,k-1},\xi_{\mathcal{S}_{t,k}})\right\|^2\right]}_{=T} \tag{A.26}$$

We continue by bounding T and get

$$\begin{aligned}
T &= \mathbb{E}_{t,k}\left[\left\|\nabla F_{\beta_{t,k}}(\boldsymbol{x}_{t,k})-\nabla F_{\beta_{t,k-1}}(\boldsymbol{x}_{t,k-1})-\widetilde{\nabla} F_{\beta_{t,k}}(\boldsymbol{x}_{t,k},\xi_{\mathcal{S}_{t,k}})+\widetilde{\nabla} F_{\beta_{t,k-1}}(\boldsymbol{x}_{t,k-1},\xi_{\mathcal{S}_{t,k}})\right\|^2\right] \\
&= \mathbb{E}_{t,k}\left[\left\|\frac{1}{K_t}\sum_{i=1}^{K_t}\nabla F_{\beta_{t,k}}(\boldsymbol{x}_{t,k})-\nabla F_{\beta_{t,k-1}}(\boldsymbol{x}_{t,k-1})-\nabla F_{\beta_{t,k}}(\boldsymbol{x}_{t,k},\xi_i)+\nabla F_{\beta_{t,k-1}}(\boldsymbol{x}_{t,k-1},\xi_i)\right\|^2\right] \qquad \text{(A.27)} \\
&= \frac{1}{K_t^2}\mathbb{E}_{t,k}\left[\sum_{i=1}^{K_t}\left\|\nabla F_{\beta_{t,k}}(\boldsymbol{x}_{t,k})-\nabla F_{\beta_{t,k-1}}(\boldsymbol{x}_{t,k-1})-\nabla F_{\beta_{t,k}}(\boldsymbol{x}_{t,k},\xi_i)+\nabla F_{\beta_{t,k-1}}(\boldsymbol{x}_{t,k-1},\xi_i)\right\|^2\right] \\
&\quad + \frac{2}{K_t^2}\mathbb{E}_{t,k}\Big[\sum_{\substack{i,j<K_t\\ i<j}}\langle\nabla F_{\beta_{t,k}}(\boldsymbol{x}_{t,k})-\nabla F_{\beta_{t,k-1}}(\boldsymbol{x}_{t,k-1})-\nabla F_{\beta_{t,k}}(\boldsymbol{x}_{t,k},\xi_i)+\nabla F_{\beta_{t,k-1}}(\boldsymbol{x}_{t,k-1},\xi_i), \\
&\qquad\qquad \nabla F_{\beta_{t,k}}(\boldsymbol{x}_{t,k})-\nabla F_{\beta_{t,k-1}}(\boldsymbol{x}_{t,k-1})-\nabla F_{\beta_{t,k}}(\boldsymbol{x}_{t,k},\xi_j)+\nabla F_{\beta_{t,k-1}}(\boldsymbol{x}_{t,k-1},\xi_j)\rangle\Big] \qquad \text{(A.28)} \\
&= \frac{1}{K_t^2}\sum_{i=1}^{K_t}\mathbb{E}_{t,k}\left[\left\|\nabla F_{\beta_{t,k}}(\boldsymbol{x}_{t,k})-\nabla F_{\beta_{t,k-1}}(\boldsymbol{x}_{t,k-1})-\nabla F_{\beta_{t,k}}(\boldsymbol{x}_{t,k},\xi_i)+\nabla F_{\beta_{t,k-1}}(\boldsymbol{x}_{t,k-1},\xi_i)\right\|^2\right] \qquad \text{(A.29)} \\
&= \frac{K_t}{K_t^2}\mathbb{E}_{t,k}\left[\left\|\nabla F_{\beta_{t,k}}(\boldsymbol{x}_{t,k})-\nabla F_{\beta_{t,k-1}}(\boldsymbol{x}_{t,k-1})-\nabla F_{\beta_{t,k}}(\boldsymbol{x}_{t,k},\xi)+\nabla F_{\beta_{t,k-1}}(\boldsymbol{x}_{t,k-1},\xi)\right\|^2\right] \\
&= \frac{1}{K_t}\mathbb{E}_{t,k}\|\nabla f(\boldsymbol{x}_{t,k})-\nabla f(\boldsymbol{x}_{t,k-1})-\nabla f(\boldsymbol{x}_{t,k},\xi)+\nabla f(\boldsymbol{x}_{t,k-1},\xi) \\
&\qquad +\nabla G_{\beta_{t,k}}(\boldsymbol{A}\boldsymbol{x}_{t,k})-\nabla G_{\beta_{t,k-1}}(\boldsymbol{A}\boldsymbol{x}_{t,k-1})-\nabla g_{\beta_{t,k}}(\boldsymbol{A}(\xi)\boldsymbol{x}_{t,k})+\nabla g_{\beta_{t,k-1}}(\boldsymbol{A}(\xi)\boldsymbol{x}_{t,k-1})\|^2 \\
&\le \underbrace{\frac{2}{K_t}\mathbb{E}_{t,k}\|\nabla f(\boldsymbol{x}_{t,k})-\nabla f(\boldsymbol{x}_{t,k-1})-\nabla f(\boldsymbol{x}_{t,k},\xi)+\nabla f(\boldsymbol{x}_{t,k-1},\xi)\|^2}_{=T_1} \\
&\quad + \underbrace{\frac{2}{K_t}\mathbb{E}_{t,k}\|\nabla G_{\beta_{t,k}}(\boldsymbol{A}\boldsymbol{x}_{t,k})-\nabla G_{\beta_{t,k-1}}(\boldsymbol{A}\boldsymbol{x}_{t,k-1})-\nabla g_{\beta_{t,k}}(\boldsymbol{A}(\xi)\boldsymbol{x}_{t,k})+\nabla g_{\beta_{t,k-1}}(\boldsymbol{A}(\xi)\boldsymbol{x}_{t,k-1})\|^2}_{=T_2}.
\end{aligned}$$

Line (A.27) comes from the use of an averaged gradient with batch size K_t. Line (A.28) comes from applying the square norm to the inner sum and linearity of expectation. Line (A.29) comes from passing the expectation inside the inner product as allowed by the independence of the samples $\boldsymbol{A}(\xi_i)$ and $\boldsymbol{A}(\xi_j)$ (if $X \perp Y$, then $\mathbb{E}[XY] = \mathbb{E}[X]\mathbb{E}[Y]$). This results in each term being

zero owing to stochastic gradient unbiasedness.

We evaluate the terms T_1 and T_2 separately:

$$
\begin{aligned}
T_2 &= \frac{2}{K_t}\mathbb{E}_{t,k}\|\nabla G_{\beta_{t,k}}(\boldsymbol{A}\boldsymbol{x}_{t,k}) - \nabla G_{\beta_{t,k-1}}(\boldsymbol{A}\boldsymbol{x}_{t,k-1}) - \nabla g_{\beta_{t,k}}(\boldsymbol{A}(\xi)\boldsymbol{x}_{t,k}) + \nabla g_{\beta_{t,k-1}}(\boldsymbol{A}(\xi)\boldsymbol{x}_{t,k-1})\|^2 \\
&= \frac{2}{K_t}\mathbb{E}_{t,k}\big[\, \big\| \nabla G_{\beta_{t,k}}(\boldsymbol{A}\boldsymbol{x}_{t,k}) - \nabla G_{\beta_{t,k-1}}(\boldsymbol{A}\boldsymbol{x}_{t,k-1}) \big\|^2 \\
&\qquad - 2\langle \nabla G_{\beta_{t,k}}(\boldsymbol{A}\boldsymbol{x}_{t,k}) - \nabla G_{\beta_{t,k-1}}(\boldsymbol{A}\boldsymbol{x}_{t,k-1}),\, \nabla g_{\beta_{t,k}}(\boldsymbol{A}(\xi)\boldsymbol{x}_{t,k}) - \nabla g_{\beta_{t,k-1}}(\boldsymbol{A}(\xi)\boldsymbol{x}_{t,k-1})\rangle \\
&\qquad + \big\| \nabla g_{\beta_{t,k}}(\boldsymbol{A}(\xi)\boldsymbol{x}_{t,k}) - \nabla g_{\beta_{t,k-1}}(\boldsymbol{A}(\xi)\boldsymbol{x}_{t,k-1}) \big\|^2 \big] \\
&= \frac{2}{K_t}\big(\big\| \nabla G_{\beta_{t,k}}(\boldsymbol{A}\boldsymbol{x}_{t,k}) - \nabla G_{\beta_{t,k-1}}(\boldsymbol{A}\boldsymbol{x}_{t,k-1}) \big\|^2 \\
&\qquad - 2\langle \nabla G_{\beta_{t,k}}(\boldsymbol{A}\boldsymbol{x}_{t,k}) - \nabla G_{\beta_{t,k-1}}(\boldsymbol{A}\boldsymbol{x}_{t,k-1}),\, \mathbb{E}_{t,k}\left[\nabla g_{\beta_{t,k}}(\boldsymbol{A}(\xi)\boldsymbol{x}_{t,k}) - \nabla g_{\beta_{t,k-1}}(\boldsymbol{A}(\xi)\boldsymbol{x}_{t,k-1})\right]\rangle \\
&\qquad + \mathbb{E}_{t,k}\left[\big\| \nabla g_{\beta_{t,k}}(\boldsymbol{A}(\xi)\boldsymbol{x}_{t,k}) - \nabla g_{\beta_{t,k-1}}(\boldsymbol{A}(\xi)\boldsymbol{x}_{t,k-1}) \big\|^2 \right] \big) \\
&= \frac{2}{K_t}\big(\mathbb{E}_{t,k}\left[\big\| \nabla g_{\beta_{t,k}}(\boldsymbol{A}(\xi)\boldsymbol{x}_{t,k}) - \nabla g_{\beta_{t,k-1}}(\boldsymbol{A}(\xi)\boldsymbol{x}_{t,k-1}) \big\|^2 \right] - \big\| \nabla G_{\beta_{t,k}}(\boldsymbol{A}\boldsymbol{x}_{t,k}) - \nabla G_{\beta_{t,k-1}}(\boldsymbol{A}\boldsymbol{x}_{t,k-1}) \big\|^2 \big) \\
&\le \frac{2}{K_t}\mathbb{E}_{t,k}\left[\big\| \nabla g_{\beta_{t,k}}(\boldsymbol{A}(\xi)\boldsymbol{x}_{t,k}) - \nabla g_{\beta_{t,k}}(\boldsymbol{A}(\xi)\boldsymbol{x}_{t,k-1}) + \nabla g_{\beta_{t,k}}(\boldsymbol{A}(\xi)\boldsymbol{x}_{t,k-1}) - \nabla g_{\beta_{t,k-1}}(\boldsymbol{A}(\xi)\boldsymbol{x}_{t,k-1}) \big\|^2 \right] \\
&= \frac{2}{K_t}\mathbb{E}_{t,k}\Big[\| \frac{1}{\beta_{t,k}}\boldsymbol{A}^\top(\xi)\boldsymbol{A}(\xi)\left(\boldsymbol{x}_{t,k} - \boldsymbol{x}_{t,k-1}\right) + \frac{1}{\beta_{t,k}}\boldsymbol{A}^\top(\xi)\left[\mathrm{proj}_{\boldsymbol{b}(\xi)}\left(\boldsymbol{A}(\xi)\boldsymbol{x}_{t,k-1}\right) - \mathrm{proj}_{\boldsymbol{b}(\xi)}\left(\boldsymbol{A}(\xi)\boldsymbol{x}_{t,k}\right)\right] \\
&\qquad + \left(\frac{1}{\beta_{t,k}} - \frac{1}{\beta_{t,k-1}}\right)\boldsymbol{A}^\top(\xi)\left[\boldsymbol{A}(\xi)\boldsymbol{x}_{t,k-1} - \mathrm{proj}_{\boldsymbol{b}(\xi)}\left(\boldsymbol{A}(\xi)\boldsymbol{x}_{t,k-1}\right)\right] \|^2 \Big] \\
&\le \frac{2}{K_t}\mathbb{E}_{t,k}\Big[\frac{3L_{\boldsymbol{A}}^2}{\beta_{t,k}^2}\|\boldsymbol{x}_{t,k} - \boldsymbol{x}_{t,k-1}\|^2 + \frac{3L_{\boldsymbol{A}}}{\beta_{t,k}^2}\| \mathrm{proj}_{\boldsymbol{b}(\xi)}\left(\boldsymbol{A}(\xi)\boldsymbol{x}_{t,k-1}\right) - \mathrm{proj}_{\boldsymbol{b}(\xi)}\left(\boldsymbol{A}(\xi)\boldsymbol{x}_{t,k}\right)\|^2 \\
&\qquad + 3L_{\boldsymbol{A}}\left(\frac{1}{\beta_{t,k}} - \frac{1}{\beta_{t,k-1}}\right)^2 \| \boldsymbol{A}(\xi)\boldsymbol{x}_{t,k-1} - \mathrm{proj}_{\boldsymbol{b}(\xi)}\left(\boldsymbol{A}(\xi)\boldsymbol{x}_{t,k-1}\right)\|^2 \Big] \\
&\le \frac{2}{K_t}\mathbb{E}_{t,k}\left[\frac{3L_{\boldsymbol{A}}^2}{\beta_{t,k}^2}\|\boldsymbol{x}_{t,k} - \boldsymbol{x}_{t,k-1}\|^2 + \frac{3L_{\boldsymbol{A}}^2}{\beta_{t,k}^2}\|\boldsymbol{x}_{t,k-1} - \boldsymbol{x}_{t,k}\|^2 \right. \\
&\qquad \left. + 3L_{\boldsymbol{A}}\left(\frac{1}{\beta_{t,k}} - \frac{1}{\beta_{t,k-1}}\right)^2 \| \boldsymbol{A}(\xi)\boldsymbol{x}_{t,k-1} - \boldsymbol{A}(\xi)\boldsymbol{x}^\star\|^2 \right] \qquad \text{(A.30)} \\
&\le \frac{2}{K_t}\Big[\frac{6L_A^2\gamma_{t,k-1}^2\mathcal{D}_{\mathcal{X}}^2}{\beta_{t,k}^2} + 3L_{\boldsymbol{A}}^2\mathcal{D}_{\mathcal{X}}^2\left(\frac{1}{\beta_{t,k}} - \frac{1}{\beta_{t,k-1}}\right)^2 \Big] \qquad \text{(A.31)} \\
&\le \frac{2L_A^2\mathcal{D}_{\mathcal{X}}^2}{\beta_0^2 K_t(K_t+k-1)}\Big[\frac{24(K_t+k)}{(K_t+k-1)} + \frac{3}{4} \Big] \qquad \text{(A.32)}
\end{aligned}
$$

$$\leq \frac{98L_{\boldsymbol{A}}^2\mathcal{D}_{\mathcal{X}}^2}{\beta_0^2 K_t(K_t+k-1)}, \tag{A.33}$$

where line (A.30) comes from the nonexpansiveness of projections and $\|\boldsymbol{A}(\xi)\boldsymbol{x}_{t,k-1} - \text{proj}_{\boldsymbol{b}(\xi)}\left(\boldsymbol{A}(\xi)\boldsymbol{x}_{t,k-1}\right)\| \leq \|\boldsymbol{A}(\xi)\boldsymbol{x}_{t,k-1} - \boldsymbol{y}\|, \forall \boldsymbol{y} \in \boldsymbol{b}(\xi)$, and line (A.31) comes from the iterate update rule and the definition of $\mathcal{D}_{\mathcal{X}}$. Line (A.32) comes from replacing the parameter rates and

$$\begin{aligned} 0 \leq \frac{1}{\beta_{t,k}} - \frac{1}{\beta_{t,k-1}} &= \frac{1}{\beta_0}\left(\sqrt{K_t+k} - \sqrt{K_t+k-1}\right) \\ &= \frac{1}{\beta_0}\left(\frac{1}{\sqrt{K_t+k} + \sqrt{K_t+k-1}}\right) \\ &\leq \frac{1}{2\beta_0\sqrt{K_t+k-1}}. \end{aligned}$$

Now we evaluate T_1 and use the fact that $\nabla f(\boldsymbol{x},\xi)$ are L_f-Lipschitz:

$$\begin{aligned} T_1 &= \frac{2}{K_t}\mathbb{E}_{t,k}\left[\|\nabla f(\boldsymbol{x}_{t,k}) - \nabla f(\boldsymbol{x}_{t,k-1}) - \nabla f(\boldsymbol{x}_{t,k},\xi) + \nabla f(\boldsymbol{x}_{t,k-1},\xi)\|^2\right] \\ &= \frac{2}{K_t}\Bigg(\|\nabla f(\boldsymbol{x}_{t,k}) - \nabla f(\boldsymbol{x}_{t,k-1})\|^2 + \mathbb{E}_{t,k}\left[\|\nabla f(\boldsymbol{x}_{t,k},\xi) - \nabla f(\boldsymbol{x}_{t,k-1},\xi)\|^2\right] \\ &\qquad - 2\langle \nabla f(\boldsymbol{x}_{t,k}) - \nabla f(\boldsymbol{x}_{t,k-1}), \mathbb{E}_{t,k}\left[\nabla f(\boldsymbol{x}_{t,k},\xi) - \nabla f(\boldsymbol{x}_{t,k-1},\xi)\right]\rangle\Bigg) \\ &\leq \frac{2}{K_t}\mathbb{E}_{t,k}\left[\|\nabla f(\boldsymbol{x}_{t,k},\xi) - \nabla f(\boldsymbol{x}_{t,k-1},\xi)\|^2\right] \\ &\leq \frac{2L_f^2}{K_t}\|\boldsymbol{x}_{t,k} - \boldsymbol{x}_{t,k-1}\|^2 \\ &\leq \frac{2L_f^2\gamma_{t,k-1}^2\mathcal{D}_{\mathcal{X}}^2}{K_t} \\ &= \frac{8L_f^2\mathcal{D}_{\mathcal{X}}^2}{K_t(K_t+k-1)^2}. \end{aligned} \tag{A.34}$$

Plugging in (A.34) and (A.33) into the expression of T, we get

$$\begin{aligned} T &\leq \frac{8L_f^2\mathcal{D}_{\mathcal{X}}^2}{K_t(K_t+k-1)^2} + \frac{98L_{\boldsymbol{A}}^2\mathcal{D}_{\mathcal{X}}^2}{\beta_0^2 K_t(K_t+k-1)} \\ &\leq \frac{\mathcal{D}_{\mathcal{X}}^2\left(8L_f^2 + \frac{98L_{\boldsymbol{A}}^2}{\beta_0^2}\right)}{K_t(K_t+k-1)}. \end{aligned} \tag{A.35}$$

Telescoping the sum in Equation (A.26) we get

$$\begin{aligned}
&\mathbb{E}_{t,1}\left[\left\|\nabla F_{\beta_{t,k}}(\boldsymbol{x}_{t,k})-\boldsymbol{v}_{t,k}\right\|^2\right] \\
&\qquad=\mathbb{E}_{t,1}\left[\mathbb{E}_{t,2}\left[\ldots\mathbb{E}_{t,k}\left[\left\|\nabla F_{\beta_{t,k}}(\boldsymbol{x}_{t,k})-\boldsymbol{v}_{t,k}\right\|^2\right]\right]\right] \\
&\qquad\le\frac{\mathcal{D}_{\mathcal{X}}^2}{K_t}\left(8L_f^2+\frac{98L_{\boldsymbol{A}}^2}{\beta_0^2}\right)\sum_{i=2}^{k}\frac{1}{K_t+i-1}+\mathbb{E}_{t,1}\left[\left\|\nabla F_{\beta_{t,1}}(\boldsymbol{x}_{t,1})-\boldsymbol{v}_{t,1}\right\|^2\right] \\
&\qquad\le\frac{\mathcal{D}_{\mathcal{X}}^2}{K_t}\left(8L_f^2+\frac{98L_{\boldsymbol{A}}^2}{\beta_0^2}\right)\sum_{i=2}^{k}\frac{1}{\frac{K_t+k}{2}}+\mathbb{E}_{t,1}\left[\left\|\nabla F_{\beta_{t,1}}(\boldsymbol{x}_{t,1})-\boldsymbol{v}_{t,1}\right\|^2\right] \qquad \text{(A.36)} \\
&\qquad=\frac{2\mathcal{D}_{\mathcal{X}}^2}{K_t}\left(8L_f^2+\frac{98L_{\boldsymbol{A}}^2}{\beta_0^2}\right)\frac{k-1}{K_t+k}+\mathbb{E}_{t,1}\left[\left\|\nabla F_{\beta_{t,1}}(\boldsymbol{x}_{t,1})-\boldsymbol{v}_{t,1}\right\|^2\right] \\
&\qquad\le\frac{2\mathcal{D}_{\mathcal{X}}^2}{K_t+k}\left(8L_f^2+\frac{98L_{\boldsymbol{A}}^2}{\beta_0^2}\right)+\mathbb{E}_{t,1}\left[\left\|\nabla F_{\beta_{t,1}}(\boldsymbol{x}_{t,1})-\boldsymbol{v}_{t,1}\right\|^2\right] \qquad \text{(A.37)}
\end{aligned}$$

where line (A.36) uses

$$\begin{aligned}
2\le k\le 2^{t-1}=K_t &\implies 2^{t-2}+1\le\frac{K_t+k}{2}\le 2^{t-1} \textbf{ and} \\
2\le i\le k\le 2^{t-1} &\implies 2^{t-1}+1\le K_t+i-1\le 2^t-1
\end{aligned}$$

and line (A.37) comes from $k-1\le K_t$. □

A.2.1 Proof of Lemma 2.2

Lemma 2.2 (Estimator variance for finite-sum problems)**.** *Consider Algorithm 2.2, and let ξ be finitely sampled from set $[n]$, $\xi_{\mathcal{Q}_t} = [n]$ and $\xi_{\mathcal{S}_{t,k}}$, such that $|\mathcal{S}_{t,k}| = K_t = 2^{t-1}$. Also, let $\gamma_{t,k} = \frac{2}{K_t+k}$, $\beta_{t,k} = \frac{\beta_0}{\sqrt{K_t+k}}$, $\beta_0 > 0$. Then, for a fixed t and for all $k \leq K_t$,*

$$\mathbb{E}\left[\left\| \nabla F_{\beta_{t,k}}(\boldsymbol{x}_{t,k}) - \boldsymbol{v}_{t,k} \right\|^2\right] \leq \frac{C_1}{K_t + k},$$

where $C_1 = 2\mathcal{D}_{\mathcal{X}}^2 \left(8L_f^2 + \frac{98L_A^2}{\beta_0^2}\right)$.

Proof. The result directly follows from the fact that we take a full gradient in the outer loop ($\xi_{\mathcal{Q}_t} = [n]$), thus zeroing out the term $\mathbb{E}_{t,1}\left[\left\| \nabla F_{\beta_{t,1}}(\boldsymbol{x}_{t,1}) - v_{t,1} \right\|^2\right]$ of Lemma A.4. Taking the full expectation on both sides gives us the stated result. □

A.2.2 Proof of Lemma 2.3

Lemma 2.3 (Estimator variance for general expectation problems)**.** *Consider Algorithm 2.2 and let* $\xi \sim P(\xi)$ *and* $\xi_{\mathcal{Q}_t}$ *such that* $|\mathcal{Q}_t| = \left\lceil \frac{2K_t}{\beta_{t,1}^2} \right\rceil$. *Also, let* $\xi_{\mathcal{S}_{t,k}}$, *such that* $|\mathcal{S}_{t,k}| = K_t = 2^{t-1}$, $\gamma_{t,k} = \frac{2}{K_t+k}$, $\beta_{t,k} = \frac{\beta_0}{\sqrt{K_t+k}}$, $\beta_0 > 0$. *Then, for a fixed* t *and for all* $k \leq K_t$,

$$\mathbb{E}\left[\left\| \nabla F_{\beta_{t,k}}(\boldsymbol{x}_{t,k}) - \boldsymbol{v}_{t,k} \right\|^2 \right] \leq \frac{C_2}{K_t + k},$$

where $C_2 = 16L_f^2\mathcal{D}_{\mathcal{X}}^2 + 2L_{\boldsymbol{A}}^2\mathcal{D}_{\mathcal{X}}^2\left(\frac{98}{\beta_0^2} + 1\right) + 2\beta_0^2\sigma_f^2$.

Proof. From the use of averaged gradient and the variance bound (A.10), we have

$$\begin{aligned}
&\mathbb{E}_{t,1}\left[\left\| \nabla F_{\beta_{t,1}}(\boldsymbol{x}_{t,1}) - \boldsymbol{v}_{t,1} \right\|^2 \right] \\
&\quad\leq \frac{1}{|\mathcal{Q}_t|}\mathbb{E}_{t,1}\left[\left\| \nabla f(\boldsymbol{x}_{t,1}) - \nabla f(\boldsymbol{x}_{t,1},\xi) + \nabla G_{\beta_{t,1}}(\boldsymbol{A}\boldsymbol{x}_{t,1}) - \nabla g_{\beta_{t,1}}(\boldsymbol{A}(\xi)\boldsymbol{x}_{t,1}) \right\|^2 \right] \\
&\quad\leq \frac{1}{|\mathcal{Q}_t|}\left(2\mathbb{E}_{t,1}\left[\left\| \nabla f(\boldsymbol{x}_{t,1}) - \nabla f(\boldsymbol{x}_{t,1},\xi) \right\|^2 \right] + 2\mathbb{E}_{t,1}\left[\left\| \nabla G_{\beta_{t,1}}(\boldsymbol{A}\boldsymbol{x}_{t,1}) - \nabla g_{\beta_{t,1}}(\boldsymbol{A}(\xi)\boldsymbol{x}_{t,1}) \right\|^2 \right) \right] \\
&\quad\leq \frac{\beta_{t,1}^2}{2K_t}\left(2\sigma_f^2 + \frac{2L_{\boldsymbol{A}}^2\mathcal{D}_{\mathcal{X}}^2}{\beta_{t,1}^2} \right) \\
&\quad\leq \frac{\beta_0^2}{2K_t(K_t+1)}\left(2\sigma_f^2 + \frac{2L_{\boldsymbol{A}}^2\mathcal{D}_{\mathcal{X}}^2(K_t+1)}{\beta_0^2} \right) \\
&\quad\leq \frac{\beta_0^2\sigma_f^2}{K_t^2} + \frac{L_{\boldsymbol{A}}^2\mathcal{D}_{\mathcal{X}}^2}{K_t} \\
&\quad\leq \frac{1}{K_t+k}\left(2\beta_0^2\sigma_f^2 + 2L_{\boldsymbol{A}}^2\mathcal{D}_{\mathcal{X}}^2 \right),
\end{aligned}$$

where we used $2K_t \geq K_t + k$ and $K_t^2 \geq K_t = \frac{2K_t}{2} \geq \frac{K_t+k}{2}$, $\forall K_t \in \mathbb{N}, K_t \geq 1, \forall k \leq K_t$. Replacing in (A.37), we obtain the desired result.

□

A.2.3 Proof of Theorem 2.2

Theorem 2.2. *Consider Algorithm 2.2 with parameters* $\gamma_{t,k} = \frac{2}{K_t+k}$, $\beta_{t,k} = \frac{\beta 0}{\sqrt{K_t+k}}$, $\beta_0 > 0$, *and* $\xi_{\mathcal{S}_{t,k}}$, *such that* $|\mathcal{S}_{t,k}| = K_t = 2^{t-1}$. *Then,*

- *For* ξ *be finitely sampled from set* $[n]$, $\xi_{\mathcal{Q}_t} = [n]$ *and* $\forall\, t \in \mathbb{N}$, $1 \le k \le 2^{t-1}$,

$$\mathbb{E}\left[S_{\beta_{t,k}}(\boldsymbol{x}_{t,k+1})\right] \le \frac{C_3}{\sqrt{K_t+k+1}},$$

where $C_3 = \max\left\{S_{\beta_{1,0}}(\boldsymbol{x}_{1,1}), 2\mathcal{D}_{\mathcal{X}}^2 L_f + 2\mathcal{D}_{\mathcal{X}}^2\sqrt{16L_f^2 + \frac{196L_{\boldsymbol{A}}^2}{\beta_0^2}} + \frac{2\mathcal{D}_{\mathcal{X}}^2 L_{\boldsymbol{A}}}{\beta_0}\right\}$;

- *For* $\xi \sim P(\xi)$, $\xi_{\mathcal{Q}_t}$ *such that* $|\mathcal{Q}_t| = \left\lceil \frac{2K_t}{\beta_{t,1}^2} \right\rceil$ *and* $\forall\, t \in \mathbb{N}$, $1 \le k \le 2^{t-1}$,

$$\mathbb{E}\left[S_{\beta_{t,k}}(\boldsymbol{x}_{t,k+1})\right] \le \frac{C_4}{\sqrt{K_t+k+1}},$$

where $C_4 = \max\left\{S_{\beta_{1,0}}(\boldsymbol{x}_{1,1}),\ 2\mathcal{D}_{\mathcal{X}}^2 L_f + \frac{2\mathcal{D}_{\mathcal{X}}^2 L_{\boldsymbol{A}}}{\beta_0} + 2\mathcal{D}_{\mathcal{X}}\sqrt{16L_f^2\mathcal{D}_{\mathcal{X}}^2 + 2L_{\boldsymbol{A}}^2\mathcal{D}_{\mathcal{X}}^2\left(\frac{98}{\beta_0^2}+1\right) + 2\beta_0^2\sigma_f^2}\right\}$.

Proof. The proof has two steps, coming from the nested loop structure of Algorithm 2.2. We first determine the recursion for $S_{\beta_{t,k}}(\boldsymbol{x}_{t,k+1})$ for all the iterates of the inner loop (constant t) and then show that the recursion holds at the 'edges' i.e. when going from $t-1$ to t.

1. Convergence recursion

1.1 Recursion of $S_{\beta_{t,k}}$ for constant t (inner loop)

Using observation (A.11), the definition of $\mathcal{D}_{\mathcal{X}}$ and the optimality of $\boldsymbol{w}_{t,k}$:

$$\begin{aligned}
&F_{\beta_{t,k}}(\boldsymbol{x}_{k+1}) \\
&\quad = \mathbb{E}_{t,k+1}\left[F_{\beta_{t,k}}(\boldsymbol{x}_{t,k+1}, \xi)\right] \\
&\quad \le \mathbb{E}_{t,k+1}\left[F_{\beta_{t,k}}(\boldsymbol{x}_{t,k}, \xi) + \langle \nabla F_{\beta_{t,k}}(\boldsymbol{x}_{t,k}, \xi),\, \boldsymbol{x}_{t,k+1} - \boldsymbol{x}_{t,k}\rangle + \frac{L_f + \frac{L_{\boldsymbol{A}}}{\beta_{t,k}}}{2}\left\| \boldsymbol{x}_{t,k+1} - \boldsymbol{x}_{t,k}\right\|^2\right] \\
&\quad \le F_{\beta_{t,k}}(\boldsymbol{x}_{t,k}) + \gamma_{t,k}\langle \nabla F_{\beta_{t,k}}(\boldsymbol{x}_{t,k}),\, \boldsymbol{w}_{t,k} - \boldsymbol{x}_{t,k}\rangle + \frac{\gamma_{t,k}^2(L_f + \frac{L_{\boldsymbol{A}}}{\beta_{t,k}})}{2}\left\| \boldsymbol{w}_{t,k} - \boldsymbol{x}_{t,k}\right\|^2 \\
&\quad \le F_{\beta_{t,k}}(\boldsymbol{x}_{t,k}) + \gamma_{t,k}\langle \nabla F_{\beta_{t,k}}(\boldsymbol{x}_{t,k}),\, \boldsymbol{w}_{t,k} - \boldsymbol{x}_{t,k}\rangle + \frac{\mathcal{D}_{\mathcal{X}}^2\gamma_{t,k}^2}{2}(L_f + \frac{L_{\boldsymbol{A}}}{\beta_{t,k}})
\end{aligned}$$

$$\leq F_{\beta_{t,k}}(\boldsymbol{x}_{t,k}) + \gamma_{t,k}\left(\langle \nabla F_{\beta_{t,k}}(\boldsymbol{x}_{t,k}) - \boldsymbol{v}_{t,k},\, \boldsymbol{w}_{t,k} - \boldsymbol{x}_{t,k}\rangle + \langle \boldsymbol{v}_{t,k},\, \boldsymbol{x}^\star - \boldsymbol{x}_{t,k}\rangle\right) + \frac{\mathcal{D}_{\mathcal{X}}^2 \gamma_{t,k}^2}{2}\left(L_f + \frac{L_{\boldsymbol{A}}}{\beta_{t,k}}\right) \quad \text{(A.38)}$$

We process the second term in (A.38) separately, using the convexity of f, observation (A.4) and noting that $\boldsymbol{v}_{t,k-1} - \boldsymbol{v}_{t,k} = \widetilde{\nabla} F_{\beta_{t,k-1}}(\boldsymbol{x}_{t,k-1}, \xi_{\mathcal{S}_{t,k}}) - \widetilde{\nabla} F_{\beta_{t,k}}(\boldsymbol{x}_{t,k}, \xi_{\mathcal{S}_{t,k}})$:

$$\begin{aligned}
&\langle \nabla F_{\beta_{t,k}}(\boldsymbol{x}_{t,k}) - \boldsymbol{v}_{t,k},\, \boldsymbol{w}_{t,k} - \boldsymbol{x}_{t,k}\rangle + \langle \boldsymbol{v}_{t,k},\, \boldsymbol{x}^\star - \boldsymbol{x}_{t,k}\rangle \\
&= \langle \nabla F_{\beta_{t,k}}(\boldsymbol{x}_{t,k}) - \boldsymbol{v}_{t,k},\, \boldsymbol{w}_{t,k} - \boldsymbol{x}^\star\rangle + \langle \nabla F_{\beta_{t,k}}(\boldsymbol{x}_{t,k}) - \boldsymbol{v}_{t,k},\, \boldsymbol{x}^\star - \boldsymbol{x}_{t,k}\rangle \\
&\qquad + \langle \boldsymbol{v}_{t,k-1} - \widetilde{\nabla} F_{\beta_{t,k-1}}(\boldsymbol{x}_{t,k-1}, \xi_{\mathcal{S}_{t,k}}) + \widetilde{\nabla} F_{\beta_k}(\boldsymbol{x}_{t,k}, \xi_{\mathcal{S}_{t,k}}),\, \boldsymbol{x}^\star - \boldsymbol{x}_{t,k}\rangle \\
&= \langle \nabla F_{\beta_{t,k}}(\boldsymbol{x}_{t,k}) - \boldsymbol{v}_{t,k},\, \boldsymbol{w}_{t,k} - \boldsymbol{x}^\star\rangle \\
&\qquad + \langle \nabla F_{\beta_{t,k}}(\boldsymbol{x}_{t,k}) - \boldsymbol{v}_{t,k} + \boldsymbol{v}_{t,k-1} - \widetilde{\nabla} F_{\beta_{t,k-1}}(\boldsymbol{x}_{t,k-1}, \xi_{\mathcal{S}_{t,k}}),\, \boldsymbol{x}^\star - \boldsymbol{x}_{t,k}\rangle \\
&\qquad + \langle \widetilde{\nabla} f(\boldsymbol{x}_{t,k}, \xi_{\mathcal{S}_{t,k}}),\, \boldsymbol{x}^\star - \boldsymbol{x}_{t,k}\rangle + \langle \boldsymbol{A}^\top(\xi_{\mathcal{S}_{t,k}}) \widetilde{\nabla} g_{\beta_{t,k}}(\boldsymbol{A}(\xi_{\mathcal{S}_{t,k}})\boldsymbol{x}_{t,k}),\, \boldsymbol{x}^\star - \boldsymbol{x}_{t,k}\rangle \\
&\leq \langle \nabla F_{\beta_{t,k}}(\boldsymbol{x}_{t,k}) - \boldsymbol{v}_{t,k},\, \boldsymbol{w}_{t,k} - \boldsymbol{x}^\star\rangle \\
&\qquad + \langle \nabla F_{\beta_{t,k}}(\boldsymbol{x}_{t,k}) - \widetilde{\nabla} F_{\beta_{t,k}}(\boldsymbol{x}_{t,k}, \xi_{\mathcal{S}_{t,k}}),\, \boldsymbol{x}_{t,k} - \boldsymbol{x}^\star\rangle \\
&\qquad + \widetilde{f}(\boldsymbol{x}^\star, \xi_{\mathcal{S}_{t,k}}) - \widetilde{f}(\boldsymbol{x}_{t,k}, \xi_{\mathcal{S}_{t,k}}) + \langle \widetilde{\nabla} g_{\beta_{t,k}}(\boldsymbol{A}(\xi_{\mathcal{S}_{t,k}})\boldsymbol{x}_{t,k}),\, \boldsymbol{A}(\xi_{\mathcal{S}_{t,k}})\boldsymbol{x}^\star - \boldsymbol{A}^\top(\xi_{\mathcal{S}_{t,k}})\boldsymbol{x}_{t,k}\rangle \\
&\leq \langle \nabla F_{\beta_{t,k}}(\boldsymbol{x}_{t,k}) - \boldsymbol{v}_{t,k},\, \boldsymbol{w}_{t,k} - \boldsymbol{x}^\star\rangle + \langle \nabla F_{\beta_{t,k}}(\boldsymbol{x}_{t,k}) - \widetilde{\nabla} F_{\beta_{t,k}}(\boldsymbol{x}_{t,k}, \xi_{\mathcal{S}_{t,k}}),\, \boldsymbol{x}_{t,k} - \boldsymbol{x}^\star\rangle \\
&\qquad + \widetilde{f}(\boldsymbol{x}^\star, \xi_{\mathcal{S}_{t,k}}) + \underbrace{\widetilde{g}(\boldsymbol{A}(\xi_{\mathcal{S}_{t,k}})\boldsymbol{x}^\star)}_{=0 \text{ a.s.}} \underbrace{- \widetilde{f}(\boldsymbol{x}_{t,k}, \xi_{\mathcal{S}_{t,k}}) - \widetilde{g}_{\beta_{t,k}}(\boldsymbol{A}(\xi_{\mathcal{S}_{t,k}})\boldsymbol{x}_{t,k})}_{=-\widetilde{F}_{\beta_{t,k}}(\boldsymbol{x}_{t,k}, \xi_{\mathcal{S}_{t,k}})} \\
&\qquad - \frac{\beta_{t,k}}{2}\left\| \boldsymbol{\lambda}^\star_{\beta_{t,k}}(\widetilde{\boldsymbol{A}(\xi_{\mathcal{S}_{t,k}})\boldsymbol{x}_{t,k}}) \right\|^2 \qquad \text{(A.39)}
\end{aligned}$$

We can now resume Equation (A.38) by plugging in the inequality in (A.39), subtracting $f^\star$ from both sides, and taking the conditional expectation $\mathbb{E}_{t,k}[\cdot] = \mathbb{E}\left[\cdot \,|\, \mathcal{F}_{t,k}\right]$.

$$\begin{aligned}
&\mathbb{E}_{t,k}\left[F_{\beta_{t,k}}(\boldsymbol{x}_{k+1}) - f^\star\right] \\
&\leq \mathbb{E}_{t,k}\Bigg[F_{\beta_{t,k}}(\boldsymbol{x}_{t,k}) + \gamma_{t,k}\left(\langle \nabla F_{\beta_{t,k}}(\boldsymbol{x}_{t,k}) - \boldsymbol{v}_{t,k},\, \boldsymbol{w}_{t,k} - \boldsymbol{x}_{t,k}\rangle + \langle \boldsymbol{v}_{t,k},\, \boldsymbol{x}^\star - \boldsymbol{x}_{t,k}\rangle\right) \\
&\qquad + \frac{\mathcal{D}_{\mathcal{X}}^2 \gamma_{t,k}^2}{2}\left(L_f + \frac{L_{\boldsymbol{A}}}{\beta_{t,k}}\right)\Bigg] - f^\star
\end{aligned}$$

$$
\begin{aligned}
&\leq F_{\beta_{t,k}}(\boldsymbol{x}_{t,k}) + \gamma_{t,k}\Bigg(\langle \nabla F_{\beta_{t,k}}(\boldsymbol{x}_{t,k}) - \boldsymbol{v}_{t,k},\, \boldsymbol{w}_{t,k} - \boldsymbol{x}^\star\rangle \\
&\qquad + \underbrace{\mathbb{E}_{t,k}\left[\langle \nabla F_{\beta_{t,k}}(\boldsymbol{x}_{t,k}) - \widetilde{\nabla} F_{\beta_{t,k}}(\boldsymbol{x}_{t,k}, \xi_{\mathcal{S}_{t,k}}),\, \boldsymbol{x}_{t,k} - \boldsymbol{x}^\star\rangle\right]}_{=0,\ \text{unbiasedness}} \\
&\qquad + \mathbb{E}_{t,k}\left[\widetilde{f}(\boldsymbol{x}^\star, \xi_{\mathcal{S}_{t,k}}) - \widetilde{F}_{\beta_{t,k}}(\boldsymbol{x}_{t,k}, \xi_{\mathcal{S}_{t,k}}) - \frac{\beta_{t,k}}{2}\left\| \boldsymbol{\lambda}^\star_{\beta_{t,k}}(\widetilde{\boldsymbol{A}(\xi_{\mathcal{S}_{t,k}})}\boldsymbol{x}_{t,k})\right\|^2\right]\Bigg) \\
&\qquad + \frac{\mathcal{D}_{\mathcal{X}}^2\gamma_{t,k}^2}{2}\left(L_f + \frac{L_{\boldsymbol{A}}}{\beta_{t,k}}\right) - f^\star
\end{aligned}
$$

$$
\begin{aligned}
&\leq F_{\beta_{t,k}}(\boldsymbol{x}_{t,k}) + \gamma_{t,k}\Bigg(\langle \nabla F_{\beta_{t,k}}(\boldsymbol{x}_{t,k}) - \boldsymbol{v}_{t,k},\, \boldsymbol{w}_{t,k} - \boldsymbol{x}^\star\rangle + f^\star - F_{\beta_{t,k}}(\boldsymbol{x}_{t,k}) \\
&\qquad - \mathbb{E}_{t,k}\left[\frac{\beta_{t,k}}{2}\left\| \boldsymbol{\lambda}^\star_{\beta_{t,k}}(\widetilde{\boldsymbol{A}(\xi_{\mathcal{S}_{t,k}})}\boldsymbol{x}_{t,k})\right\|^2\right]\Bigg) + \frac{\mathcal{D}_{\mathcal{X}}^2\gamma_{t,k}^2}{2}\left(L_f + \frac{L_{\boldsymbol{A}}}{\beta_{t,k}}\right) - f^\star
\end{aligned}
$$

$$
\begin{aligned}
&= (1-\gamma_{t,k})(F_{\beta_{t,k}}(\boldsymbol{x}_{t,k}) - f^\star) + \gamma_{t,k}\langle \nabla F_{\beta_{t,k}}(\boldsymbol{x}_{t,k}) - \boldsymbol{v}_{t,k},\, \boldsymbol{w}_{t,k} - \boldsymbol{x}^\star\rangle \\
&\qquad - \frac{\gamma_{t,k}\beta_{t,k}}{2}\mathbb{E}_{t,k}\left[\left\| \boldsymbol{\lambda}^\star_{\beta_{t,k}}(\widetilde{\boldsymbol{A}(\xi_{\mathcal{S}_{t,k}})}\boldsymbol{x}_{t,k})\right\|^2\right] + \frac{\mathcal{D}_{\mathcal{X}}^2\gamma_{t,k}^2}{2}\left(L_f + \frac{L_{\boldsymbol{A}}}{\beta_{t,k}}\right)
\end{aligned}
$$

Using property (A.12) we note that

$$
\begin{aligned}
F_{\beta_{t,k}}(\boldsymbol{x}_{t,k}) &= \mathbb{E}_{t,k}\left[\widetilde{f}(\boldsymbol{x}_{t,k}, \xi_{\mathcal{S}_{t,k}}) + \widetilde{g}_{\beta_{t,k}}(\boldsymbol{A}(\xi_{\mathcal{S}_{t,k}})\boldsymbol{x}_{t,k})\right] \\
&\leq \mathbb{E}_{t,k}\left[\widetilde{f}(\boldsymbol{x}_{t,k}, \xi_{\mathcal{S}_{t,k}}) + \widetilde{g}_{\beta_{t,k-1}}(\boldsymbol{A}(\xi_{\mathcal{S}_{t,k}})\boldsymbol{x}_{t,k}) + \frac{\beta_{t,k-1} - \beta_{t,k}}{2}\left\| \boldsymbol{\lambda}^\star_{\beta_{t,k}}(\widetilde{\boldsymbol{A}(\xi_{\mathcal{S}_{t,k}})}\boldsymbol{x}_{t,k})\right\|^2\right] \\
&= F_{\beta_{t,k-1}}(\boldsymbol{x}_{t,k}) + \mathbb{E}_{t,k}\left[\frac{\beta_{t,k-1} - \beta_{t,k}}{2}\left\| \boldsymbol{\lambda}^\star_{\beta_{t,k}}(\widetilde{\boldsymbol{A}(\xi_{\mathcal{S}_{t,k}})}\boldsymbol{x}_{t,k})\right\|^2\right].
\end{aligned}
$$

Using the above and the definition of $\mathcal{D}_{\mathcal{X}}$, we continue the inequality as

$$
\begin{aligned}
&\mathbb{E}_{t,k}\left[F_{\beta_{t,k}}(\boldsymbol{x}_{k+1}) - f^\star\right] \\
&\leq (1-\gamma_{t,k})(F_{\beta_{t,k-1}}(\boldsymbol{x}_{t,k}) - f^\star) + \gamma_{t,k}\mathcal{D}_{\mathcal{X}}\left\| \nabla F_{\beta_{t,k}}(\boldsymbol{x}_{t,k}) - \boldsymbol{v}_{t,k}\right\| \\
&+ \frac{(1-\gamma_{t,k})(\beta_{t,k-1} - \beta_{t,k}) - \gamma_{t,k}\beta_{t,k}}{2}\mathbb{E}_{t,k}\left[\left\| \boldsymbol{\lambda}^\star_{\beta_{t,k}}(\widetilde{\boldsymbol{A}(\xi_{\mathcal{S}_{t,k}})}\boldsymbol{x}_{t,k})\right\|^2\right] + \frac{\mathcal{D}_{\mathcal{X}}^2\gamma_{t,k}^2}{2}\left(L_f + \frac{L_{\boldsymbol{A}}}{\beta_{t,k}}\right). \quad \text{(A.40)}
\end{aligned}
$$

Using the stated parameter rates, we notice that $(1-\gamma_{t,k})(\beta_{t,k-1}-\beta_{t,k})-\gamma_{t,k}\beta_{t,k}<0$, as follows:

$$\begin{aligned}
&\left(1-\frac{2}{K_t+k}\right)\left(\frac{\beta_0}{\sqrt{K_t+k-1}}-\frac{\beta_0}{\sqrt{K_t+k}}\right)-\frac{2\beta_0}{(K_t+k)\sqrt{K_t+k}} \\
&\qquad=\frac{\beta_0}{\sqrt{K_t+k-1}}-\frac{\beta_0}{\sqrt{K_t+k}}-\frac{2\beta_0}{(K_t+k)\sqrt{K_t+k-1}} \\
&\qquad=\beta_0\frac{K_t+k-\sqrt{K_t+k}\sqrt{K_t+k-1}-2}{(K_t+k)\sqrt{K_t+k-1}} \\
&\qquad=\beta_0\frac{(K_t+k-1)-2\sqrt{\frac{K_t+k}{4}}\sqrt{K_t+k-1}+\frac{K_t+k}{4}-\frac{K_t+k}{4}-1}{(K_t+k)\sqrt{K_t+k-1}} \\
&\qquad=\beta_0\frac{(\sqrt{K_t+k-1}-\frac{\sqrt{K_t+k}}{2})^2-\frac{K_t+k}{4}-1}{(K_t+k)\sqrt{K_t+k-1}} \\
&\qquad=\beta_0\frac{(\sqrt{K_t+k-1}-\frac{\sqrt{K_t+k}}{2}-\frac{\sqrt{K_t+k}}{2})(\sqrt{K_t+k-1}-\frac{\sqrt{K_t+k}}{2}+\frac{\sqrt{K_t+k}}{2})-1}{(K_t+k)\sqrt{K_t+k-1}} \\
&\qquad=\beta_0\frac{\overbrace{(\sqrt{K_t+k-1}-\sqrt{K_t+k})}^{<0}\sqrt{K_t+k-1}-1}{(K_t+k)\sqrt{K_t+k-1}} \\
&\qquad<0
\end{aligned} \tag{A.41}$$

Finally, noting the definition of $S_{\beta_{t,k}}(\boldsymbol{x}_{t,k+1})$ and taking full expectation on both sides, we arrive at

$$\begin{aligned}
&\mathbb{E}\left[S_{\beta_{t,k}}(\boldsymbol{x}_{t,k+1})\right] \\
&\quad\leq(1-\gamma_{t,k})\mathbb{E}\left[S_{\beta_{t,k-1}}(\boldsymbol{x}_{t,k})\right]+\gamma_{t,k}\mathcal{D}_{\mathcal{X}}\mathbb{E}\left[\left\|\nabla F_{\beta_{t,k}}(\boldsymbol{x}_{t,k})-\boldsymbol{v}_{t,k}\right\|\right]+\frac{\mathcal{D}_{\mathcal{X}}^2\gamma_{t,k}^2}{2}\left(L_f+\frac{L_{\boldsymbol{A}}}{\beta_{t,k}}\right).
\end{aligned} \tag{A.42}$$

1.2 Recursion of $S_{\beta_{t,k}}$ at the 'edges'

We now want to show that the same recursion holds when going for $S_{\beta_{t,1}}(\boldsymbol{x}_{t,2})$ and $S_{\beta_{t-1,K_{t-1}}}(\boldsymbol{x}_{t-1,K_{t-1}+1})$. We follow similar steps as in the previous section (which we shorten now for conciseness). Using smoothness and $\boldsymbol{x}_{t,1}=\boldsymbol{x}_{t-1,K_{t-1}+1}$ (from Algorithm 2.2), we get

$$F_{\beta_{t,1}}(\boldsymbol{x}_{t,2})\leq F_{\beta_{t,1}}(\boldsymbol{x}_{t,1})+\gamma_{t,1}\langle\nabla F_{\beta_{t,1}}(\boldsymbol{x}_{t,1}),\,\boldsymbol{w}_{t,1}-\boldsymbol{x}_{t,1}\rangle+\frac{\mathcal{D}_{\mathcal{X}}^2\gamma_{t,1}^2}{2}\left(L_f+\frac{L_{\boldsymbol{A}}}{\beta_{t,1}}\right). \tag{A.43}$$

Since $\boldsymbol{v}_{t,1} = \nabla F_{\beta_{t,1}}(\boldsymbol{x}_{t,1})$ and $\boldsymbol{w}_{t,1} = \mathrm{LMO}_{\mathcal{X}}(\boldsymbol{v}_{t,1})$, it holds that $\langle \nabla F_{\beta_{t,1}}(\boldsymbol{x}_{t,1}), \boldsymbol{w}_{t,1} - \boldsymbol{x}_{t,1}\rangle \leq \langle \nabla F_{\beta_{t,1}}(\boldsymbol{x}_{t,1}), \boldsymbol{x}^\star - \boldsymbol{x}_{t,1}\rangle$. Further using the definition of F_β, the convexity of f and property (A.4) we get

$$
\begin{aligned}
&\langle \nabla F_{\beta_{t,1}}(\boldsymbol{x}_{t,1}),\, \boldsymbol{w}_{t,1} - \boldsymbol{x}_{t,1}\rangle \\
&\qquad \leq \langle \nabla F_{\beta_{t,1}}(\boldsymbol{x}_{t,1}),\, \boldsymbol{x}^\star - \boldsymbol{x}_{t,1}\rangle \\
&\qquad = \langle \nabla f(\boldsymbol{x}_{t,1}) + \nabla_{\boldsymbol{x}} G_{\beta_{t,1}}(\boldsymbol{A}\boldsymbol{x}_{t,1}),\, \boldsymbol{x}^\star - \boldsymbol{x}_{t,1}\rangle \\
&\qquad \leq f^\star - f(\boldsymbol{x}_{t,1}) + \mathbb{E}_{t,1}\left[\langle \widetilde{\nabla}_{\boldsymbol{x}} g_{\beta_{t,1}}(\boldsymbol{A}(\xi_{\mathcal{Q}_t})\boldsymbol{x}_{t,1}),\, \boldsymbol{x}^\star - \boldsymbol{x}_{t,1}\rangle\right] \\
&\qquad \leq f^\star - f(\boldsymbol{x}_{t,1}) + \mathbb{E}_{t,1}\left[\underbrace{\widetilde{g}(\boldsymbol{A}(\xi_{\mathcal{Q}_t})\boldsymbol{x}^\star)}_{=0 \text{ a.s.}} - \widetilde{g}_{\beta_{t,1}}(\boldsymbol{A}(\xi_{\mathcal{Q}_t})\boldsymbol{x}_{t,1}) - \frac{\beta_{t,1}}{2}\left\| \widetilde{\boldsymbol{\lambda}^\star_{\beta_{t,1}}(\boldsymbol{A}(\xi_{\mathcal{Q}_t})\boldsymbol{x}_{t,1})} \right\|^2\right] \\
&\qquad \leq f^\star \underbrace{- f(\boldsymbol{x}_{t,1}) - G_{\beta_{t,1}}(\boldsymbol{A}\boldsymbol{x}_{t,1})}_{=-F_{\beta_{t,1}}(x_t,1)} - \frac{\beta_{t,1}}{2}\mathbb{E}_{t,1}\left[\left\| \widetilde{\boldsymbol{\lambda}^\star_{\beta_{t,1}}(\boldsymbol{A}(\xi_{\mathcal{Q}_t})\boldsymbol{x}_{t,1})} \right\|^2\right].
\end{aligned} \tag{A.44}
$$

We remark that we can still transition from $F_{\beta_{t,1}}(\boldsymbol{x}_{t,1})$ to $F_{\beta_{t-1,K_{t-1}}}(\boldsymbol{x}_{t,1})$ using property (A.12) since the β's are 'continuous' at the edge, i.e., $\beta_{t-1,K_{t-1}} = \frac{\beta_0}{\sqrt{K_{t-1}+K_{t-1}}} = \frac{\beta_0}{\sqrt{K_t}}$ and $\beta_{t,1} = \frac{\beta_0}{\sqrt{K_t+1}}$. It thus holds that

$$
\begin{aligned}
&F_{\beta_{t,1}}(\boldsymbol{x}_{t,1}) \\
&\qquad = \mathbb{E}_{t,1}\left[\widetilde{f}(\boldsymbol{x}_{t,1}, \xi_{\mathcal{Q}_t}) + \widetilde{g}_{\beta_{t,1}}(\boldsymbol{A}(\xi_{\mathcal{Q}_t})\boldsymbol{x}_{t,1})\right] \\
&\qquad \leq \mathbb{E}_{t,1}\left[\widetilde{f}(\boldsymbol{x}_{t,1}, \xi_{\mathcal{Q}_t}) + \widetilde{g}_{\beta_{t-1,K_{t-1}}}(\boldsymbol{A}(\xi_{\mathcal{Q}_t})\boldsymbol{x}_{t,1}) + \frac{\beta_{t-1,K_{t-1}} - \beta_{t,1}}{2}\left\| \widetilde{\boldsymbol{\lambda}^\star_{\beta_{t,1}}(\boldsymbol{A}(\xi_{\mathcal{Q}_t})\boldsymbol{x}_{t,1})} \right\|^2\right] \\
&\qquad = F_{\beta_{t-1,K_{t-1}}}(\boldsymbol{x}_{t,1}) + \mathbb{E}_{t,1}\left[\frac{\beta_{t-1,K_{t-1}} - \beta_{t,1}}{2}\left\| \widetilde{\boldsymbol{\lambda}^\star_{\beta_{t,1}}(\boldsymbol{A}(\xi_{\mathcal{Q}_t})\boldsymbol{x}_{t,1})} \right\|^2\right].
\end{aligned} \tag{A.45}
$$

Using (A.45) and (A.44), (A.43) becomes

$$
\begin{aligned}
F_{\beta_{t,1}}(\boldsymbol{x}_{t,2}) &\leq (1-\gamma_{t,1})F_{\beta_{t,1}}(\boldsymbol{x}_{t,1}) + \gamma_{t,1}f^\star - \frac{\gamma_{t,1}\beta_{t,1}}{2}\mathbb{E}\left[\left\| \widetilde{\boldsymbol{\lambda}^\star_{\beta_{t,1}}(\boldsymbol{A}(\xi_{\mathcal{Q}_t})\boldsymbol{x}_{t,1})} \right\|^2\right] + \frac{\mathcal{D}_{\mathcal{X}}^2\gamma_{t,1}^2}{2}\left(L_f + \frac{L_{\boldsymbol{A}}}{\beta_{t,1}}\right) \\
&\leq (1-\gamma_{t,1})F_{\beta_{t-1,K_{t-1}}}(\boldsymbol{x}_{t,1}) + \gamma_{t,1}f^\star \\
&\quad + \underbrace{\frac{(1-\gamma_{t,1})(\beta_{t-1,K_{t-1}} - \beta_{t,1}) - \gamma_{t,1}\beta_{t,1}}{2}}_{<0, \text{ as before}}\mathbb{E}_{t,1}\left[\left\| \widetilde{\boldsymbol{\lambda}^\star_{\beta_{t,1}}(\boldsymbol{A}(\xi_{\mathcal{Q}_t})\boldsymbol{x}_{t,1})} \right\|^2\right] + \frac{\mathcal{D}_{\mathcal{X}}^2\gamma_{t,1}^2}{2}\left(L_f + \frac{L_{\boldsymbol{A}}}{\beta_{t,1}}\right).
\end{aligned}
$$

Finally, subtracting $f^\star$ from both sides and taking the expectation, we have

$$\mathbb{E}\left[S_{\beta_{t,1}}(\boldsymbol{x}_{t,2})\right] \leq (1-\gamma_{t,1})\mathbb{E}\left[S_{\beta_{t-1,K_{t-1}}}(\boldsymbol{x}_{t,1})\right] + \frac{\mathcal{D}_{\mathcal{X}}^2\gamma_{t,1}^2}{2}\left(L_f + \frac{L_{\boldsymbol{A}}}{\beta_{t,1}}\right). \tag{A.46}$$

2. Convergence rates for the finite sum case For ease, we first cast the index pairs (t,k) to their corresponding global index counterparts (in a sense, we flatten the double loop structure). The variables indexed by (t,k) can be seen as equivalently indexed by $\rho(t,k) = K_t + k \coloneqq 2^{t-1} + k,\ t \in \mathbb{N},\ k \in [2^{t-1}]$.

The following properties hold for ρ:

- $\rho(t,k+1) = \rho(t,k) + 1$
- $\rho(t-1, K_{t-1}+1) = \rho(t-1,K_{t-1}) + 1 = \rho(t,1)$ (the 'increment-by-one' rule holds between the last iteration of epoch $t-1$ and the first iteration of epoch t)

In other words, $\rho(t,k)$ returns for iteration (t,k) its global index since the beginning of Algorithm 2.2.

We use this new indexing scheme and its properties to rewrite relations A.42 and A.46 into a single, global inequality. Note that here ρ should be read as $\rho(t,k)$, for some given, arbitrary t,k.

$$\mathbb{E}\left[S_{\beta_\rho}(\boldsymbol{x}_{\rho+1})\right] \leq (1-\gamma_\rho)\mathbb{E}\left[S_{\beta_{\rho-1}}(\boldsymbol{x}_\rho)\right] + \gamma_\rho\mathcal{D}_{\mathcal{X}}\mathbb{E}\left[\left\|\nabla F_{\beta_\rho}(\boldsymbol{x}_\rho) - \boldsymbol{v}_\rho\right\|\right] + \frac{\mathcal{D}_{\mathcal{X}}^2\gamma_\rho^2}{2}\left(L_f + \frac{L_{\boldsymbol{A}}}{\beta_\rho}\right) \tag{A.47}$$

Further replacing the parameter rates and the variance bound of Lemma 2.2 (subject to Jensen's inequality):

$$\begin{aligned}\mathbb{E}\left[S_{\beta_\rho}(\boldsymbol{x}_{\rho+1})\right] &= \left(1-\frac{2}{\rho}\right)\mathbb{E}\left[S_{\beta_{\rho-1}}(\boldsymbol{x}_\rho)\right] + \frac{2\mathcal{D}_{\mathcal{X}}^2\sqrt{16L_f^2 + \frac{196L_{\boldsymbol{A}}^2}{\beta_0^2}}}{\rho\sqrt{\rho}} + \frac{2\mathcal{D}_{\mathcal{X}}^2L_f}{\rho^2} + \frac{2\mathcal{D}_{\mathcal{X}}^2L_{\boldsymbol{A}}}{\beta_0\rho\sqrt{\rho}} \\ &\leq \left(1-\frac{2}{\rho}\right)\mathbb{E}\left[S_{\beta_{\rho-1}}(\boldsymbol{x}_\rho)\right] + \frac{1}{\rho^{3/2}}\left(2\mathcal{D}_{\mathcal{X}}^2L_f + 2\mathcal{D}_{\mathcal{X}}^2\sqrt{16L_f^2 + \frac{196L_{\boldsymbol{A}}^2}{\beta_0^2}} + \frac{2\mathcal{D}_{\mathcal{X}}^2L_{\boldsymbol{A}}}{\beta_0}\right)\end{aligned}$$

We can now apply Lemma A.2, with $\alpha = 1,\ \beta = 3/2,\ b = 2\mathcal{D}_{\mathcal{X}}^2L_f + 2\mathcal{D}_{\mathcal{X}}^2\sqrt{16L_f^2 + \frac{196L_{\boldsymbol{A}}^2}{\beta_0^2}} + \frac{2\mathcal{D}_{\mathcal{X}}^2L_{\boldsymbol{A}}}{\beta_0}$, $c = 2,\ k_0 = 0$ and

$C_3 = \max\left\{S_{\beta_{1,0}}(\boldsymbol{x}_{1,1}),\, 2\mathcal{D}_{\mathcal{X}}^2 L_f + 2\mathcal{D}_{\mathcal{X}}^2\sqrt{16L_f^2 + \frac{196L_{\boldsymbol{A}}^2}{\beta_0^2}} + \frac{2\mathcal{D}_{\mathcal{X}}^2 L_{\boldsymbol{A}}}{\beta_0}\right\}$ to get

$$\mathbb{E}\left[F_{\beta_\rho}(\boldsymbol{x}_{\rho+1}) - f^\star\right] \leq \frac{C_3}{\sqrt{\rho+1}}$$

$$\Updownarrow$$

$$\mathbb{E}\left[S_{\beta_{t,k}}(\boldsymbol{x}_{t,k+1})\right] \leq \frac{C_3}{\sqrt{K_t+k+1}}.$$

3. Convergence rates for the general expectation case

Following the same steps for the general expectation case, we get

$$\begin{aligned}\mathbb{E}\left[S_{\beta_\rho}(\boldsymbol{x}_{\rho+1})\right] &= \left(1-\frac{2}{\rho}\right)\mathbb{E}\left[S_{\beta_{\rho-1}}(\boldsymbol{x}_\rho)\right] + \frac{2\mathcal{D}_{\mathcal{X}}\sqrt{16L_f^2\mathcal{D}_{\mathcal{X}}^2 + 2L_{\boldsymbol{A}}^2\mathcal{D}_{\mathcal{X}}^2\left(\frac{98}{\beta_0^2}+1\right) + 2\beta_0^2\sigma_f^2}}{\rho\sqrt{\rho}} \\ &\qquad + \frac{2\mathcal{D}_{\mathcal{X}}^2 L_f}{\rho^2} + \frac{2\mathcal{D}_{\mathcal{X}}^2 L_{\boldsymbol{A}}}{\beta_0\rho\sqrt{\rho}} \\ &\leq \left(1-\frac{2}{\rho}\right)\mathbb{E}\left[S_{\beta_{\rho-1}}(\boldsymbol{x}_\rho)\right] + \frac{1}{\rho^{3/2}}\Bigg(2\mathcal{D}_{\mathcal{X}}^2 L_f + \frac{2\mathcal{D}_{\mathcal{X}}^2 L_{\boldsymbol{A}}}{\beta_0} \\ &\qquad + 2\mathcal{D}_{\mathcal{X}}\sqrt{16L_f^2\mathcal{D}_{\mathcal{X}}^2 + 2L_{\boldsymbol{A}}^2\mathcal{D}_{\mathcal{X}}^2\left(\frac{98}{\beta_0^2}+1\right) + 2\beta_0^2\sigma_f^2}\Bigg).\end{aligned}$$

We can now apply Lemma A.2, with $b = 2\mathcal{D}_{\mathcal{X}}^2 L_f + \frac{2\mathcal{D}_{\mathcal{X}}^2 L_{\boldsymbol{A}}}{\beta_0} + 2\mathcal{D}_{\mathcal{X}}\sqrt{16L_f^2\mathcal{D}_{\mathcal{X}}^2 + 2L_{\boldsymbol{A}}^2\mathcal{D}_{\mathcal{X}}^2\left(\frac{98}{\beta_0^2}+1\right) + 2\beta_0^2\sigma_f^2}$, $C_4 = \max\left\{S_{\beta_{1,0}}(x_{1,1}),\, 2\mathcal{D}_{\mathcal{X}}^2 L_f + \frac{2\mathcal{D}_{\mathcal{X}}^2 L_{\boldsymbol{A}}}{\beta_0} + 2\mathcal{D}_{\mathcal{X}}\sqrt{16L_f^2\mathcal{D}_{\mathcal{X}}^2 + 2L_{\boldsymbol{A}}^2\mathcal{D}_{\mathcal{X}}^2\left(\frac{98}{\beta_0^2}+1\right) + 2\beta_0^2\sigma_f^2}\right\}$ and $c = 2$, $\alpha = 1$, $\beta = 3/2$, to get

$$\mathbb{E}\left[S_{\beta_{t,k}}(\boldsymbol{x}_{1,k+1})\right] \leq \frac{C_4}{\sqrt{K_t+k+1}} \qquad \square$$

A.2.4 Proof of Corollary 2.2

Corollary 2.2. *The expected convergence in terms of objective suboptimality and feasibility of Algorithm 2.2 is, respectively,*

$$\left|\mathbb{E}\left[f(\boldsymbol{x}_{t,k})\right] - f^\star\right| \qquad \in \mathcal{O}\left((K_t+k)^{-1/2}\right)$$

$$\sqrt{\mathbb{E}\left[\operatorname{dist}(\boldsymbol{A}(\xi)\boldsymbol{x}_{t,k}, \boldsymbol{b}(\xi))^2\right]} \qquad \in \mathcal{O}\left((K_t+k)^{-1/2}\right)$$

for both the finite sum and the general expectation setting. Consequently, the oracle complexities are given by #(IFO) $\in \mathcal{O}\left(n\log_2(\epsilon^{-2}) + \epsilon^{-4}\right)$ *and #(LMO)* $\in \mathcal{O}\left(\epsilon^{-2}\right)$ *for the finite-sum setting, and by #(SFO)* $\in \mathcal{O}\left(\epsilon^{-4}\right)$ *and #(LMO)* $\in \mathcal{O}\left(\epsilon^{-2}\right)$ *for the expectation setting.*

Proof. A simple application of Lemma 3.1 of Fercoq et al. [81] for the previously derived convergence bounds of the smoothed gap, along with our chosen decrease rate for β yields the stated results.

For the oracle complexities, we choose a total number of outer loops T_ϵ in order to achieve a desired ϵ-accuracy.

$$\frac{1}{\sqrt{K_t+k}} \leq \epsilon \implies \frac{1}{\epsilon^2} \leq K_t + k \leq 2^t \implies T_\epsilon \geq \log_2\left(\epsilon^{-2}\right)$$

We can now state the corresponding complexity in terms of #(IFO) and #(LMO) for the finite-sum case of Algorithm 2.2:

$$\begin{aligned}
\#(\text{IFO}) &= \sum_{t=1}^{T_\epsilon}\left(n + \sum_{k=2}^{K_t} K_t\right) \\
&= \sum_{t=1}^{T_\epsilon}\left(n + 2^{2(t-1)}\right) \\
&= nT_\epsilon + \mathcal{O}\left(2^{2T_\epsilon}\right) \in \mathcal{O}\left(\epsilon^{-4}\right)
\end{aligned}$$

$$\#(\text{LMO}) = \sum_{t=1}^{T_\epsilon} K_t \leq 2K_{T_\epsilon} = 2^{T_\epsilon} \in \mathcal{O}\left(\epsilon^{-2}\right)$$

For the general expectation case, following the same steps, we get:

$$\#(\text{SFO}) = \sum_{t=1}^{T_\epsilon}\left(|\mathcal{Q}_t| + \sum_{k=2}^{K_t} K_t\right)$$

$$
\begin{aligned}
&= \sum_{t=1}^{T_\epsilon} \left(\left\lceil \frac{2K_t}{\beta_{t,1}^2} \right\rceil + 2^{2(t-1)} \right) \\
&\leq \sum_{t=1}^{T_\epsilon} \left(\frac{2K_t}{\beta_{t,1}^2} + 1 + 2^{2(t-1)} \right) \\
&= \sum_{t=1}^{T_\epsilon} \left(\frac{2^t(2^{t-1}+1)}{\beta_0^2} + 1 + 2^{2(t-1)} \right) \\
&= \underbrace{\frac{1}{\beta_0^2} \sum_{t=1}^{T_\epsilon} 2^{2t-1}}_{\substack{\in \mathcal{O}(2^{2T_\epsilon}) \\ \equiv \mathcal{O}(\epsilon^{-4})}} + \underbrace{\frac{1}{\beta_0^2} \sum_{t=1}^{T_\epsilon} 2^t}_{\substack{\in \mathcal{O}(2^{T_\epsilon}) \\ \equiv \mathcal{O}(\epsilon^{-2})}} + \underbrace{T_\epsilon}_{\in \mathcal{O}(\log_2(\epsilon^{-2}))} + \underbrace{\sum_{t=1}^{T_\epsilon} 2^{2(t-1)}}_{\substack{\in \mathcal{O}(2^{2T_\epsilon}) \\ \equiv \mathcal{O}(\epsilon^{-4})}} \\
&\in \mathcal{O}\left(\epsilon^{-4}\right)
\end{aligned}
$$

$$
\#(\text{LMO}) = \sum_{t=1}^{T_\epsilon} K_t \leq 2K_{T_\epsilon} = 2^{T_\epsilon} \in \mathcal{O}\left(\epsilon^{-2}\right) \qquad \square
$$

A.3 Analysis of H-SAG-CGM

A.3.1 Proof of Lemma 2.4

Lemma 2.4. *Consider H-SAG-CGM (Algorithm 2.3). Then, for all $k \geq 1$, it holds that*

$$S_{\beta_k}(\boldsymbol{x}_{k+1}) \leq (1-\gamma_k) S_{\beta_{k-1}}(\boldsymbol{x}_k) + \gamma_k D_{\mathcal{X}} \mathbb{E}\left[\|\nabla F_{\beta_k}(\boldsymbol{x}_k) - \boldsymbol{d}_k\|\right] + \frac{\gamma_k^2 D_{\mathcal{X}}^2 L_{F_{\beta_k}}}{2},$$

where $L_{F_{\beta_k}} = \frac{\|\boldsymbol{H}\| L_f}{n} + \frac{\|\boldsymbol{A}\|}{\beta_k m}$ represents the smoothness constant of the surrogate objective F_{β_k}.

Proof. We follow the steps laid out in Theorem 4.1 by Vladarean et al. [219], which in turn builds upon Theorem 9 of Locatello et al. [142].

We use the quadratic upper bound ensured by the fact that F_{β_k} is $L_{F_{\beta_k}}$-smooth:

$$F_{\beta_k}(\boldsymbol{x}_{k+1}) \leq F_{\beta_k}(\boldsymbol{x}_k) + \langle \nabla F_{\beta_k}(\boldsymbol{x}_k), \boldsymbol{x}_{k+1} - \boldsymbol{x}_k \rangle + \frac{L_{F_{\beta_k}}}{2} \|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k\|^2 \tag{A.48}$$

$$\leq F_{\beta_k}(\boldsymbol{x}_k) + \gamma_k \langle \nabla F_{\beta_k}(\boldsymbol{x}_k), \boldsymbol{w}_k - \boldsymbol{x}_k \rangle + \frac{\gamma_k^2 L_{F_{\beta_k}} \mathcal{D}_{\mathcal{X}}^2}{2} \tag{A.49}$$

where the second line follows from the boundedness of $\mathcal{X}$.

Next, we use the rule for change of β in smoothing (see (A.5)), which gives

$$F_{\beta_k}(\boldsymbol{x}_{k+1}) \leq F_{\beta_{k-1}}(\boldsymbol{x}_k) + \frac{\beta_{k-1} - \beta_k}{2} \|\boldsymbol{\lambda}^\star_{\beta_k}(\boldsymbol{A}\boldsymbol{x}_k)\|^2 + \gamma_k \langle \nabla F_{\beta_k}(\boldsymbol{x}_k), \boldsymbol{w}_k - \boldsymbol{x}_k \rangle + \frac{\gamma_k^2 L_{F_{\beta_k}} \mathcal{D}_{\mathcal{X}}^2}{2}, \tag{A.50}$$

where $y^\star_{\beta_k}$ is defined as in (A.2).

Then, we bound the term $\langle \nabla F_{\beta_k}(\boldsymbol{x}_k), \boldsymbol{w}_k - \boldsymbol{x}_k \rangle$ as follows:

$$\langle \nabla F_{\beta_k}(\boldsymbol{x}_k), \boldsymbol{w}_k - \boldsymbol{x}_k \rangle = \langle \nabla F_{\beta_k}(\boldsymbol{x}_k) - \boldsymbol{d}_k, \boldsymbol{w}_k - \boldsymbol{x}_k \rangle + \langle \boldsymbol{d}_k, \boldsymbol{w}_k - \boldsymbol{x}_k \rangle \tag{A.51}$$

$$= \langle \nabla F_{\beta_k}(\boldsymbol{x}_k) - \boldsymbol{d}_k, \boldsymbol{w}_k - \boldsymbol{x}^\star \rangle + \langle \nabla F_{\beta_k}(\boldsymbol{x}_k) - \boldsymbol{d}_k, \boldsymbol{x}^\star - \boldsymbol{x}_k \rangle + \langle \boldsymbol{d}_k, \boldsymbol{w}_k - \boldsymbol{x}_k \rangle \tag{A.52}$$

$$\leq \langle \nabla F_{\beta_k}(\boldsymbol{x}_k) - \boldsymbol{d}_k, \boldsymbol{w}_k - \boldsymbol{x}^\star \rangle + \langle \nabla F_{\beta_k}(\boldsymbol{x}_k) - \boldsymbol{d}_k, \boldsymbol{x}^\star - \boldsymbol{x}_k \rangle + \langle \boldsymbol{d}_k, \boldsymbol{x}^\star - \boldsymbol{x}_k \rangle \tag{A.53}$$

$$= \langle \nabla F_{\beta_k}(\boldsymbol{x}_k) - \boldsymbol{d}_k, \boldsymbol{w}_k - \boldsymbol{x}^\star \rangle + \langle \nabla F_{\beta_k}(\boldsymbol{x}_k), \boldsymbol{x}^\star - \boldsymbol{x}_k \rangle \tag{A.54}$$

where the inequality follows by the definition of $\boldsymbol{w}_k$.

Now, we focus on the term $\langle \nabla F_{\beta_k}(\boldsymbol{x}_k), \boldsymbol{x}^\star - \boldsymbol{x}_k \rangle$ and bound it as follows:

$$\langle \nabla F_{\beta_k}(\boldsymbol{x}_k), \boldsymbol{x}^\star - \boldsymbol{x}_k \rangle = \langle \boldsymbol{H}^\top \nabla f(\boldsymbol{H}\boldsymbol{x}_k) + \boldsymbol{A}^\top \nabla g_{\beta_k}(\boldsymbol{A}\boldsymbol{x}_k), \boldsymbol{x}^\star - \boldsymbol{x}_k \rangle \tag{A.55}$$

$$= \langle \nabla f(\boldsymbol{H}\boldsymbol{x}_k), \boldsymbol{H}(\boldsymbol{x}^\star - \boldsymbol{x}_k) \rangle + \langle \nabla g_{\beta_k}(\boldsymbol{A}\boldsymbol{x}_k), \boldsymbol{A}(\boldsymbol{x}^\star - \boldsymbol{x}_k) \rangle \tag{A.56}$$

$$\leq f(\boldsymbol{H}\boldsymbol{x}^\star) - f(\boldsymbol{H}\boldsymbol{x}_k) + g(\boldsymbol{A}\boldsymbol{x}^\star) - g_{\beta_k}(\boldsymbol{A}\boldsymbol{x}_k) - \frac{\beta_k}{2} \|\boldsymbol{\lambda}^\star_{\beta_k}(\boldsymbol{A}\boldsymbol{x}_k)\|^2 \tag{A.57}$$

$$= F^\star - F_{\beta_k}(\boldsymbol{x}_k) - \frac{\beta_k}{2}\|\boldsymbol{\lambda}^\star_{\beta_k}(\boldsymbol{A}\boldsymbol{x}_k)\|^2, \tag{A.58}$$

where the inequality holds due to the convexity of f and g and the smoothing property in (A.4).

Combining all these bounds and subtracting $F^\star$ from both sides, we get

$$\begin{aligned} F_{\beta_k}(\boldsymbol{x}_{k+1}) - F^\star \leq (1-\gamma_k)\left(F_{\beta_{k-1}}(\boldsymbol{x}_k) - F^\star\right) + \gamma_k\langle\nabla F_{\beta_k}(\boldsymbol{x}_k) - \boldsymbol{d}_k, \boldsymbol{w}_k - \boldsymbol{x}^\star\rangle \\ + \frac{1}{2}((1-\gamma_k)(\beta_{k-1}-\beta_k) - \gamma_k\beta_k)\|\boldsymbol{\lambda}^\star_{\beta_k}(\boldsymbol{A}\boldsymbol{x}_k)\|_2^2 + \frac{\gamma_k^2 L_{F_{\beta_k}}\mathcal{D}_{\mathcal{X}}^2}{2} \end{aligned} \tag{A.59}$$

We cannot bound $\|\boldsymbol{\lambda}^\star_{\beta_k}(\boldsymbol{A}\boldsymbol{x}_k)\|_2^2$ in general, so we choose γ_k and β_k carefully to vanish this term. Let $\gamma_k = \frac{2}{k+1}$ and $\beta_k = \frac{\beta_0}{\sqrt{k+1}}$ for an arbitrary $\beta_0 > 0$. Then,

$$(1-\gamma_k)(\beta_{k-1}-\beta_k) - \gamma_k\beta_k = \frac{\beta_0}{\sqrt{k}}\left(\frac{k-1}{k+1} - \frac{\sqrt{k}}{\sqrt{k+1}}\right) < 0, \qquad \text{for all } k \geq 1. \tag{A.60}$$

Finally, taking expectation on both sides and applying the definition of $S_\beta(\boldsymbol{x}) \coloneqq \mathbb{E}\left[F_\beta(\boldsymbol{x}) - F^\star\right]$ we arrive at our stated result:

$$S_{\beta_k}(\boldsymbol{x}_{k+1}) \leq (1-\gamma_k)S_{\beta_{k-1}}(\boldsymbol{x}_k) + \gamma_k\mathbb{E}[\langle\nabla F_{\beta_k}(\boldsymbol{x}_k) - \boldsymbol{d}_k, \boldsymbol{w}_k - \boldsymbol{x}^\star\rangle] + \frac{\gamma_k^2 L_{F_{\beta_k}}\mathcal{D}_{\mathcal{X}}^2}{2}. \qquad \square$$

A.3.2 Proof of Lemma 2.6

The following Lemma will be needed in the subsequent characterization of the estimator variance.

Lemma A.2 (Adaptation of the similar intermediary result from the proof of Lemma 3 of Négiar et al. [159])**.** *Let $\rho \in (0,1)$, $C > 0$ and $\{u_k\}_{k\in\mathbb{N}}$ be a sequence such that*

$$u_k \leq \rho(u_{k-1} + \frac{1}{\sqrt{k}}C). \tag{A.61}$$

Then, it holds that

$$u_k \leq \rho^k u_0 + \frac{2C\rho}{\sqrt{k}(1-\rho)}. \tag{A.62}$$

Proof. Unrolling the recurrence yields

$$u_k \leq \rho^k u_0 + C\sum_{i=1}^{k} \frac{\rho^{k-i+1}}{\sqrt{i}} \tag{A.63}$$

Observe that ρ^{k+1-i} is a monotonically increasing with i because $\rho \in (0,1)$. Therefore,

$$\frac{1}{\sum_{i=1}^{k}\frac{1}{\sqrt{i}}}\sum_{i=1}^{k}\frac{\rho^{k-i+1}}{\sqrt{i}} \leq \frac{1}{k}\sum_{i=1}^{k}\rho^{k-i+1} = \frac{1}{k}\sum_{i=1}^{k}\rho^{i} \tag{A.64}$$

since the left side of the inequality is a weighted average of ρ^{k-i+1} with decreasing weights and the right side is the simple average with uniform weights. The equality holds simply by change of indices. Now, we rearrange as

$$\sum_{i=1}^{k}\frac{\rho^{k-i+1}}{\sqrt{i}} \leq \frac{1}{k}\left(\sum_{i=1}^{k}\frac{1}{\sqrt{i}}\right)\left(\sum_{i=1}^{k}\rho^{i}\right) \leq \frac{2\rho}{\sqrt{k}(1-\rho)} \tag{A.65}$$

We complete the proof by combining (A.63) and (A.65). □

Lemma 2.6. *Consider H-SAG-CGM (Algorithm 2.3) and the SAG estimator γ_k defined in* (2.17)*. Then, for all $k \geq 2$,*

$$\mathbb{E}[\|\nabla g_{\beta_k}(\boldsymbol{A}\boldsymbol{x}_k) - \boldsymbol{q}_k\|_1] \leq \left(1 - \tfrac{1}{m}\right)^k \|\nabla g_{\beta_0}(\boldsymbol{A}\boldsymbol{x}_0) - \boldsymbol{q}_0\|_1 + \frac{C}{\sqrt{k}}$$

where $C = 10\beta_0^{-1}\mathcal{D}_1(\boldsymbol{A})$ and the expectation is taken over all previous steps of the algorithm.

Proof of Lemma 2.6 for indicator functions

First, we prove Lemma 2.6 for the case in which g is an indicator function of a set $\mathcal{K} \in \mathbb{R}^m$, with $\mathcal{K} \coloneqq \mathcal{K}_1 \times \ldots \times \mathcal{K}_m$, $\mathcal{K}_i \in \mathbb{R}$, $\forall i \in [m]$. Observe that

$$\mathbb{E}_k[|\nabla g_{\beta_k}(\boldsymbol{A}\boldsymbol{x}_k)_j - \boldsymbol{q}_{k,j}|] = \frac{1}{m}0 + \frac{m-1}{m}|\nabla g_{\beta_k}(\boldsymbol{A}\boldsymbol{x}_k)_j - \boldsymbol{q}_{k-1,j}|. \tag{A.66}$$

Summing over all coordinates gives

$$\begin{aligned}
&\mathbb{E}[\|\nabla g_{\beta_k}(\boldsymbol{A}\boldsymbol{x}_k) - \boldsymbol{q}_k\|_1] \\
&\qquad = \frac{m-1}{m}\mathbb{E}[\|\nabla g_{\beta_k}(\boldsymbol{A}\boldsymbol{x}_k) - \boldsymbol{q}_{k-1}\|_1] \\
&\qquad = \frac{m-1}{m}\mathbb{E}[\|\nabla g_{\beta_k}(\boldsymbol{A}\boldsymbol{x}_k) - \nabla g_{\beta_{k-1}}(\boldsymbol{A}\boldsymbol{x}_{k-1}) + \nabla g_{\beta_{k-1}}(\boldsymbol{A}\boldsymbol{x}_{k-1}) - \boldsymbol{q}_{k-1}\|_1] \\
&\qquad \leq \frac{m-1}{m}\Big(\mathbb{E}[\|\nabla g_{\beta_{k-1}}(\boldsymbol{A}\boldsymbol{x}_{k-1}) - \boldsymbol{q}_{k-1}\|_1] + \mathbb{E}[\|\nabla g_{\beta_k}(\boldsymbol{A}\boldsymbol{x}_k) - \nabla g_{\beta_{k-1}}(\boldsymbol{A}\boldsymbol{x}_{k-1})\|_1]\Big).
\end{aligned} \tag{A.67}$$

Now, we focus on the last term and bound it as follows:

$$\begin{aligned}
&\|\nabla g_{\beta_k}(\boldsymbol{A}\boldsymbol{x}_k) - \nabla g_{\beta_{k-1}}(\boldsymbol{A}\boldsymbol{x}_{k-1})\|_1 \\
&\qquad = \|\nabla g_{\beta_k}(\boldsymbol{A}\boldsymbol{x}_k) \pm \nabla g_{\beta_k}(\boldsymbol{A}\boldsymbol{x}_{k-1}) - \nabla g_{\beta_{k-1}}(\boldsymbol{A}\boldsymbol{x}_{k-1})\|_1 \\
&\qquad \leq \|\nabla g_{\beta_k}(\boldsymbol{A}\boldsymbol{x}_k) - \nabla g_{\beta_k}(\boldsymbol{A}\boldsymbol{x}_{k-1})\|_1 + \|\nabla g_{\beta_k}(\boldsymbol{A}\boldsymbol{x}_{k-1}) - \nabla g_{\beta_{k-1}}(\boldsymbol{A}\boldsymbol{x}_{k-1})\|_1 \qquad \text{(A.68)} \\
&\qquad \leq \frac{1}{m\beta_k}\|\boldsymbol{A}(\boldsymbol{x}_{k-1} - \boldsymbol{x}_k)\|_1 + \frac{1}{m}\Big(\frac{1}{\beta_k} - \frac{1}{\beta_{k-1}}\Big)\|\boldsymbol{A}\boldsymbol{x}_{k-1} - \mathrm{proj}_K(\boldsymbol{A}\boldsymbol{x}_{k-1})\|_1 \\
&\qquad \leq \frac{\gamma_{k-1}}{m\beta_k}\mathcal{D}_1(\boldsymbol{A}) + \frac{1}{m}\Big(\frac{1}{\beta_k} - \frac{1}{\beta_{k-1}}\Big)\|\boldsymbol{A}\boldsymbol{x}_{k-1} - \boldsymbol{A}\boldsymbol{x}^\star\|_1 \\
&\qquad \leq \frac{\mathcal{D}_1(\boldsymbol{A})}{m}\Big(\frac{\gamma_{k-1}}{\beta_k} + \frac{1}{\beta_k} - \frac{1}{\beta_{k-1}}\Big) \qquad \text{(A.69)}
\end{aligned}$$

where the third inequality is due to the fact that $\mathcal{K} = \mathcal{K}_1 \times \mathcal{K}_2 \times \cdots \times \mathcal{K}_m$. Simplifying further: $\frac{\gamma_{k-1}}{\beta_k} + \frac{1}{\beta_k} - \frac{1}{\beta_{k-1}} = \frac{2}{k}\frac{\sqrt{k+1}}{\beta_0} + \frac{\sqrt{k+1}}{\beta_0} - \frac{\sqrt{k}}{\beta_0} < \frac{2}{k}\frac{\sqrt{k}+1}{\beta_0} + \frac{\sqrt{k}\sqrt{k+1}}{\beta_0\sqrt{k}} - \frac{k}{\beta_0\sqrt{k}} < \frac{2}{\beta_0\sqrt{k}} + \frac{2}{\beta_0 k} + \frac{k+1}{\beta_0\sqrt{k}} - \frac{k}{\beta_0\sqrt{k}} < \frac{5}{\beta_0\sqrt{k}}$, gives

$$\|\nabla g_{\beta_k}(\boldsymbol{A}\boldsymbol{x}_k) - \nabla g_{\beta_{k-1}}(\boldsymbol{A}\boldsymbol{x}_{k-1})\|_1 \leq \frac{5D_1(\boldsymbol{A})}{m\beta_0\sqrt{k}}. \tag{A.70}$$

Substituting this back into (A.67), we get

$$\mathbb{E}[\|\nabla g_{\beta_k}(\boldsymbol{A}\boldsymbol{x}_k) - \boldsymbol{q}_k\|_1] \leq \frac{m-1}{m}\Big(\mathbb{E}[\|\nabla g_{\beta_{k-1}}(\boldsymbol{A}\boldsymbol{x}_{k-1}) - \boldsymbol{q}_{k-1}\|_1] + \frac{5D_2(\boldsymbol{A})\sqrt{m}}{\beta_0\sqrt{k}}\Big). \tag{A.71}$$

This is in the form of (A.61). We conclude the proof by applying Lemma A.2:

$$\mathbb{E}[\|\nabla g_{\beta_k}(\boldsymbol{A}\boldsymbol{x}_k) - \boldsymbol{q}_k\|_1] \leq \left(\frac{m-1}{m}\right)^k \mathbb{E}[\|\nabla g_{\beta_0}(\boldsymbol{A}\boldsymbol{x}_0) - \boldsymbol{q}_0\|_1] + \frac{10D_2(\boldsymbol{A})\sqrt{m}(m-1)}{\beta_0\sqrt{k}}.$$

Proof of Lemma 2.6 for Lipschitz continuous functions

Suppose g is Lipschitz continuous with parameter L_g. Then, from (A.6), we get

$$\underbrace{f(\boldsymbol{H}\boldsymbol{x}_{k+1}) + g(\boldsymbol{A}\boldsymbol{x}_{k+1})}_{F(\boldsymbol{x}_{k+1})} \leq \underbrace{f(\boldsymbol{H}\boldsymbol{x}_{k+1}) + g_{\beta_k}(\boldsymbol{A}\boldsymbol{x}_{k+1})}_{F_{\beta_k}(\boldsymbol{x}_{k+1})} + \frac{\beta_k}{2} L_g^2 = F_{\beta_k}(\boldsymbol{x}_{k+1}) + \frac{\beta_0 L_g^2}{2\sqrt{k+1}}. \tag{A.72}$$

We achieve the desired bound by subtracting $F^\star$ and taking expectation on both sides:

$$\mathbb{E}[F(\boldsymbol{x}_{k+1}) - F^\star] \leq S_{\beta_k}(\boldsymbol{x}_{k+1}) + \frac{\beta_0 L_g^2}{2\sqrt{k+1}}. \tag{A.73}$$

To bound S_{β_k}, we can follow the proof of Lemma 2.6 up to (A.68), which we repeat here for convenience:

$$\|\nabla g_{\beta_k}(\boldsymbol{A}\boldsymbol{x}_k) - \nabla g_{\beta_k}(\boldsymbol{A}\boldsymbol{x}_{k-1})\|_1 + \|\nabla g_{\beta_k}(\boldsymbol{A}\boldsymbol{x}_{k-1}) - \nabla g_{\beta_{k-1}}(\boldsymbol{A}\boldsymbol{x}_{k-1})\|_1$$

Recall that $\nabla g_\beta(\boldsymbol{z}) = \beta^{-1}(\boldsymbol{z} - \operatorname{prox}_{\beta g}(\boldsymbol{z}))$. The first term can be bounded using the $1/\beta$-smoothness of g_β. For the second term, recall the well-established fact that $\operatorname{prox}_g(\boldsymbol{z}) = \lambda \operatorname{prox}_{g/\lambda}(x/\lambda)$ for any $\lambda > 0$. Thus,

$$\begin{aligned}
\nabla g_{\beta_k}(\boldsymbol{A}\boldsymbol{x}_{k-1}) &= {\beta_k}^{-1}(\boldsymbol{A}\boldsymbol{x}_{k-1} - \operatorname{prox}_{\beta_k g}(\boldsymbol{A}\boldsymbol{x}_{k-1})) && \text{(A.74)}\\
&= {\beta_k}^{-1}(\boldsymbol{A}\boldsymbol{x}_{k-1} - \frac{\beta_k}{\beta_{k-1}} \operatorname{prox}_{\beta_{k-1} g}(\frac{\beta_{k-1}}{\beta_k}\boldsymbol{A}\boldsymbol{x}_{k-1})) && \text{(A.75)}\\
&= \nabla g_{\beta_{k-1}}(\frac{\beta_{k-1}}{\beta_k}\boldsymbol{A}\boldsymbol{x}_{k-1}) && \text{(A.76)}
\end{aligned}$$

Thus,

$$\begin{aligned}
&\|\nabla g_{\beta_k}(\boldsymbol{A}\boldsymbol{x}_k) - \nabla g_{\beta_k}(\boldsymbol{A}\boldsymbol{x}_{k-1})\|_1 + \|\nabla g_{\beta_k}(\boldsymbol{A}\boldsymbol{x}_{k-1}) - \nabla g_{\beta_{k-1}}(\boldsymbol{A}\boldsymbol{x}_{k-1})\|_1 && \text{(A.77)}\\
&\leq \frac{1}{m\beta_k}\|\boldsymbol{A}(\boldsymbol{x}_k - \boldsymbol{x}_{k-1})\|_1 + \frac{1}{m\beta_{k-1}}(\frac{\beta_{k-1}}{\beta_k} - 1)\|\boldsymbol{A}\boldsymbol{x}_{k-1}\|_1 && \text{(A.78)}\\
&\leq \frac{\gamma_{k-1}}{m\beta_k}\mathcal{D}_1(\boldsymbol{A}) + \frac{1}{m}(\frac{1}{\beta_k} - \frac{1}{\beta_{k-1}})\|\boldsymbol{A}\boldsymbol{x}_{k-1}\|_1 && \text{(A.79)}\\
&\leq \frac{\mathcal{D}_1(\boldsymbol{A})}{m}\left(\frac{\gamma_{k-1}}{\beta_k} + \frac{1}{\beta_k} - \frac{1}{\beta_{k-1}}\right) && \text{(A.80)}
\end{aligned}$$

Note that this is identical to (A.69) in Lemma A.2. Thus, the rest of Lemma A.2 can be applied to arrive at the same bound.

A.3.3 Proof of Theorem 2.3

Theorem 2.3. *The sequence generated by H-SAG-CGM (Algorithm 2.3) satisfies, for all $k \geq 2$,*

$$S_{\beta_k}(\boldsymbol{x}_{k+1}) \leq \frac{C_1}{\sqrt{k}} + \frac{C_2}{k} + \frac{C_3}{k^2},$$

for the following constants

- $C_1 = \beta_0^{-1}(2\mathcal{D}_{\mathcal{X}}^2 \| \boldsymbol{A} \| + 10\mathcal{D}_1(\boldsymbol{A}))$;
- $C_2 = 8L_f \mathcal{D}_1(\boldsymbol{H})\mathcal{D}_\infty(\boldsymbol{H}) + 2n^{-1}L_f \| \boldsymbol{H} \| \mathcal{D}_{\mathcal{X}}^2$;
- $C_3 = 2n^2\mathcal{D}_\infty(\boldsymbol{H})\big(\|\nabla f(\boldsymbol{H}\boldsymbol{x}_1) - \boldsymbol{p}_0\|_1 + 32L_f\mathcal{D}_1(\boldsymbol{H})\big) + 2m^2\mathcal{D}_\infty(\boldsymbol{A})\|\nabla g_{\beta_0}(\boldsymbol{A}\boldsymbol{x}_1) - \boldsymbol{q}_0\|_1$.

Proof. Our aim is to get a rate on the smoothed gap $S_{\beta_k}(\boldsymbol{x}_{k+1})$. We start from Lemma 2.4:

$$\begin{aligned} &S_{\beta_k}(\boldsymbol{x}_{k+1}) \\ &\quad \leq (1-\gamma_k) S_{\beta_{k-1}}(\boldsymbol{x}_k) + \gamma_k \mathbb{E}[\langle \nabla F_{\beta_k}(\boldsymbol{x}_k) - \boldsymbol{d}_k, \boldsymbol{w}_k - \boldsymbol{x}^\star \rangle] + \frac{\gamma_k^2}{2}\left(\frac{\| \boldsymbol{H} \| L_f}{n} + \frac{\| \boldsymbol{A} \|}{\beta_k}\right)\mathcal{D}_{\mathcal{X}}^2. \end{aligned} \tag{A.81}$$

We multiply both sides by $k(k+1)$ and unroll the recurrence to get

$$\begin{aligned} &k(k+1)S_{\beta_k}(\boldsymbol{x}_{k+1}) \\ &\quad \leq (k-1)kS_{\beta_{k-1}}(\boldsymbol{x}_k) + 2k\mathbb{E}[\langle \nabla F_{\beta_k}(\boldsymbol{x}_k) - \boldsymbol{d}_k, \boldsymbol{w}_k - \boldsymbol{x}^\star \rangle] + \frac{2k}{k+1}\left(\frac{\| \boldsymbol{H} \| L_f}{n} + \frac{\| \boldsymbol{A} \|}{\beta_k}\right)\mathcal{D}_{\mathcal{X}}^2 \\ &\quad \leq \underbrace{\sum_{i=1}^{k} 2i\mathbb{E}[\langle \nabla F_{\beta_i}(\boldsymbol{x}_i) - \boldsymbol{d}_i, \boldsymbol{w}_i - \boldsymbol{x}^\star \rangle]}_{\text{(A)}} + \underbrace{\sum_{i=1}^{k} \frac{2i}{i+1}\left(\frac{\| \boldsymbol{H} \| L_f}{n} + \frac{\| \boldsymbol{A} \|}{\beta_i}\right)\mathcal{D}_{\mathcal{X}}^2}_{\text{(B)}}. \end{aligned} \tag{A.82}$$

We focus on the term (B), and we use Lemma A.1 to obtain

$$\text{(B)} = 2\mathcal{D}_{\mathcal{X}}^2\left(\frac{\| \boldsymbol{H} \| L_f}{n}\sum_{i=1}^{k}\frac{i}{i+1} + \frac{\| \boldsymbol{A} \|}{\beta_0}\sum_{i=1}^{k}\frac{i}{\sqrt{i+1}}\right) \leq 2\mathcal{D}_{\mathcal{X}}^2\left(\frac{\| \boldsymbol{H} \| L_f}{n}k + \frac{\| \boldsymbol{A} \|}{\beta_0}k\sqrt{k+1}\right). \tag{A.83}$$

We get an upper-bound on the variance term (A) as follows:

$$\begin{aligned} &\mathbb{E}[\langle \nabla F_{\beta_k}(\boldsymbol{x}_k) - \boldsymbol{d}_k, \boldsymbol{w}_k - \boldsymbol{x}^\star \rangle] \\ &\qquad = \mathbb{E}[\langle \boldsymbol{H}^\top(\nabla f(\boldsymbol{H}\boldsymbol{x}_k) - \boldsymbol{p}_k) + \boldsymbol{A}^\top(\nabla g_{\beta_k}(\boldsymbol{A}\boldsymbol{x}_k) - \boldsymbol{q}_k), \boldsymbol{w}_k - \boldsymbol{x}^\star \rangle] \\ &\qquad = \mathbb{E}[\langle \nabla f(\boldsymbol{H}\boldsymbol{x}_k) - \boldsymbol{p}_k, \boldsymbol{H}(\boldsymbol{w}_k - \boldsymbol{x}^\star) \rangle + \langle \nabla g_{\beta_k}(\boldsymbol{A}\boldsymbol{x}_k) - \boldsymbol{q}_k, \boldsymbol{A}(\boldsymbol{w}_k - \boldsymbol{x}^\star) \rangle] \\ &\qquad \leq \mathbb{E}[\left\| \nabla f(\boldsymbol{H}\boldsymbol{x}_k) - \boldsymbol{p}_k \right\|_1 \left\| \boldsymbol{H}(\boldsymbol{w}_k - \boldsymbol{x}^\star) \right\|_\infty + \left\| \nabla g_{\beta_k}(\boldsymbol{A}\boldsymbol{x}_k) - \boldsymbol{q}_k \right\|_1 \left\| \boldsymbol{A}(\boldsymbol{w}_k - \boldsymbol{x}^\star) \right\|_\infty] \\ &\qquad \leq \mathbb{E}[\left\| \nabla f(\boldsymbol{H}\boldsymbol{x}_k) - \boldsymbol{p}_k \right\|_1]\mathcal{D}_\infty(\boldsymbol{H}) + \mathbb{E}[\left\| \nabla g_{\beta_k}(\boldsymbol{A}\boldsymbol{x}_k) - \boldsymbol{q}_k \right\|_1]\mathcal{D}_\infty(\boldsymbol{A}) \end{aligned} \tag{A.84}$$

where the first inequality is the Hölder's inequality, and the second one is based on the bounded-

ness of $\mathcal{X}$.

Then, by Lemma 2.5, we have

$$\begin{aligned}&\mathbb{E}[\|\nabla f(\boldsymbol{H}\boldsymbol{x}_k)-\boldsymbol{p}_k\|_1]\\&\qquad\le\left(1-\tfrac{1}{n}\right)^k\|\nabla f(Xw_1)-\boldsymbol{p}_0\|_1+\frac{2L_f\mathcal{D}_1(\boldsymbol{H})}{n}\left(\left(1-\tfrac{1}{n}\right)^{k/2}\log k+\frac{2(n-1)}{k}\right)\end{aligned}\tag{A.85}$$

And by Lemma 2.6, we have

$$\mathbb{E}[\|\nabla g_{\beta_k}(\boldsymbol{A}\boldsymbol{x}_k)-\boldsymbol{q}_k\|_1]\le\left(1-\tfrac{1}{m}\right)^k\mathbb{E}[\|\nabla g_{\beta_0}(Aw_1)-\boldsymbol{q}_0\|_1]+\frac{10D_2(\boldsymbol{A})\sqrt{m}(m-1)}{\beta_0\sqrt{k}}.\tag{A.86}$$

Finally, we substitute (A.85) and (A.86) back into (A.84) to get

$$\begin{aligned}\text{(A)}&\le 2\mathcal{D}_\infty(\boldsymbol{H})\left[\|\nabla f(\boldsymbol{H}\boldsymbol{x}_1)-\boldsymbol{p}_0\|_1\sum_{i=1}^{k}i\left(1-\tfrac{1}{n}\right)^i+\frac{2L_f\mathcal{D}_1(\boldsymbol{H})}{n}\sum_{i=1}^{k}\left(i\left(1-\tfrac{1}{n}\right)^{i/2}\log i+2(n-1)\right)\right]\\&\quad+2\mathcal{D}_\infty(\boldsymbol{A})\left[\left\|\nabla g_{\beta_0}(\boldsymbol{A}\boldsymbol{x}_1)-\boldsymbol{q}_0\right\|_1\sum_{i=1}^{k}i\left(1-\tfrac{1}{m}\right)^i+\frac{10D_2(\boldsymbol{A})\sqrt{m}(m-1)}{\beta_0}\sum_{i=1}^{k}\sqrt{i}\right]\end{aligned}\tag{A.87}$$

$$\begin{aligned}&\le 2\mathcal{D}_\infty(\boldsymbol{H})\left[\|\nabla f(\boldsymbol{H}\boldsymbol{x}_1)-\boldsymbol{p}_0\|_1\,n^2+\frac{2L_f\mathcal{D}_1(\boldsymbol{H})}{n}\left(16n^3+2(n-1)k\right)\right]\\&\quad+2\mathcal{D}_\infty(\boldsymbol{A})\left[\left\|\nabla g_{\beta_0}(\boldsymbol{A}\boldsymbol{x}_1)-\boldsymbol{q}_0\right\|_1 m^2+\frac{10D_2(\boldsymbol{A})\sqrt{m}(m-1)}{\beta_0}k^{3/2}\right]\end{aligned}\tag{A.88}$$

$$\begin{aligned}&\le 2\mathcal{D}_\infty(\boldsymbol{H})\left[\|\nabla f(\boldsymbol{H}\boldsymbol{x}_1)-\boldsymbol{p}_0\|_1\,n^2+4L_f\mathcal{D}_1(\boldsymbol{H})\left(8n^2+k\right)\right]\\&\quad+2\mathcal{D}_\infty(\boldsymbol{A})\left[\left\|\nabla g_{\beta_0}(\boldsymbol{A}\boldsymbol{x}_1)-\boldsymbol{q}_0\right\|_1 m^2+\frac{10D_2(\boldsymbol{A})\,m^{3/2}}{\beta_0}k^{3/2}\right]\end{aligned}\tag{A.89}$$

where we use Lemma A.1 for the second inequality.

Combining this with the bound on the smoothness term (B) from (A.82) gives the desired result:

$$\begin{aligned}S_{\beta_k}(\boldsymbol{x}_{k+1})&\le\frac{2\mathcal{D}_\infty(\boldsymbol{H})}{k(k+1)}\left\{\|\nabla f(\boldsymbol{H}\boldsymbol{x}_1)-\boldsymbol{p}_0\|_1\,n^2+4L_f\mathcal{D}_1(\boldsymbol{H})\left(8n^2+k\right)\right.\\&\qquad\left.+2\mathcal{D}_\infty(\boldsymbol{A})\left[\left\|\nabla g_{\beta_0}(\boldsymbol{A}\boldsymbol{x}_1)-\boldsymbol{q}_0\right\|_1 m^2+\frac{10D_2(\boldsymbol{A})\,m^{3/2}}{\beta_0}k^{3/2}\right]\right\}\\&\quad+\frac{2\mathcal{D}_{\mathcal{X}}^2}{k(k+1)}\left(\frac{\|\boldsymbol{H}\|L_f}{n}k+\frac{\|\boldsymbol{A}\|}{\beta_0}k\sqrt{k+1}\right)\\&\le\frac{C_3}{k(k+1)}+\frac{C_2}{k+1}+\frac{C_1}{\sqrt{k+1}},\end{aligned}$$

where

$$C_3 = 2n^2\mathcal{D}_\infty(\boldsymbol{H})\big(\|\nabla f(\boldsymbol{H}\boldsymbol{x}_1) - \boldsymbol{p}_0\|_1 + 32L_f\mathcal{D}_1(\boldsymbol{H})\big) + 2m^2\mathcal{D}_\infty(\boldsymbol{A})\|\nabla g_{\beta_0}(\boldsymbol{A}\boldsymbol{x}_1) - \boldsymbol{q}_0\|_1$$

$$C_2 = 8L_f\mathcal{D}_1(\boldsymbol{H})\mathcal{D}_\infty(\boldsymbol{H}) + 2n^{-1}L_f\|\boldsymbol{H}\|\mathcal{D}_{\mathcal{X}}^2$$

$$C_1 = \beta_0^{-1}(2\mathcal{D}_{\mathcal{X}}^2\|\boldsymbol{A}\| + 10\mathcal{D}_1(\boldsymbol{A})).$$ □

A.3.4 Proof of Corollary 2.3

Corollary 2.3. *Suppose $g : \mathbb{R}^m \to \mathbb{R}$ is L_g-Lipschitz continuous. Then, the estimates generated by H-SAG-CGM (Algorithm 2.3) satisfy*

$$\mathbb{E}[F(\boldsymbol{x}_{k+1}) - F^\star] \leq \frac{C_1}{\sqrt{k}} + \frac{C_2}{k} + \frac{C_3}{k^2} + \frac{\beta_0 L_g^2}{2\sqrt{k}}$$

where the constants C_1, C_2 and C_2 are defined in Theorem 2.3.

Proof. Suppose g is L_g-Lipschitz continuous. Then, from (A.6) we get

$$\mathbb{E}F(\boldsymbol{x}_{k+1}) - F^\star = \mathbb{E}[f(\boldsymbol{H}\boldsymbol{x}_{k+1}) + g(\boldsymbol{A}\boldsymbol{x}_{k+1})] - F^\star \tag{A.90}$$

$$\leq \mathbb{E}[f(\boldsymbol{H}\boldsymbol{x}_{k+1}) + g_{\beta_k}(\boldsymbol{A}\boldsymbol{x}_{k+1})] - F^\star + \frac{\beta_k L_g^2}{2} \tag{A.91}$$

$$= S_{\beta_k}(\boldsymbol{x}_{k+1}) + \frac{\beta_0 L_g^2}{2\sqrt{k+1}}. \qquad \square$$

A.3.5 Proof of Corollary 2.4

Corollary 2.4. *Suppose g is the indicator function of a closed and convex set $\mathcal{K} \in \mathbb{R}^m$, $\mathcal{K} \coloneqq \mathcal{K}_1 \times \ldots \times \mathcal{K}_m$, $\mathcal{K}_i \subseteq \mathbb{R}$, $\forall i \in [m]$. Then, for H-SAG-CGM (Algorithm 2.3), we have a lower bound on the suboptimality as $\mathbb{E}\left[f(\boldsymbol{H}\boldsymbol{x}_{k+1}) - f(\boldsymbol{H}\boldsymbol{x}^\star)\right] \geq -\|\boldsymbol{\lambda}^\star\|\mathbb{E}[\operatorname{dist}(\boldsymbol{A}\boldsymbol{x}_{k+1}, \mathcal{K})]$, where $\boldsymbol{\lambda}^\star$ is a solution of the dual problem, and the following upper bounds on the suboptimality and feasibility:*

$$\mathbb{E}\left[f(\boldsymbol{H}\boldsymbol{x}_{k+1}) - f(\boldsymbol{H}\boldsymbol{x}^\star)\right] \leq \frac{C_1 + \beta_0}{\sqrt{k}} + \frac{C_2}{k} + \frac{C_3}{k^2}, \text{ and}$$
$$\mathbb{E}[\operatorname{dist}(\boldsymbol{A}\boldsymbol{x}_{k+1}, \mathcal{K})] \leq \frac{C_4}{\sqrt{k}} + \frac{\sqrt{2C_2}}{k^{3/4}} + \frac{\sqrt{2C_3}}{k^{5/4}},$$

where the constants C_1, C_2 and C_3 are defined in Theorem 2.3 and $C_4 = \left(\frac{3\beta_0\|\boldsymbol{\lambda}^\star\|}{2} + \sqrt{2C_1}\right)$.

Proof. Suppose $g(\boldsymbol{z}) = \iota_{\mathcal{K}}(\boldsymbol{z})$, the indicator function of a closed and convex set. We can write the Lagrangian as

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{r}, \boldsymbol{\lambda}) \coloneqq f(\boldsymbol{H}\boldsymbol{x}) + \langle \boldsymbol{A}\boldsymbol{x} - \boldsymbol{r}, \boldsymbol{\lambda}\rangle, \qquad \boldsymbol{x} \in \mathcal{X},\ \boldsymbol{r} \in \mathcal{K}. \tag{A.92}$$

From the Lagrange saddle point theory, we have

$$f(\boldsymbol{H}\boldsymbol{x}^\star) \leq \mathcal{L}(\boldsymbol{x}, \boldsymbol{r}, \boldsymbol{\lambda}^\star) \leq f(\boldsymbol{H}\boldsymbol{x}) + \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{r}\|\|\boldsymbol{\lambda}^\star\|, \qquad \forall \boldsymbol{x} \in \mathcal{X} \text{ and } \forall \boldsymbol{r} \in \mathcal{K}. \tag{A.93}$$

Letting $\boldsymbol{x} = \boldsymbol{x}_{k+1} \in \mathcal{X}$ and $\boldsymbol{r} = \operatorname{proj}_{\mathcal{K}}(\boldsymbol{A}\boldsymbol{x}_{k+1}) \in \mathcal{K}$, taking expectation on both sides and rearranging, we get

$$\mathbb{E}\left[f(\boldsymbol{H}\boldsymbol{x}_{k+1}) - f(\boldsymbol{H}\boldsymbol{x}^\star)\right] \geq -\|\boldsymbol{\lambda}^\star\|\,\mathbb{E}[\operatorname{dist}(\boldsymbol{A}\boldsymbol{x}_{k+1}, \mathcal{K})] \tag{A.94}$$

This is the desired lower bound on the objective residual.

Next, we derive an upper bound on the objective residual. By definition of g_β (see (A.1)) for $\iota_{\mathcal{K}}$,

$$g_\beta(\boldsymbol{A}\boldsymbol{x}) = \frac{1}{2\beta}\operatorname{dist}(\boldsymbol{A}\boldsymbol{x}, \mathcal{K})^2. \tag{A.95}$$

Note that $f(\boldsymbol{H}\boldsymbol{x}^\star) = F(\boldsymbol{x}^\star)$ since $g(\boldsymbol{A}\boldsymbol{x}^\star) = 0$. Then,

$$\mathbb{E}[f(\boldsymbol{H}\boldsymbol{x}_{k+1}) - f(\boldsymbol{H}\boldsymbol{x}^\star)] = \mathbb{E}[F_{\beta_k}(\boldsymbol{x}_{k+1}) - F^\star - g_{\beta_k}(\boldsymbol{A}\boldsymbol{x}_{k+1})] \tag{A.96}$$
$$\leq S_{\beta_k}(\boldsymbol{x}_{k+1}) - \frac{1}{2\beta_k}\mathbb{E}[\operatorname{dist}(\boldsymbol{A}\boldsymbol{x}_{k+1}, \mathcal{K})^2] \tag{A.97}$$
$$\leq S_{\beta_k}(\boldsymbol{x}_{k+1}). \tag{A.98}$$

Finally, we derive the convergence rate of the infeasibility error. To this end, we combine (A.94) and (A.97):

$$-\|\boldsymbol{\lambda}^\star\|\,\mathbb{E}[\operatorname{dist}(\boldsymbol{A}\boldsymbol{x}_{k+1}, \mathcal{K})] \leq S_{\beta_k}(\boldsymbol{x}_{k+1}) - \frac{1}{2\beta_k}\mathbb{E}[\operatorname{dist}(\boldsymbol{A}\boldsymbol{x}_{k+1}, \mathcal{K})^2] \tag{A.99}$$

We rearrange and apply Jensen's inequality to $\mathbb{E}[\text{dist}(\boldsymbol{A}\boldsymbol{x}_{k+1}, \mathcal{K})^2]$, and we get a second order inequality with respect to $\mathbb{E}[\text{dist}(\boldsymbol{A}\boldsymbol{x}_{k+1}, \mathcal{K})]$:

$$\frac{1}{2\beta_k} \underbrace{\mathbb{E}[\text{dist}(\boldsymbol{A}\boldsymbol{x}_{k+1}, \mathcal{K})]^2}_{t^2} - \|\boldsymbol{\lambda}^\star\| \underbrace{\mathbb{E}[\text{dist}(\boldsymbol{A}\boldsymbol{x}_{k+1}, \mathcal{K})]}_{t} - S_{\beta_k}(\boldsymbol{x}_{k+1}) \leq 0. \tag{A.100}$$

By solving this inequality for t, we achieve the desired bound:

$$\mathbb{E}[\text{dist}(\boldsymbol{A}\boldsymbol{x}_{k+1}, \mathcal{K})] \leq \beta_k \left(\|\boldsymbol{\lambda}^\star\| + \sqrt{\|\boldsymbol{\lambda}^\star\|^2 + \frac{2S_{\beta_k}(\boldsymbol{x}_{k+1})}{\beta_k}} \right) \leq 2\beta_k \|\boldsymbol{\lambda}^\star\| + \sqrt{2\beta_k S_{\beta_k}(\boldsymbol{x}_{k+1})}, \tag{A.101}$$

where we used $\sqrt{a^2 + b^2} \leq a + b$ for $a, b \geq 0$ in the last inequality to simplify the terms. □

B Appendix for Chapter 3

B.1 Proofs

Assumption 3.1.a implies global progress bounds on our fully composite objective with an inner linearization of $\boldsymbol{f}$, as stated in the following Lemma B.1. This lemma provides a basis for all our convergence results.

Lemma B.1. *Let $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{X}$ and $\gamma \in [0,1]$. Denote $\boldsymbol{y}_\gamma = \boldsymbol{x} + \gamma(\boldsymbol{y} - \boldsymbol{x})$. Then, it holds*

$$\varphi(\boldsymbol{y}_\gamma) \quad \leq \quad F\big(\boldsymbol{f}(x) + \nabla \boldsymbol{f}(\boldsymbol{x})(\boldsymbol{y}_\gamma - \boldsymbol{x}),\, \boldsymbol{y}_\gamma\big) \,+\, \tfrac{\gamma^2}{2}\mathcal{S}. \tag{B.1}$$

Proof. Note that the subhomogenity assumption (3.6) is equivalent to the following useful inequality for the outer component of the objective see (Theorem 7.1 in [73]):

$$F(\boldsymbol{u} + t\boldsymbol{v}, \boldsymbol{x}) \quad \leq \quad F(\boldsymbol{u}, \boldsymbol{x}) + tF(\boldsymbol{v}, \boldsymbol{x}), \qquad \forall \boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^n,\, \boldsymbol{x} \in X,\, t \geq 0. \tag{B.2}$$

Then, we have

$$\begin{aligned}
&\varphi(\boldsymbol{y}_\gamma) \;\equiv\; F(\boldsymbol{f}(\boldsymbol{y}_\gamma), \boldsymbol{y}_\gamma) \\
&= F\big(\boldsymbol{f}(\boldsymbol{x}) + \nabla \boldsymbol{f}(\boldsymbol{x})(\boldsymbol{y}_\gamma - \boldsymbol{x}) + \tfrac{\gamma^2}{2} \cdot \tfrac{2}{\gamma^2}\big[\boldsymbol{f}(\boldsymbol{y}_\gamma) - \boldsymbol{f}(\boldsymbol{x}) - \nabla f(\boldsymbol{x})(\boldsymbol{y}_\gamma - \boldsymbol{x})\big],\, \boldsymbol{y}_\gamma\big) \\
&\overset{(B.2)}{\leq} F\big(\boldsymbol{f}(\boldsymbol{x}) + \nabla \boldsymbol{f}(\boldsymbol{x})(\boldsymbol{y}_\gamma - \boldsymbol{x}),\, \boldsymbol{y}_\gamma\big) + \tfrac{\gamma^2}{2} F\big(\tfrac{2}{\gamma^2}\big[\boldsymbol{f}(\boldsymbol{y}_\gamma) - \boldsymbol{f}(\boldsymbol{x}) - \nabla f(\boldsymbol{x})(\boldsymbol{y}_\gamma - \boldsymbol{x})\big],\, \boldsymbol{y}_\gamma\big) \\
&\leq F\big(\boldsymbol{f}(\boldsymbol{x}) + \nabla \boldsymbol{f}(\boldsymbol{x})(\boldsymbol{y}_\gamma - \boldsymbol{x}),\, \boldsymbol{y}_\gamma\big) + \tfrac{\gamma^2}{2}\mathcal{S},
\end{aligned}$$

which is the desired bound. □

B.1.1 Proof of Theorem 3.1

Theorem 3.1. *Let Assumptions 3.1, 3.1.a, and 3.2 be satisfied. Let* $\gamma_k \coloneqq \min\left\{1, \frac{\mathcal{G}_k}{\mathcal{S}}\right\}$ ***or*** $\gamma_k \coloneqq \frac{2}{2+k}$. *Then, for* $k \geq 1$ *it holds that*

$$\varphi(\boldsymbol{y}_k) - \varphi^\star \;\leq\; \frac{2\mathcal{S}}{1+k} \qquad \textit{and} \qquad \min_{1\leq i\leq k} \mathcal{G}_i \;\leq\; \frac{6\mathcal{S}}{k}. \tag{3.16}$$

Proof. Indeed, for one iteration of the method, we have

$$\begin{aligned}
\varphi(\boldsymbol{y}_{k+1}) &\overset{(B.1)}{\leq} F\big(\boldsymbol{f}(\boldsymbol{y}_k) + \nabla \boldsymbol{f}(\boldsymbol{y}_k)(\boldsymbol{y}_{k+1} - \boldsymbol{y}_k), \boldsymbol{y}_{k+1}\big) + \tfrac{\gamma_k^2}{2}\mathcal{S} \\
&= F\big((1-\gamma_k)\boldsymbol{f}(\boldsymbol{y}_k) + \gamma_k(\boldsymbol{f}(\boldsymbol{y}_k) + \nabla \boldsymbol{f}(\boldsymbol{y}_k)(\boldsymbol{x}_{k+1} - \boldsymbol{y}_k)), \\
&\qquad (1-\gamma_k)\boldsymbol{y}_k + \gamma_k \boldsymbol{x}_{k+1}\big) + \tfrac{\gamma_k^2}{2}\mathcal{S} \\
&\overset{(*)}{\leq} \varphi(\boldsymbol{y}_k) + \gamma_k\Big[F\big(\boldsymbol{f}(\boldsymbol{y}_k) + \nabla \boldsymbol{f}(\boldsymbol{y}_k)(\boldsymbol{x}_{k+1} - \boldsymbol{y}_k), \boldsymbol{x}_{k+1}\big) - \varphi(\boldsymbol{y}_k)\Big] + \tfrac{\gamma_k^2}{2}\mathcal{S} \\
&\equiv \varphi(\boldsymbol{y}_k) - \gamma_k \mathcal{G}_k + \tfrac{\gamma_k^2}{2}\mathcal{S},
\end{aligned}$$

where we used in $(*)$ that $F(\cdot,\cdot)$ is jointly convex. Hence, we obtain the following inequality for the progress of one step, for $k \geq 0$:

$$\varphi(\boldsymbol{y}_k) - \varphi(\boldsymbol{y}_{k+1}) \;\geq\; \gamma_k \mathcal{G}_k - \tfrac{\gamma_k^2}{2}\mathcal{S}. \tag{B.3}$$

Now, let us choose use an auxiliary sequence $A_k \coloneqq k \cdot (k+1)$ and $a_{k+1} \coloneqq A_{k+1} - A_k = 2(k+1)$. Then,

$$\tfrac{a_{k+1}}{A_{k+1}} \;=\; \tfrac{2}{2+k},$$

which is one of the possible choices for γ_k. Note that for the other choice, we set $\gamma_k = \min\{1, \frac{\mathcal{G}_k}{\mathcal{S}}\}$, which maximizes the right hand side of (B.3) with respect to $\gamma_k \in [0,1]$. Hence, in both cases we have that

$$\varphi(\boldsymbol{y}_k) - \varphi(\boldsymbol{y}_{k+1}) \;\geq\; \tfrac{a_{k+1}}{A_{k+1}}\mathcal{G}_k - \tfrac{a_{k+1}^2}{2A_{k+1}^2}\mathcal{S}, \tag{B.4}$$

or, rearranging the terms,

$$A_{k+1}\big[\varphi(\boldsymbol{y}_{k+1}) - \varphi^\star\big] \overset{(B.4)}{\leq} A_{k+1}\big[\varphi(\boldsymbol{y}_k) - \varphi^\star\big] - a_{k+1}\mathcal{G}_k + \tfrac{a_{k+1}^2}{2A_{k+1}}\mathcal{S}$$

$$\overset{(3.15)}{\leq} A_k\big[\varphi(\boldsymbol{y}_k) - \varphi^\star\big] + \tfrac{a_{k+1}^2}{2A_{k+1}}\mathcal{S}.$$

Telescoping this bound for the first $k \geq 1$ iterations, we get

$$\varphi(\boldsymbol{y}_k) - \varphi^\star \;\leq\; \tfrac{\mathcal{S}}{2A_k} \cdot \sum_{i=1}^{k} \tfrac{a_i^2}{A_i} \;=\; \tfrac{2\mathcal{S}}{k(k+1)} \cdot \sum_{i=1}^{k} \tfrac{i}{i+1} \;\leq\; \tfrac{2\mathcal{S}}{k+1}. \tag{B.5}$$

It remains to prove the convergence in terms of the accuracy measure $\mathcal{G}_k$. For that, we telescope the bound (B.4), which is

$$a_{k+1}\mathcal{G}_k \;\leq\; a_{k+1}\varphi(\boldsymbol{y}_k) + A_k\varphi(\boldsymbol{y}_k) - A_{k+1}\varphi(\boldsymbol{y}_{k+1}) + \tfrac{a_{k+1}^2}{A_{k+1}}\tfrac{\mathcal{S}}{2}, \tag{B.6}$$

for the $k \geq 1$ iterations, and use the convergence for the functional residual (B.5):

$$\sum_{i=1}^{k} a_{i+1} \cdot \min_{1\leq i\leq k} \mathcal{G}_i \;\leq\; \sum_{i=1}^{k} a_{i+1}\mathcal{G}_i$$

$$\overset{(B.6)}{\leq} a_1\big[\varphi(\boldsymbol{y}_1) - \varphi^\star\big] + \sum_{i=1}^{k} a_{i+1}\big[\varphi(\boldsymbol{y}_i) - \varphi^\star\big] + \tfrac{\mathcal{S}}{2}\sum_{i=1}^{k}\tfrac{a_{i+1}^2}{A_{i+1}}$$

$$\overset{(B.5)}{\leq} 2\mathcal{S}\cdot\Big(1 + \sum_{i=1}^{k}\tfrac{a_{i+1}}{i+1} + \sum_{i=1}^{k}\tfrac{i}{i+1}\Big) \;\leq\; 2\mathcal{S}\cdot(1+3k).$$

To finish the proof, we need to divide both sides by $\sum_{i=1}^{k} a_{i+1} = A_{k+1} - a_1 = k(3+k)$. □

B.1.2 Proof of Theorem 3.2

Theorem 3.2. *Let Assumptions 3.1 and 3.1.a be satisfied. Let* $\gamma_k \coloneqq \min\left\{1, \frac{\mathcal{G}_k}{\mathcal{S}}\right\}$ ***or*** $\gamma_k \coloneqq \frac{1}{\sqrt{1+k}}$. *Then, for all* $k \geq 1$ *it holds that*

$$\min_{0\leq i\leq k} \mathcal{G}_i \quad \leq \quad \frac{\varphi(\boldsymbol{y}_0)-\varphi^\star+0.5\mathcal{S}(1+\ln(k+1))}{\sqrt{k+1}}. \tag{3.17}$$

Proof. As in the proof of the previous theorem, our main inequality (B.3) on the progress of one step is:

$$\varphi(\boldsymbol{y}_k) - \varphi(\boldsymbol{y}_{k+1}) \quad \geq \quad \gamma_k \mathcal{G}_k - \tfrac{\gamma_k^2}{2}\mathcal{S},$$

where we can substitute $\gamma_k = \frac{1}{\sqrt{k+1}}$ for both strategies of choosing this parameter.

Summing up this bound for the first $k+1$ iterations, we obtain

$$\sum_{i=0}^{k} \gamma_i \mathcal{G}_i \quad \leq \quad \varphi(\boldsymbol{y}_0) - \varphi(\boldsymbol{y}_{k+1}) + \tfrac{\mathcal{S}}{2}\sum_{i=0}^{k}\gamma_i^2. \tag{B.7}$$

Using the bound $\varphi(\boldsymbol{y}_{k+1}) \geq \varphi^\star$ and our value of γ_i, we get

$$\min_{0\leq i\leq k} \mathcal{G}_i \cdot \sqrt{k+1} \quad \leq \quad \sum_{i=0}^{k} \tfrac{\mathcal{G}_i}{\sqrt{1+i}} \overset{\text{(B.7)}}{\leq} \varphi(\boldsymbol{y}_0) - \varphi^\star + \tfrac{\mathcal{S}}{2}\sum_{i=0}^{k}\tfrac{1}{1+i}$$

$$\leq \quad \varphi(\boldsymbol{y}_0) - \varphi^\star + \tfrac{\mathcal{S}}{2}(1+\ln(k+1)),$$

which is (3.17). □

B.1.3 Proof of Theorem 3.3

Theorem 3.3. *Let Assumptions 3.1, 3.1.b, and 3.2 be satisfied. We choose* $\gamma_k \coloneqq \frac{3}{k+3}$, $\beta_k \coloneqq cF(\boldsymbol{L})\gamma_k$ *and* $\eta_k \coloneqq \frac{\delta}{3(k+1)(k+2)}$ *where* $\delta > 0$ *and* $c \geq 0$ *are chosen constants, and* $F(\boldsymbol{L}) \coloneqq \sup_{\boldsymbol{x}\in\mathcal{X}} F(\boldsymbol{L}, \boldsymbol{x})$. *Then, for all* $k \geq 1$ *it holds that*

$$\varphi(\boldsymbol{y}_k) - \varphi^\star \quad \leq \quad \frac{\delta + 8cF(\boldsymbol{L})\mathcal{D}_{\mathcal{X}}^2}{(k+2)(k+3)} + \frac{2\max\{0, 1-c\}F(\boldsymbol{L})\mathcal{D}_{\mathcal{X}}^2}{k+3}.$$

Proof. Let us consider one iteration of the method for some $k \geq 0$.

Since all the components of $\boldsymbol{f}$ have the Lipschitz continuous gradients, it holds that

$$\boldsymbol{f}(\boldsymbol{y}_{k+1}) \quad \leq \quad \boldsymbol{f}(\boldsymbol{z}_{k+1}) + \nabla\boldsymbol{f}(\boldsymbol{z}_{k+1})(\boldsymbol{y}_{k+1} - \boldsymbol{z}_{k+1}) + \tfrac{\boldsymbol{L}}{2}\|\boldsymbol{y}_{k+1} - \boldsymbol{z}_{k+1}\|^2,$$

where the vector inequality is component-wise. Then, using the properties of F, we have

$$\begin{aligned}
\varphi(\boldsymbol{y}_{k+1}) \quad &= \quad F(\boldsymbol{f}(\boldsymbol{y}_{k+1}), \boldsymbol{y}_{k+1}) \\
&\overset{(3.8),(B.2)}{\leq} \quad F(\boldsymbol{f}(\boldsymbol{z}_{k+1}) + \nabla\boldsymbol{f}(\boldsymbol{z}_{k+1})(\boldsymbol{y}_{k+1} - \boldsymbol{z}_{k+1}), \boldsymbol{y}_{k+1}) + \tfrac{F(\boldsymbol{L})}{2}\|\boldsymbol{y}_{k+1} - \boldsymbol{z}_{k+1}\|^2 \\
&= \quad F\big((1-\gamma_k)\big[\boldsymbol{f}(\boldsymbol{z}_{k+1}) + \nabla\boldsymbol{f}(\boldsymbol{z}_{k+1})(\boldsymbol{y}_k - \boldsymbol{z}_{k+1})\big] \\
&\qquad + \gamma_k\big[\boldsymbol{f}(\boldsymbol{z}_{k+1}) + \nabla\boldsymbol{f}(\boldsymbol{z}_{k+1})(\boldsymbol{x}_{k+1} - \boldsymbol{z}_{k+1})\big], \\
&\qquad (1-\gamma_k)\boldsymbol{y}_k + \gamma_k\boldsymbol{x}_{k+1}\big) + \tfrac{\gamma_k^2 F(\boldsymbol{L})}{2}\|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k\|^2 \\
&\leq \quad (1-\gamma_k)F(\boldsymbol{f}(\boldsymbol{z}_{k+1}) + \nabla\boldsymbol{f}(\boldsymbol{z}_{k+1})(\boldsymbol{y}_k - \boldsymbol{z}_{k+1}), \boldsymbol{y}_k) \\
&\qquad + \gamma_k F(\boldsymbol{f}(\boldsymbol{z}_{k+1}) + \nabla\boldsymbol{f}(\boldsymbol{z}_{k+1})(\boldsymbol{x}_{k+1} - \boldsymbol{z}_{k+1}), \boldsymbol{x}_{k+1}) + \tfrac{\gamma_k^2 F(\boldsymbol{L})}{2}\|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k\|^2 \\
&\leq \quad (1-\gamma_k)\varphi(\boldsymbol{y}_k) + \gamma_k F(\boldsymbol{f}(\boldsymbol{z}_{k+1}) + \nabla\boldsymbol{f}(\boldsymbol{z}_{k+1})(\boldsymbol{x}_{k+1} - \boldsymbol{z}_{k+1}), \boldsymbol{x}_{k+1}) \\
&\qquad + \tfrac{\gamma_k^2 F(\boldsymbol{L})}{2}\|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k\|^2,
\end{aligned}$$

where the second equality comes from the update rule of $\boldsymbol{y}_{k+1}$, the second inequality comes from the joint convexity in Assumption 3.1, the third inequality comes from convexity of the components of $\boldsymbol{f}$ and monotonicity of F.

Since we are introducing a norm-regularized minimization subproblem for the purpose of acceleration, the term $\frac{\gamma_k^2 F(\boldsymbol{L})}{2}\|\boldsymbol{x}_{k+1}-\boldsymbol{x}_k\|^2$ can be further upper bounded using η_k-approximate guarantee (3.20), as follows, for any $\boldsymbol{x}\in\mathcal{X}$:

$$\begin{aligned}
\tfrac{\gamma_k^2 F(\boldsymbol{L})}{2}\|\boldsymbol{x}_{k+1}-\boldsymbol{x}_k\|^2 &= \left(\tfrac{\gamma_k^2 F(\boldsymbol{L})}{2}-\tfrac{\beta_k\gamma_k}{2}\right)\|\boldsymbol{x}_{k+1}-\boldsymbol{x}_k\|^2+\tfrac{\beta_k\gamma_k}{2}\|\boldsymbol{x}_{k+1}-\boldsymbol{x}_k\|^2\\
&\le \tfrac{\beta_k\gamma_k}{2}\|\boldsymbol{x}_{k+1}-\boldsymbol{x}_k\|_2^2+\tfrac{\gamma_k^2 F(\boldsymbol{L})(1-c)}{2}\|\boldsymbol{x}_{k+1}-\boldsymbol{x}_k\|^2\\
&= \tfrac{\beta_k\gamma_k}{2}\left(\|\boldsymbol{x}-\boldsymbol{x}_k\|_2^2-\|\boldsymbol{x}-\boldsymbol{x}_{k+1}\|^2-2\langle\boldsymbol{x}_k-\boldsymbol{x}_{k+1},\boldsymbol{x}_{k+1}-\boldsymbol{x}\rangle\right)\\
&\quad+\tfrac{\gamma_k^2 F(\boldsymbol{L})(1-c)}{2}\|\boldsymbol{x}_{k+1}-\boldsymbol{x}_k\|^2\\
&\overset{(3.20)}{\le} \tfrac{\beta_k\gamma_k}{2}\left(\|\boldsymbol{x}-\boldsymbol{x}_k\|_2^2-\|\boldsymbol{x}-\boldsymbol{x}_{k+1}\|^2\right)+\tfrac{\gamma_k^2 F(\boldsymbol{L})(1-c)}{2}\|\boldsymbol{x}_{k+1}-\boldsymbol{x}_k\|^2\\
&\quad+\gamma_k F(\boldsymbol{f}(\boldsymbol{z}_{k+1})+\nabla\boldsymbol{f}(\boldsymbol{z}_{k+1})(\boldsymbol{x}-\boldsymbol{z}_{k+1}),\boldsymbol{x})\\
&\quad-\gamma_k F(\boldsymbol{f}(\boldsymbol{z}_{k+1})+\nabla\boldsymbol{f}(\boldsymbol{z}_{k+1})(\boldsymbol{x}_{k+1}-\boldsymbol{z}_{k+1}),\boldsymbol{x}_{k+1})+\gamma_k\eta_k,
\end{aligned}$$

where we used our choice $\beta_k = cF(\boldsymbol{L})\gamma_k$.

Therefore, by combining these two bounds together, we obtain

$$\begin{aligned}
\varphi(\boldsymbol{y}_{k+1}) &\le (1-\gamma_k)\varphi(\boldsymbol{y}_k)+\gamma_k F(\boldsymbol{f}(\boldsymbol{z}_{k+1})+\nabla\boldsymbol{f}(\boldsymbol{z}_{k+1})(\boldsymbol{x}-\boldsymbol{z}_{k+1}),\boldsymbol{x})\\
&\quad+\tfrac{\beta_k\gamma_k}{2}\left(\|\boldsymbol{x}-\boldsymbol{x}_k\|^2-\|\boldsymbol{x}-\boldsymbol{x}_{k+1}\|^2\right)+\tfrac{\gamma_k^2 F(\boldsymbol{L})(1-c)}{2}\|\boldsymbol{x}_{k+1}-\boldsymbol{x}_k\|^2+\gamma_k\eta_k\\
&\le (1-\gamma_k)\varphi(\boldsymbol{y}_k)+\gamma_k\varphi(\boldsymbol{x})+\tfrac{\beta_k\gamma_k}{2}\left(\|\boldsymbol{x}-\boldsymbol{x}_k\|^2-\|\boldsymbol{x}-\boldsymbol{x}_{k+1}\|^2\right)\\
&\quad+\tfrac{\gamma_k^2 F(\boldsymbol{L})(1-c)}{2}\|\boldsymbol{x}_{k+1}-\boldsymbol{x}_k\|^2+\gamma_k\eta_k,
\end{aligned}$$

for all $\boldsymbol{x}\in\mathcal{X}$, where we used convexity of $\boldsymbol{f}$ and monotonicity of F.

We now subtract $\varphi(\boldsymbol{x})$ from both sides, let $\boldsymbol{x} = \boldsymbol{x}^\star$ and denote $\varepsilon_k := \varphi(\boldsymbol{y}_k) - \varphi^\star$, which gives

$$\begin{aligned} \varepsilon_{k+1} \quad \leq \quad & (1-\gamma_k)\varepsilon_k + \tfrac{\gamma_k \beta_k}{2}\left(\|\boldsymbol{x}_k - \boldsymbol{x}^\star\|^2 - \|\boldsymbol{x}_{k+1} - \boldsymbol{x}^\star\|^2\right) \\ & + \tfrac{\gamma_k^2 F(\boldsymbol{L})(1-c)}{2}\|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k\|^2 + \gamma_k \eta_k. \end{aligned} \tag{B.8}$$

We now move on to choosing the parameters γ_k, η_k and β_k in a way that allows us to accelerate. For more flexibility, we let $\gamma_k := \frac{a_{k+1}}{A_{k+1}}$, for some sequences $\{a_k\}_{k\geq 0}$ and $\{A_k\}_{k\geq 0}$ that will be defined later. Then (B.8) becomes:

$$\begin{aligned} A_{k+1}\varepsilon_{k+1} \quad \leq \quad & A_0\epsilon_0 + \sum_{i=0}^{k} a_{i+1}\eta_i + \tfrac{1}{2}\sum_{i=0}^{k} a_{i+1}\beta_i\left(\|\boldsymbol{x}_i - \boldsymbol{x}^\star\|^2 - \|\boldsymbol{x}_{i+1} - \boldsymbol{x}^\star\|^2\right) \\ & + \tfrac{F(\boldsymbol{L})(1-c)}{2}\sum_{i=0}^{k}\tfrac{a_{i+1}^2}{A_{i+1}}\|\boldsymbol{x}_{i+1} - \boldsymbol{x}_i\|^2 \\ \leq \quad & A_0\epsilon_0 + \sum_{i=0}^{k} a_{i+1}\eta_i + \tfrac{a_1\beta_0}{2}\|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|^2 + \tfrac{1}{2}\textstyle\sum_{i=1}^{k}\left(a_{i+1}\beta_i - a_i\beta_{i-1}\right)\|\boldsymbol{x}_i - \boldsymbol{x}^\star\|^2 \\ & + \tfrac{F(\boldsymbol{L})(1-c)}{2}\sum_{i=0}^{k}\tfrac{a_{i+1}^2}{A_{i+1}}\|\boldsymbol{x}_{i+1} - \boldsymbol{x}_i\|^2 \end{aligned}$$

and therefore, we have

$$\begin{aligned} \varepsilon_{k+1} \quad \leq \quad & \tfrac{A_0\epsilon_0}{A_{k+1}} + \tfrac{1}{A_{k+1}}\sum_{i=0}^{k} a_{i+1}\eta_i + \tfrac{a_1\beta_0}{2A_{k+1}}\|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|^2 \\ & + \tfrac{1}{2A_{k+1}}\sum_{i=1}^{k}\left(a_{i+1}\beta_i - a_i\beta_{i-1}\right)\|\boldsymbol{x}_i - \boldsymbol{x}^\star\|^2 + \tfrac{F(\boldsymbol{L})(1-c)}{2A_{k+1}}\sum_{i=0}^{k}\tfrac{a_{i+1}^2}{A_{i+1}}\|\boldsymbol{x}_{i+1} - \boldsymbol{x}_i\|^2. \end{aligned}$$

We wish to choose sequences A_k, a_k, β_k and η_k such that we obtain a $\mathcal{O}\left(1/k^2\right)$ rate of convergence on the functional residual of $\varphi(\cdot)$. The constraint we require on the sequences is $\gamma_k F(\boldsymbol{L}) \leq \beta_k$. The following choices

$$\eta_k \quad = \quad \tfrac{\delta}{a_{k+1}}, \text{ for some constant } \delta > 0,$$

$$\beta_k \quad = \quad cF(\boldsymbol{L})\gamma_k, \text{ for some constant } c > 0$$

$$a_{k+1} \quad = \quad A_{k+1} - A_k \;=\; \tfrac{3A_{k+1}}{k+3}, \quad A_{k+1} \;=\; (k+1)(k+2)(k+3),$$

give us the desired outcome, since equation (B.9) becomes:

$$\varepsilon_{k+1} \leq \frac{\delta}{(k+2)(k+3)} + \frac{3cF(\boldsymbol{L}) \| \boldsymbol{x}_0 - \boldsymbol{x}^\star \|^2}{(k+1)(k+2)(k+3)} + \frac{5ckF(\boldsymbol{L})\mathcal{D}_{\mathcal{X}}^2}{(k+1)(k+2)(k+3)}$$

$$+ \frac{9F(\boldsymbol{L})(1-c)}{2(k+1)(k+2)(k+3)} \sum_{i=0}^{k} (i+1) \| \boldsymbol{x}_{i+1} - \boldsymbol{x}_i \|^2$$

$$\leq \frac{\delta}{(k+2)(k+3)} + \frac{8cF(\boldsymbol{L})\mathcal{D}_{\mathcal{X}}^2}{(k+2)(k+3)} + \frac{2\max\{0, 1-c\}F(\boldsymbol{L})\mathcal{D}_{\mathcal{X}}^2}{k+3}$$

since $a_{i+1}\beta_i - a_i\beta_{i-1} = \frac{9cF(\boldsymbol{L})\left(i^2+5i+4\right)}{i^2+5i+6} < 9cF(\boldsymbol{L})$ and $\| \boldsymbol{x}_i - \boldsymbol{x}^\star \|^2 \leq \mathcal{D}_{\mathcal{X}}^2$. □

B.1.4 Proof of Theorem 3.4

Theorem 3.4. *Let Assumptions 3.1, 3.1.b, and 3.2 be satisfied. Then, for all $t \geq 1$ it holds that*

$$P(\boldsymbol{u}_t) - P^\star \;\leq\; \tfrac{2\beta\mathcal{D}_{\mathcal{X}}^2}{t+1} \qquad \textit{and} \qquad \min_{1\leq i\leq t} \mathcal{G}_t \;\leq\; \tfrac{6\beta\mathcal{D}_{\mathcal{X}}^2}{t}.$$

Consequently, Algorithm 3.3 returns an η-approximate solution according to condition (3.20) *after at most $\mathcal{O}\left(\frac{\beta\mathcal{D}_{\mathcal{X}}^2}{\eta}\right)$ iterations.*

Proof. Let us introduce our subproblem in a general form, that is

$$s^\star \;=\; \min_{\boldsymbol{u}\in\mathcal{X}}\Big\{ s(\boldsymbol{u}) \;\coloneqq\; r(\boldsymbol{u}) + h(\boldsymbol{u}), \Big\} \tag{B.9}$$

where $r(\cdot)$ is a differentiable convex function, whose gradient is Lipschitz continuous with constant $\beta > 0$, and $h(\boldsymbol{u})$ is a general proper closed convex function, not necessarily differentiable.

In our case, for computing the inexact proximal step (3.19), we set

$$r(\boldsymbol{u}) \;\coloneqq\; \tfrac{\beta}{2}\|\boldsymbol{u}-\boldsymbol{x}\|^2,$$

$$h(\boldsymbol{u}) \;\coloneqq\; F(\boldsymbol{f}(\boldsymbol{z}) + \nabla\boldsymbol{f}(\boldsymbol{z})(\boldsymbol{u}-\boldsymbol{z}), \boldsymbol{u}),$$

for a fixed $\boldsymbol{x}$ and $\boldsymbol{z}$.

Then, in each iteration of Algorithm 3.3, we compute, for $t \geq 0$:

$$\boldsymbol{v}_{t+1} \;\in\; \arg\min_{\boldsymbol{u}\in\mathcal{X}}\Big\{ \langle \nabla r(\boldsymbol{u}_t), \boldsymbol{u}\rangle + h(\boldsymbol{u}) \Big\}. \tag{B.10}$$

The optimality condition for this operation is (see, e.g. Theorem 3.1.23 in [168])

$$\langle \nabla r(\boldsymbol{u}_t), \boldsymbol{u} - \boldsymbol{v}_{t+1}\rangle + h(\boldsymbol{u}) \;\geq\; h(\boldsymbol{v}_{t+1}), \qquad \forall\, \boldsymbol{u}\in\mathcal{X}. \tag{B.11}$$

Therefore, employing the Lipschitz continuity of the gradient of $r(\cdot)$, we have

$$\begin{aligned} s(\boldsymbol{u}_{t+1}) &\leq r(\boldsymbol{u}_t) + \langle \nabla r(\boldsymbol{u}_t), \boldsymbol{u}_{t+1} - \boldsymbol{u}_t \rangle + \frac{\beta}{2}\|\boldsymbol{u}_{t+1} - \boldsymbol{u}_t\|^2 + h(\boldsymbol{u}_{t+1}) \\ &= r(\boldsymbol{u}_t) + \alpha_t \langle \nabla r(\boldsymbol{u}_t), \boldsymbol{v}_{t+1} - \boldsymbol{u}_t \rangle + \frac{\beta\alpha_t^2}{2}\|\boldsymbol{v}_{t+1} - \boldsymbol{u}_t\|^2 \\ &\quad + h(\alpha_t \boldsymbol{v}_{t+1} + (1-\alpha_t)\boldsymbol{u}_t) \qquad \text{(B.12)} \\ &\leq s(\boldsymbol{u}_t) + \alpha_t\big(\langle \nabla r(\boldsymbol{u}_t), \boldsymbol{v}_{t+1} - \boldsymbol{u}_t \rangle + h(\boldsymbol{v}_{t+1}) - h(\boldsymbol{u}_t)\big) + \frac{\beta\alpha^2\mathcal{D}_{\mathcal{X}}^2}{2} \\ &\equiv s(\boldsymbol{u}_t) - \alpha_t \mathcal{G}_t + \frac{\beta\alpha_t^2\mathcal{D}_{\mathcal{X}}^2}{2}, \end{aligned}$$

where the last equality comes from the definition of $\mathcal{G}_t$ in Algorithm 3.3.

Note that α_t is defined as the minimizer of $\frac{\beta\alpha_t^2}{2}\|\boldsymbol{v}_{t+1} - \boldsymbol{u}_t\|^2 - \alpha_t\mathcal{G}_t$ and hence, for any other $\rho_t \in [0,1]$ it will hold that:

$$s(\boldsymbol{u}_{t+1}) \leq s(\boldsymbol{u}_t) - \rho_t\mathcal{G}_t + \frac{\beta\rho_t^2\mathcal{D}_{\mathcal{X}}^2}{2}. \qquad \text{(B.13)}$$

At the same time,

$$\begin{aligned} \mathcal{G}_t &\coloneqq h(\boldsymbol{u}_t) - h(\boldsymbol{v}_{t+1}) - \langle \nabla r(\boldsymbol{u}_t), \boldsymbol{v}_{t+1} - \boldsymbol{u}_t \rangle \\ &\overset{\text{(B.10)}}{\geq} h(\boldsymbol{u}_t) - h(\boldsymbol{u}) - \langle \nabla r(\boldsymbol{u}_t), \boldsymbol{u} - \boldsymbol{u}_t \rangle \qquad \text{(B.14)} \\ &\geq s(\boldsymbol{u}_t) - s(\boldsymbol{u}), \qquad \forall \boldsymbol{u} \in \mathcal{X} \end{aligned}$$

where the last line follows from the convexity of $r(\boldsymbol{u})$. Letting $\boldsymbol{u} \coloneqq \boldsymbol{u}^\star$ (solution to (B.9)) in (B.14) and further substituting it into (B.13) and subtracting $s^\star$ from both sides, we obtain

$$[s(\boldsymbol{u}_{t+1}) - s^\star] \leq (1-\rho_t)\,[s(\boldsymbol{u}_t) - s^\star] + \frac{\beta\rho_t^2\mathcal{D}_{\mathcal{X}}^2}{2}. \qquad \text{(B.15)}$$

Now, let us choose $\rho_t \coloneqq \frac{a_{t+1}}{A_{t+1}}$ for sequences $A_t \coloneqq t\cdot(t+1)$, and $a_{t+1} \coloneqq A_{t+1} - A_t = 2(t+1)$. Then,

$\rho_t := \frac{2}{2+t}$, $t \geq 0$. Using this choice, inequality (B.15) can be rewritten as

$$A_{t+1}\big[s(\boldsymbol{u}_{t+1}) - s^\star\big] \leq A_t\big[s(\boldsymbol{u}_t) - s^\star\big] + \frac{a_{t+1}^2 \beta \mathcal{D}_{\mathcal{X}}^2}{2A_{t+1}}$$

Telescoping this inequality for the first iterations, we obtain, for $t \geq 1$:

$$s(\boldsymbol{u}_t) - s^\star \leq \frac{\beta \mathcal{D}_{\mathcal{X}}^2}{2A_t} \cdot \sum_{i=1}^{t} \frac{a_i^2}{A_i} = \frac{\beta \mathcal{D}_{\mathcal{X}}^2}{2t(t+1)} \cdot \sum_{i=1}^{t} \frac{4i}{i+1} \leq \frac{2\beta \mathcal{D}_{\mathcal{X}}^2}{t+1}. \tag{B.16}$$

This is the global convergence in terms of the functional residual. It remains to justify the convergence for the accuracy certificates $\mathcal{G}_t$. Multiplying (B.13) by A_{t+1}, we obtain

$$a_{t+1}\mathcal{G}_t \leq a_{t+1} s(\boldsymbol{u}_t) + A_t s(\boldsymbol{u}_t) - A_{t+1} s(\boldsymbol{u}_{t+1}) + \frac{a_{t+1}^2}{A_{t+1}} \frac{\beta \mathcal{D}_{\mathcal{X}}^2}{2}. \tag{B.17}$$

Telescoping this bound, we get, for $t \geq 1$:

$$\begin{aligned}
\sum_{i=1}^{t} a_{i+1} \cdot \min_{1 \leq i \leq t} \mathcal{G}_i &\leq \sum_{i=1}^{t} a_{i+1}\mathcal{G}_i \\
&\overset{(B.17)}{\leq} a_1\big[s(\boldsymbol{u}_1) - s^\star\big] + \sum_{i=1}^{t} a_{i+1}\big[s(\boldsymbol{u}_i) - s^\star\big] + \frac{\beta \mathcal{D}_{\mathcal{X}}^2}{2} \sum_{i=1}^{t} \frac{a_{i+1}^2}{A_{i+1}} \\
&\overset{(B.16)}{\leq} 2\beta \mathcal{D}_{\mathcal{X}}^2 \cdot \Big(1 + \sum_{i=1}^{t} \frac{a_{i+1}}{i+1} + \frac{1}{4} \sum_{i=1}^{t} \frac{a_{i+1}^2}{A_{i+1}}\Big) \\
&\leq 2\beta \mathcal{D}_{\mathcal{X}}^2 \cdot (1 + 3t).
\end{aligned}$$

Dividing both sides by $\sum_{i=1}^{t} a_{i+1} = A_{t+1} - A_1 = t(3+t)$ completes the proof we finally get:

$$\min_{1 \leq i \leq t} \mathcal{G}_i \leq \frac{6\beta \mathcal{D}_{\mathcal{X}}^2}{t}.$$

□

B.1.5 Proof of Proposition 3.1

Proposition 3.1. *Let* $\gamma_k \coloneqq \frac{1}{\sqrt{1+k}}$. *Then, for the iterations* (3.18)*, under Assumption 3.1.b and for all* $k \geq 1$*, it holds that*

$$\min_{0\leq i\leq k} \max_{\boldsymbol{y}\in\mathcal{X}} \langle \nabla\Phi(\boldsymbol{y}_i), \boldsymbol{y}_i - \boldsymbol{y}\rangle \quad \leq \quad \mathcal{O}\left(\frac{\ln(k)}{\sqrt{k}}\right).$$

Proof. In our case, we have $\varphi(\boldsymbol{x}) \equiv \|\boldsymbol{f}(\boldsymbol{x})\|_2$. Using Lemma B.1, we obtain

$$\begin{aligned} \varphi(\boldsymbol{y}_{k+1}) \quad &\leq \quad \|\boldsymbol{f}(\boldsymbol{y}_k) + \nabla f(\boldsymbol{y}_k)(\boldsymbol{y}_{k+1} - \boldsymbol{y}_k)\|_2 + \tfrac{\gamma_k^2}{2}\mathcal{S} \\ &= \quad \|\boldsymbol{f}(\boldsymbol{y}_k) + \gamma_k \nabla f(\boldsymbol{y}_k)(\boldsymbol{x}_{k+1} - \boldsymbol{y}_k)\|_2 + \tfrac{\gamma_k^2}{2}\mathcal{S}, \end{aligned} \tag{B.18}$$

where $\boldsymbol{x}_{k+1} \in \mathcal{X}$ is the point such that $\boldsymbol{y}_{k+1} = \boldsymbol{y}_k + \gamma_k(\boldsymbol{x}_{k+1} - \boldsymbol{y}_k)$. Using convexity of the function $g(\boldsymbol{x}) \coloneqq \|\boldsymbol{f}(\boldsymbol{y}_k) + \gamma_k \nabla \boldsymbol{f}(\boldsymbol{y}_k)(\boldsymbol{x} - \boldsymbol{y}_k)\|_2$, we get that

$$\begin{aligned} \varphi(\boldsymbol{y}_k) \quad &= \quad g(\boldsymbol{y}_k) \;\geq\; g(\boldsymbol{x}_{k+1}) + \langle g'(\boldsymbol{x}_{k+1}), \boldsymbol{y}_k - \boldsymbol{x}_{k+1}\rangle \\ &= \quad \|\boldsymbol{f}(\boldsymbol{y}_k) + \gamma_k \nabla f(\boldsymbol{y}_k)(\boldsymbol{x}_{k+1} - \boldsymbol{y}_k)\|_2 + \langle g'(\boldsymbol{x}_{k+1}), \boldsymbol{y}_k - \boldsymbol{x}_{k+1}\rangle, \end{aligned}$$

where the subgradient $g'(\boldsymbol{x}_{k+1}) = \gamma_k \nabla \boldsymbol{f}(\boldsymbol{y}_k)^\top \frac{\boldsymbol{f}_{k+1}}{\|\boldsymbol{f}_{k+1}\|_2}$ with $\boldsymbol{f}_{k+1} \coloneqq \boldsymbol{f}(\boldsymbol{y}_k) + \gamma_k \nabla \boldsymbol{f}(\boldsymbol{y}_k)(\boldsymbol{x}_{k+1} - \boldsymbol{y}_k)$, satisfies the stationary condition for the method step:

$$\langle g'(\boldsymbol{x}_{k+1}), \boldsymbol{x} - \boldsymbol{x}_{k+1}\rangle \quad \geq \quad 0, \qquad \forall \boldsymbol{x} \in \mathcal{X}. \tag{B.19}$$

A few comments are in order now about the use of the subgradient above. Note that we wish to impose an assumption on $\boldsymbol{f}$ which can ensure that $\boldsymbol{f}(\boldsymbol{y}_k) + \gamma_k \nabla \boldsymbol{f}(\boldsymbol{y}_k)(\boldsymbol{x} - \boldsymbol{y}_k) \neq \boldsymbol{0} \in \mathbb{R}^n$. First, some preliminaries. Under Assumption 3.1.b on $\boldsymbol{f}$, it holds that:

$$\exists \mathcal{F} \in (0,\infty) \text{ s.t. } \left\|\boldsymbol{f}(x)\right\| \leq \mathcal{F}, \forall \boldsymbol{x} \in \mathcal{X} \quad \text{by continuity of } \boldsymbol{f} \tag{B.20}$$

$$\exists \mathcal{H} \in (0,\infty) \text{ s.t. } \left\|\nabla \boldsymbol{f}(x)\right\| \leq \mathcal{H}, \forall \boldsymbol{x} \in \mathcal{X} \quad \text{by continuous differentiability of } \boldsymbol{f} \tag{B.21}$$

From here, we can bound the products between Jacobians and iterates as follows:

$$\left\|\nabla \boldsymbol{f}(\boldsymbol{x})(\boldsymbol{y} - \boldsymbol{z})\right\| \leq \left\|\nabla \boldsymbol{f}(\boldsymbol{x})\right\| \left\|\boldsymbol{y} - \boldsymbol{z}\right\| \leq \mathcal{H}\mathcal{D}_{\mathcal{X}}, \quad \forall \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z} \in \mathcal{X}. \tag{B.22}$$

Thus, without loss of generality, we can shift $\boldsymbol{f}$ by a constant vector of identical values depending on $\mathcal{H}\mathcal{D}_{\mathcal{X}}$ such that we ensure, for example, $\boldsymbol{f}(\boldsymbol{y}_k) + \gamma_k \nabla \boldsymbol{f}(\boldsymbol{y}_k)(\boldsymbol{x} - \boldsymbol{y}_k) > \boldsymbol{0}$ component-wise.

Hence, combining these observations with (B.18), we have

$$\begin{aligned}\varphi(\boldsymbol{y}_k)-\varphi(\boldsymbol{y}_{k+1}) &\geq \langle g'(\boldsymbol{x}_{k+1}),\boldsymbol{y}_k-\boldsymbol{x}_{k+1}\rangle-\tfrac{\gamma_k^2}{2}\mathcal{S}\\ &\overset{\text{(B.19)}}{\geq} \max_{\boldsymbol{x}\in\mathcal{X}}\langle g'(\boldsymbol{x}_{k+1}),\boldsymbol{y}_k-\boldsymbol{x}\rangle-\tfrac{\gamma_k^2}{2}\mathcal{S}.\end{aligned}$$

Then, by lower bounding appropriately using (B.20) and (B.21), we get:

$$\begin{aligned}\varphi(\boldsymbol{y}_k)-\varphi(\boldsymbol{y}_{k+1}) &\geq \tfrac{\gamma_k}{\mathcal{F}+\mathcal{H}\mathcal{D}_{\mathcal{X}}}\max_{\boldsymbol{y}\in\mathcal{X}}\langle\nabla\boldsymbol{f}(\boldsymbol{y}_k)^\top\boldsymbol{f}(\boldsymbol{y}_k),\boldsymbol{y}_k-\boldsymbol{y}\rangle-\gamma_k^2\Big(\tfrac{\mathcal{H}\mathcal{D}_{\mathcal{X}}^2}{\mathcal{F}+\mathcal{H}\mathcal{D}_{\mathcal{X}}}+\tfrac{\mathcal{S}}{2}\Big)\\ &= \tfrac{\gamma_k}{\mathcal{F}+\mathcal{H}\mathcal{D}_{\mathcal{X}}}\max_{\boldsymbol{y}\in\mathcal{X}}\langle\nabla\Phi(\boldsymbol{y}_k),\boldsymbol{y}_k-\boldsymbol{y}\rangle-\gamma_k^2\Big(\tfrac{\mathcal{H}\mathcal{D}_{\mathcal{X}}^2}{\mathcal{F}+\mathcal{H}\mathcal{D}_{\mathcal{X}}}+\tfrac{\mathcal{S}}{2}\Big).\end{aligned}$$

Substituting $\gamma_k := \frac{1}{\sqrt{1+k}}$ and telescoping this bound would lead to the desired global convergence (for the details, see the end of the proof of Theorem 3.2). □

B.2 Interpretation of the generalized gap in non-convex settings

While we cannot make any strong claims about the meaning of $\mathcal{G}_k$ in general, we can provide an additional observation for this quantity when the outer component F is smooth inside a ball included in $\mathcal{X}$.

Thus, consider a ball of radius ε centered at $\boldsymbol{y}_k$ denoted by $B(\boldsymbol{y}_k, \varepsilon) = \{\boldsymbol{x} \in \mathbb{R}^d : \|\boldsymbol{x} - \boldsymbol{y}_k\| \leq \varepsilon\}$, and set $\mathcal{B} = B(\boldsymbol{y}_k, \varepsilon) \cap \mathcal{X}$. Assuming that $F(\boldsymbol{u}, \boldsymbol{x})$ is differentiable at all points from $\mathbb{R}^n \times \mathcal{B}$, and that its gradient is Lipschitz continuous with constant L_F, we have for any $\boldsymbol{x} \in \mathcal{B} \subseteq \mathcal{X}$:

$$
\begin{aligned}
\mathcal{G}_k &= \max_{\boldsymbol{x} \in \mathcal{X}} \Big[\varphi(\boldsymbol{y}_k) - F\big(\boldsymbol{f}(\boldsymbol{y}_k) + \nabla \boldsymbol{f}(\boldsymbol{y}_k)(\boldsymbol{x} - \boldsymbol{y}_k), \boldsymbol{x}\big) \Big] \\
&\geq \max_{\boldsymbol{x} \in \mathcal{B}} \Big[\varphi(\boldsymbol{y}_k) - F(\boldsymbol{f}(\boldsymbol{y}_k), \boldsymbol{y}_k) - \langle \tfrac{\partial F}{\partial \boldsymbol{u}}(\boldsymbol{f}(\boldsymbol{y}_k), \boldsymbol{y}_k), \nabla \boldsymbol{f}(\boldsymbol{y}_k)(\boldsymbol{x} - \boldsymbol{y}_k) \rangle \\
&\qquad - \langle \tfrac{\partial F}{\partial \boldsymbol{x}}(\boldsymbol{f}(\boldsymbol{y}_k), \boldsymbol{y}_k), \boldsymbol{x} - \boldsymbol{y}_k \rangle - \tfrac{L_F}{2} \big(\|\nabla \boldsymbol{f}(\boldsymbol{y}_k)\|^2 + 1 \big) \cdot \varepsilon^2 \Big] \\
&= \max_{\boldsymbol{x} \in \mathcal{B}} \Big[\langle \nabla \varphi(\boldsymbol{y}_k), \boldsymbol{y}_k - \boldsymbol{x} \rangle \Big] - \tfrac{L_F}{2} \big(\|\nabla \boldsymbol{f}(\boldsymbol{y}_k)\|^2 + 1 \big) \cdot \varepsilon^2 .
\end{aligned}
$$

Hence, for a small enough ball, $\mathcal{G}_k$ is an $\mathcal{O}\left(\epsilon^2\right)$-approximation of the original FW gap restricted to the considered neighbourhood. If, in addition, the composite function φ is convex in $\mathcal{B}$ and there is a local optimum $\boldsymbol{x}^\star \in \mathcal{B}$, then $\mathcal{G}_k$ is an $\mathcal{O}\left(\epsilon^2\right)$-approximation of functional suboptimality.

C Appendix for Chapter 4

C.1 Proof of Lemma 4.1

Lemma 4.1. *Consider APDA along with Assumptions 4.1 and 4.2 and* $(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{X} \times \mathcal{Y}$. *Then, for all* k *and* $\eta_k \in \left(\frac{\beta\tau_k \|\boldsymbol{A}\|}{1-c}, \frac{1-2\tau_k L_k}{2\tau_k \|\boldsymbol{A}\|}\right)$,

$$\begin{aligned}\|\boldsymbol{x}_{k+1}-\boldsymbol{x}\|^2 + \frac{1}{\beta}\left\|\boldsymbol{y}_{k+1}-\boldsymbol{y}\right\|^2 + \left(1-\eta_k\tau_k\|\boldsymbol{A}\|-\tau_k L_k\right)\|\boldsymbol{x}_{k+1}-\boldsymbol{x}_k\|^2 \\ + \frac{\eta_k-\tau_k\beta\|\boldsymbol{A}\|}{\beta\eta_k}\left\|\boldsymbol{y}_{k+1}-\boldsymbol{y}_k\right\|^2 + 2\tau_k(1+\theta_k)P_{\boldsymbol{x},\boldsymbol{y}}(\boldsymbol{x}_k) + 2\tau_k D_{\boldsymbol{x},\boldsymbol{y}}(\boldsymbol{y}_{k+1}) \\ \leq \|\boldsymbol{x}_k-\boldsymbol{x}\|^2 + \frac{1}{\beta}\left\|\boldsymbol{y}_k-\boldsymbol{y}\right\|^2 + \tau_k L_k\|\boldsymbol{x}_k-\boldsymbol{x}_{k-1}\|^2 + 2\tau_k\theta_k P_{\boldsymbol{x},\boldsymbol{y}}(\boldsymbol{x}_{k-1}).\end{aligned}$$

Moreover, it holds that:

1) $\tau_k L_k < \frac{1}{2} < 1-\eta_k\tau_k\|\boldsymbol{A}\|-\tau_k L_k$,
2) $\frac{1}{\beta} - \frac{\tau_k\|\boldsymbol{A}\|}{\eta_k} > \frac{c}{\beta} > 0$.

Proof. Using the primal update rule, we have

$$\|\boldsymbol{x}_{k+1}-\boldsymbol{x}\|^2 = \|\boldsymbol{x}_k-\boldsymbol{x}\|^2 + \|\boldsymbol{x}_{k+1}-\boldsymbol{x}_k\|^2 - 2\tau_k\langle\nabla f(\boldsymbol{x}_k) + \boldsymbol{A}^\top\boldsymbol{y}_{k+1}, \boldsymbol{x}_k-\boldsymbol{x}\rangle. \tag{C.1}$$

We address each term in the RHS separately. Using the convexity of f we bound the last term of (C.1):

$$-2\tau_k\langle\nabla f(\boldsymbol{x}_k) + \boldsymbol{A}^\top\boldsymbol{y}_{k+1}, \boldsymbol{x}_k-\boldsymbol{x}\rangle \leq 2\tau_k\left(f(\boldsymbol{x})-f(\boldsymbol{x}_k)\right) + 2\tau_k\langle\boldsymbol{A}(\boldsymbol{x}-\boldsymbol{x}_k), \boldsymbol{y}_{k+1}\rangle. \tag{C.2}$$

For the second term of (C.1) we use an expansion similar to the analysis in [147] along with the

primal update rule:

$$\begin{aligned}\| \boldsymbol{x}_{k+1} - \boldsymbol{x}_k \|^2 &= 2 \| \boldsymbol{x}_{k+1} - \boldsymbol{x}_k \|^2 - \| \boldsymbol{x}_{k+1} - \boldsymbol{x}_k \|^2 \\ &= 2\tau_k \langle \nabla f(\boldsymbol{x}_k) + \boldsymbol{A}^\top \boldsymbol{y}_{k+1}, \boldsymbol{x}_k - \boldsymbol{x}_{k+1} \rangle - \| \boldsymbol{x}_{k+1} - \boldsymbol{x}_k \|^2 \\ &= 2\tau_k \langle \nabla f(\boldsymbol{x}_k) - \nabla f(\boldsymbol{x}_{k-1}), \boldsymbol{x}_k - \boldsymbol{x}_{k+1} \rangle + 2\tau_k \langle \boldsymbol{A}^\top \boldsymbol{y}_{k+1} - \boldsymbol{A}^\top \boldsymbol{y}_k, \boldsymbol{x}_k - \boldsymbol{x}_{k+1} \rangle \\ &\quad + 2\tau_k \langle \nabla f(\boldsymbol{x}_{k-1}) + \boldsymbol{A}^\top \boldsymbol{y}_k, \boldsymbol{x}_k - \boldsymbol{x}_{k+1} \rangle - \| \boldsymbol{x}_{k+1} - \boldsymbol{x}_k \|^2 . \end{aligned} \tag{C.3}$$

Notice that the first term in (C.3) gives us the opportunity to insert a dependence on the local Lipschitz constant L_k. Using Cauchy-Schwarz, the definition of L_k and Young's inequality, we indeed take this opportunity and get:

$$\begin{aligned}\langle \nabla f(\boldsymbol{x}_k) - \nabla f(\boldsymbol{x}_{k-1}), \boldsymbol{x}_k - \boldsymbol{x}_{k+1} \rangle &\leq L_k \| \boldsymbol{x}_k - \boldsymbol{x}_{k-1} \| \| \boldsymbol{x}_{k+1} - \boldsymbol{x}_k \| \\ &\leq \frac{L_k}{2} \left(\| \boldsymbol{x}_k - \boldsymbol{x}_{k-1} \|^2 + \| \boldsymbol{x}_{k+1} - \boldsymbol{x}_k \|^2 \right). \end{aligned} \tag{C.4}$$

Similarly, we bound the second term in (C.3) and obtain:

$$\langle \boldsymbol{A}^\top \boldsymbol{y}_{k+1} - \boldsymbol{A}^\top \boldsymbol{y}_k, \boldsymbol{x}_k - \boldsymbol{x}_{k+1} \rangle \leq \frac{\| \boldsymbol{A} \| \eta}{2} \| \boldsymbol{x}_{k+1} - \boldsymbol{x}_k \|^2 + \frac{\| \boldsymbol{A} \|}{2\eta} \left\| \boldsymbol{y}_{k+1} - \boldsymbol{y}_k \right\|^2, \tag{C.5}$$

where $\eta > 0$ is a free parameter coming from Young's inequality.

Finally, for the third term in (C.3) we use the update rule and the convexity of f:

$$\begin{aligned}\langle \nabla f(\boldsymbol{x}_{k-1}) + \boldsymbol{A}^\top \boldsymbol{y}_k, \boldsymbol{x}_k - \boldsymbol{x}_{k+1} \rangle &= \langle \frac{1}{\tau_{k-1}} (\boldsymbol{x}_{k-1} - \boldsymbol{x}_k), \tau_k (\nabla f(\boldsymbol{x}_k) + \boldsymbol{A}^\top \boldsymbol{y}_{k+1}) \rangle \\ &\leq \theta_k \left(f(\boldsymbol{x}_{k-1}) - f(\boldsymbol{x}_k) \right) + \theta_k \langle \boldsymbol{A}(\boldsymbol{x}_{k-1} - \boldsymbol{x}_k), \boldsymbol{y}_{k+1} \rangle. \end{aligned} \tag{C.6}$$

Replacing (C.4), (C.5) and (C.6) into (C.3), we get

$$\begin{aligned}\| \boldsymbol{x}_{k+1} - \boldsymbol{x}_k \|^2 \leq \tau_k L_k \| \boldsymbol{x}_k - \boldsymbol{x}_{k-1} \|^2 &+ \left(\tau_k \| \boldsymbol{A} \| \eta + \tau_k L_k - 1 \right) \| \boldsymbol{x}_{k+1} - \boldsymbol{x}_k \|^2 \\ &+ \frac{\tau_k \| \boldsymbol{A} \|}{\eta} \left\| \boldsymbol{y}_{k+1} - \boldsymbol{y}_k \right\|^2 + 2\tau_k \theta_k \left(f(\boldsymbol{x}_{k-1}) - f(\boldsymbol{x}_k) \right) \\ &+ 2\tau_k \theta_k \langle \boldsymbol{A}(\boldsymbol{x}_{k-1} - \boldsymbol{x}_k), \boldsymbol{y}_{k+1} \rangle. \end{aligned} \tag{C.7}$$

Finally, replacing (C.7) and (C.2) back into (C.1) and using the fact that $\theta_k \langle \boldsymbol{A}(\boldsymbol{x}_{k-1} - \boldsymbol{x}_k), \boldsymbol{y}_{k+1} \rangle + \langle \boldsymbol{A}(\boldsymbol{x} - \boldsymbol{x}_k), \boldsymbol{y}_{k+1} \rangle = -\langle \boldsymbol{A}(\widetilde{\boldsymbol{x}}_k - \boldsymbol{x}), \boldsymbol{y}_{k+1} \rangle$, we obtain the inequality for the primal iterate sequence:

$$\begin{aligned}\| \boldsymbol{x}_{k+1} - \boldsymbol{x} \|^2 \leq \| \boldsymbol{x}_k - \boldsymbol{x} \|^2 + \tau_k L_k \| \boldsymbol{x}_k - \boldsymbol{x}_{k-1} \|^2 &+ \left(\tau_k \| \boldsymbol{A} \| \eta + \tau_k L_k - 1 \right) \| \boldsymbol{x}_{k+1} - \boldsymbol{x}_k \|^2 \\ &+ \frac{\tau_k \| \boldsymbol{A} \|}{\eta} \left\| \boldsymbol{y}_{k+1} - \boldsymbol{y}_k \right\|^2 + 2\tau_k \theta_k \left(f(\boldsymbol{x}_{k-1}) - f(\boldsymbol{x}_k) \right) + 2\tau_k \left(f(\boldsymbol{x}) - f(\boldsymbol{x}_k) \right) \end{aligned}$$

$$-2\tau_k\langle \boldsymbol{A}(\widetilde{\boldsymbol{x}}_k - \boldsymbol{x}), \boldsymbol{y}_{k+1}\rangle. \quad \text{(C.8)}$$

We now seek a similar result for the dual sequence. For this, we use the following characterization of the proximal operator:

$$\boldsymbol{u} = \text{prox}_{g^*}(\boldsymbol{x}) \iff \langle \boldsymbol{u} - \boldsymbol{x}, \boldsymbol{z} - \boldsymbol{u}\rangle \geq g^*(\boldsymbol{u}) - g^*(\boldsymbol{z})\ \forall \boldsymbol{z}. \quad \text{(C.9)}$$

Thus, letting $\boldsymbol{u} = \boldsymbol{y}_{k+1}$, $\boldsymbol{x} = \boldsymbol{y}_k + \sigma_k \boldsymbol{A}\widetilde{\boldsymbol{x}}_k$ and $\boldsymbol{z} = \boldsymbol{y}$ in (C.9), we obtain:

$$g^*(\boldsymbol{y}) \geq g^*(\boldsymbol{y}_{k+1}) + \langle \frac{1}{\sigma_k}(\boldsymbol{y}_k - \boldsymbol{y}_{k+1}), \boldsymbol{y} - \boldsymbol{y}_{k+1}\rangle + \langle \boldsymbol{A}\widetilde{\boldsymbol{x}}_k, \boldsymbol{y} - \boldsymbol{y}_{k+1}\rangle.$$

Using the cosine rule for the second term, the fact that $\sigma_k = \beta\tau_k$ and multiplying both sides by $2\tau_k > 0$, we obtain:

$$\frac{1}{\beta}\left\| \boldsymbol{y}_{k+1} - \boldsymbol{y} \right\|^2 \leq \frac{1}{\beta}\left\| \boldsymbol{y}_k - \boldsymbol{y} \right\|^2 - \frac{1}{\beta}\left\| \boldsymbol{y}_{k+1} - \boldsymbol{y}_k \right\|^2 + 2\tau_k(g^*(\boldsymbol{y}) - g^*(\boldsymbol{y}_{k+1})) + 2\tau_k\langle \boldsymbol{A}\widetilde{\boldsymbol{x}}_k, \boldsymbol{y}_{k+1} - \boldsymbol{y}\rangle. \quad \text{(C.10)}$$

Summing (C.10) with (C.8) we obtain the following recurrence:

$$\begin{aligned}
&\| \boldsymbol{x}_{k+1} - \boldsymbol{x} \|^2 + \frac{1}{\beta}\left\| \boldsymbol{y}_{k+1} - \boldsymbol{y} \right\|^2 \\
&\quad \leq \| \boldsymbol{x}_k - \boldsymbol{x} \|^2 + \frac{1}{\beta}\left\| \boldsymbol{y}_k - \boldsymbol{y} \right\|^2 + \tau_k L_k \| \boldsymbol{x}_k - \boldsymbol{x}_{k-1} \|^2 \\
&\qquad + \left(\tau_k \| \boldsymbol{A} \| \eta + \tau_k L_k - 1\right) \| \boldsymbol{x}_{k+1} - \boldsymbol{x}_k \|^2 + \left(\frac{\tau_k \| \boldsymbol{A} \|}{\eta} - \frac{1}{\beta}\right)\left\| \boldsymbol{y}_{k+1} - \boldsymbol{y}_k \right\|^2 \\
&\qquad\quad + 2\tau_k \left(\theta_k \left(f(\boldsymbol{x}_{k-1}) - f(\boldsymbol{x}_k)\right) + f(\boldsymbol{x}) - f(\boldsymbol{x}_k) + g^*(\boldsymbol{y}) - g^*(\boldsymbol{y}_{k+1})\right) \\
&\qquad\quad + 2\tau_k\langle \boldsymbol{A}\boldsymbol{x}, \boldsymbol{y}_{k+1}\rangle - 2\tau_k\langle \boldsymbol{A}\widetilde{\boldsymbol{x}}_k, \boldsymbol{y}\rangle. \quad \text{(C.11)}
\end{aligned}$$

We further process the terms involving f and g^* on the right-hand side in order to form the $P_{\boldsymbol{x},\boldsymbol{y}}(\cdot)$ and $D_{\boldsymbol{x},\boldsymbol{y}}(\cdot)$:

$$f(\boldsymbol{x}) - f(\boldsymbol{x}_k) = -P_{\boldsymbol{x},\boldsymbol{y}}(\boldsymbol{x}_k) + \langle \boldsymbol{A}(\boldsymbol{x}_k - \boldsymbol{x}), \boldsymbol{y}\rangle,$$

$$\theta_k(f(\boldsymbol{x}_{k-1}) - f(\boldsymbol{x}_k)) = \theta_k P_{\boldsymbol{x},\boldsymbol{y}}(\boldsymbol{x}_{k-1}) - \theta_k P_{\boldsymbol{x},\boldsymbol{y}}(\boldsymbol{x}_k) + \langle \theta_k \boldsymbol{A}(\boldsymbol{x}_k - \boldsymbol{x}_{k-1}), \boldsymbol{y}\rangle,$$

$$g^*(\boldsymbol{y}) - g^*(\boldsymbol{y}_{k+1}) = -D_{\boldsymbol{x},\boldsymbol{y}}(\boldsymbol{y}_{k+1}) - \langle \boldsymbol{A}\boldsymbol{x}, \boldsymbol{y}_{k+1} - \boldsymbol{y}\rangle.$$

Replacing the above expressions into (C.11) and noting that $\langle \boldsymbol{A}(\widetilde{\boldsymbol{x}}_k - \boldsymbol{x}), \boldsymbol{y}\rangle - \langle \boldsymbol{A}\boldsymbol{x}, \boldsymbol{y}_{k+1} - \boldsymbol{y}\rangle +$

$\langle \boldsymbol{A}\boldsymbol{x}, \boldsymbol{y}_{k+1}\rangle - \langle \boldsymbol{A}\widetilde{\boldsymbol{x}}_k, \boldsymbol{y}\rangle = 0$, we obtain:

$$\begin{aligned}
&\|\boldsymbol{x}_{k+1}-\boldsymbol{x}\|^2 + \frac{1}{\beta}\left\|\boldsymbol{y}_{k+1}-\boldsymbol{y}\right\|^2 + \left(1-\tau_k\|\boldsymbol{A}\|\eta-\tau_k L_k\right)\|\boldsymbol{x}_{k+1}-\boldsymbol{x}_k\|^2 \\
&\quad + \left(\frac{1}{\beta}-\frac{\tau_k\|\boldsymbol{A}\|}{\eta}\right)\left\|\boldsymbol{y}_{k+1}-\boldsymbol{y}_k\right\|^2 + 2\tau_k(1+\theta_k)P_{\boldsymbol{x},\boldsymbol{y}}(\boldsymbol{x}_k) + 2\tau_k D_{\boldsymbol{x},\boldsymbol{y}}(\boldsymbol{y}_{k+1}) \\
&\qquad \le \|\boldsymbol{x}_k-\boldsymbol{x}\|^2 + \frac{1}{\beta}\left\|\boldsymbol{y}_k-\boldsymbol{y}\right\|^2 + \tau_k L_k\|\boldsymbol{x}_k-\boldsymbol{x}_{k-1}\|^2 + 2\tau_k\theta_k P_{\boldsymbol{x},\boldsymbol{y}}(\boldsymbol{x}_{k-1}).
\end{aligned} \tag{C.12}$$

What is left in order to obtain the stated result is to choose η, possibly depending on k, such that the corresponding terms are positive. First, note that $\tau_k L_k\|\boldsymbol{x}_k-\boldsymbol{x}_{k-1}\|^2 < \frac{1}{2}\|\boldsymbol{x}_k-\boldsymbol{x}_{k-1}\|^2$ because $z\mapsto\frac{z}{2\sqrt{z^2+a}}$, $a>0$ is an increasing function whose limit at ∞ is $\frac{1}{2}$ and we have:

$$\tau_k L_k \le \frac{L_k}{2\sqrt{L_k^2+(\beta/(1-c))\|\boldsymbol{A}\|^2}} < \frac{1}{2}. \tag{C.13}$$

Next we need to choose $\eta=\eta_k$ (iteration-dependent) to satisfy:

$$\begin{cases} \frac{1}{\beta}-\frac{\tau_k\|\boldsymbol{A}\|}{\eta_k} > 0, \\ 1-\tau_k\|\boldsymbol{A}\|\eta_k-\tau_k L_k > \frac{1}{2}. \end{cases}$$

However, for theoretical purposes related to controlling the sequence $\left\|\boldsymbol{y}_{k+1}-\boldsymbol{y}_k\right\|^2$, we strengthen the first inequality to $\dfrac{1}{\beta}-\dfrac{\tau_k\|\boldsymbol{A}\|}{\eta_k} > \dfrac{c}{\beta}$, $c\in(0,1)$. In practice, this constant is chosen as small as possible. The new conditions to be satisfied are:

$$\begin{cases} \frac{1}{\beta}-\frac{\tau_k\|\boldsymbol{A}\|}{\eta_k} > \frac{c}{\beta}, \\ 1-\tau_k\|\boldsymbol{A}\|\eta_k-\tau_k L_k > \frac{1}{2}, \end{cases} \iff \begin{cases} \eta_k > \frac{\beta\tau_k\|\boldsymbol{A}\|}{1-c}, \\ \eta_k < \frac{1-2\tau_k L_k}{2\tau_k\|\boldsymbol{A}\|}. \end{cases} \tag{C.14}$$

The question we need to answer therefore is: given the expression of τ_k, is the interval always valid for choosing $\eta_k\in\left(\dfrac{\beta\tau_k\|\boldsymbol{A}\|}{1-c}, \dfrac{1-2\tau_k L_k}{2\tau_k\|\boldsymbol{A}\|}\right)$?

To answer, we form the corresponding quadratic inequality in τ_k:

$$\frac{\beta\tau_k\|\boldsymbol{A}\|}{1-c} - \frac{1-2\tau_k L_k}{2\tau_k\|\boldsymbol{A}\|} < 0 \iff \frac{2\beta\tau_k^2\|\boldsymbol{A}\|^2}{1-c} + 2\tau_k L_k - 1 < 0, \tag{C.15}$$

whose 2 real roots are given by:

$$\begin{cases} \tau_{k,1} = \frac{1}{L_k - \sqrt{L_k^2 + 2(\beta/(1-c))\|\boldsymbol{A}\|^2}} < 0, \\ \tau_{k,2} = \frac{1}{L_k + \sqrt{L_k^2 + 2(\beta/(1-c))\|\boldsymbol{A}\|^2}} > 0. \end{cases}$$

For inequality (C.15) to be satisfied, we need:

$$\tau_k \in (0, \tau_{k,2}) = \left(0, \frac{1}{L_k + \sqrt{L_k^2 + 2(\beta/(1-c))\|\boldsymbol{A}\|^2}}\right), \forall k. \tag{C.16}$$

The lower bound for τ_k trivially holds, and for the upper bound, we make the following observation:

$$\begin{aligned} L_k + \sqrt{L_k^2 + 2(\beta/(1-c))\|\boldsymbol{A}\|^2} &= \frac{2\left[\sqrt{L_k^2} + \sqrt{L_k^2 + 2(\beta/(1-c))\|\boldsymbol{A}\|^2}\right]}{2} \\ &\overset{\text{Jensen}}{<} 2\sqrt{\frac{2L_k^2 + 2(\beta/(1-c))\|\boldsymbol{A}\|^2}{2}} \\ &= 2\sqrt{L_k^2 + (\beta/(1-c))\|\boldsymbol{A}\|^2}. \end{aligned}$$

Here Jensen's inequality holds strictly because function $\sqrt{\cdot}$ is strictly concave and $L_k^2 \neq L_k^2 + 2\|\boldsymbol{A}\|^2\beta$. Thus, we obtain:

$$0 < \tau_k \leq \frac{1}{2\sqrt{L_k^2 + \|\boldsymbol{A}\|^2\beta}} < \frac{1}{L_k + \sqrt{L_k^2 + 2\beta\|\boldsymbol{A}\|^2}} = \tau_{k,2} \ \forall k.$$

It follows that we can find an $\eta_k \in \left(\frac{\beta\tau_k\|\boldsymbol{A}\|}{1-c}, \frac{1-2\tau_k L_k}{2\tau_k\|\boldsymbol{A}\|}\right), \forall k$, which implies that conditions (C.14) can always be satisfied. This concludes the proof. □

C.2 Proof of Theorem 4.1

Theorem 4.1. *Consider APDA along with Assumptions 4.1 and 4.2, and let* $(\boldsymbol{x}^\star, \boldsymbol{y}^\star) \in \mathcal{X} \times \mathcal{Y}$ *be a saddle point of problem* (4.2)*. Then,*

1) ***Boundedness.*** *The sequence* $\{(\boldsymbol{x}_k, \boldsymbol{y}_k)\}$ *is bounded. Specifically, for all* k*,*

$$\left\| \boldsymbol{x}_k - \boldsymbol{x}^\star \right\|^2 + \left\| \boldsymbol{y}_k - \boldsymbol{y}^\star \right\|^2 \leq M,$$

where $M := \| \boldsymbol{x}_1 - \boldsymbol{x}^\star \|^2 + \frac{1}{\beta} \left\| \boldsymbol{y}_1 - \boldsymbol{y}^\star \right\|^2 + \frac{1}{2} \| \boldsymbol{x}_1 - \boldsymbol{x}_0 \|^2 < \infty$.

2) ***Convergence to a saddle point.*** *The sequence* $\{(\boldsymbol{x}_k, \boldsymbol{y}_k)\}$ *converges to a saddle point of* (4.2)*.*

3) ***Ergodic convergence.*** *Let* $S_k := \sum_{i=1}^{k} \tau_i$, $\overline{\boldsymbol{x}}_k := \frac{1}{S_k} \left(\tau_k (1+\theta_k) \boldsymbol{x}_k + \sum_{i=1}^{k-1} \left(\tau_i (1+\theta_i) - \tau_{i+1} \theta_{i+1} \right) \boldsymbol{x}_i \right)$ *and* $\overline{\boldsymbol{y}}_k := \frac{1}{S_k} \sum_{i=1}^{k} \tau_i \boldsymbol{y}_{i+1}$*. Then, for any bounded* $B_1 \times B_2 \in \mathcal{X} \times \mathcal{Y}$ *and for all* k*,*

$$\mathcal{G}_{B_1 \times B_2}(\overline{\boldsymbol{x}}_k, \overline{\boldsymbol{y}}_k) \;\leq\; \frac{M(B_1, B_2) \sqrt{L^2 + (\beta/(1-c)) \| \boldsymbol{A} \|^2}}{k},$$

where L *is the Lipschitz constant of* ∇f *over the compact set* $\overline{conv}(\{\boldsymbol{x}^\star, \boldsymbol{x}_0, \boldsymbol{x}_1, \ldots\})$ *and* $M(B_1, B_2) = \sup_{(\boldsymbol{x}, \boldsymbol{y}) \in B_1 \times B_2} \| \boldsymbol{x}_1 - \boldsymbol{x} \|^2 + \frac{1}{\beta} \left\| \boldsymbol{y}_1 - \boldsymbol{x} \right\|^2 + \frac{1}{2} \| \boldsymbol{x}_1 - \boldsymbol{x}_0 \|^2$.

Proof. 1) Sequence boundedness. Using the inequality of Lemma (4.1) with $(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x}^\star, \boldsymbol{y}^\star)$ and the fact that $\tau_k L_k < \frac{1}{2}$, $\forall k$, unrolling it over the iterations and rearranging the terms we obtain:

$$\begin{aligned}
&\left\| \boldsymbol{x}_{k+1} - \boldsymbol{x}^\star \right\|^2 + \frac{1}{\beta} \left\| \boldsymbol{y}_{k+1} - \boldsymbol{y}^\star \right\|^2 + \left(1 - \eta_k \tau_k \| \boldsymbol{A} \| - \tau_k L_k\right) \| \boldsymbol{x}_{k+1} - \boldsymbol{x}_k \|^2 \\
&+ \sum_{i=1}^{k-1} \left(\frac{1}{2} - \eta_i \tau_i \| \boldsymbol{A} \| - \tau_i L_i \right) \| \boldsymbol{x}_{i+1} - \boldsymbol{x}_i \|^2 + \frac{c}{\beta} \sum_{i=1}^{k} \left\| \boldsymbol{y}_{i+1} - \boldsymbol{y}_i \right\|^2 + 2\tau_k (1+\theta_k) P_{\boldsymbol{x}^\star, \boldsymbol{y}^\star}(\boldsymbol{x}_k) \\
&\quad + 2 \sum_{i=2}^{k-1} \left(\tau_i (1+\theta_i) - \tau_{i+1} \theta_{i+1} \right) P_{\boldsymbol{x}^\star, \boldsymbol{y}^\star}(\boldsymbol{x}_i) + 2 \sum_{i=1}^{k} \tau_i D_{\boldsymbol{x}^\star, \boldsymbol{y}^\star}(\boldsymbol{y}_{i+1}) \\
&\qquad \leq \left\| \boldsymbol{x}_1 - \boldsymbol{x}^\star \right\|^2 + \frac{1}{\beta} \left\| \boldsymbol{y}_1 - \boldsymbol{y}^\star \right\|^2 + \frac{1}{2} \| \boldsymbol{x}_1 - \boldsymbol{x}_0 \|^2 + 2\tau_1 \theta_1 P_{\boldsymbol{x}^\star, \boldsymbol{y}^\star}(\boldsymbol{x}_0). \qquad \text{(C.17)}
\end{aligned}$$

All the terms on the left hand-side of (C.17) are non-negative:

$$\begin{cases} \frac{1}{\beta} - \frac{\tau_k \| \boldsymbol{A} \|}{\eta_i} > \frac{c}{\beta} > 0, \ \forall i, & \text{(by Lemma 4.1)} \\ \frac{1}{2} - \eta_i \tau_i \| \boldsymbol{A} \| - \tau_i L_i > 0, \ \forall i, & \text{(by Lemma 4.1)} \\ \tau_{i+1}\theta_{i+1} \leq \tau_i \sqrt{1+\theta_i}\, \theta_{i+1} \leq \tau_i (1+\theta_i), & \text{(by stepsize update rule)} \\ P_{\boldsymbol{x}^\star, \boldsymbol{y}^\star}(\boldsymbol{x}) \geq 0, \ D_{\boldsymbol{x}^\star, \boldsymbol{y}^\star}(\boldsymbol{y}) \geq 0 \ \ \forall \boldsymbol{x}, \boldsymbol{y}. & \text{(by the saddle point property)} \end{cases}$$

Also, by our parameter setup, we have that $\theta_1 = 0$. Consequently, it holds that:

$$\left\| \boldsymbol{x}_{k+1} - \boldsymbol{x}^\star \right\|^2 + \frac{1}{\beta} \left\| \boldsymbol{y}_{k+1} - \boldsymbol{y}^\star \right\|^2 \leq M < \infty \ \forall k,$$

where $M := \| \boldsymbol{x}_1 - \boldsymbol{x}^\star \|^2 + \frac{1}{\beta} \left\| \boldsymbol{y}_1 - \boldsymbol{y}^\star \right\|^2 + \frac{1}{2} \| \boldsymbol{x}_1 - \boldsymbol{x}_0 \|^2$, which implies that the sequence is bounded.

We make the following remarks which will be useful for the remainder of the theorem's proof:

- Boundedness of $\{\boldsymbol{x}_k\}$ together with the local Lipschitz continuity of ∇f from Assumption 4.1 implies that there exists $L > 0$ such that f is L-smooth over $\overline{\text{conv}}(\{\boldsymbol{x}^\star, \boldsymbol{x}_0, \boldsymbol{x}_1, \ldots\})$. Furthermore, $L \geq L_k \ \forall k$.

- A consequence of the prior point is that τ_k has a uniform and positive lower bound. By the definition of APDA, it holds that:

$$\tau_1 = \frac{1}{2\sqrt{L_1^2 + (\beta/(1-c)) \| \boldsymbol{A} \|^2}} \geq \frac{1}{2\sqrt{L^2 + (\beta/(1-c)) \| \boldsymbol{A} \|^2}}$$

  and, from the definition of τ_k, it is straightforward to see that at every iteration, we either explicitly increase τ_k relative to τ_{k-1} or otherwise set it to an expression dictated by the local smoothness constant L_k. Thus it holds that:

$$\tau_k \geq \frac{1}{2\sqrt{L^2 + (\beta/(1-c)) \| \boldsymbol{A} \|^2}}, \ \forall k. \tag{C.18}$$

- Furthermore, the existence of L guarantees that $\tau_k L_k$ can have a tighter upper bound than the $1/2$ shown before, as follows:

$$\begin{aligned} \tau_k L_k &\leq \frac{L_k}{2\sqrt{L_k^2 + (\beta/(1-c)) \| \boldsymbol{A} \|^2}} \\ &\leq \frac{L}{2\sqrt{L^2 + (\beta/(1-c)) \| \boldsymbol{A} \|^2}}, . \end{aligned} \tag{C.19}$$

where we used the fact that $z \mapsto \frac{z}{2\sqrt{z^2+a}}$, $a > 0$ is an increasing function.

- Finally, due to the point above, we can uniformly lower bound the coefficients of terms $\| \boldsymbol{x}_{k+1} - \boldsymbol{x}_k \|^2$ on the LHS of (C.17), and thus obtain:

$$\frac{1}{2}\left(1 - \frac{L}{\sqrt{L^2 + (\beta/(1-c)) \| \boldsymbol{A} \|^2}}\right) \sum_{i=1}^{k-1} \| \boldsymbol{x}_i - \boldsymbol{x}_{i+1} \|^2 + \frac{c}{\beta} \sum_{i=1}^{k} \left\| \boldsymbol{y}_{i+1} - \boldsymbol{y}_i \right\|^2 \leq M,$$

which conveniently ensures that:

$$\begin{cases} \lim\limits_{k\to\infty} \| \boldsymbol{x}_k - \boldsymbol{x}_{k-1} \|^2 = 0, \\ \lim\limits_{k\to\infty} \left\| \boldsymbol{y}_k - \boldsymbol{y}_{k-1} \right\|^2 = 0. \end{cases} \tag{C.20}$$

2) Convergence to a saddle point. Let $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$ be an arbitrary cluster point of the sequence $\{(\boldsymbol{x}_k, \boldsymbol{y}_k)\}$. Since we have shown that the sequence is bounded, then there must exist a subsequence $\{(\boldsymbol{x}_{k_i}, \boldsymbol{y}_{k_i})\}$, such that $\lim_{i\to\infty}(\boldsymbol{x}_{k_i}, \boldsymbol{y}_{k_i}) = (\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$. We wish to prove that $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$ is a saddle point of (4.2).

More precisely, we wish to prove that $P_{\hat{\boldsymbol{x}},\hat{\boldsymbol{y}}}(\boldsymbol{x}) \geq 0$ and $D_{\hat{\boldsymbol{x}},\hat{\boldsymbol{y}}}(\boldsymbol{y}) \geq 0$ for $\forall \boldsymbol{x}, \boldsymbol{y}$, respectively. For convenience, we remind the reader of the definitions of these two quantities:

$$\begin{aligned} P_{\hat{\boldsymbol{x}},\hat{\boldsymbol{y}}}(\boldsymbol{x}) &= f(\boldsymbol{x}) - f(\hat{\boldsymbol{x}}) + \langle \boldsymbol{A}(\boldsymbol{x} - \hat{\boldsymbol{x}}), \hat{\boldsymbol{y}} \rangle, \\ D_{\hat{\boldsymbol{x}},\hat{\boldsymbol{y}}}(\boldsymbol{y}) &= g^*(\boldsymbol{y}) - g^*(\hat{\boldsymbol{y}}) - \langle \boldsymbol{A}\hat{\boldsymbol{x}}, \boldsymbol{y} - \hat{\boldsymbol{y}} \rangle. \end{aligned}$$

We start with $P_{\hat{\boldsymbol{x}},\hat{\boldsymbol{y}}}(\boldsymbol{x})$:

$$\begin{aligned} P_{\hat{\boldsymbol{x}},\hat{\boldsymbol{y}}}(\boldsymbol{x}) &= f(\boldsymbol{x}) - f(\hat{\boldsymbol{x}}) + \langle \boldsymbol{A}(\boldsymbol{x} - \hat{\boldsymbol{x}}), \hat{\boldsymbol{y}} \rangle \\ &= \lim_{i\to\infty} f(\boldsymbol{x}) - f(\boldsymbol{x}_{k_i}) + \langle \boldsymbol{A}(\boldsymbol{x} - \boldsymbol{x}_{k_i}), \boldsymbol{y}_{k_i} \rangle && \text{(Continuity of } f\text{)} \\ &\geq \lim_{i\to\infty} \langle \nabla f(\boldsymbol{x}_{k_i}) + \boldsymbol{A}^\top \boldsymbol{y}_{k_i+1}, \boldsymbol{x} - \boldsymbol{x}_{k_i} \rangle + \langle \boldsymbol{A}^\top (\boldsymbol{y}_{k_i} - \boldsymbol{y}_{k_i+1}), \boldsymbol{x} - \boldsymbol{x}_{k_i} \rangle && \text{(Convexity of } f\text{)} \\ &= \lim_{i\to\infty} \langle \frac{\boldsymbol{x}_{k_i+1} - \boldsymbol{x}_{k_i}}{\tau_{k_i}}, \boldsymbol{x} - \boldsymbol{x}_{k_i} \rangle + \langle \boldsymbol{A}^\top (\boldsymbol{y}_{k_i} - \boldsymbol{y}_{k_i+1}), \boldsymbol{x} - \boldsymbol{x}_{k_i} \rangle && \text{(Primal update rule)} \\ &= 0. && \text{(By (C.18), (C.20))} \end{aligned}$$

Showing the analogous result for $D_{\hat{\boldsymbol{x}},\hat{\boldsymbol{y}}}(\boldsymbol{y})$ relies on similar arguments, with the additional requirement that θ_k is uniformly upper bounded. From the update rule of τ_k we have:

$$\theta_k = \frac{\tau_k}{\tau_{k-1}} \leq \sqrt{1 + \theta_{k-1}} \implies \theta_k \leq \sqrt{1 + \ldots + \sqrt{1 + \theta_2}} \leq \underbrace{\sqrt{1 + \ldots + \sqrt{1 + 1}}}_{k-2 \text{ times}} \leq 2, \tag{C.21}$$

where the second to last inequality comes from the way APDA's first two iterations are set up.

Therefore, we have that $\forall \boldsymbol{y} \in \mathcal{Y}$:

$$
\begin{aligned}
D_{\hat{\boldsymbol{x}},\hat{\boldsymbol{y}}}(\boldsymbol{y}) &= g^*(\boldsymbol{y}) - g^*(\hat{\boldsymbol{y}}) - \langle \boldsymbol{A}\hat{\boldsymbol{x}}, \boldsymbol{y} - \hat{\boldsymbol{y}} \rangle \\
&\geq g^*(\boldsymbol{y}) - \liminf_{i\to\infty} g^*(\boldsymbol{y}_{k_i}) - \langle \boldsymbol{A} \liminf_{i\to\infty} \boldsymbol{x}_{k_i}, \boldsymbol{y} - \liminf_{i\to\infty} \boldsymbol{y}_{k_i} \rangle && \text{(l.s.c. of } g^*\text{)} \\
&= \limsup_{i\to\infty} g^*(\boldsymbol{y}) - g^*(\boldsymbol{y}_{k_i}) - \langle \boldsymbol{A}\boldsymbol{x}_{k_i}, \boldsymbol{y} - \boldsymbol{y}_{k_i} \rangle \\
&\geq \limsup_{i\to\infty} \langle \frac{\boldsymbol{y}_{k_i-1} - \boldsymbol{y}_{k_i}}{\sigma_{k_i-1}}, \boldsymbol{y} - \boldsymbol{y}_{k_i} \rangle + \langle \boldsymbol{A}(\widetilde{\boldsymbol{x}}_{k_i-1} - \boldsymbol{x}_{k_i}), \boldsymbol{y} - \boldsymbol{y}_{k_i} \rangle && \text{(Poperty (C.9))} \\
&= \limsup_{i\to\infty} \langle \frac{\boldsymbol{y}_{k_i-1} - \boldsymbol{y}_{k_i}}{\beta\tau_{k_i-1}}, \boldsymbol{y} - \boldsymbol{y}_{k_i} \rangle + \langle \boldsymbol{A}\left[\boldsymbol{x}_{k_i-1} - \boldsymbol{x}_{k_i} + \theta_{k_i-1}(\boldsymbol{x}_{k_i-1} - \boldsymbol{x}_{k_i-2})\right], \boldsymbol{y} - \boldsymbol{y}_{k_i} \rangle \\
&= 0. && \text{(By (C.18), (C.20), (C.21))}
\end{aligned}
$$

3) Gap rate. Unrolling the inequality of Lemma 4.1 for some $(\boldsymbol{x}, \boldsymbol{y}) \in B_1 \times B_2$, we obtain:

$$
\begin{aligned}
&\|\boldsymbol{x}_{k+1} - \boldsymbol{x}\|^2 + \frac{1}{\beta}\left\|\boldsymbol{y}_{k+1} - \boldsymbol{x}\right\|^2 + \left(1 - \eta_k\tau_k\|\boldsymbol{A}\| - \tau_k L_k\right)\|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k\|^2 \\
&+ \sum_{i=1}^{k-1}\left(\frac{1}{2} - \eta_i\tau_i\|\boldsymbol{A}\| - \tau_i L_i\right)\|\boldsymbol{x}_{i+1} - \boldsymbol{x}_i\|^2 + \frac{c}{\beta}\sum_{i=1}^{k}\left\|\boldsymbol{y}_{i+1} - \boldsymbol{y}_i\right\|^2 + 2\tau_k(1+\theta_k)P_{\boldsymbol{x},\boldsymbol{y}}(\boldsymbol{x}_k) \\
&\quad + 2\sum_{i=2}^{k-1}\left(\tau_i(1+\theta_i) - \tau_{i+1}\theta_{i+1}\right)P_{\boldsymbol{x},\boldsymbol{y}}(\boldsymbol{x}_i) + 2\sum_{i=1}^{k}\tau_i D_{\boldsymbol{x},\boldsymbol{y}}(\boldsymbol{y}_{i+1}) \\
&\quad \leq \|\boldsymbol{x}_1 - \boldsymbol{x}\|^2 + \frac{1}{\beta}\left\|\boldsymbol{y}_1 - \boldsymbol{y}\right\|^2 + \frac{1}{2}\|\boldsymbol{x}_1 - \boldsymbol{x}_0\|^2. \qquad \text{(C.22)}
\end{aligned}
$$

First, note that due to $\theta_1 = 0$, the following holds:

$$
\tau_k(1+\theta_k) + \sum_{i=1}^{k-1}\left(\tau_i(1+\theta_i) - \tau_{i+1}\theta_{i+1}\right) = \sum_{i=1}^{k}\tau_i =: S_k.
$$

Second, since all the terms on the LHS of (C.22) except those involving $P_{\boldsymbol{x},\boldsymbol{y}}(\cdot)$ and $D_{\boldsymbol{x},\boldsymbol{y}}(\cdot)$ are non-negative and, for fixed $(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{X} \times \mathcal{Y}$ the functions $P_{\boldsymbol{x},\boldsymbol{y}}(\cdot)$ and $D_{\boldsymbol{x},\boldsymbol{y}}(\cdot)$ are convex, we have:

$$
\begin{aligned}
&2S_k\left(P_{\boldsymbol{x},\boldsymbol{y}}(\overline{\boldsymbol{x}}_k) + D_{\boldsymbol{x},\boldsymbol{y}}(\overline{\boldsymbol{y}}_k)\right) \\
&\leq 2\tau_k(1+\theta_k)P_{\boldsymbol{x},\boldsymbol{y}}(\boldsymbol{x}_k) + 2\sum_{i=2}^{k-1}\left(\tau_i(1+\theta_i) - \tau_{i+1}\theta_{i+1}\right)P_{\boldsymbol{x},\boldsymbol{y}}(\boldsymbol{x}_i) + 2\sum_{i=1}^{k}\tau_i D_{\boldsymbol{x},\boldsymbol{y}}(\boldsymbol{y}_{i+1}) \\
&\leq \|\boldsymbol{x}_1 - \boldsymbol{x}\|^2 + \frac{1}{\beta}\left\|\boldsymbol{y}_1 - \boldsymbol{y}\right\|^2 + \frac{1}{2}\|\boldsymbol{x}_1 - \boldsymbol{x}_0\|^2. \qquad \text{(C.23)}
\end{aligned}
$$

Lastly, since $\tau_k \geq \frac{1}{2\sqrt{L^2 + (\beta/(1-c))\|\boldsymbol{A}\|^2}}$, $\forall k$, we have that $S_k \geq \frac{k}{2\sqrt{L^2 + (\beta/(1-c))\|\boldsymbol{A}\|^2}}$ and the rate for the restricted gap is:

$$
\begin{aligned}
&\mathcal{G}_{B_1\times B_2}(\overline{\boldsymbol{x}}_k,\overline{\boldsymbol{y}}_k)\\
&\quad=\sup_{(\boldsymbol{x},\boldsymbol{y})\in B_1\times B_2} P_{\boldsymbol{x},\boldsymbol{y}}(\overline{\boldsymbol{x}}_k)+D_{\boldsymbol{x},\boldsymbol{y}}(\overline{\boldsymbol{y}}_k)\\
&\quad\le\sup_{(\boldsymbol{x},\boldsymbol{y})\in B_1\times B_2} \frac{\left(\|\boldsymbol{x}_1-\boldsymbol{x}\|^2+\frac{1}{\beta}\left\|\boldsymbol{y}_1-\boldsymbol{y}\right\|^2+\frac{1}{2}\|\boldsymbol{x}_1-\boldsymbol{x}_0\|^2\right)\sqrt{L^2+(\beta/(1-c))\|\boldsymbol{A}\|^2}}{k}\\
&\quad=\frac{M(B_1,B_2)\sqrt{L^2+(\beta/(1-c))\|\boldsymbol{A}\|^2}}{k},
\end{aligned}
$$

which concludes the proof of the theorem. □

C.3 Proof of Theorem 4.2

Before proving the result of Theorem 4.2, a few remarks are in order. First, the boundedness result of Theorem 4.1 point 1) also holds for constant $c = 0$, since this constant was required only for proving convergence to a saddle point in point 2) of the theorem. Second, taking a stepsize smaller than the originally considered τ_k will not change the validity of Lemma 4.1 or the boundedness result of Theorem 4.1, as it remains within the interval given in (C.16).

Consequently, for studying APDA under the additional Assumption 4.3 we can simplify the stepsize expression by taking $c = 0$, since now we will prove convergence of the iterates directly by using the strong convexity and full row-rank assumptions. Specifically, we consider τ_k as defined in (4.8), which is smaller than the one originally considered and, due to the above remarks, it ensures that APDA produces a bounded sequence. It follows that, under the local smoothness and local strong convexity assumptions, there exist constant L and μ such that f is L-smooth and μ-strongly convex over $\overline{\text{conv}}(\{\boldsymbol{x}^\star, \boldsymbol{x}_0, \boldsymbol{x}_1, \ldots\})$. This observation suffices to show linear convergence in Theorem 4.2.

Theorem 4.2. *Consider APDA along with Assumptions 4.1, 4.2 and 4.3. Let* $(\boldsymbol{x}^\star, \boldsymbol{y}^\star) \in \mathcal{X} \times \mathcal{Y}$ *be a saddle point of problem* (4.2). *Furthermore, let* τ_k *be defined by* (4.8) *and let* $s := \sqrt{4L^2 + \beta \| \boldsymbol{A} \|^2}$ *and* $t := \sqrt{4\mu^2 + \beta \| \boldsymbol{A} \|^2}$, *where* μ, L *are the strong convexity and smoothness constants of* f *over the compact set* $\overline{conv}(\{\boldsymbol{x}^\star, \boldsymbol{x}_0, \boldsymbol{x}_1, \ldots\})$. *Then, for all* k,

$$\left\| \boldsymbol{x}_k - \boldsymbol{x}^\star \right\|^2 + \frac{1}{\beta} \left\| \boldsymbol{y}_k - \boldsymbol{y}^\star \right\|^2 \leq \left(1 - \min\{p, q, r\}\right)^k M,$$

where the rate constants are given by

$$p = \frac{1}{2}, \quad q = \frac{\mu}{4s}, \quad r = \frac{\beta \sigma_{\min}^2(\boldsymbol{A}) \mu}{\beta \sigma_{\min}^2(\boldsymbol{A}) \mu + 8 s^2 t + 4 L^2 s},$$

and $M = \| \boldsymbol{x}_2 - \boldsymbol{x}^\star \|^2 + \left(\frac{1}{\beta} + T\right) \left\| \boldsymbol{y}_2 - \boldsymbol{y}^\star \right\|^2 + \frac{1}{2} \| \boldsymbol{x}_2 - \boldsymbol{x}_1 \|^2 + 2\tau_1 P_{\boldsymbol{x}^\star, \boldsymbol{y}^\star}(\boldsymbol{x}_1)$, $T = \dfrac{\sigma_{\min}^2(\boldsymbol{A}) \mu}{8 s^2 t + 4 L^2 s}$, *with* $\sigma_{\min}(\boldsymbol{A})$ *representing the smallest singular value of* $\boldsymbol{A}$.

Proof. The outline of the proof is first arriving at a strengthened version of the inequality in Lemma 4.1, then showing that the inequality expresses a contraction.

Since this new stepsize still ensures the boundedness result of Theorem 4.1, there exist μ and L such that f is μ-strongly convex and L-Lipschitz smooth over the compact set $\overline{\text{conv}}(\{\boldsymbol{x}^\star, \boldsymbol{x}_0, \boldsymbol{x}_1, \ldots\})$. From these properties, it follows that, for all k:

$$\begin{aligned} 2\tau_k \langle \nabla f(\boldsymbol{x}_k), \boldsymbol{x}^\star - \boldsymbol{x}_k \rangle &\leq 2\tau_k \left(f(\boldsymbol{x}^\star) - f(\boldsymbol{x}_k) \right) - \mu \tau_k \left\| \boldsymbol{x}_k - \boldsymbol{x}^\star \right\|^2, \\ 2\tau_k \langle \nabla f(\boldsymbol{x}_k), \boldsymbol{x}^\star - \boldsymbol{x}_k \rangle &\leq 2\tau_k \left(f(\boldsymbol{x}^\star) - f(\boldsymbol{x}_k) \right) - \frac{\tau_k}{L} \left\| \nabla f(\boldsymbol{x}_k) - \nabla f(\boldsymbol{x}^\star) \right\|^2. \end{aligned}$$

Summing these two inequalities and dividing by 2, we obtain a stronger version of equation (C.2):

$$-2\tau_k\langle\nabla f(\boldsymbol{x}_k)+\boldsymbol{A}^\top\boldsymbol{y}_{k+1},\boldsymbol{x}_k-\boldsymbol{x}^\star\rangle\leq 2\tau_k\big(f(\boldsymbol{x}^\star)-f(\boldsymbol{x}_k)\big)-\frac{\tau_k\mu}{2}\left\|\boldsymbol{x}_k-\boldsymbol{x}^\star\right\|^2 - \frac{\tau_k}{2L}\left\|\nabla f(\boldsymbol{x}_k)-\nabla f(\boldsymbol{x}^\star)\right\|^2+2\tau_k\langle\boldsymbol{A}(\boldsymbol{x}^\star-\boldsymbol{x}_k),\boldsymbol{y}_{k+1}\rangle. \tag{C.24}$$

We further bound the term $\left\|\nabla f(\boldsymbol{x}_k)-\nabla f(\boldsymbol{x}^\star)\right\|^2$ in (C.24):

$$\begin{aligned}
\left\|\nabla f(\boldsymbol{x}^\star)-\nabla f(\boldsymbol{x}_k)\right\|^2 &= \left\|\boldsymbol{A}^\top(\boldsymbol{y}_{k+1}-\boldsymbol{y}^\star)-\frac{\boldsymbol{x}_k-\boldsymbol{x}_{k+1}}{\tau_k}\right\|^2 && \text{(C.25)}\\
&\geq \left\|\boldsymbol{A}^\top(\boldsymbol{y}_{k+1}-\boldsymbol{y}^\star)\right\|^2+\frac{1}{\tau_k^2}\left\|\boldsymbol{x}_{k+1}-\boldsymbol{x}_k\right\|^2 \\
&\quad -\frac{2}{\tau_k}\left\|\boldsymbol{A}^\top(\boldsymbol{y}_{k+1}-\boldsymbol{y}^\star)\right\|\left\|\boldsymbol{x}_{k+1}-\boldsymbol{x}_k\right\| && \text{(C.26)}\\
&\geq \left\|\boldsymbol{A}^\top(\boldsymbol{y}_{k+1}-\boldsymbol{y}^\star)\right\|^2+\frac{1}{\tau_k^2}\left\|\boldsymbol{x}_{k+1}-\boldsymbol{x}_k\right\|^2 \\
&\quad -\left(\frac{1}{\xi+1}\left\|\boldsymbol{A}^\top(\boldsymbol{y}_{k+1}-\boldsymbol{y}^\star)\right\|^2+\frac{\xi+1}{\tau_k^2}\left\|\boldsymbol{x}_{k+1}-\boldsymbol{x}_k\right\|^2\right) && \text{(C.27)}\\
&\geq \frac{\xi\sigma_{\min}^2(\boldsymbol{A})}{\xi+1}\left\|\boldsymbol{y}_{k+1}-\boldsymbol{y}^\star\right\|^2-\frac{\xi}{\tau_k^2}\left\|\boldsymbol{x}_{k+1}-\boldsymbol{x}_k\right\|^2, && \text{(C.28)}
\end{aligned}$$

where line (C.25) comes from the primal iterate update rule and the optimality condition (4.4); line (C.26) comes from developing the square and applying Cauchy-Schwarz; line (C.27) comes from applying Young's inequality with constant $1+\xi$, where $\xi>0$; line (C.28) comes from the assumption of $\boldsymbol{A}$ having full-row rank, which implies that $\left\|\boldsymbol{A}^\top(\boldsymbol{y}_{k+1}-\boldsymbol{y}^\star)\right\|^2\geq\sigma_{\min}^2(\boldsymbol{A})\left\|\boldsymbol{y}_{k+1}-\boldsymbol{y}^\star\right\|^2$.

Finally, setting $\xi=2\tau_k^2L_kL$ we obtain that:

$$\begin{aligned}
-2\tau_k\langle\nabla f(\boldsymbol{x}_k)+\boldsymbol{A}^\top\boldsymbol{y}_{k+1},\boldsymbol{x}_k-\boldsymbol{x}^\star\rangle\leq{}& 2\tau_k\big(f(\boldsymbol{x}^\star)-f(\boldsymbol{x}_k)\big)-\frac{\tau_k\mu}{2}\left\|\boldsymbol{x}_k-\boldsymbol{x}^\star\right\|^2 \\
&-\frac{\tau_k^3L_k\sigma_{\min}^2(\boldsymbol{A})}{1+2\tau_k^2L_kL}\left\|\boldsymbol{y}_{k+1}-\boldsymbol{y}^\star\right\|^2+\tau_kL_k\left\|\boldsymbol{x}_{k+1}-\boldsymbol{x}_k\right\|^2 \\
&+2\tau_k\langle\boldsymbol{A}(\boldsymbol{x}^\star-\boldsymbol{x}_k),\boldsymbol{y}_{k+1}\rangle. \qquad \text{(C.29)}
\end{aligned}$$

Replacing inequality (C.2) with inequality (C.29) in the proof of Lemma 4.1 and keeping everything else identical, we obtain a strengthened version of Lemma's 4.1 result:

$$\left\|\boldsymbol{x}_{k+1}-\boldsymbol{x}^\star\right\|^2+\left(\frac{1}{\beta}+\frac{\tau_k^3L_k\sigma_{\min}^2(\boldsymbol{A})}{1+2\tau_k^2L_kL}\right)\left\|\boldsymbol{y}_{k+1}-\boldsymbol{y}^\star\right\|^2+\left(1-\eta_k\tau_k\left\|\boldsymbol{A}\right\|-2\tau_kL_k\right)$$
$$\left\|\boldsymbol{x}_{k+1}-\boldsymbol{x}_k\right\|^2+\frac{\eta_k-\tau_k\beta\left\|\boldsymbol{A}\right\|}{\beta\eta_k}\left\|\boldsymbol{y}_{k+1}-\boldsymbol{y}_k\right\|^2+2\tau_k(1+\theta_k)P_{\boldsymbol{x}^\star,\boldsymbol{y}^\star}(\boldsymbol{x}_k)+2\tau_kD_{\boldsymbol{x}^\star,\boldsymbol{y}^\star}(\boldsymbol{y}_{k+1})$$

$$\leq \left(1 - \frac{\mu\tau_k}{2}\right) \left\| \boldsymbol{x}_k - \boldsymbol{x}^\star \right\|^2 + \frac{1}{\beta} \left\| \boldsymbol{y}_k - \boldsymbol{y}^\star \right\|^2 + \tau_k L_k \left\| \boldsymbol{x}_k - \boldsymbol{x}_{k-1} \right\|^2 + 2\tau_k \theta_k P_{\boldsymbol{x}^\star, \boldsymbol{y}^\star}(\boldsymbol{x}_{k-1}). \tag{C.30}$$

In order to show that this is, in fact, a contraction, we note a few properties of the terms in (C.30):

a) It holds that $1 - \eta_k \tau_k \left\| \boldsymbol{A} \right\| - 2\tau_k L_k > 1/2$ and $\frac{\eta_k - \tau_k \beta \left\| \boldsymbol{A} \right\|}{\beta \eta_k} > 0$ since:

$$\begin{cases} 1 - \eta_k \tau_k \left\| \boldsymbol{A} \right\| - 2\tau_k L_k > 1/2 \\ \frac{\eta_k - \tau_k \beta \left\| \boldsymbol{A} \right\|}{\beta \eta_k} > 0 \end{cases} \iff \begin{cases} \eta_k < \frac{1}{2\tau_k \left\| \boldsymbol{A} \right\|} - \frac{2L_k}{\left\| \boldsymbol{A} \right\|} \\ \eta_k > \tau_k \beta \left\| \boldsymbol{A} \right\| \end{cases}$$

$$\iff 2\beta \left\| \boldsymbol{A} \right\|^2 \tau_k^2 + 4L_k \tau_k - 1 < 0,$$

which holds for any $\tau_k \in \left(0, \frac{1}{2L_k + \sqrt{4L_k^2 + 2\beta \left\| \boldsymbol{A} \right\|^2}} \right)$. Our choice of τ_k belongs to this interval and therefore ensures the stated properties;

b) It holds that $\tau_k L_k < 1/4$, by the same observation as that in (C.13) but with a different limit constant given by the new stepsize;

c) It holds that $\tau_k \theta_k \leq \tau_{k-1} \sqrt{1 + \theta_{k-1}/2}\, \theta_k \leq \tau_{k-1}(1 + \theta_{k-1}/2)$, by the definitions of τ_k and θ_k;

d) It holds that:

$$\frac{1}{2\sqrt{4L^2 + \beta \left\| \boldsymbol{A} \right\|^2}} \leq \tau_k \leq \frac{1}{2\sqrt{4\mu^2 + \beta \left\| \boldsymbol{A} \right\|^2}}, \tag{C.31}$$

by the existence of μ and L over $\overline{\text{conv}}(\{\boldsymbol{x}^\star, \boldsymbol{x}_0, \boldsymbol{x}_1, \ldots\})$ and a similar argument to that in (C.18), plus the fact that under strong convexity $\left\| \nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y}) \right\| \geq \mu \left\| \boldsymbol{x} - \boldsymbol{y} \right\|$;

e) It holds that:

$$\frac{\mu}{2\sqrt{4\mu^2 + \beta \left\| \boldsymbol{A} \right\|^2}} \leq \tau_k L_k \leq \frac{L}{2\sqrt{4L^2 + \beta \left\| \boldsymbol{A} \right\|^2}}, \tag{C.32}$$

by a similar argument to that in (C.19).

Using properties a), b), c) in the list above and ignoring the positive terms on the LHS that do not have a correspondent on the RHS of (C.30), the main inequality becomes:

$$\left\| \boldsymbol{x}_{k+1} - \boldsymbol{x}^\star \right\|^2 + \left(\frac{1}{\beta} + T\right) \left\| \boldsymbol{y}_{k+1} - \boldsymbol{y}^\star \right\|^2 + \frac{1}{2} \left\| \boldsymbol{x}_{k+1} - \boldsymbol{x}_k \right\|^2 + 2\tau_k (1 + \theta_k) P_{\boldsymbol{x}^\star, \boldsymbol{y}^\star}(\boldsymbol{x}_k)$$

$$\leq \left(1 - \frac{\mu\tau_k}{2}\right) \left\| \boldsymbol{x}_k - \boldsymbol{x}^\star \right\|^2 + \frac{1}{\beta} \left\| \boldsymbol{y}_k - \boldsymbol{y}^\star \right\|^2 + \frac{1}{2}\left(1 - \frac{1}{2}\right) \left\| \boldsymbol{x}_k - \boldsymbol{x}_{k-1} \right\|^2$$

$$+ 2\tau_{k-1}(1 + \theta_{k-1}/2) P_{\boldsymbol{x}^\star, \boldsymbol{y}^\star}(\boldsymbol{x}_{k-1}), \tag{C.33}$$

where T is given by:

$$\begin{aligned} T &:= \frac{\sigma_{\min}^2(\boldsymbol{A})\mu}{8s^2t + 4L^2s} \\ &= \frac{\sigma_{\min}^2(\boldsymbol{A})\mu}{8(4L^2 + \beta\|\boldsymbol{A}\|^2)\sqrt{4\mu^2 + \beta\|\boldsymbol{A}\|^2} + 4L^2\sqrt{4L^2 + \beta\|\boldsymbol{A}\|^2}} \\ &\leq \frac{\tau_k^3 L_k \sigma_{\min}^2(\boldsymbol{A})}{1 + 2\tau_k^2 L_k L}, \qquad \text{(by d) and e) above)} \end{aligned}$$

where we used the definitions of $s = \sqrt{4L^2 + \beta\|\boldsymbol{A}\|^2}$ and $t = \sqrt{4\mu^2 + \beta\|\boldsymbol{A}\|^2}$ to simplify notations.

We thus have the following contractions in (C.33):

- For $\frac{1}{2}\|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k\|^2$ it is: $1 - \underbrace{\frac{1}{2}}_{=:p}$;
- For $\left(\frac{1}{\beta} + T\right)\left\|\boldsymbol{y}_{k+1} - \boldsymbol{y}^\star\right\|^2$ it is:

$$\begin{aligned} 1 + \frac{1}{1 + T\beta} &= 1 - \frac{T\beta}{1 + T\beta} \\ &= 1 - \underbrace{\frac{\beta\sigma_{\min}^2(\boldsymbol{A})\mu}{\sigma_{\min}^2(\boldsymbol{A})\mu + 8s^2t + 4L^2s}}_{=:r} \end{aligned}$$

- For $\|\boldsymbol{x}_{k+1} - \boldsymbol{x}^\star\|^2$ it is:

$$1 - \frac{\mu\tau_k}{2} \leq 1 - \underbrace{\frac{\mu}{4s}}_{=:q}$$

- For $2\tau_k(1 + \theta_k)P_{\boldsymbol{x}^\star,\boldsymbol{y}^\star}(\boldsymbol{x}_k)$ it is:

$$\begin{aligned} \frac{1 + \theta_{k-1}/2}{1 + \theta_{k-1}} &= 1 - \frac{\theta_{k-1}}{2(1 + \theta_{k-1})} \\ &\leq 1 - \frac{t}{s} \qquad \text{(By def. of } \theta_{k-1} \text{ and property 5.)} \end{aligned}$$

Note that for the latter two contractions above, it always holds that $\mu/(4s) < t/s$, so in the final bound, we can ignore the latter. Finally, denoting the LHS of inequality (C.33) as E_{k+1}, we have that:

$$E_{k+1} \leq \left(1 - \min\{p, q, r\}\right)^{k+1} M.$$

where $M = E_2$ and we used the fact that $\theta_1 = 0$. □

Bibliography

[1] Zeeshan Akhtar and Ketan Rajawat. ""Zeroth and First Order Stochastic Frank-Wolfe Algorithms for Constrained Optimization"". In: arXiv preprint arXiv:2107.06534 (2021). arXiv: 2107.06534 [math.OC] (cit. on p. 34).

[2] Abdo Y Alfakih, Amir Khandani, and Henry Wolkowicz. "Solving Euclidean distance matrix completion problems via semidefinite programming". In: *Computational optimization and applications* (1999) (cit. on p. 16).

[3] Zeyuan Allen-Zhu and Yang Yuan. "Improved SVRG for non-strongly-convex or sum-of-non-convex objectives". In: *International conference on machine learning*. 2016, pp. 1080–1089 (cit. on p. 18).

[4] Andreas Argyriou, Marco Signoretto, and Johan Suykens. "Hybrid conditional gradient-smoothing algorithms with applications to sparse and low rank regularization". In: *Regularization, Optimization, Kernels, and Support Vector Machines* (2014), pp. 53–82 (cit. on p. 46).

[5] Sanjeev Arora, Elad Hazan, and Satyen Kale. "Fast algorithms for approximate semidefinite programming using the multiplicative weights update method". In: *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS'05)*. IEEE. 2005, pp. 339–348 (cit. on p. 17).

[6] Sanjeev Arora, Satish Rao, and Umesh Vazirani. "Expander flows, geometric embeddings and graph partitioning". In: *Journal of the ACM (JACM)* 56.2 (2009), pp. 1–37 (cit. on p. 1).

[7] Sanjeev Arora, Satish Rao, and Umesh Vazirani. "Expander flows, geometric embeddings and graph partitioning". In: *Journal of the ACM (JACM)* 56.2 (2009), p. 5 (cit. on pp. 4, 10, 16, 30).

[8] Claudine Badue, Rânik Guidolini, Raphael Vivacqua Carneiro, Pedro Azevedo, Vinicius B Cardoso, Avelino Forechi, Luan Jesus, Rodrigo Berriel, Thiago M Paixao, Filipe Mutz, et al. "Self-driving cars: A survey". In: *Expert Systems with Applications* 165 (2021), p. 113816 (cit. on p. 2).

[9] Krishnakumar Balasubramanian and Saeed Ghadimi. "Zeroth-order (non)-convex stochastic optimization via conditional gradient and gradient updates". In: *Advances in Neural Information Processing Systems*. 2018, pp. 3455–3464 (cit. on p. 18).

[10] Cynthia Barnhart, Peter Belobaba, and Amedeo R Odoni. "Applications of operations research in the air transport industry". In: *Transportation science* 37.4 (2003), pp. 368–391 (cit. on p. 1).

[11] Solon Barocas, Moritz Hardt, and Arvind Narayanan. *Fairness and Machine Learning. fairmlbook.org, 2019*. 2018 (cit. on p. 3).

[12] Heinz H Bauschke, Patrick L Combettes, et al. *Convex analysis and monotone operator theory in Hilbert spaces*. Vol. 408. Springer (cit. on p. 21).

[13] Aqeel Ahmed Bazmi and Gholamreza Zahedi. "Sustainable energy systems: Role of optimization modeling techniques in power generation and supply – A review". In: *Renewable and sustainable energy reviews* 15.8 (2011), pp. 3480–3500 (cit. on p. 1).

[14] Amir Beck and Marc Teboulle. "A fast iterative shrinkage-thresholding algorithm for linear inverse problems". In: *SIAM journal on imaging sciences* 2.1 (2009), pp. 183–202 (cit. on pp. 6, 45, 60, 70, 75).

[15] Aharon Ben-Tal and Marc Teboulle. "Hidden convexity in some nonconvex quadratically constrained quadratic programming". In: *Mathematical Programming* 72.1 (1996), pp. 51–63 (cit. on p. 4).

[16] Dimitri P Bertsekas. "Nonlinear programming". In: *Journal of the Operational Research Society* 48.3 (1997), pp. 334–334 (cit. on p. 2).

[17] Dimitris Bertsimas, Guglielmo Lulli, and Amedeo Odoni. "An integer optimization approach to large-scale air traffic flow management". In: *Operations research* 59.1 (2011), pp. 211–227 (cit. on p. 1).

[18] Jérôme Bolte, Zheng Chen, and Edouard Pauwels. "The multiproximal linearization method for convex composite problems". In: *Mathematical Programming* 182.1 (2020), pp. 1–36 (cit. on p. 45).

[19] André B Bondi. "Characteristics of scalability and their impact on performance". In: *Proceedings of the 2nd international workshop on Software and performance*. 2000, pp. 195–203 (cit. on p. 3).

[20] Radu Ioan Boţ, Sorin-Mihai Grad, and Gert Wanka. "A new constraint qualification for the formula of the subdifferential of composed convex functions in infinite dimensional spaces". In: *Mathematische Nachrichten* 281.8 (2008), pp. 1088–1107 (cit. on p. 45).

[21] Radu Ioan Boţ, Sorin-Mihai Grad, and Gert Wanka. "New constraint qualification and conjugate duality for composed convex optimization problems". In: *Journal of Optimization Theory and Applications* 135 (2007), pp. 241–255 (cit. on p. 45).

[22] Léon Bottou, Frank E Curtis, and Jorge Nocedal. "Optimization methods for large-scale machine learning". In: *SIAM review* 60.2 (2018), pp. 223–311 (cit. on p. 5).

[23] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004 (cit. on pp. 2, 4, 49).

[24] Gábor Braun, Alejandro Carderera, Cyrille W Combettes, Hamed Hassani, Amin Karbasi, Aryan Mokhtari, and Sebastian Pokutta. "Conditional gradient methods". In: *arXiv preprint arXiv:2211.14103* (2022) (cit. on pp. 7, 18, 44).

[25] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. "Language models are few-shot learners". In: *Advances in neural information processing systems* 33 (2020), pp. 1877–1901 (cit. on p. 3).

[26] Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. "Sparks of artificial general intelligence: Early experiments with gpt-4". In: *arXiv preprint arXiv:2303.12712* (2023) (cit. on p. 2).

[27] Samuel Burer and Renato DC Monteiro. "Local minima and convergence in low-rank semidefinite programming". In: *Mathematical Programming* 103.3 (2005), pp. 427–444 (cit. on p. 16).

[28] James V Burke. "Descent methods for composite nondifferentiable optimization problems". In: *Mathematical Programming* 33.3 (1985), pp. 260–279 (cit. on p. 45).

[29] James V Burke. "Second order necessary and sufficient conditions for convex composite NDO". In: *Mathematical programming* 38 (1987), pp. 287–302 (cit. on p. 45).

[30] James V Burke and Michael C Ferris. "A Gauss-Newton method for convex composite optimization". In: *Mathematical Programming* 71.2 (1995), pp. 179–194 (cit. on p. 50).

[31] James V Burke, Hoheisel Tim, and Quang V Nguyen. "A study of convex convex-composite functions via infimal convolution with applications". In: *Mathematics of Operations Research* 46.4 (2021), pp. 1324–1348 (cit. on p. 45).

[32] Valentina Cacchiani, Dennis Huisman, Martin Kidd, Leo Kroon, Paolo Toth, Lucas Veelenturf, and Joris Wagenaar. "An overview of recovery models and algorithms for real-time railway rescheduling". In: *Transportation Research Part B: Methodological* 63 (2014), pp. 15–37 (cit. on p. 1).

[33] Emmanuel Candes and Benjamin Recht. "Exact matrix completion via convex optimization". In: *Communications of the ACM* 55.6 (2012), pp. 111–119 (cit. on p. 4).

[34] Florin Capitanescu. "Critical review of recent advances and further developments needed in AC optimal power flow". In: *Electric Power Systems Research* 136 (2016), pp. 57–68 (cit. on p. 1).

[35] Alberto Caprara, Matteo Fischetti, Paolo Toth, Daniele Vigo, and Pier Luigi Guida. "Algorithms for railway crew management". In: *Mathematical programming* 79 (1997), pp. 125–141 (cit. on p. 1).

[36] Alberto Caprara, Leo Kroon, Michele Monaci, Marc Peeters, and Paolo Toth. "Passenger railway optimization". In: *Handbooks in operations research and management science* 14 (2007), pp. 129–187 (cit. on p. 1).

[37] Alejandro Carderera, Jelena Diakonikolas, Cheuk Yin Lin, and Sebastian Pokutta. "Parameter-free Locally Accelerated Conditional Gradients". In: *arXiv preprint arXiv:2102.06806* (2021) (cit. on pp. 7, 46).

[38] Simon Caton and Christian Haas. "Fairness in machine learning: A survey". In: *ACM Computing Surveys* (2020) (cit. on p. 78).

[39] Volkan Cevher, Stephen Becker, and Mark Schmidt. "Convex optimization for big data: Scalable, randomized, and parallel algorithms for big data analytics". In: *IEEE Signal Processing Magazine* 31.5 (2014), pp. 32–43 (cit. on p. 5).

[40] Antonin Chambolle. "An algorithm for total variation minimization and applications". In: *Journal of Mathematical imaging and vision* 20.1 (2004), pp. 89–97 (cit. on pp. 61, 65).

[41] Antonin Chambolle, Vicent Caselles, Daniel Cremers, Matteo Novaga, and Thomas Pock. "An introduction to total variation for image analysis". In: *Theoretical foundations and numerical methods for sparse recovery* 9.263-340 (2010), p. 227 (cit. on p. 73).

[42] Antonin Chambolle and Thomas Pock. "A first-order primal-dual algorithm for convex problems with applications to imaging". In: *Journal of mathematical imaging and vision* 40.1 (2011), pp. 120–145 (cit. on pp. 6, 60, 64).

[43] Antonin Chambolle and Thomas Pock. "An introduction to continuous optimization for imaging". In: *Acta Numerica* 25 (2016), pp. 161–319 (cit. on pp. 60, 73).

[44] Antonin Chambolle and Thomas Pock. "On the ergodic convergence rates of a first-order primal–dual algorithm". In: *Mathematical Programming* 159.1 (2016), pp. 253–287 (cit. on pp. 6, 60, 64, 69, 70).

[45] Chih-Chung Chang and Chih-Jen Lin. "LIBSVM: A library for support vector machines". In: *ACM Transactions on Intelligent Systems and Technology* 2 (3 2011). Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm, 27:1–27:27 (cit. on p. 70).

[46] Vaggos Chatziafratis, Rad Niazadeh, and Moses Charikar. "Hierarchical clustering with structural constraints". In: *arXiv preprint arXiv:1805.09476* (2018) (cit. on p. 30).

[47] Peijun Chen, Jianguo Huang, and Xiaoqun Zhang. "A primal–dual fixed point algorithm for convex separable minimization with applications to image restoration". In: *Inverse Problems* 29.2 (2013), p. 025011 (cit. on pp. 12, 63, 69).

[48] Zhaoyue Chen and Yifan Sun. "Accelerating Frank-Wolfe via Averaging Step Directions". In: *arXiv preprint arXiv:2205.11794* (2022) (cit. on p. 46).

[49] Lenaic Chizat and Francis Bach. "On the global convergence of gradient descent for over-parameterized models using optimal transport". In: *Advances in neural information processing systems* 31 (2018) (cit. on p. 4).

[50] Kenneth L Clarkson. "Coresets, sparse greedy approximation, and the Frank-Wolfe algorithm". In: *ACM Transactions on Algorithms (TALG)* 6.4 (2010), p. 63 (cit. on p. 18).

[51] Jens Clausen, Allan Larsen, Jesper Larsen, and Natalia J Rezanova. "Disruption management in the airline industry – Concepts, models and methods". In: *Computers & Operations Research* 37.5 (2010), pp. 809–821 (cit. on p. 1).

[52] Cyrille Combettes and Sebastian Pokutta. “Boosting Frank-Wolfe by chasing gradients”. In: *International Conference on Machine Learning*. PMLR. 2020, pp. 2111–2121 (cit. on p. 46).

[53] Cyrille W Combettes and Sebastian Pokutta. “Complexity of linear minimization and projection on some sets”. In: *Operations Research Letters* (2021) (cit. on pp. 6, 7, 44).

[54] Patrick L Combettes and Jean-Christophe Pesquet. “Proximal splitting methods in signal processing”. In: *Fixed-point algorithms for inverse problems in science and engineering*. Springer, 2011, pp. 185–212 (cit. on p. 60).

[55] Laurent Condat. “A primal–dual splitting method for convex optimization involving Lipschitzian, proximable and linear composite terms”. In: *Journal of optimization theory and applications* 158.2 (2013), pp. 460–479 (cit. on pp. 6, 12, 60, 62, 75).

[56] Laurent Condat. “Discrete total variation: New definition and minimization”. In: *SIAM Journal on Imaging Sciences* 10.3 (2017), pp. 1258–1290 (cit. on p. 73).

[57] Jean-François Cordeau, Paolo Toth, and Daniele Vigo. “A survey of optimization models for train routing and scheduling”. In: *Transportation science* 32.4 (1998), pp. 380–404 (cit. on p. 1).

[58] Ying Cui, Jong-Shi Pang, and Bodhisattva Sen. “Composite difference-max programs for modern statistical estimation problems”. In: *SIAM Journal on Optimization* 28.4 (2018), pp. 3344–3374 (cit. on p. 45).

[59] George B Dantzig. “A theorem on linear inequalities”. In: *unpublished report* (1948) (cit. on p. 2).

[60] George B Dantzig. “Linear programming”. In: *Operations research* 50.1 (2002), pp. 42–47 (cit. on p. 2).

[61] George B Dantzig. “Maximization of a linear function of variables subject to linear inequalities”. In: *Activity analysis of production and allocation* 13 (1951), pp. 339–347 (cit. on p. 2).

[62] George B Dantzig. “Origins of the simplex method”. In: *A history of scientific computing*. 1990, pp. 141–151 (cit. on p. 2).

[63] George B Dantzig. “Reminiscences about the origins of linear programming”. In: *Mathematical Programming The State of the Art: Bonn 1982*. Springer, 1983, pp. 78–86 (cit. on p. 2).

[64] George Bernard Dantzig and Richard Cottle. *The basic George B. Dantzig*. Stanford University Press, 2003 (cit. on p. 2).

[65] Sanjoy Dasgupta. “A cost function for similarity-based hierarchical clustering”. In: *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*. 2016, pp. 118–127 (cit. on p. 30).

[66] Xolani Dastile, Turgay Celik, and Moshe Potsane. “Statistical and machine learning models in credit scoring: A systematic literature survey”. In: *Applied Soft Computing* 91 (2020), p. 106263 (cit. on p. 2).

[67] Samuel Daulton, David Eriksson, Maximilian Balandat, and Eytan Bakshy. "Multi-objective bayesian optimization over high-dimensional search spaces". In: *Uncertainty in Artificial Intelligence*. PMLR. 2022, pp. 507–517 (cit. on p. 50).

[68] Welington De Oliveira. "Short Paper-A note on the Frank–Wolfe algorithm for a class of nonconvex and nonsmooth optimization problems". In: *Open Journal of Mathematical Optimization* 4 (2023), pp. 1–10 (cit. on pp. 11, 46, 47, 78).

[69] Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. "SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives". In: *Advances in neural information processing systems*. 2014, pp. 1646–1654 (cit. on p. 18).

[70] Olivier Devolder. "Exactness, inexactness and stochasticity in first-order methods for large-scale convex optimization". PhD thesis. ICTEAM and CORE, Université catholique de Louvain, 2013 (cit. on p. 54).

[71] Jelena Diakonikolas, Alejandro Carderera, and Sebastian Pokutta. "Locally accelerated conditional gradients". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2020, pp. 1737–1747 (cit. on pp. 7, 46).

[72] Steven Diamond and Stephen Boyd. "CVXPY: A Python-embedded modeling language for convex optimization". In: *Journal of Machine Learning Research* 17.83 (2016), pp. 1–5 (cit. on p. 56).

[73] Nikita Doikov and Yurii Nesterov. "High-Order Optimization Methods for Fully Composite Problems". In: *SIAM Journal on Optimization* 32.3 (2022), pp. 2402–2427 (cit. on pp. 11, 45–47, 49, 51, 121).

[74] Gideon Dresdner, Maria-Luiza Vladarean, Gunnar Rätsch, Francesco Locatello, Volkan Cevher, and Alp Yurtsever. "Faster One-Sample Stochastic Conditional Gradient Method for Composite Convex Minimization". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2022, pp. 8439–8457 (cit. on pp. ix, 15).

[75] Yoel Drori, Shoham Sabach, and Marc Teboulle. "A simple algorithm for a class of nonsmooth convex–concave saddle-point problems". In: *Operations Research Letters* 43.2 (2015), pp. 209–214 (cit. on pp. 12, 63).

[76] Dmitriy Drusvyatskiy and Adrian S Lewis. "Error bounds, quadratic growth, and linear convergence of proximal methods". In: *Mathematics of Operations Research* 43.3 (2018), pp. 919–948 (cit. on p. 45).

[77] Dmitriy Drusvyatskiy and Courtney Paquette. "Efficiency of minimizing compositions of convex functions and smooth maps". In: *Mathematical Programming* 178.1 (2019), pp. 503–558 (cit. on pp. 11, 45, 47).

[78] John C Duchi and Feng Ruan. "Solving (most) of a set of quadratic equalities: Composite optimization for robust phase retrieval". In: *Information and Inference: A Journal of the IMA* 8.3 (2019), pp. 471–529 (cit. on p. 50).

[79] Majid Eskandarpour, Pierre Dejax, Joe Miemczyk, and Olivier Péton. "Sustainable supply chain network design: An optimization-oriented review". In: *Omega* 54 (2015), pp. 11–32 (cit. on p. 1).

[80] Cong Fang, Chris Junchi Li, Zhouchen Lin, and Tong Zhang. "Spider: Near-optimal non-convex optimization via stochastic path-integrated differential estimator". In: *Advances in Neural Information Processing Systems*. 2018, pp. 689–699 (cit. on pp. 10, 17, 18, 21, 23, 40).

[81] Olivier Fercoq, Ahmet Alacaoglu, Ion Necoara, and Volkan Cevher. "Almost surely constrained convex optimization". In: *arXiv preprint arXiv:1902.00126* (2019) (cit. on pp. 10, 16–20, 22, 82, 108).

[82] David A. Ferrucci. "Introduction to "This is Watson"". In: *IBM Journal of Research and Development* 56.3.4 (2012), pp. 1–1 (cit. on p. 2).

[83] C Fienup and J Dainty. "Phase retrieval and image reconstruction for astronomy". In: *Image recovery: theory and application* 231 (1987), p. 275 (cit. on p. 72).

[84] Marguerite Frank and Philip Wolfe. "An algorithm for quadratic programming". In: *Naval research logistics quarterly* 3.1-2 (1956), pp. 95–110 (cit. on pp. 6, 44, 58).

[85] Marguerite Frank and Philip Wolfe. "An algorithm for quadratic programming". In: *Naval Research Logistics Quarterly* 3 (1956), pp. 95–110. DOI: 10.1002/nav.3800030109. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/nav.3800030109. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/nav.3800030109 (cit. on pp. 17, 18).

[86] Robert M Freund, Paul Grigas, and Rahul Mazumder. "An extended Frank–Wolfe method with "in-face" directions, and its application to low-rank matrix completion". In: *SIAM Journal on optimization* 27.1 (2017), pp. 319–346 (cit. on p. 7).

[87] David Gale, Harold W Kuhn, and Albert W Tucker. "Linear programming and the theory of games". In: *Activity analysis of production and allocation* 13 (1951), pp. 317–335 (cit. on p. 2).

[88] Dan Garber. "Faster Projection-free Convex Optimization over the Spectrahedron". In: *Advances in Neural Information Processing Systems* 29 (2016), pp. 874–882 (cit. on p. 33).

[89] Dan Garber. "Projection-free Algorithms for Convex Optimization and Online Learning". PhD thesis. Technion-Israel Institute of Technology, 2016 (cit. on p. 18).

[90] Dan Garber and Elad Hazan. "A linearly convergent variant of the conditional gradient algorithm under strong convexity, with applications to online and stochastic optimization". In: *SIAM Journal on Optimization* 26.3 (2016), pp. 1493–1528 (cit. on p. 46).

[91] Dan Garber and Elad Hazan. "Faster rates for the frank-wolfe method over strongly-convex sets". In: *International Conference on Machine Learning*. PMLR. 2015, pp. 541–549 (cit. on p. 7).

[92] Dan Garber and Elad Hazan. "Sublinear time algorithms for approximate semidefinite programming". In: *Mathematical Programming* 158.1-2 (2016), pp. 329–361 (cit. on p. 17).

[93] Dan Garber and Noam Wolf. "Frank-Wolfe with a Nearest Extreme Point Oracle". In: *arXiv preprint arXiv:2102.02029* (2021) (cit. on p. 78).

[94] Gauthier Gidel, Fabian Pedregosa, and Simon Lacoste-Julien. "Frank-Wolfe Splitting via Augmented Lagrangian Method". In: *International Conference on Artificial Intelligence and Statistics*. 2018, pp. 1456–1465 (cit. on p. 18).

[95] Giorgio Giorgi and Tinne Hoff Kjeldsen. "A historical view of nonlinear programming: Traces and emergence". In: *Traces and Emergence of Nonlinear Programming*. Springer, 2013, pp. 1–43 (cit. on p. 2).

[96] Michel X Goemans and David P Williamson. "Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming". In: *Journal of the ACM (JACM)* 42.6 (1995), pp. 1115–1145 (cit. on pp. 4, 16).

[97] AA Goldstein. "Convex programming in Hilbert space". In: *Bulletin of the American Mathematical Society* 70.5 (1964), pp. 709–710 (cit. on p. 5).

[98] Tom Goldstein, Min Li, and Xiaoming Yuan. "Adaptive primal-dual splitting methods for statistical learning and image processing". In: *Advances in Neural Information Processing Systems*. 2015, pp. 2089–2097 (cit. on p. 61).

[99] Tom Goldstein, Min Li, Xiaoming Yuan, Ernie Esser, and Richard Baraniuk. "Adaptive primal-dual hybrid gradient methods for saddle-point problems". In: *arXiv preprint arXiv:1305.0546* (2013) (cit. on p. 61).

[100] Michael Grant and Stephen Boyd. *CVX: Matlab Software for Disciplined Convex Programming, version 2.1*. http://cvxr.com/cvx. Mar. 2014 (cit. on p. 27).

[101] Elad Hazan. "Sparse approximate solutions to semidefinite programs". In: *Latin American symposium on theoretical informatics*. Springer. 2008, pp. 306–316 (cit. on pp. 10, 18, 33).

[102] Elad Hazan and Haipeng Luo. "Variance-reduced and projection-free stochastic optimization". In: *International Conference on Machine Learning*. 2016, pp. 1263–1271 (cit. on pp. 18, 19).

[103] Elad E Hazan and Satyen Kale. "Projection-free online learning". In: *29th International Conference on Machine Learning, ICML 2012*. 2012, pp. 521–528 (cit. on p. 18).

[104] Yuping He and John Mcphee. "Multidisciplinary optimization of multibody systems with application to the design of rail vehicles". In: *Multibody System Dynamics* 14 (2005), pp. 111–135 (cit. on p. 1).

[105] Donald W. Hearn. "The Gap Function of a Convex Program". In: *Operations Research Letters* 1.2 (Apr. 1982), pp. 67–71. DOI: 10.1016/0167-6377(82)90049-9 (cit. on p. 51).

[106] Tao Hong and Shu Fan. "Probabilistic electric load forecasting: A tutorial review". In: *International Journal of Forecasting* 32.3 (2016), pp. 914–938 (cit. on p. 1).

[107] Gregory Hornby, Al Globus, Derek Linden, and Jason Lohn. "Automated antenna design with evolutionary algorithms". In: *Space 2006*. 2006, p. 7242 (cit. on p. 1).

[108] Qixing Huang, Yuxin Chen, and Leonidas Guibas. "Scalable semidefinite relaxation for maximum a posterior estimation". In: *International Conference on Machine Learning*. 2014, pp. 64–72 (cit. on p. 16).

[109] Garud Iyengar, David J Phillips, and Cliff Stein. "Feasible and accurate algorithms for covering semidefinite programs". In: *Scandinavian Workshop on Algorithm Theory*. Springer. 2010, pp. 150–162 (cit. on p. 31).

[110] Arthur Jacot, Franck Gabriel, and Clément Hongler. "Neural tangent kernel: Convergence and generalization in neural networks". In: *Advances in neural information processing systems* 31 (2018) (cit. on p. 4).

[111] Martin Jaggi. "Revisiting Frank-Wolfe: Projection-Free Sparse Convex Optimization". In: *Proceedings of the 30th International Conference on Machine Learning*. Ed. by Sanjoy Dasgupta and David McAllester. Vol. 28. Proceedings of Machine Learning Research 1. Atlanta, Georgia, USA: PMLR, June 2013, pp. 427–435. URL: http://proceedings.mlr.press/v28/jaggi13.html (cit. on pp. 17, 18).

[112] Martin Jaggi. "Revisiting Frank-Wolfe: Projection-free sparse convex optimization". In: *International Conference on Machine Learning*. 2013, pp. 427–435 (cit. on pp. 7, 11, 44, 46, 49, 58).

[113] Martin Jaggi and Marek Sulovskỳ. "A simple algorithm for nuclear norm regularized problems". In: *Proceedings of the 27th International Conference on International Conference on Machine Learning*. 2010 (cit. on p. 33).

[114] Yiding Jiang, Behnam Neyshabur, Hossein Mobahi, Dilip Krishnan, and Samy Bengio. "Fantastic Generalization Measures and Where to Find Them". In: *International Conference on Learning Representations*. 2019 (cit. on p. 3).

[115] Rie Johnson and Tong Zhang. "Accelerating stochastic gradient descent using predictive variance reduction". In: *Advances in neural information processing systems*. 2013, pp. 315–323 (cit. on pp. 18, 23).

[116] Michael I. Jordan. "Artificial Intelligence—The Revolution Hasn't Happened Yet". In: *Harvard Data Science Review* 1.1 (July 2019). https://hdsr.mitpress.mit.edu/pub/wot7mkc1 (cit. on p. 2).

[117] Armand Joulin, Kevin Tang, and Li Fei-Fei. "Efficient image and video co-localization with frank-wolfe algorithm". In: *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VI 13*. Springer. 2014, pp. 253–268 (cit. on p. 7).

[118] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. "Highly accurate protein structure prediction with AlphaFold". In: *Nature* 596.7873 (2021), pp. 583–589 (cit. on p. 2).

[119] Leonid V Kantorovich. "The mathematical method of production planning and organization". In: *Management Science* 6.4 (1939), pp. 363–422 (cit. on p. 2).

[120] Ali Kavis. "Universal and adaptive methods for robust stochastic optimization". In: (2023), p. 235. DOI: https://doi.org/10.5075/epfl-thesis-9077. URL: http://infoscience.epfl.ch/record/304455 (cit. on p. 8).

[121] Gaetan KW Kenway and Joaquim RRA Martins. "Multipoint high-fidelity aerostructural optimization of a transport aircraft configuration". In: *Journal of Aircraft* 51.1 (2014), pp. 144–160 (cit. on p. 1).

[122] Thomas Kerdreux, Alexandre d'Aspremont, and Sebastian Pokutta. "Local and Global Uniform Convexity Conditions". In: *ArXiv* abs/2102.05134 (2021) (cit. on p. 38).

[123] Thomas Kerdreux, Alexandre d'Aspremont, and Sebastian Pokutta. "Projection-free optimization on uniformly convex sets". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2021, pp. 19–27 (cit. on p. 7).

[124] Jürgen M Kleinhans, Georg Sigl, Frank M Johannes, and Kurt J Antreich. "GORDIAN: VLSI placement by quadratic programming and slicing optimization". In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 10.3 (1991), pp. 356–365 (cit. on p. 1).

[125] Walid Klibi, Alain Martel, and Adel Guitouni. "The design of robust value-creating supply chain networks: a critical review". In: *European Journal of Operational Research* 203.2 (2010), pp. 283–293 (cit. on p. 1).

[126] Junpei Komiyama, Akiko Takeda, Junya Honda, and Hajime Shimao. "Nonconvex optimization for regression with fairness constraints". In: *International conference on machine learning*. PMLR. 2018, pp. 2737–2746 (cit. on p. 78).

[127] Nikos Komodakis and Jean-Christophe Pesquet. "Playing with duality: An overview of recent primal? dual approaches for solving large-scale optimization problems". In: *IEEE Signal Processing Magazine* 32.6 (2015), pp. 31–54 (cit. on pp. 12, 60).

[128] Timo Kreimeier, Sebastian Pokutta, Andrea Walther, and Zev Woodstock. "On a Frank-Wolfe Approach for Abs-smooth Functions". In: *arXiv preprint arXiv:2303.09881* (2023) (cit. on pp. 11, 47, 51).

[129] John W Labadie. "Optimal operation of multireservoir systems: State-of-the-art review". In: *Journal of water resources planning and management* 130.2 (2004), pp. 93–111 (cit. on p. 1).

[130] Simon Lacoste-Julien. "Convergence rate of Frank-Wolfe for non-convex objectives". In: *arXiv preprint arXiv:1607.00345* (2016) (cit. on pp. 46, 51, 52).

[131] Simon Lacoste-Julien and Martin Jaggi. "On the global linear convergence of Frank-Wolfe optimization variants". In: *arXiv preprint arXiv:1511.05932* (2015) (cit. on p. 7).

[132] Simon Lacoste-Julien, Martin Jaggi, Mark Schmidt, and Patrick Pletscher. "Block-coordinate Frank-Wolfe optimization for structural SVMs". In: *International Conference on Machine Learning*. PMLR. 2013, pp. 53–61 (cit. on p. 7).

[133] Simon Lacoste-Julien, Mark Schmidt, and Francis Bach. "A simpler approach to obtaining an O (1/t) convergence rate for the projected stochastic subgradient method". In: *arXiv preprint arXiv:1212.2002* (2012) (cit. on p. 6).

[134] Guanghui Lan. "The complexity of large-scale convex programming under a linear optimization oracle". In: *arXiv preprint arXiv:1309.5550* (2013) (cit. on pp. 7, 18, 38, 45, 46).

[135] Guanghui Lan and Yi Zhou. "Conditional gradient sliding for convex optimization". In: *SIAM Journal on Optimization* 26.2 (2016), pp. 1379–1409 (cit. on pp. 7, 11, 18, 46, 53, 54, 58).

[136] Puya Latafat, Andreas Themelis, Lorenzo Stella, and Panagiotis Patrinos. "Adaptive proximal algorithms for convex optimization under local Lipschitz continuity of the gradient". In: *arXiv preprint arXiv:2301.04431* (2023) (cit. on p. 75).

[137] Yann LeCun and Corinna Cortes. "MNIST handwritten digit database". In: (2010). URL: http://yann.lecun.com/exdb/mnist/ (cit. on p. 29).

[138] Michel Ledoux. *The concentration of measure phenomenon*. 89. American Mathematical Soc., 2001 (cit. on p. 27).

[139] Claude Lemaréchal. "Cauchy and the gradient method". In: *Doc Math Extra* 251.254 (2012), p. 10 (cit. on pp. 5, 44).

[140] Evgeny S Levitin and Boris T Polyak. "Constrained minimization methods". In: *USSR Computational mathematics and mathematical physics* 6.5 (1966), pp. 1–50 (cit. on pp. 2, 5, 6).

[141] Ya-Feng Liu, Xin Liu, and Shiqian Ma. "On the nonergodic convergence rate of an inexact augmented lagrangian framework for composite convex programming". In: *Mathematics of Operations Research* 44.2 (2019), pp. 632–650 (cit. on p. 18).

[142] Francesco Locatello, Alp Yurtsever, Olivier Fercoq, and Volkan Cevher. "Stochastic Frank-Wolfe for Composite Convex Minimization". In: *Advances in Neural Information Processing Systems* (2019) (cit. on pp. 34, 36, 39, 40, 110).

[143] Francesco Locatello, Alp Yurtsever, Olivier Fercoq, and Volkan Cevher. "Stochastic Frank-Wolfe for composite convex minimization". In: *Advances in Neural Information Processing Systems* 32 (2019) (cit. on pp. 10, 17–19, 26, 27, 29, 90).

[144] Ignace Loris and Caroline Verhoeven. "On a generalization of the iterative soft-thresholding algorithm for the case of non-separable penalty". In: *Inverse Problems* 27.12 (2011), p. 125007 (cit. on pp. 12, 63).

[145] Mehrdad Mahdavi, Lijun Zhang, and Rong Jin. "Mixed optimization for smooth functions". In: *Advances in neural information processing systems*. 2013, pp. 674–682 (cit. on p. 18).

[146] Yura Malitsky. "Golden ratio algorithms for variational inequalities". In: *Mathematical Programming* 184.1 (2020), pp. 383–410 (cit. on pp. 12, 62).

[147] Yura Malitsky and Konstantin Mishchenko. "Adaptive Gradient Descent without Descent". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 6702–6712 (cit. on pp. 62, 69, 72, 78, 135).

[148] Yura Malitsky and Konstantin Mishchenko. "Adaptive Proximal Gradient Method for Convex Optimization". In: *arXiv preprint arXiv:2308.02261* (2023) (cit. on p. 75).

[149] Yura Malitsky and Thomas Pock. "A first-order primal-dual algorithm with linesearch". In: *SIAM Journal on Optimization* 28.1 (2018), pp. 411–432 (cit. on pp. 12, 61, 63).

[150] M Teresa Melo, Stefan Nickel, and Francisco Saldanha-Da-Gama. "Facility location and supply chain management–A review". In: *European journal of operational research* 196.2 (2009), pp. 401–412 (cit. on p. 1).

[151] Pedro Mendes and Douglas Kell. "Non-linear optimization of biochemical pathways: applications to metabolic engineering and parameter estimation." In: *Bioinformatics (Oxford, England)* 14.10 (1998), pp. 869–883 (cit. on p. 1).

[152] Francesco Mezzadri. "How to generate random matrices from the classical compact groups". In: *arXiv preprint math-ph/0609050* (2006) (cit. on p. 57).

[153] Kaisa Miettinen. *Nonlinear multiobjective optimization*. Vol. 12. Springer Science & Business Media, 1999 (cit. on pp. 11, 50).

[154] Dustin G Mixon, Soledad Villar, and Rachel Ward. "Clustering subgaussian mixtures by semidefinite programming". In: *arXiv preprint arXiv:1602.06612* (2016) (cit. on pp. 27, 29).

[155] Aryan Mokhtari, Hamed Hassani, and Amin Karbasi. "Stochastic conditional gradient methods: From convex minimization to submodular maximization". In: *arXiv preprint arXiv:1804.09554* (2018) (cit. on pp. 10, 19, 21, 22, 40, 83).

[156] Daniel K Molzahn, Florian Dörfler, Henrik Sandberg, Steven H Low, Sambuddha Chakrabarti, Ross Baldick, and Javad Lavaei. "A survey of distributed optimization and control algorithms for electric power systems". In: *IEEE Transactions on Smart Grid* 8.6 (2017), pp. 2941–2962 (cit. on p. 1).

[157] Jean Jacques Moreau. "Fonctions convexes duales et points proximaux dans un espace hilbertien". In: *Comptes rendus hebdomadaires des séances de l'Académie des sciences* 255 (1962), pp. 2897–2899 (cit. on p. 6).

[158] Jean-Jacques Moreau. "Proximité et dualité dans un espace hilbertien". In: *Bulletin de la Société mathématique de France* 93 (1965), pp. 273–299 (cit. on p. 6).

[159] Geoffrey Négiar, Gideon Dresdner, Alicia Yi-Ting Tsai, Laurent El Ghaoui, Francesco Locatello, and Fabian Pedregosa. "Stochastic Frank-Wolfe for Constrained Finite-Sum Minimization". In: *The International Conference on Machine Learning (ICML)* (2020) (cit. on pp. 33, 34, 36, 37, 112).

[160] Arkadi Nemirovski. "Information-based complexity of convex programming". In: *Lecture notes* 834 (1995) (cit. on p. 45).

[161] Arkadi Nemirovski and David Yudin. “Problem complexity and method efficiency in optimization”. In: (1983) (cit. on pp. 2, 9, 45).

[162] Arkadii Semenovich Nemirovsky and David Borisovich Yudin. “Problem complexity and method efficiency in optimization.” In: (1983) (cit. on pp. 4, 20).

[163] YE Nesterov. “A method for solving the convex programming problem with convergence rate O (1/kˆ 2)”. In: *Dokl. Akad. Nauk SSSR*. Vol. 269. 1983, pp. 543–547 (cit. on pp. 6, 46, 53).

[164] Yu Nesterov. “Gradient methods for minimizing composite functions”. In: *Mathematical Programming* 140.1 (2013), pp. 125–161 (cit. on pp. 45, 60).

[165] Yurii Nesterov. “Complexity bounds for primal-dual methods minimizing the model of objective function”. In: *Mathematical Programming* 171.1 (2018), pp. 311–330 (cit. on pp. 7, 10, 20, 44).

[166] Yurii Nesterov. “Effective methods in nonlinear programming”. In: *Moscow, Radio i Svyaz* (1989) (cit. on p. 45).

[167] Yurii Nesterov. “Smooth minimization of non-smooth functions”. In: *Mathematical Programming* (2005) (cit. on pp. 19, 79, 80).

[168] Yurii Nesterov. *Lectures on convex optimization*. Vol. 137. Springer, 2018 (cit. on pp. 2, 129).

[169] Yurii Nesterov. “Modified Gauss–Newton scheme with worst case guarantees for global performance”. In: *Optimisation Methods and Software* 22.3 (2007), pp. 469–483 (cit. on p. 50).

[170] Yurii Nesterov and Arkadii Nemirovskii. *Interior-point polynomial algorithms in convex programming*. SIAM, 1994 (cit. on p. 50).

[171] John von Neumann. “Discussion of a maximization problem”. In: *John von Neumann: Collected Works*. Ed. by Abraham Haskel Taub. Vol. 6. Oxford: Pergamon Press, 1963, pp. 89–95 (cit. on p. 2).

[172] Kee Yuan Ngiam and Wei Khor. “Big data and machine learning algorithms for health-care delivery”. In: *The Lancet Oncology* 20.5 (2019), e262–e273 (cit. on p. 2).

[173] Lam M Nguyen, Jie Liu, Katya Scheinberg, and Martin Takáč. “SARAH: A novel method for machine learning problems using stochastic recursive gradient”. In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org. 2017, pp. 2613–2621 (cit. on pp. 18, 23).

[174] Jorge Nocedal and Stephen J Wright. *Numerical optimization*. Springer, 1999 (cit. on p. 2).

[175] OpenAI. *GPT-4 Technical Report*. 2023. arXiv: 2303.08774 [cs.CL] (cit. on p. 2).

[176] OpenAI. *Introducing ChatGPT*. 2022. URL: https://openai.com/blog/chatgpt (visited on 09/19/2023) (cit. on p. 2).

[177] Nicolas Papernot, Patrick McDaniel, Arunesh Sinha, and Michael P Wellman. "Sok: Security and privacy in machine learning". In: *2018 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE. 2018, pp. 399–414 (cit. on p. 3).

[178] Neal Parikh, Stephen Boyd, et al. "Proximal algorithms". In: *Foundations and trends® in Optimization* 1.3 (2014), pp. 127–239 (cit. on pp. 32, 60).

[179] Andrei Patrascu and Ion Necoara. "Nonasymptotic convergence of stochastic proximal point methods for constrained convex optimization." In: *Journal of Machine Learning Research* 18 (2017), pp. 198–1 (cit. on pp. 10, 16, 17).

[180] François-Pierre Paty and Marco Cuturi. "Subspace robust Wasserstein distances". In: *International conference on machine learning*. PMLR. 2019, pp. 5072–5081 (cit. on p. 7).

[181] Fabian Pedregosa and Gauthier Gidel. "Adaptive three operator splitting". In: *International Conference on Machine Learning*. PMLR. 2018, pp. 4085–4094 (cit. on p. 61).

[182] Jiming Peng and Yu Wei. "Approximating k-means-type clustering via semidefinite programming". In: *SIAM journal on optimization* 18.1 (2007), pp. 186–205 (cit. on pp. 4, 16, 29).

[183] Teemu Pennanen. "Graph-convex mappings and K-convex functions." In: *Journal of Convex Analysis* 6.2 (1999), pp. 235–266 (cit. on p. 45).

[184] Mert Pilanci and Tolga Ergen. "Neural networks are convex regularizers: Exact polynomial-time convex optimization formulations for two-layer networks". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 7695–7705 (cit. on p. 4).

[185] Boris T Polyak. "History of mathematical programming in the USSR: analyzing the phenomenon". In: *Mathematical Programming* 91.3 (2002), pp. 401–416 (cit. on p. 2).

[186] Boris T Polyak. "Introduction to optimization". In: (1987) (cit. on p. 2).

[187] Chao Qu, Yan Li, and Huan Xu. "Non-convex conditional gradient sliding". In: *international conference on machine learning*. PMLR. 2018, pp. 4208–4217 (cit. on p. 46).

[188] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. "Zero-shot text-to-image generation". In: *International Conference on Machine Learning*. PMLR. 2021, pp. 8821–8831 (cit. on p. 2).

[189] Deepti Rani and Maria Madalena Moreira. "Simulation–optimization modeling: a survey and potential application in reservoir systems operation". In: *Water resources management* 24 (2010), pp. 1107–1138 (cit. on p. 1).

[190] Sathya N Ravi, Maxwell D Collins, and Vikas Singh. "A deterministic nonsmooth frank wolfe algorithm with coreset guarantees". In: *Informs Journal on Optimization* 1.2 (2019), pp. 120–142 (cit. on p. 46).

[191] Herbert Robbins and Sutton Monro. "A stochastic approximation method". In: *The annals of mathematical statistics* (1951), pp. 400–407 (cit. on p. 8).

[192] R Tyrrell Rockafellar. *Convex analysis*. Vol. 36. Princeton university press, 1970 (cit. on p. 50).

[193] R Tyrrell Rockafellar. *Convex analysis*. Vol. 11. Princeton university press, 1997 (cit. on p. 2).

[194] R Tyrrell Rockafellar and Roger J-B Wets. *Variational analysis*. Vol. 317. Springer Science & Business Media, 2009 (cit. on p. 2).

[195] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. "High-resolution image synthesis with latent diffusion models". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 10684–10695 (cit. on p. 2).

[196] Ryan A. Rossi and Nesreen K. Ahmed. "The Network Data Repository with Interactive Graph Analytics and Visualization". In: *AAAI*. 2015. URL: http://networkrepository.com (cit. on pp. xvii, 31, 32).

[197] Nicolas L Roux, Mark Schmidt, and Francis R Bach. "A stochastic gradient method with an exponential convergence _rate for finite training sets". In: *Advances in neural information processing systems*. 2012, pp. 2663–2671 (cit. on p. 18).

[198] Olga Russakovsky et al. "ImageNet Large Scale Visual Recognition Challenge". In: *International Journal of Computer Vision (IJCV)* 115.3 (2015), pp. 211–252. DOI: 10.1007/s11263-015-0816-y (cit. on p. 3).

[199] Mark Schmidt, Nicolas Le Roux, and Francis Bach. "Minimizing finite sums with the stochastic average gradient". In: *Mathematical Programming* 162.1-2 (2017), pp. 83–112 (cit. on pp. 10, 18, 34, 41).

[200] Bernhard Schölkopf and Alexander J Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002 (cit. on p. 4).

[201] Andrew W Senior, Richard Evans, John Jumper, James Kirkpatrick, Laurent Sifre, Tim Green, Chongli Qin, Augustin Žídek, Alexander WR Nelson, Alex Bridgland, et al. "Improved protein structure prediction using potentials from deep learning". In: *Nature* 577.7792 (2020), pp. 706–710 (cit. on p. 2).

[202] Khushro Shahookar and Pinaki Mazumder. "VLSI cell placement techniques". In: *ACM Computing Surveys (CSUR)* 23.2 (1991), pp. 143–220 (cit. on p. 1).

[203] Vaishaal Shankar, Rebecca Roelofs, Horia Mania, Alex Fang, Benjamin Recht, and Ludwig Schmidt. "Evaluating machine accuracy on imagenet". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 8634–8644 (cit. on p. 5).

[204] N Shor. "Application of the subgradient method for the solution of network transport problems, Notes of Sc". In: *Seminar, Ukrainian Acad. of Sci., Kiew*. 1962 (cit. on pp. 6, 44, 47, 58).

[205] NZ Shor, Krzysztof C Kiwiel, and Andrzej Ruszcayński. *Minimization methods for non-differentiable functions*. 1985 (cit. on p. 56).

[206] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. "Mastering the game of go without human knowledge". In: *nature* 550.7676 (2017), pp. 354–359 (cit. on p. 2).

[207] Antonio Silveti-Falls, Cesare Molinari, and Jalal Fadili. "Generalized Conditional Gradient with Augmented Lagrangian for Composite Minimization". In: *arXiv preprint arXiv:1901.01287* (2019) (cit. on p. 18).

[208] Bismark Singh and Mark Eisner. *A Brief History of Optimization and Mathematical Programming*. 2020. URL: https://www.informs.org/Explore/History-of-O.R.-Excellence/O.R.-Methodologies/Optimization-Mathematical-Programming (visited on 09/19/2023) (cit. on p. 2).

[209] Brent Smith and Greg Linden. "Two decades of recommender systems at Amazon. com". In: *Ieee internet computing* 21.3 (2017), pp. 12–18 (cit. on p. 2).

[210] Ju Sun, Qing Qu, and John Wright. "A geometric analysis of phase retrieval". In: *Foundations of Computational Mathematics* 18.5 (2018), pp. 1131–1198 (cit. on p. 73).

[211] Kiran K Thekumparampil, Prateek Jain, Praneeth Netrapalli, and Sewoong Oh. "Projection efficient subgradient method and optimal nonsmooth frank-wolfe method". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 12211–12224 (cit. on pp. 7, 11, 46, 47).

[212] Quoc Tran-Dinh, Olivier Fercoq, and Volkan Cevher. "A smooth primal-dual optimization framework for nonsmooth composite convex minimization". In: *SIAM Journal on Optimization* 28.1 (2018), pp. 96–134 (cit. on pp. 19, 35, 38, 79).

[213] Quoc Tran-Dinh, Nhan Pham, and Lam Nguyen. "Stochastic Gauss-Newton algorithms for nonconvex compositional optimization". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 9572–9582 (cit. on p. 50).

[214] Oleg Trott and Arthur J Olson. "AutoDock Vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading". In: *Journal of computational chemistry* 31.2 (2010), pp. 455–461 (cit. on p. 1).

[215] Paul Tseng. "On accelerated proximal gradient methods for convex-concave optimization". In: *submitted to SIAM Journal on Optimization* 2.3 (2008) (cit. on p. 6).

[216] Alan Turing. "Machinery and Intelligence". In: *Mind: A Quarterly Review of Psychology and Philosophy* 59.236 (1950), pp. 433–460 (cit. on p. 2).

[217] Vladimir N Vapnik. *The nature of statistical learning theory*. 1995 (cit. on p. 4).

[218] Aditya Varre, Maria-Luiza Vladarean, Loucas Pillaud-Vivien, and Nicolas Flammarion. "On the spectral bias of two-layer linear networks". In: *Advances in Neural Information Processing Systems* (2023) (cit. on p. ix).

[219] Maria-Luiza Vladarean, Ahmet Alacaoglu, Ya-Ping Hsieh, and Volkan Cevher. "Conditional gradient methods for stochastically constrained convex minimization". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 9775–9785 (cit. on pp. ix, 15, 33, 34, 36, 38–40, 110).

[220] Maria-Luiza Vladarean, Nikita Doikov, Martin Jaggi, and Nicolas Flammarion. "Linearization Algorithms for Fully Composite Optimization". In: *Proceedings of Thirty Sixth Conference on Learning Theory*. Ed. by Gergely Neu and Lorenzo Rosasco. Vol. 195. Proceedings of Machine Learning Research. PMLR, July 2023, pp. 3669–3695. URL: https://proceedings.mlr.press/v195/vladarean23a.html (cit. on pp. ix, 43).

[221] Maria-Luiza Vladarean, Yura Malitsky, and Volkan Cevher. "A first-order primal-dual method with adaptivity to local smoothness". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 6171–6182 (cit. on pp. ix, 59).

[222] Bang Công Vũ. "A splitting algorithm for dual monotone inclusions involving cocoercive operators". In: *Advances in Computational Mathematics* 38.3 (2013), pp. 667–681 (cit. on pp. 6, 12, 60, 62, 75).

[223] Adriaan Walther. "The question of phase retrieval in optics". In: *Optica Acta: International Journal of Optics* 10.1 (1963), pp. 41–49 (cit. on p. 72).

[224] Kilian Q Weinberger and Lawrence K Saul. "Distance metric learning for large margin nearest neighbor classification." In: *Journal of Machine Learning Research* 10.2 (2009) (cit. on p. 16).

[225] Ross Wightman, Hugo Touvron, and Hervé Jégou. "Resnet strikes back: An improved training procedure in timm". In: *arXiv preprint arXiv:2110.00476* (2021) (cit. on p. 3).

[226] Yong Xia. "A survey of hidden convex optimization". In: *Journal of the Operations Research Society of China* 8.1 (2020), pp. 1–28 (cit. on p. 4).

[227] Lin Xiao and Tong Zhang. "A proximal stochastic gradient method with progressive variance reduction". In: *SIAM Journal on Optimization* 24.4 (2014), pp. 2057–2075 (cit. on p. 18).

[228] Yangyang Xu. "Accelerated first-order primal-dual proximal methods for linearly constrained composite convex programming". In: *SIAM Journal on Optimization* 27.3 (2017), pp. 1459–1484 (cit. on p. 26).

[229] Yangyang Xu. "Primal-dual stochastic gradient method for convex programs with many functional constraints". In: *arXiv preprint arXiv:1802.02724* (2018) (cit. on pp. 16–18).

[230] Liuqin Yang, Defeng Sun, and Kim-Chuan Toh. "SDPNAL+: a majorized semismooth Newton-CG augmented Lagrangian method for semidefinite programming with nonnegative constraints". In: *Mathematical Programming Computation* 7.3 (2015), pp. 331–366 (cit. on p. 27).

[231] Alp Yurtsever, Olivier Fercoq, and Volkan Cevher. "A Conditional-Gradient-Based Augmented Lagrangian Framework". In: *International Conference on Machine Learning*. 2019, pp. 7272–7281 (cit. on pp. 18, 46).

[232] Alp Yurtsever, Olivier Fercoq, Francesco Locatello, and Volkan Cevher. "A conditional gradient framework for composite convex minimization with applications to semidefinite programming". In: *Proceedings of the 35th International Conference on Machine Learning* (2018) (cit. on pp. 10, 17–19, 26, 27, 29, 33, 38, 40, 46).

[233] Alp Yurtsever, Suvrit Sra, and Volkan Cevher. "Conditional Gradient Methods via Stochastic Path-Integrated Differential Estimator". In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. Long Beach, California, USA: PMLR, June 2019, pp. 7282–7291. URL: http://proceedings.mlr.press/v97/yurtsever19b.html (cit. on pp. 19, 23, 24).

[234] Alp Yurtsever, Suvrit Sra, and Volkan Cevher. "Conditional Gradient Methods via Stochastic Path-Integrated Differential Estimator". In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. Long Beach, California, USA: PMLR, June 2019, pp. 7282–7291. URL: http://proceedings.mlr.press/v97/yurtsever19b.html (cit. on p. 46).

[235] Alp Yurtsever, Joel A Tropp, Olivier Fercoq, Madeleine Udell, and Volkan Cevher. "Scalable semidefinite programming". In: *SIAM Journal on Mathematics of Data Science* 3.1 (2021), pp. 171–200 (cit. on pp. 9, 33).

[236] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. "Understanding deep learning (still) requires rethinking generalization". In: *Communications of the ACM* 64.3 (2021), pp. 107–115 (cit. on p. 3).

[237] Mingrui Zhang, Zebang Shen, Aryan Mokhtari, Hamed Hassani, and Amin Karbasi. "One sample stochastic frank-wolfe". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2020 (cit. on p. 19).

[238] Mingrui Zhang, Zebang Shen, Aryan Mokhtari, Hamed Hassani, and Amin Karbasi. "One Sample Stochastic Frank-Wolfe". In: *arXiv preprint arXiv:1910.04322* (2019) (cit. on p. 19).

[239] Richard Zhang and Daniel Golovin. "Random hypervolume scalarizations for provable multi-objective black box optimization". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 11096–11105 (cit. on p. 50).

[240] Qing Zhao, Stefan E Karisch, Franz Rendl, and Henry Wolkowicz. "Semidefinite programming relaxations for the quadratic assignment problem". In: *Journal of Combinatorial Optimization* (1998) (cit. on p. 16).

[241] Renbo Zhao and Robert M Freund. "Analysis of the Frank–Wolfe method for convex composite optimization involving a logarithmically-homogeneous barrier". In: *Mathematical Programming* (2022), pp. 1–41 (cit. on p. 46).

Curriculum Vitae

Maria-Luiza Vladarean

Education

École Polytechnique Fédérale de Lausanne (EPFL) *Lausanne, Switzerland*
Ph.D. in Computer Science *2018 – 2023*
- Advisor: Nicolas Flammarion
- Thesis: Scalable Constrained Optimization

École Polytechnique Fédérale de Lausanne (EPFL) *Lausanne, Switzerland*
M.Sc. in Computer Science, Minor in Biocomputing *2013 – 2016*
- Thesis: *SwapIt – a Recommender System for Exchange Platforms*, Advisors: Michele Catasta, Karl Aberer
- GPA: 5.66/6.00

University of Bucharest *Bucharest, Romania*
B.Sc. in Computer Science *2010 – 2013*
- Thesis: *Personalized Emotion Recognition of a Photographed Subject*, Advisor: Radu Gramatovici
- GPA: 9.91/10.00 (top 2% of cohort)

Work Experience

09.2018 – 03.2024	**EPFL School of Computer Science**, Graduate Teaching Assistant	Lausanne, Switzerland
01.2017 – 08.2018	**Amazon Video**, Software Engineer, Full-time	London, UK
02.2015 – 09.2015	**Bloomberg**, Software Engineer, Intern	London, UK
07.2011 – 08.2011	**LMS (now Siemens Industry Software)**, Software Engineer, Intern	Brasov, Romania

Fellowships & Awards

2023	**NeurIPS Top Reviewer (top 8%)**, NeurIPS	
2022	**ICML Outstanding Reviewer (top 10%)**, ICML	
2018	**Fellowship for Doctoral Studies**, EPFL IC School	
2013–2016	**Master Excellence Fellowship**, EPFL	*CHF 16,000/ academic year*
2010–2013	**Tuition waiver for undergraduate study**, University of Bucharest	

Teaching

All listed classes were taught at EPFL.

Fall 2022	**Machine learning (CS-433)**, Teaching Assistant	*School of Computer Science*
Fall 2021	**Mathematics of Data (EE-556)**, Teaching Assistant	*School of Engineering*
Spring 2021	**Analysis II (MATH-106C)**, Teaching Assistant	*School of Engineering*
Fall 2020	**Mathematics of Data (EE-556)**, Teaching Assistant	*School of Engineering*
Fall 2019	**Mathematics of Data (EE-556)**, Teaching Assistant	*School of Engineering*
Spring 2019	**Object-oriented programming (CS-108)**, Teaching Assistant	*School of Computer Science*

Publications

* authors contributed equally

Peer-reviewed Conference Publications

2023 Aditya Varre, **Maria-Luiza Vladarean**, Loucas Pillaud-Vivien, and Nicolas Flammarion. *On the spectral bias of two-layer linear networks*. Advances in Neural Information Processing Systems (NeurIPS). Accessible at: https://openreview.net/forum?id=FFdrXkm3Cz.

2023 **Maria-Luiza Vladarean**, Nikita Doikov, Martin Jaggi, and Nicolas Flammarion. *Linearization Algorithms for Fully Composite Optimization*. Conference on Learning Theory (COLT). Accessible at: https://proceedings.mlr.press/v195/vladarean23a.html.

2022 Gideon Dresdner, **Maria-Luiza Vladarean**, Gunnar Rätsch, Francesco Locatello, Volkan Cevher, and Alp Yurtsever. *Faster One-Sample Stochastic Conditional Gradient Method for Composite Convex Minimization*. International Conference on Artificial Intelligence and Statistics (AISTATS). Accessible at: https://proceedings.mlr.press/v151/dresdner22a.

2021 **Maria-Luiza Vladarean**, Yura Malitsky, and Volkan Cevher. *A first-order primal-dual method with adaptivity to local smoothness*. Advances in Neural Information Processing Systems (NeurIPS). Accessible at: https://proceedings.neurips.cc/paper_files/paper/2021/hash/310b60949d2b6096903d7e8a539b20f5-Abstract.html.

2020 **Maria-Luiza Vladarean**, Ahmet Alacaoglu, Ya-Ping Hsieh, and Volkan Cevher. *Conditional gradient methods for stochastically constrained convex minimization*. International Conference on Machine Learning (ICML). Accessible at: https://proceedings.mlr.press/v119/vladarean20a.

2017 Jérémie Rappaz*, **Maria-Luiza Vladarean***, Julian McAuley, and Michele Catasta. *Bartering books to beers: a recommender system for exchange platforms*. International Conference on Web Search and Data Mining (WSDM). Accessible at: https://dl.acm.org/doi/10.1145/3018661.3018696.