[EURO Journal on Computational Optimization 9 (2021) 100015](https://doi.org/10.1016/j.ejco.2021.100015)

Contents lists available at [ScienceDirect](http://www.ScienceDirect.com/)

EURO Journal on Computational Optimization

journal homepage: [www.elsevier.com/locate/ejco](http://www.elsevier.com/locate/ejco)

First-Order Methods for Convex Optimization

Pavel Dvurechensky[a](#_bookmark0),[b](#_bookmark1),[c](#_bookmark2), Shimrit Shtern[d](#_bookmark3), Mathias Staudigl[e](#_bookmark4),[∗](#_bookmark5)

a *Weierstrass Institute for Applied Analysis and Stochastics, Mohrenstr. 39, Berlin 10117, Germany*

b *Institute for Information Transmission Problems RAS, Bolshoy Karetny per. 19, build.1, Moscow, 127051, Russia*

c *Moscow Institute of Physics and Technology, 9 Institutskiy per., Dolgoprudny, Moscow Region, 141701, Russia*

d *Faculty of Industrial Engineering and Management, Technion - Israel Institute of Technology, Haifa, Israel*

e *Maastricht University, Department of Data Science and Knowledge Engineering (DKE) and Mathematics Centre Maastricht (MCM), Paul-Henri Spaaklaan 1, Maastricht 6229 EN, The Netherlands*

a r t i c l e i n f o a b s t r a c t

*2010 MSC:*

90C25

90C30

90C06

68Q25

65Y20

68W40

*Keywords:*

Convex Optimization Composite Optimization First-Order Methods Numerical Algorithms Convergence Rate Proximal Mapping Proximity Operator Bregman Divergence

First-order methods for solving convex optimization problems have been at the forefront of mathematical opti- mization in the last 20 years. The rapid development of this important class of algorithms is motivated by the success stories reported in various applications, including most importantly machine learning, signal process- ing, imaging and control theory. First-order methods have the potential to provide low accuracy solutions at low computational complexity which makes them an attractive set of tools in large-scale optimization problems. In this survey, we cover a number of key developments in gradient-based optimization methods. This includes non-Euclidean extensions of the classical proximal gradient method, and its accelerated versions. Additionally we survey recent developments within the class of projection-free methods, and proximal versions of primal- dual schemes. We give complete proofs for various key results, and highlight the unifying aspects of several optimization algorithms.

### Introduction

The traditional standard in convex optimization was to trans- late a problem into a conic program and solve it using a primal- [dual interior point method (IPM). The monograph Nesterov and Ne- mirovski (1994) was instrumental in setting this standard. The primal-](#_bookmark230) dual formulation is a mathematically elegant and powerful approach as these conic problems can then be solved to high accuracy when the di- mension of the problem is of moderate size. This philosophy culminated into the development of a robust technology for solving convex opti- mization problems which is nowadays the computational backbone of [many specialized solution packages like MOSEK (Andersen and Ander- sen, 2000), or SeDuMi (](#_bookmark112)[Sturm,](#_bookmark220) [1999). However, in general, the iteration](#_bookmark112)

mension. As a result, as the dimension *𝑛* of optimization problems grows, costs of interior point methods grow non-linearly with the problem’s di-

off-the shelve interior point methods eventually become impractical. As an illustration, the computational complexity of a single step of many

standardized IPMs scales like *𝑛*3, corresponding roughly to the complex-

ity of inverting an *𝑛* × *𝑛* matrix. This means that for already quite small problems of size like *𝑛* = 102, we would need roughly 106 arithmetic op-

erations just to compute a single iterate. From a practical viewpoint, such a scaling is not acceptable. An alternative solution approach, par- ticularly attractive for such ”large-scale” problems, are *first-order meth- ods* (FOMs). These are iterative schemes with computationally cheap iterations usually known to yield low-precision solutions within reason- able computation time. The success-story of FOMs went hand-in-hand with the fast progresses made in data science, analytics and machine learning. In such data-driven optimization problems, the trade-off be- tween fast iterations and low accuracy is particularly pronounced, as these problems usually feature high-dimensional decision variables. In these application domains precision is usually considered to be a sub- ordinate goal because of the inherent randomness of the problem data, which makes it unreasonable to minimize with accuracy below the sta- tistical error.

The development of first-order methods for convex optimization problems is still a very vibrant field, with a lot of stimulus from the

∗ Corresponding author.

*E-mail addresses:* [pavel.dvurechensky@wias-berlin.de](mailto:pavel.dvurechensky@wias-berlin.de) (P. Dvurechensky), [shimrits@technion.ac.il](mailto:shimrits@technion.ac.il) (S. Shtern), [m.staudigl@maastrichtuniversity.nl](mailto:m.staudigl@maastrichtuniversity.nl) (M. Staudigl).

<https://doi.org/10.1016/j.ejco.2021.100015>

Received 30 December 2020; Received in revised form 19 August 2021; Accepted 11 October 2021

2192-4406/© 2021 The Author(s). Published by Elsevier Ltd on behalf of Association of European Operational Research Societies (EURO). This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

already mentioned applications in machine learning, statistics, optimal control, signal processing, imaging, and many more, see e.g. the recent [review papers on optimization for machine learning (Bottou et al., 2018; Jain and Kar, 2017; Wright, 2018). Naturally, any attempt to try to sur-](#_bookmark124) vey this lively scientific field is already doomed from the beginning to be a failure, if one is not willing to make restrictions on the topics cov- ered. Hence, in this survey we tried to give a largely self-contained and concise summary of some important families of FOMs, which we believe have had an ever-lasting impact on the modern perspective of continu- ous optimization. Before we give an outline of what is covered in this survey, it is therefore fair to mention explicitly, what is NOT covered in the pages to come. One major restriction we imposed on ourselves is the concentration on *deterministic* optimization algorithms. This is in- deed a significant cut in terms of topics, since the field of stochastic op- timization and randomized algorithms has particularly been at the fore- front of recent progresses made. Nonetheless, we intentionally made this cut, since most of the developments within stochastic optimization al- gorithms are based on deterministic counterparts, and actually in many cases one can think of deterministic algorithms as the mean-field equiv- alent of a stochastic optimization technique. As a well-known example, we can mention the celebrated stochastic approximation theory initi- ated by [Robbins and Monro (1951)](#_bookmark242), with its close connection to deter- ministic gradient descent. See [Benveniste et al. (1990)](#_bookmark113); [Kushner (1984)](#_bookmark185); [Ljung et al. (2012)](#_bookmark199), for classical references from the point of view of systems theory and optimization, and [Benaïm, 1998](#_bookmark114) for its deep con- nection with deterministic dynamical systems. This link has gained sig- nificant relevance in various stochastic optimization models recently [(Davis et al., 2020; Duchi and Ruan, 2018; Mertikopoulos and Staudigl,](#_bookmark167)

timization, mainly because of its good scalability properties and small iteration costs. Conceptually, it is an interesting optimization method, as it allows us to solve convex programming problems with complicated geometry on which proximal operators are not easy to evaluate. This, in fact, applies to many important domains, like the Spectrahedron, or domains defined via intersections of several half spaces. CG is also rele- vant when the iterates should preserve structural features of the desired solution, like sparsity. [Section 5](#_bookmark60) gives a comprehensive account of this versatile method.

All the methods we discussed so far generally provide sublinear con-

ity of *𝑂*(1∕*𝜀*). In his influential paper ([Nesterov, 1983](#_bookmark216)), Nesterov pub- vergence guarantees in terms of function values with iteration complex- lished an optimal method with iteration complexity of *𝑂*(1∕ *𝜀*) to reach an *𝜀*-optimal solution. This was the starting point for the development

*√*

of acceleration techniques for given FOMs. [Section 6](#_bookmark72) summarizes the recent developments in this field.

With writing this survey, we tried to give a holistic presentation of the main methods in use. At various stages in the survey, we establish connections, if not equivalences, between various methods. For many of the key results we provide self-contained proofs.

*Notation* We use standard notation and concepts from convex and variational analysis, which, unless otherwise specified, can all be found [in the monographs (Bauschke and Combettes, 2016; Hiriart-Urrut and](#_bookmark96)

we let 𝖵 represent a finite-dimensional vector space of dimension *𝑛* with [Lemaréchal, 2001; Rockafellar and Wets, 1998). Throughout this article,](#_bookmark96) norm ⋅ . We will write 𝖵∗ for the (algebraic) dual space of 𝖵 with

duality*‖*pa*‖*iring *⟨𝑦, 𝑥⟩* between *𝑦* ∈ 𝖵∗ and *𝑥* ∈ 𝖵. The dual norm of *𝑦* ∈ 𝖵∗

is *𝑦* = sup{ *𝑦, 𝑥 𝑥* ≤ 1}. The set of proper lower semi-continuous

*⟨*

functions *𝑓* ∶ 𝖵 → (−∞*,* ∞] is denoted as Γ0(𝖵). The (effective) domain of a function *𝑓* ∈ Γ (𝖵) is defined as dom *𝑓* = {*𝑥* ∈ 𝖵 *𝑓* (*𝑥*) *<* ∞}. For a

*‖ ‖*∗ *⟩| ‖ ‖*

[2018a; 2018b). Some excellent references on stochastic optimization](#_bookmark167) are [Shapiro et al., 2009](#_bookmark250) and [Lan, 2020](#_bookmark188). Furthermore, we excluded some

important classes of alternating minimization methods, such as block- coordinate descent, and variations thereof. Section 14 in the beautiful book ([Beck, 2017](#_bookmark97)) gives a thorough account of these methods, and we urge the interested reader to start reading there.

So, what is it that we actually do in this survey? Four seemingly dif-

0

given continuously differentiable function *𝑓*

gradient vector

*(*  *𝜕𝑓*  *𝜕𝑓 )⊤*

∇*𝑓* (*𝑥*1*,* … *, 𝑥𝑛* ) =

*,* … *,*

*𝜕𝑥*1 *𝜕𝑥𝑛*

*.*

∶ 𝖷

*⊆* 𝖵 →*|*

ℝ we denote its

ferent optimization algorithms are surveyed, all of which belong now to the standard toolkit of mathematical programmers. After introduc- ing the (standard) notation that will be used in this survey, we give a precise formulation of the model problem for which modern convex optimization algorithms are developed. In particular, we focus on the general composite convex optimization model, including smooth and non-smooth terms. This model is rich enough to capture a significant class of convex optimization problems. Non-smoothness is an impor- tant feature of the model, as it allows us to incorporate constraints via penalty and barrier functions. Non-smooth optimization methods also gained a lot of attention in statistical and machine learning where regu- larization functions are usually included in the estimation part in order to promote sparsity or other a-priori relevant information about the es- timator to be obtained. An eﬃcient way to deal with non-smoothness is provided by the use of *proximal operators*, a key methodological contri- bution born within convex analysis (see [Rockafellar and Wets (1998)](#_bookmark247) for an historical overview). [Section 3](#_bookmark12) introduces the general non-Euclidean proximal setup, which describes the mathematical framework within which the celebrated *Mirror Descent* and *Bregman proximal gradient meth- ods* are analyzed nowadays. These tools achieved extreme popularity in [online learning and convex optimization (Bubeck, 2015; Juditsky and Nemirovski, 2011a; 2011b). The main idea behind this technology is to](#_bookmark129) exploit favorable structure in the problem’s geometry to boost the prac- tical performance of gradient-based methods. The proximal revolution has also influenced the further development of primal-dual optimization methods based on augmented Lagrangians. We review proximal variants of the celebrated Alternating Direction Method of Multipliers (ADMM) in [Section 4](#_bookmark52). We then move on to give an in-depth presentation of projection-free optimization methods based on linear minimization ora-

cles, the classical *Conditional Gradient* (CG) (a.k.a Frank-Wolfe) method

The subdifferential at a point *𝑥* ∈ 𝖷 *⊆* 𝖵 of a convex function *𝑓* ∶ 𝖵 →

ℝ ∪ {+∞} is denoted as

*𝜕𝑓* (*𝑥*) = {*𝑝* ∈ 𝖵∗ *𝑓* (*𝑦*) ≥ *𝑓* (*𝑥*) + *𝑝, 𝑦* − *𝑥* ∀*𝑦* ∈ 𝖵}*.* (1.1)

*| ⟨ ⟩*

The elements of *𝜕𝑓* (*𝑥*) are called subgradients.

Given some set 𝖷 *⊆* 𝖵, denote its relative interior as relint(𝖷). Recall As a notational convention, we write matrices in bold capital fonts.

that, if the dimension of the set 𝖷 agrees with the dimension of the

interior, which we denote as int(𝖷). Hence, the two notions differ only ground space 𝖵, then the relative interior coincides with the topological

We denote the closure as cl(𝖷). The boundary of 𝖷 is defined in the usual in situations where 𝖷 is contained in a lower-dimensional submanifold. way bd(𝖷) = cl(𝖷) ⧵ int(𝖷).

### Composite convex optimization

In this survey we focus on the generic optimization problem

min{Ψ(*𝑥*) ∶= *𝑓* (*𝑥*) + *𝑟*(*𝑥*)}*.* (P)

*𝑥*∈𝖷

At many stages of this survey, the following properties are imposed on the data of the minimization problem [(P)](#_bookmark6):

### Assumption 1.

1. 𝖷 *⊆* 𝖵 is a nonempty closed convex set embedded in a finite- dimensional real vector space 𝖵;
2. *𝑓* ∶ 𝖵 → ℝ is convex and continuously differentiable on a neighbor- hood of 𝖷. Furthermore, it possesses a *𝐿𝑓* -Lipschitz continuous gra-

dient on 𝖷:

*‖ 𝑓*

– ′

*‖*∗

and its recent variants. CG gained extreme popularity in large-scale op-

(∀*𝑥, 𝑥*′ ∈ 𝖷) ∶

∇ (*𝑥*) − ∇*𝑓* (*𝑥*′)

≤ *𝐿*

*𝑓 ‖𝑥 𝑥 ‖*

(2.1)

1. *𝑟* ∈ Γ0(𝖵) and *𝜇*-strongly convex on 𝖵 for some *𝜇* ≥ 0 with respect to a norm ⋅ on 𝖵. This means that for all *𝑥, 𝑦* ∈ dom *𝑟*, and any selection *𝑟* (*𝑥*) ∈ *𝜕𝑟*(*𝑥*), we have

′*‖ ‖*

*𝑟*(*𝑦*) ≥ *𝑟*(*𝑥*) + *𝑟*′(*𝑥*)*, 𝑦* − *𝑥* + *𝜇 𝑥* − *𝑦* 2*.*

*⟨ ⟩ ‖ ‖*

2

If *𝜇* = 0 then the function *𝑟* is called convex.

We are interested in problems which are feasible.

**Assumption 2.** dom *𝑟* ∩ 𝖷 ≠ ∅.

In many recent applications, the smooth function *𝑓* represents a data fidelity term and the non-smooth part *𝑟* takes the role of a penalty func- tion or regularizer. The most important examples of function *𝑟* are as

follows:

[Wajs (2005)](#_bookmark157). This led to a rich interplay between convex programming on the one hand and machine learning and signal/image processing on the other hand. Indeed, several work-horse models in these application domains are of the composite type

Ψ(*𝑥*) = *𝑔*(**𝐀***𝑥*) + *𝑟*(*𝑥*) (2.7)

where *𝑔* ∶ 𝖤 → ℝ is a smooth function defined on a finite-dimensional set 𝖤 (usually of lower dimension than 𝖵), and **𝐀** ∈ B𝖫(𝖵*,* 𝖤) is bounded linear operator mapping points *𝑥* ∈ 𝖵 to elements **𝐀***𝑥* ∈ 𝖤. Convexity al-

lows us to switch between primal and dual formulations freely, so that the above problem can be equivalently considered as a convex-concave minimax problem

min max{*𝑟*(*𝑥*) + **𝐀***𝑥, 𝑦* − *𝑔*∗(*𝑦*)} (2.8)

* *𝑟* is an indicator function of a closed convex set 𝖢 *⊂* 𝖵 with 𝖢 ∩ 𝖷 ≠

*{* (2.2)

*𝑥*∈𝖷 *𝑦*∈𝖤 *⟨ ⟩*

∅:

*𝑟*(*𝑥*) = *𝛿*𝖢(*𝑥*) ∶=

0 if *𝑥* ∈ 𝖢*,*

+∞ if *𝑥* ∉ 𝖢*.*

Such minimax formulations have been of key importance in signal pro- [cessing and machine learning (Juditsky et al., 2013; Juditsky and Ne- mirovski, 2011b), game theory (](#_bookmark202)[Sorin,](#_bookmark215) [2000), decomposition methods](#_bookmark202) ([Tseng, 1991](#_bookmark234)) and its very recent innovation around generative adver-

* [*𝑟* is a self-concordant barrier (Nesterov, 2018b; Nesterov and Ne-](#_bookmark228)

[≠](#_bookmark228)

[mirovski, 1994) for a closed convex set 𝖢 *⊂* 𝖵 with 𝖢 ∩ 𝖷 ∅.](#_bookmark228)

* *𝑟* is a nonsmooth convex function with relatively simple structure.

For example, it could be a norm regularization like the celebrated

𝓁 -regularizer *𝑟*(*𝑥*) = *𝑥 .* This regularizer plays a fundamental role

1 [*‖ ‖*](#_bookmark126)1

in high-dimensional statistics ([Bühlmann and van de Geer, 2011](#_bookmark131)) and signal processing ([Bruckstein et al., 2009; Daubechies et al., 2004](#_bookmark126)).

sarial networks ([Goodfellow et al., 2014](#_bookmark184)).

Another canonical class of optimization problems in machine learn- ing is the finite-sum model

Ψ(*𝑥*) = 1 *∑ 𝑓* (*𝑥*) + *𝑟*(*𝑥*)*,* (2.9)

*𝑁*

*𝑁*

*𝑖*=1

*𝑖*

*cone* associated with the closed convex set 𝖷 *⊆* 𝖵 as For characterizing solutions to our problem [(P)](#_bookmark6), define the *tangent*

𝖳𝖢 (*𝑥*) ∶= *{*{*𝑣* = *𝑡*(*𝑥*′ − *𝑥*)*|𝑥*′ ∈ 𝖷*, 𝑡* ≥ 0} *⊆* 𝖵 if *𝑥* ∈ 𝖷*,*

𝖷

∅ else

which comes from supervised learning, where *𝑓𝑖* (*𝑥*) corresponds to the loss incurred on the *𝑖*-th data sample using a hypothesis parameter- ized by the decision variable *𝑥*. Typically, *𝑁* is an extremely large

number as it corresponds to the size of the data set. The recent liter-

ature on variance reduction techniques and distributed optimization is

and the *normal cone*

*{*{*𝑝* ∈ 𝖵∗*|* sup

*⟨𝑝, 𝑣⟩*

𝖭𝖢𝖷(*𝑥*) ∶=

∅

*𝑣*∈𝖳𝖢𝖷(*𝑥*)

≤ 0} if *𝑥* ∈ 𝖷

else.

very active in making such large scale optimization problems tractable. Surveys on the latest developments in these fields can be found in

[Gower et al. (2020)](#_bookmark191) and the recent comprehensive textbook [Lan (2020)](#_bookmark188).

We remark that *𝜕𝛿*𝖷(*𝑥*) = 𝖭𝖢𝖷(*𝑥*) for all *𝑥* ∈ 𝖷.

Given the feasible set 𝖷 *⊆* 𝖵, we denote the value function

Ψ (𝖷) ∶= inf Ψ(*𝑥*)*.* (2.3)

min

*𝑥*∈𝖷

We are focusing in this survey on problems which are solvable.

### The Proximal Gradient Method

* 1. *Motivation*

In the context of the composite optimization problem [(P)](#_bookmark6), a classical

**Assumption 3.** 𝖷∗ ∶= {*𝑥* ∈ 𝖷 Ψ(*𝑥*) = Ψ

*|*

min

(𝖷)} ≠ ∅*.*

and very powerful idea is to construct numerical optimization meth- ods by exploiting problem structure. Following this philosophy, we de-

Given the standing hypothesis on the functions *𝑓* and *𝑟*, it is easy

to see that 𝖷∗ is always a closed convex set. Moreover, if *𝜇 >* 0, then problem [(P)](#_bookmark6) is *strongly convex*, and so 𝖷∗ is a singleton.

Optimality conditions for problem [(P)](#_bookmark6) can be formulated using dif- ferential calculus tools from convex analysis. [(P)](#_bookmark6) can be treated as an

linearization of the smooth part, the non-smooth part *𝑟* ∈ Γ0(𝖵), and a termine the position of the next iterate by minimizing the sum of the quadratic regularization term with weight *𝛾 >* 0:

*𝑥*+(*𝛾*) = argmin{*𝑓* (*𝑥*) + ∇*𝑓* (*𝑥*)*, 𝑢* − *𝑥* + *𝑟*(*𝑢*) + 1 *𝑢* − *𝑥* 2}*.* (3.1)

unconstrained problem by augmenting the objective function Ψ = *𝑟* + *𝑓*

*𝑢*∈𝖷 *⟨*

*⟩* 2*𝛾 ‖ ‖*2

by the non-smooth penalty *𝛿*𝖷. Our main problem becomes then

min(*𝑓* (*𝑥*) + *𝑟*(*𝑥*) + *𝛿* (*𝑥*))*.* (2.4)

*𝑥*∈𝖵

𝖷

Fermat’s rule says that *𝑥*∗ ∈ 𝖵 is a solution to [(2.4)](#_bookmark15) if and only if 0 ∈

*𝜕*Ψ(*𝑥*∗) + 𝖭𝖢𝖷(*𝑥*∗) ([Rockafellar and Wets (1998](#_bookmark247), Theorem 8.15)). In gen-

eral, the subdifferential operator is not linear, and we only have a ”fuzzy

sum rule” *𝜕*Ψ(*𝑥*) *⊆* ∇*𝑓* (*𝑥*) + *𝜕𝑟*(*𝑥*). However, we know that *𝑟* is finite at *𝑥*

∗

Disregarding terms which do not influence the computation of the so-

set constraint into the non-smooth part by defining *𝜙*(*𝑥*) = *𝑟*(*𝑥*) + *𝛿*𝖷(*𝑥*), lution of this strongly convex minimization problem, and absorbing the

we see that [(3.1)](#_bookmark14) can be equivalently written as

*𝑥*+ (*𝛾*) = argmin *{𝛾𝜙*(*𝑢*) + 1 *‖𝑢* − (*𝑥* − *𝛾*∇*𝑓* (*𝑥*))*‖*2 *}.* (3.2)

*𝑢*∈𝖵

2

2

and *𝑓* is smooth on a neighborhood containing 𝖷 ([Assumptions 1](#_bookmark7)(b)). Hence, by [Rockafellar and Wets (1998](#_bookmark247), Exercise 8.8c), we have *𝜕*Ψ(*𝑥*∗) =

∇*𝑓* (*𝑥*∗) + *𝜕𝑟*(*𝑥*∗). Therefore, Fermat’s optimality condition becomes

0 ∈ ∇*𝑓* (*𝑥*∗) + *𝜕𝑟*(*𝑥*∗) + 𝖭𝖢𝖷(*𝑥*∗)*.* (2.5)

This means that there exists *𝜉* ∈ *𝜕𝑟*(*𝑥*∗) such that

interesting geometric principles acting here. Indeed, if *𝑟* would be a This way of writing the updating scheme immediately reveals some

finite constant on 𝖷 (say 0 for concreteness), then the rule [(3.2)](#_bookmark16) is nothing else than the Euclidean projection of the directional vector

*𝑥* − *𝛾*∇*𝑓* (*𝑥*) onto the set 𝖷. In this case, the minimization routine re-

turns the classical projected gradient step *𝑥*+(*𝛾*) = *𝑃*𝖷(*𝑥* − *𝛾*∇*𝑓* (*𝑥*)), where

∇ (*𝑥*∗) + *𝜉, 𝑣* ≥ 0 ∀*𝑣* ∈ 𝖳𝖢 (*𝑥*∗)*.* (2.6) 𝖷

*⟨ 𝑓 ⟩* 𝖷

*𝑃* (*𝑥*) = argmin

1

*𝑦* − *𝑥* 2. Iterating the map *𝑥* ↦ *𝑃* ◦(Id −*𝛾*∇*𝑓* ) gen-

*𝑦*∈𝖷 2 *‖ ‖*2 𝖷

The structured composite optimization problem [(P)](#_bookmark6) has attracted a lot of interest in convex programming over the last 20 years moti- vated by a number of important applications, see, e.g. [Combettes and](#_bookmark157)

cases where the non-smooth function *𝑟* is non-trivial over the domain erates the classical *gradient projection method*. A new obstacle arises in

𝖷. A fundamental idea, going back to [Moreau (1965)](#_bookmark207), is to define the

*proximity operator* Prox*𝜙* ∶ 𝖵 → 𝖵 associated with a function *𝜙* ∈ Γ0(𝖵)

as[1](#_bookmark23)

Prox (*𝑥*) ∶= argmin *{𝜙*(*𝑢*) + 1 *‖𝑢* − *𝑥‖*2*}.* (3.3)

*𝜙*

*𝑢*∈𝖵

2

2

footprint on the overall iteration complexity of the method, as we will demonstrate in this section. The point of departure of Bregman Proximal

of the domain 𝖷, see e.g. [Auslender and Teboulle (2009)](#_bookmark133). This can not only positively affect the per-iteration complexity, but also will have a

**Remark 3.1.** The classical Moreau proximity operator of *𝜙* is, in gen- eral, explicitly computable when *𝜙* is norm like, or when *𝜙* is the in-

dicator function of sets whose geometry is favorable to Euclidean pro- jections. Although quite frequent in applications (orthant, second-order

cone, 𝓁1 norm), these prox-friendly functions are very scarce, see, e.g.,

[Combettes and Wajs (2005](#_bookmark157), Section 2.6). A significant improvement will

be made in [Section 3.2](#_bookmark20), where a general Bregman proximal framework will be introduced.

The value function

1

algorithms is to introduce a *distance generating function ℎ* ∶ 𝖵 → (−∞*,* ∞], which is a barrier-type mapping suitably chosen to capture geometric

features of the set 𝖷.

**Definition 3.1.** Let 𝖷 be a closed convex subset of 𝖵. We say that *ℎ* ∈ Γ0(𝖵) is a *distance generating function (DGF)* with modulus *𝛼 >* 0 with

respect to ⋅ on 𝖷 if

*‖ ‖*

1. *ℎ* is closed, convex and proper;
2. 𝖷 *⊆* dom *ℎ*;

≠

1. the set 𝖷◦ = {*𝑥* ∈ 𝖷 *𝜕ℎ*(*𝑥*) ∅} is nonempty and convex;

*𝜙* (*𝑥*) = inf{*𝜙*(*𝑢*) +

– 2}

*𝑢*

*𝛾*

2*𝛾*

*‖𝑢 𝑥‖*

1. *ℎ* restricted to 𝖷◦

is*|*continuously differentiable and strongly convex

is called the *Moreau envelope* of the function *𝜙*, and is an important

smoothing and regularization tool, frequently employed in numerical

analysis. Indeed, for a function *𝜙* ∈ Γ (𝖵) and *𝛾 >* 0, its Moreau enve-

under the norm ⋅ with parameter *𝛼*:

(∀*𝑥, 𝑥*′ ∈ 𝖷◦) ∶ ∇ (*𝑥*) − ∇*ℎ*(*𝑥*′)*, 𝑥* − *𝑥*′ ≥

*⟨ ℎ*

*‖*

*𝛼*

*⟩*

*‖ ‖*

*𝛼‖𝑥*

– *𝑥*′ 2*.*

0

lope is finite everywhere, convex and has *𝛾*−1-Lipschitz continuous gra- [dient on 𝖵 given by ∇*𝜙𝛾* (*𝑥*) = 1 (*𝑥* − Prox*𝛾𝜙*(*𝑥*)) Bauschke and Combettes](#_bookmark96)

[*𝛾*](#_bookmark96)

[(2016, Prop. 12.30).](#_bookmark96)

In the context of minimization the composite model Ψ = *𝑓* + *𝑟*, the

gradient-based methods. Choosing *𝜙* = *𝑟* + *𝛿*F in [(3.3)](#_bookmark18), and replacing the proximity operator is the key actor in generating a large family of generic input with the specific input *𝑥* − *𝛾*∇*𝑓* (*𝑥*), we are in the frame-

work of the *Proximal Gradient Method* (PGM). PGM is a very powerful method which received enormous interest in optimization and its ap- plications. For a survey in the context of signal processing we refer the reader to [Combettes and Pesquet (2011)](#_bookmark155). A general survey on proximal operators can be found in [Beck (2017)](#_bookmark97); [Parikh and Boyd (2014)](#_bookmark237).

We denote by  (𝖷) the set of DGFs on 𝖷.

[(1970, Section 23-25), we know that dom(*𝜕ℎ*) *⊂* dom *ℎ*. Hence, 𝖷◦ = From classical differential theory of convex functions Rockafellar](#_bookmark245) dom(*𝜕ℎ*) ∩ 𝖷 is contained in the set dom *ℎ* ∩ 𝖷, which in turn agrees with

𝖷 thanks to property (b). Restricted to 𝖷◦ the DGF *ℎ* is continuously

differentiable.

In many proximal settings we are interested in DGFs which act as

function *ℎ* are captured by its scaling near bd(𝖷), usually encoded in barriers on the feasible set 𝖷. Naturally, the barrier properties of the

terms of the notion of *essential smoothness* ([Rockafellar (1970](#_bookmark245), Section 26).)

**Definition 3.2** (Essential smoothness)**.** *ℎ* ∈  (𝖷) is *essentially smooth*

*𝛼*

**Input:** *𝑥*0 ∈ 𝖷*.* **The Proximal Gradient Method (PGM)**

**General step:** For *𝑘* = 0*,* 1*,* … do: Choose *𝛾𝑘 >* 0.

set *𝑥𝑘*+1 = Prox*𝛾 𝜙 𝑥𝑘* − *𝛾𝑘*∇*𝑓* (*𝑥𝑘*) .

*(*

*)*

if it satisfies the following three conditions:

1. int(dom *ℎ*) ≠ ∅;
2. *ℎ* is differentiable throughout int(dom *ℎ*);
3. lim*𝑖*→∞ *‖*∇*ℎ*(*𝑥𝑖*)*‖* = +∞ whenever *𝑥*1*, 𝑥*2*,* … is a sequence in int(dom *ℎ*)

converging to a boundary point *𝑥* of int(dom *ℎ*).

Given *ℎ* ∈  (𝖷), its *Bregman divergence 𝐷* ∶ dom *ℎ* × dom(*𝜕ℎ*) → ℝ

*𝑘*

*𝛼*

*ℎ*

**Remark 3.2.** In the fully non-smooth case, i.e. when *𝑓* = 0 in our model

is defined as

*𝐷* (*𝑢, 𝑥*) ∶= *ℎ*(*𝑢*) − *ℎ*(*𝑥*) − ∇*ℎ*(*𝑥*)*, 𝑢* − *𝑥 .* (3.4)

problem, PGM reduces to a classical recursion known as the *proximal ℎ* *⟨ ⟩*

The *𝛼*-strong convexity of the DGF ensures that

*point* method:

*𝑥𝑘*+1 = Prox (*𝑥𝑘*) = argmin{*𝜙*(*𝑢*) + 1

*𝑢* − *𝑥𝑘* 2}*.*

(∀*𝑥* ∈ dom(*𝜕ℎ*)*,* ∀*𝑢* ∈ dom *ℎ*) ∶ *𝐷* (*𝑢, 𝑥*) ≥ *𝛼*

*𝑢* − *𝑥* 2*.* (3.5)

*𝑘 𝑢*∈𝖵

*𝛾 𝜙*

2*𝛾𝑘 ‖ ‖*

*ℎ* 2 *‖ ‖*

This scheme has been first proposed by [Martinet (1970)](#_bookmark203) and [Rockafellar (1976b)](#_bookmark248).

* 1. *Bregman Proximal Setup*

Hence, *𝐷ℎ*(*𝑥, 𝑥*) = 0 for *𝑥* ∈ dom(*𝜕ℎ*), but in general it is not a symmetric

function and it does not satisfy a triangle inequality. This disqualifies

*𝐷ℎ* from carrying the label of a metric, but it can still be interpreted as

a distance measure.

*𝛼* 𝖷 is known to be differentiable on 𝖵

with a -Lipschitz continuous

*𝛼*

∗

 *⟨ ⟩*

The convex conjugate *ℎ* (*𝑦*) = sup*𝑥*∈𝖵{ *𝑥, 𝑦* − *ℎ*(*𝑥*)} for a function *ℎ* ∈ ( ) ∗ 1

the 𝓁 -norm 1 *𝑢* − 2

The basic idea behind non-Euclidean extensions of PGM is to replace

gradient ([Rockafellar and Wets (1998](#_bookmark247), Proposition 12.60)):

2 2 *‖ 𝑥‖* by a different distance-like function which is tai-

lored to the geometry of the feasible set 𝖷 *⊆* 𝖵. These non-Euclidean

*ℎ*∗(*𝑦* ) ≤ *ℎ*∗(*𝑦* ) + ∇*ℎ*∗(*𝑦* )*, 𝑦*

+ 1

2 *.* (3.6)

distance-like functions that will be used are *Bregman divergences*. The

– *𝑦*

*𝑦*

– *𝑦*

2 1 *⟨*

1 2 1*⟩*

2*𝛼 ‖* 2

1*‖*∗

transition from Euclidean to non-Euclidean distance measures is mo- tivated by the usefulness and flexibility of the latter in computational perspectives and potentials for improving convergence properties for specific application domains. In particular, the move from Euclidean to non-Euclidean distance measures allows to adapt the algorithm to the underlying geometry, typically explicitly embodied in the description

for all *𝑦*1*, 𝑦*2 ∈ 𝖵∗.

It will be instructive to go over some standard examples of dis-

tance generating functions. See also [Combettes and Wajs (2005)](#_bookmark157), [Bauschke and Combettes (2016)](#_bookmark96), and [Ben-Tal and Nemirovski (2020)](#_bookmark111).

**Example 3.1** (Euclidean Projection)**.** We begin by revisiting the 𝓁2- projection on some closed convex subset 𝖷 *⊂* 𝖵 = ℝ*𝑛*. Letting *ℎ*(*𝑥*) =

1 2

∈

1 The repository <http://proximity-operator.net/index.html> provides codes

◦ = 𝖷. Moreover, for *𝑥* ∈ 𝖷◦,

2 *‖𝑥‖*2 for *𝑥*

and explicit expressions for proximity operators of many standard functions. A [useful MATLAB implementation of proximal methods is described in Beck and](#_bookmark98)

the DGF is 1-strongly convex and continuously differentiable with

∇*ℎ*(*𝑥*) = *𝑥*. The associated Bregman divergence is the Euclidean distance

𝖵, we readily see that 𝖷

[Guttmann-Beck (2019).](#_bookmark98)

*𝐷* (*𝑢, 𝑥*) = 1 *𝑢* − *𝑥* 2 for all *𝑢, 𝑥* ∈ 𝖷.

*ℎ* 2 *‖ ‖*2

**Example 3.2** (Ent*∑*ropic Regularization)**.** Let 𝖷 = {*𝑥* ∈ ℝ*𝑛 | ∑𝑛*

*𝑛*

+ *𝑖*=1

= 1} denote the unit simplex in =

*𝑥𝑖* =

([Bauschke et al., 2003; Censor and Zenios, 1992](#_bookmark93)) onto the set 𝖷. It should

ℝ+ *𝑥*

1} =

*𝑛* ∩{ ∈

*𝑛*

ℝ *| 𝑖*=1 *𝑥𝑖*

𝖵 ℝ*𝑛*. De-

fine the function *𝜓* ∶ ℝ → [0*,* ∞] as

be pointed out that under the standard Euclidean setup described in

*⎧⎪𝑡* ln(*𝑡*) − *𝑡* if *𝑡 >* 0*,*

[Example 3.1](#_bookmark22) the Bregman proximal operator boils down to the Moreau proximal operator [(3.3)](#_bookmark18). As we already alluded to, the main rationale

for the introduction of Bregman proximal operators is that it allows us

*𝜓* (*𝑡*) =

*⎨*

0 if *𝑡* = 0*,*

*⎪⎩*+∞ else.

of 𝖷. Below, we give examples for which Prox*ℎ* (*𝑥*) is easy to compute in to define a projection framework which can be adapted to the geometry

*∑*

*𝜙*

*𝑛*

*𝑖*=1

*𝜓* (*𝑥𝑖* ). En-

closed form, whereas the standard Moreau proximal map is not explic- itly known (and would thus require a numerical procedure, implying a

dowing the ground space 𝖵 with the 𝓁1 norm *‖* ⋅ *‖ ‖* ⋅ *‖*1, i*∑*t can be

As DGF consider the *Boltzmann-Shannon entropy ℎ*(*𝑥*) ∶=

=

shown that *ℎ* ∈  (𝖷) with dom *ℎ* = ℝ*𝑛*

and 𝖷◦ = {*𝑥* ∈ ℝ*𝑛*

*𝑛*

*𝑥* =

nested scheme if used in an algorithm).

1} 1 + ++ *|*

*𝑖*=1 *𝑖*

=

. Indeed, on 𝖷◦ the function *ℎ* is continuously differentiable with

∇*ℎ*(*𝑥*) = [ln(*𝑥*1 )*,* … *,* ln(*𝑥𝑛* )]*⊤*. Furthermore, *𝜕ℎ*(0) = ∅, so that dom(*𝜕ℎ*) =

*𝑛* . The resulting Bregman divergence is the *Kullback-Leibler diver-*

ℝ

++

*gence*

*𝑛* *( 𝑢 ) 𝑛*

*∑*

*∑*

**Example 3.6.** In the following examples we assume that 𝖵 ℝ for sim- plicity.

1. Let *𝜙*(*𝑥*) = *𝛾 𝑥* − *𝜉* where *𝛾, 𝜉 >* 0. Take *ℎ*(*𝑥*) = *𝑥* ln(*𝑥*)*,* dom *ℎ* = [0*,* ∞).

*| |*

Then

*⎧⎪*

*𝐷ℎ* (*𝑢, 𝑥*) =

*𝑖*=1

*𝑢𝑖* ln

*𝑖*

*𝑥𝑖*

+

*𝑖*=1

(*𝑥𝑖* − *𝑢𝑖* )*.*

*∏*

*𝑛*

[*𝑎 , 𝑏* ], where

*𝑏*, define the *Fermi-Dirac entropy*

*𝑎, 𝑏 ,*

Prox*𝜙*(*𝑥*) = 21

1 + 4*𝛾𝑥*2 − 1 .

Prox*ℎ* (*𝑥*) =

exp(*𝛾*)*𝑥* if *𝑥 <* exp(−*𝛾*)*𝜉,*

*𝜉* if *𝑥* ∈ [exp(−*𝛾*)*𝜉,* exp(*𝛾*)*𝜉*]*,*

*𝜙*

*⎨*

exp(−*𝛾*)*𝑥* if *𝑥 >* exp(*𝛾*)*𝜉.*

*⎪*

1. Let *𝜙*(*𝑥*) = *𝛾 𝑥*2 for *𝛾 >* 0, and *ℎ*(*𝑥*) = − ln(*𝑥*)*,* dom *ℎ* = (0*,* ∞). Then

0 ≤ ≤

**Example 3.3** (Box Constraints)**.** Let 𝖵 = ℝ*𝑛* and 𝖷 =

*𝑎𝑖*

0 ≤ ≤

*𝑖*=1 *𝑖 𝑖 ⎩*

*𝛾 𝑥*

*𝑎*

( − ) ln( −

*𝑏𝑖* . Given parameters

*⎧⎪ 𝑡*

*𝑎*

*𝑡*

) + (

– ) ln( − )

∈ ( )

2 *(√ )*

*𝜓𝑎,𝑏*

*𝑎*

*𝑡*

if *𝑡*

(*𝑡*) ∶=

0 if *𝑡* ∈ {*𝑎, 𝑏*}*,*

*⎪*+∞

*⎨*

*𝑏*

*𝑡*

*𝑏*

Note that f*⎩*or *𝑡* ∈ (*𝑎, 𝑏*), we have *𝜓* ′

*𝑎,𝑏*

*𝑏*−*𝑡*

else

(*𝑡*) = ln *( 𝑡*−*𝑎 )*, and *𝜕𝜓𝑎,𝑏* (*𝑡*) = ∅ for *𝑡* ∈

* 1. *Bregman proximal gradient method*

For solving our main problem [(P)](#_bookmark6), a special selection of the function

ℝ ⧵ (*𝑎, 𝑏*). Accordin*∏*gly, the function *ℎ*(*𝑥*) = *∑𝑛*

0 𝖷 *⟨ ⟩*

dom

◦

*𝑛*

◦

*𝑖*=1

*𝑖 𝑖*

gradient *𝛾*∇*𝑓* (*𝑥*) with a general dual vector *𝑦* ∈ 𝖵∗, we obtain the *prox-*

*𝑖*=1 (*𝑎𝑖 , 𝑏𝑖* ). On 𝖷 , the gradient mapping is ∇*ℎ*(*𝑥*) =

*𝜓𝑎 ,𝑏* (*𝑥𝑖* ) is a DGF on 𝖷 =

*𝜙* ∈ Γ (𝖵) in [(3.7)](#_bookmark26) is *𝜙*(*𝑢*) = *𝛾*(*𝑟* + *𝛿* )(*𝑢*) + *𝛾*∇*𝑓* (*𝑥*)*, 𝑢* − *𝑥* . Replacing the

*𝜓* ′

[

*𝑎*1 *,𝑏*1

*𝑥 , , 𝜓* ′

*𝑎𝑛 ,𝑏𝑛*

( 1) …

*ℎ* with 𝖷

=

(*𝑥𝑛* )]*⊤*.

*mapping*

P *ℎ*

(*𝑥, 𝑦*) ∶= argmin{*𝛾𝑟*(*𝑢*) + *𝑦, 𝑢* − *𝑥* + *𝐷* (*𝑢, 𝑥*)}*.* (3.8)

*⟨ ⟩*

**Example 3.4** (Semidefinite Constraints)**.** Let 𝖵 = S*𝑛* be the set of real

symmetric matrices and 𝖷 = S*𝑛* be the cone of real symmetric positive

*𝛾𝑟*

*𝑢*∈𝖷

*ℎ*

* ∗

+

**𝐀 𝐁**

*⟨ ⟩*

= tr(**𝐀𝐁**).

The prox-mapping takes as inputs a ”primal-dual” pair (*𝑥, 𝑦*) ∈ 𝖷

× 𝖵

semi-definite matrices equipped with the inner product *,*

Define the negative von Neumann entropy *ℎ*(**𝐗**) = tr[**𝐗** log(**𝐗**)], which

entropy. It can be verified that dom *ℎ* = 𝖷 and ∇*ℎ*(**𝐗**) = log(**𝐗**) + **𝐈** for can be seen as the matrix-equivalent of the negative Boltzmann-Shannon

**𝐗** ∈ S*𝑛* . Hence, dom *ℎ* = 𝖷, and 𝖷◦ = S*𝑛* , the cone of positive definite matrices. For **𝐗** ∈ S++, the corresponding Bregman divergence is given

++ ′ *𝑛* ++

by

*𝐷ℎ* (**𝐗**′*,* **𝐗**) = tr[**𝐗**′ log(**𝐗**′) − **𝐗**′ log(**𝐗**) + **𝐗**′ − **𝐗**]

where *𝑥* is the current iterate, and *𝑦* is a dual variable representing a

“gradient signal” we obtain on the smooth part of the minimization

problem [(P)](#_bookmark6) (usually obtained after consulting a black-box oracle). Var- ious conditions on the well-posedness of the prox-mapping have been stated in the literature. We will not repeat them here, but rather refer to the recent survey ([Teboulle, 2018](#_bookmark228)). Below we give some examples.

**Example 3.7** (Moreau Proximal Operator)**.** Let 𝖵 = ℝ*𝑛* and 𝖷 a

nonempty, closed and convex set in 𝖵. Let ⋅ = ⋅ , and *ℎ*(*𝑥*) =

*‖ ‖ ‖ ‖ ‖*2

1

mains. Pinsker’s inequality ([Cesa-Bianchi and Lugosi (2006)](#_bookmark144) ) says that *ℎ* See [Doljansky and Teboulle (1998)](#_bookmark172) for further examples on matrix do-

*𝑥‖*2. The prox-mapping [(3.8)](#_bookmark25) reduces in this case to

2

is 1 -strongly convex with respect to the nuclear norm *‖***𝐗***‖*1 = *∑ |𝜆* (**𝐗**)*|*

2

= {

P

*ℎ*

2

*𝑛*

*𝑖*

*𝑖*

for **𝐗** ∈ S , i.e.

**Example 3.8** (Simplex Constraints)**.** Let 𝖵 = ℝ*𝑛* with 𝓁 -norm

*𝑥*

ℝ+

*𝑖*=1 *𝑥𝑖*

*𝛾𝑟*(*𝑥, 𝑦*) = Prox*𝛾*(*𝑟*+*𝛿*𝖷 )(*𝑥* − *𝑦*)*.*

*‖* ⋅ *‖* . Consider the set 𝖷

∈ *𝑛 | ∑𝑛*

1

= 1}

⋅

=

and endow this set

*‖ ‖*

*𝐷* (**𝐗** *,* **𝐗**) ≥ **𝐗** − **𝐗** *.*

*ℎ ‖ ‖*1

′

1

′

1

2

*𝑖 ∑*

*𝑖*=1

a standard calculation gives rise to the prox-mapping

with the Boltzmann-Shannon entropy *ℎ*(*𝑥*) = *∑𝑛*

*𝑥𝑖* ln(*𝑥𝑖* ). For *𝑟*(*𝑥*) = 0,

{*𝑥* ∈ 𝖵 *𝑥 >* (*𝑥*2 + …+ *𝑥*2

**Example 3.5** (2nd order cone constraints)**.** Let 𝖵 = ℝ*𝑛* and *𝐿𝑛*

∶=

)1∕2 }

++ *𝑥 𝑒𝑦𝑖*

*| 𝑛* 1

Let 𝖷 = cl(*𝐿𝑛*

). Denote by **𝐉***𝑛* be the *𝑛* × *𝑛* diagonal matrix with −1

++

*𝑛*−1

the interior of the second-order cone.

[P *ℎ* (*𝑥, 𝑦*)] = *𝑖*

1 ≤ *𝑖* ≤ *𝑛, 𝑥* ∈ 𝖷*, 𝑦* ∈ 𝖵∗*.* (3.9)

– ln( **𝐉** *𝑥, 𝑥* ) + *𝛼 𝑥* 2. Then *ℎ* ∈  (𝖷) with dom *ℎ* = 𝖷◦ = *𝐿𝑛*

*𝛿*𝖷

*𝑛*

*𝑗*=1

*𝑥𝑗 𝑒𝑦𝑗*

in its first *𝑛* − 1 diagonal entries and 1 in the last one. Define *ℎ*(*𝑥*) =

*⟨ ⟩ ‖ ‖*

*⊂* 𝖷. The

*𝑛* 2 2 *𝛼*

associated Bregman divergence is

*𝐷* (*𝑥, 𝑢*) = − ln *( ⟨***𝐉***𝑛 𝑥, 𝑥⟩ )* + 2 *⟨***𝐉***𝑛 𝑥, 𝑢⟩* − 2 + *𝛼 ‖𝑥* − *𝑢‖*2 *.*

*⟨ 𝑛*

*⟩*

++ [ponentiated gradient descent (Beck and Teboulle, 2003; Juditsky et al.,](#_bookmark101)

[2005).](#_bookmark101)

This mapping plays a key role in optimization, where it is known as ex-

**Example 3.9** (Box Constraints)**.** Consider the setting introduced in

*ℎ* **𝐉** *𝑢, 𝑢*

*⟨ 𝑛 ⟩*

**𝐉** *𝑢, 𝑢* 2 2

[Example 3.3](#_bookmark24). For *𝑟*(*𝑥*) = 0, one can compute

The proximal framework for general conic constraints has been devel-

oped in [Auslender and Teboulle (2006b)](#_bookmark132).

[P *ℎ* (*𝑥, 𝑦*)] = *𝑎* +

*𝑏𝑖* − *𝑎𝑖*

1 ≤ *𝑖* ≤ *𝑛, 𝑥* ∈ 𝖷*, 𝑦* ∈ 𝖵∗*.*

*𝛿*𝖷

*𝑖 𝑖*

1 + *𝑏𝑖* −*𝑥𝑖* exp(*𝑦* )

Once we endow our set 𝖷 with a DGF, the technology generating a gradient method in this non-Euclidean setting is the *Bregman proximal*

*operator* ([Teboulle, 1992](#_bookmark227)) applied to the function *𝜙* ∈ Γ (𝖵):

*𝑥𝑖* −*𝑎𝑖 𝑖*

If *𝛾 >* 0 is a step-size parameter and *𝑦* = *𝛾*∇*𝑓* (*𝑥*), then we obtain the

P

*Bregman proximal map 𝑇 ℎ*(*𝑥*) ∶= *ℎ* (*𝑥, 𝛾*∇*𝑓* (*𝑥*)) for all *𝑥* ∈ 𝖷. Iterating

0 *𝛾*

*𝛾𝑟*

Prox*ℎ* (*𝑥*) ∶= argmin{*𝜙*(*𝑢*) + *𝐷ℎ* (*𝑢, 𝑥*)}*.* (3.7)

*𝜙 𝑢*∈𝖵

If *𝜙*(*𝑥*) = *𝛿*𝖷(*𝑥*) is the indicator function of the closed convex set 𝖷 *⊂* 𝖵,

then the Bregman proximal operator defines the *Bregman projection*

this map generates a discrete-time dynamical system known as the *Breg- man proximal gradient method* (BPGM).

assumption that the prox-mapping *ℎ*(*𝑥, 𝑦*) can be evaluated eﬃciently **Remark 3.3.** For simple implementation, BPGM relies on the structural

*𝑟*

P

(3≤*.*5) *𝑓* (*𝑢*) − *𝑓* (*𝑥*+) + *𝐿𝑓 𝐷* (*𝑥*+*, 𝑥*)*.*

### The Bregman Proximal Gradient Method (BPGM)



**Input:** *ℎ* ∈ (𝖷). Pick *𝑥*0 ∈ dom(*𝑟*) ∩ 𝖷◦*.*

*𝛼 ℎ*

*𝛼*

**General step:** For *𝑘* = 0*,* 1*,* … do: choose *𝛾 >* 0.

update *𝑥𝑘*+1 = P *ℎ* ( *𝑘* ∇ ( *𝑘* ))

*𝑘*

*𝑥 , 𝛾 𝑓 𝑥* .

Using this estimate in relation [(3.12)](#_bookmark31), we obtain, for all *𝑢* ∈ dom *ℎ* ∩ 𝖷,

*𝛾*(Ψ(*𝑥*+) − Ψ(*𝑢*)) ≤ *𝐷* (*𝑢, 𝑥*) − *𝐷* (*𝑢, 𝑥*+) − *(*1 − *𝛾 𝐿𝑓 )𝐷* (*𝑥*+*, 𝑥*)*.* (3.17)

*ℎ*

*ℎ*

*𝛼*

*ℎ*

*𝛾𝑘 𝑟 𝑘*

If *𝛾* ∈ (0*,*  *𝛼* ], then the above yields

*𝑓*

*𝐿*

*𝛾*(Ψ(*𝑥*+) − Ψ(*𝑢*)) *𝐷* (*𝑢, 𝑥*) − *𝐷* (*𝑢, 𝑥*+)*, 𝑢* ∈ dom *ℎ* ∩ 𝖷*.*

≤

on the trajectory {(*𝑥𝑘, 𝛾* ∇*𝑓* (*𝑥𝑘*)) 0 ≤ *𝑘* ≤ *𝐾* ∈ ℕ∗}. This, often somewhat *ℎ ℎ*

*𝑘* *|*

hidden, assumption is known in the literature as the ”prox-friendliness” assumption, a terminology apparently coined by [Cox et al. (2014)](#_bookmark158)).

*𝑥*

display as

*𝑥 , 𝑥*

*𝑥*

*𝛾*

*𝛾𝑘*

Setting = *𝑘* + = *𝑘*+1 and = , one can reformulate the previous

* + 1. *Basic Complexity Properties*

To assess the iteration complexity of BPGM, let us start with some preparatory estimates. The first-order optimality condition for the point

*𝑥* = P (*𝑥, 𝛾*∇*𝑓* (*𝑥*)) is given by

+ *ℎ*

*𝛾𝑟*

*𝛾* Ψ(*𝑥𝑘*+1) − Ψ(*𝑢*) ≤ *𝐷* (*𝑢, 𝑥𝑘*) − *𝐷* (*𝑢, 𝑥𝑘*+1)*, 𝑢* ∈ dom *ℎ* ∩ 𝖷*.*

Choosing *𝑢* = *𝑥𝑘*, we readily see *𝛾* (Ψ(*𝑥𝑘*+1) − Ψ(*𝑥𝑘*)) ≤ −*𝐷* (*𝑥𝑘, 𝑥𝑘*+1) ≤ 0,

*𝑘 ( ) ℎ ℎ*

*𝑘 ℎ*

i.e. the sequence of function values {Ψ(*𝑥𝑘*)}*𝑘*∈ℕ is non-increasing. On the other hand, for a general reference point *𝑢* ∈ dom *ℎ* ∩ 𝖷, we also see that

*𝑁* −1 *𝑁* −1

0 ∈ *𝛾𝜕𝑟*(*𝑥*+) + *𝛾*∇*𝑓* (*𝑥*) + ∇*ℎ*(*𝑥*+) − ∇*ℎ*(*𝑥*) + 𝖭𝖢𝖷(*𝑥*+)

*∑ (*Ψ(*𝑥𝑘*+1 ) − Ψ(*𝑢*)*)* ≤ *∑*  1 *[𝐷ℎ* (*𝑢, 𝑥𝑘* ) − *𝐷ℎ* (*𝑢, 𝑥𝑘*+1 )*]*

= + +

*𝑘*=0

*𝑘*=0 *𝛾𝑘*

*𝛾*(*𝜕𝑟* + 𝖭𝖢𝖷)(*𝑥* ) + *𝛾*∇*𝑓* (*𝑥*) + ∇*ℎ*(*𝑥* ) − ∇*ℎ*(*𝑥*)*.*

+

–

= 1

*𝐷ℎ* (*𝑢, 𝑥*0 ) −

1 *𝐷ℎ* (*𝑢, 𝑥𝑁* ) +

*𝑁**∑*−2 *(*  1

1 *)𝐷ℎ* (*𝑢, 𝑥𝑘*+1 )*.*

Whence, there exists *𝜉* ∈ *𝜕𝑟*(*𝑥* ) such that, for all *𝑢* ∈ 𝖷,

*𝛾*0

*𝛾𝑁* −1

*𝑘*=0

*𝛾𝑘*+1

*𝛾𝑘*

*⟨𝛾𝜉*

*⟩*

+ *𝛾*∇*𝑓* (*𝑥*) + ∇*ℎ*(*𝑥*+) − ∇*ℎ*(*𝑥*)*, 𝑥*+ − *𝑢*

≤ 0*.* (3.10)

Assuming a constant step size policy *𝛾𝑘* = *𝛾*, this gives us

Via the subgradient inequality for the convex function *𝑥* ↦ *𝜙*(*𝑥*), we ob-

tain for all *𝑢* ∈ 𝖷:

*⟩*

*𝑟*(*𝑥*+) − *𝑟*(*𝑢*) ≤ ∇*𝑓* (*𝑥*)*, 𝑢* − *𝑥*+

+ 1 ∇*ℎ*(*𝑥*+) − ∇*ℎ*(*𝑥*)*, 𝑢* − *𝑥*+

*.* (3.11)

*𝑘*=0

*𝛾 ℎ*

*𝑘 𝑘*

*∑ (*Ψ(*𝑥𝑘*+1) − Ψ(*𝑢*)*)* ≤ 1 0

*𝑁* −1

*𝐷* (*𝑢, 𝑥* )*.*

*𝑘*+1 *𝑘*

*𝑘*+1

*⟨ ⟩ 𝛾 ⟨*

Define the function gap *𝑠*

Ψ(*𝑥𝑘*) 0, and therefore

≤

∶= Ψ(*𝑥* ) − Ψ(*𝑢*), then *𝑠*

– *𝑠*

= Ψ(*𝑥* ) −

For further analysis, we need the celebrated three-point identity, due to [Chen and Teboulle (1993)](#_bookmark148), whose simple proof we omit.

*𝑠𝑁* ≤ 1

*𝑁* −1

*𝑠𝑘*+1 =

*∑* 1

*𝑁* −1

*𝐷ℎ* (*𝑢, 𝑥* )

*∑*[Ψ(*𝑥𝑘*+1) − Ψ(*𝑢*)] ≤ 1 0

**Lemma 3.3** (3-point lemma)**.** *For all 𝑥, 𝑦* ∈ 𝖷◦ *and 𝑧* ∈ dom *ℎ we have*

*𝑁 𝑘*=0

*𝑁 𝑘*=0

*𝑁𝛾*

*𝐷* (*𝑧, 𝑥*) − *𝐷* (*𝑧, 𝑦*) − *𝐷* (*𝑦, 𝑥*) = ∇*ℎ*(*𝑥*) − ∇*ℎ*(*𝑦*)*, 𝑦* − *𝑧 .*

*⟨*

*⟩*

for all *𝑢* ∈ dom *ℎ* ∩ 𝖷. As an attractive step size choice, we may take the

*ℎ ℎ ℎ*

□

greedy choice *𝛾*  *𝛼* . However, we need to know the Lipschitz constant

*𝐿𝑓*

=

Thanks to [Lemma 3.3](#_bookmark30), relation [(3.11)](#_bookmark29) reads as

*𝛾*(*𝑟*(*𝑥*+) − *𝑟*(*𝑢*)) ≤ *𝛾* ∇*𝑓* (*𝑥*)*, 𝑢* − *𝑥*+ + *𝐷* (*𝑢, 𝑥*) − *𝐷* (*𝑢, 𝑥*+) − *𝐷* (*𝑥*+*, 𝑥*)

*⟨ ⟩ ℎ ℎ ℎ*

of the gradient map of the smooth part *𝑓* of the minimization problem

[(P)](#_bookmark6) to make this an implementable solution strategy.

**Proposition 3.4.** *Consider problem* [*(P)*](#_bookmark6) *with Assumptions* [*1*](#_bookmark7)*-*[*3*](#_bookmark13) *in place. Let*

*ℎ* ∈  (𝖷)*. If BPGM is run with the constant step size*

= *𝛼 , then for any*

for all *𝑢* ∈ dom *ℎ* ∩ 𝖷*.*

(3.12)

*𝛼*

*𝑥*∗ ∈ 𝖷∗*, we have*

*𝑘 𝐿𝑓*

**Remark 3.4.** Note that if *𝑥*+ is calculated inexactly in the sense that instead of [(3.10)](#_bookmark28), for some *𝜉* ∈ *𝜕𝑟*(*𝑥*+), it holds that

for some Δ ≥ 0, then instead of [(3.12)](#_bookmark31) we have

*⟩*

Ψ(*𝑥𝑘*) − Ψmin

(𝖷) ≤ *𝐿𝑓 𝐷* (*𝑥*∗*, 𝑥*0)*.* (3.18)

*𝛼𝑘*

*ℎ*

∇*𝑓* (*𝑥*) + ∇*ℎ*(*𝑥*+) − ∇*ℎ*(*𝑥*)*, 𝑥*+ − *𝑢*

*⟨𝛾𝜉 𝛾*

+

≤ Δ (3.13)

This global sublinear rate of convergence for the Euclidean setting is due to [Beck and Teboulle (2009b)](#_bookmark104); [Nesterov (2013)](#_bookmark223).

*𝛾*(*𝑟*(*𝑥*+) − *𝑟*(*𝑢*)) ≤ *𝛾* ∇*𝑓* (*𝑥*)*, 𝑢* − *𝑥*+ + *𝐷* (*𝑢, 𝑥*) − *𝐷* (*𝑢, 𝑥*+) − *𝐷* (*𝑥*+*, 𝑥*) + Δ*.*

*⟨ ⟩ ℎ ℎ ℎ*

strongly-convex with *𝜇 >* 0 it is possible to obtain linear convergence

rate of BPGM, i.e. Ψ(*𝑥𝑘*) − Ψ (𝖷) 2*𝐿* exp(−*𝑘𝜇*∕*𝐿* )*𝐷* (*𝑥*∗*, 𝑥*0).

**Remark 3.5.** Under additional assumption that the objective Ψ is *𝜇*-

≤

(3.14)

See [Auslender and Teboulle (2006b)](#_bookmark132) for an explicit analysis of the error-

min *𝑓*

* + 1. *Subgradient and Mirror Descent*

*𝑓 ℎ*

prone implementation.

[Assumption 1](#_bookmark7)(a) gives rise to the the classical ”descent Lemma” ([Nesterov, 2018b](#_bookmark228)):

*𝑓* (*𝑥*+) ≤ *𝑓* (*𝑥*) + ∇*𝑓* (*𝑥*)*, 𝑥*+ − *𝑥* + *𝐿𝑓 𝑥*+ − *𝑥* 2*.* (3.15)

*⟨ ⟩ ‖ ‖*

2

Additionally, for all *𝑢* ∈ 𝖷, differential convexity of *𝑓* on 𝖵 implies (cf.

[(1.1)](#_bookmark8))

*𝑓* (*𝑢*) ≥ *𝑓* (*𝑥*) + ∇*𝑓* (*𝑥*)*, 𝑢* − *𝑥 .* (3.16)

*⟨ ⟩*

This allows us to bound

*𝑓* (*𝑥*)*, 𝑢* − *𝑥*+ = ∇*𝑓* (*𝑥*)*, 𝑥* − *𝑥*+ + ∇*𝑓* (*𝑥*)*, 𝑢* − *𝑥*

∇

*⟨*

(3≤*.*15)

*⟩*

*⟨*

*⟩*

*⟩*

*⟨*

In the previous subsections we focused on the setting of problem

[(P)](#_bookmark6) with smooth part *𝑓* and obtained for BPGM a convergence rate

*𝑂*(1∕*𝑘*). The same method actually works for non-smooth convex op- timization problems when *𝑓* has bounded subgradients In this setting BPGM with a different choice of the step-size (*𝛾𝑘* )*𝑘* is known as the *Mir-*

*ror Descent* (MD) method ([Nemirovski and Yudin, 1983](#_bookmark213)). A version of this method for convex composite non-smooth optimization was pro- posed in [Duchi et al. (2010)](#_bookmark178), and an overview of Subgradient/Mirror Descent type of methods for non-smooth problems can be found in [Beck (2017)](#_bookmark97); [Dvurechensky et al. (2020b)](#_bookmark156); [Lan (2020)](#_bookmark188). The main dif- ference between BPGM and MD is that one replaces the assumption that

∇*𝑓* is Lipschitz continuous with the assumption that *𝑓* is subdifferen-

≤

tiable with bounded subgradients, i.e. *𝑓* ′(*𝑥*) *𝑀* for all *𝑥* ∈ 𝖷 and

*‖ ‖*∗ *𝑓*

*𝑓* ′(*𝑥*) ∈ *𝜕𝑓* (*𝑥*). For a given sequence of step-sizes (*𝛾𝑘* )*𝑘* one defines the

2 *‖ ‖ ⟨*

*𝑓* (*𝑥*) − *𝑓* (*𝑥*+) +

*𝐿𝑓*

*𝑥*

+

– *𝑥*

2

+ ∇*𝑓* (*𝑥*)*, 𝑢* − *𝑥*

*⟩* next test point as

(3≤*.*16) *𝑓* (*𝑢*) − *𝑓* (*𝑥*+) + *𝐿𝑓 𝑥*+ − *𝑥* 2

*‖ ‖*

2

*𝑢*

*𝑘*

*𝑘*

*ℎ*

*𝛾𝑘 𝑟*

*𝑘*

*𝑥𝑘*+1 = argmin *{⟨𝛾 𝑓* ′(*𝑥𝑘*)*, 𝑢* − *𝑥𝑘⟩* + *𝛾 𝑟*(*𝑢*) + *𝐷* (*𝑢, 𝑥𝑘*)*}* = P *ℎ* (*𝑥𝑘, 𝛾 𝑓* ′(*𝑥𝑘*))*.*

policy like *𝛾𝑘* ∼ *𝑘*−1∕2. Under such a specification, the MD sequence (*𝑥𝑘*)*𝑘* A typical choice for the step size sequence is a monotonically decreasing can be shown to converge with rate *𝑂*(1∕ *𝑘*) to the solution, which

*√*

is optimal in this setting. A proof of this result can be patterned via a suitable adaption of the arguments employed in our analysis of the Dual Averaging Method in [Section 3.4](#_bookmark40).

* + 1. *Potential Improvements due to relative smoothness*

A key pillar of the complexity analysis of BPGM was the descent inequality [(3.15)](#_bookmark34), which is available thanks to the as-

sumed Lipschitz continuity of the gradient ∇*𝑓* . The influential work

[Bauschke et al. (2016)](#_bookmark94) introduced a very clever construction which al-

lows one to relax this restrictive assumption.[2](#_bookmark41) The elegant observation

The first important observation is an extended version of the fundamen- tal inequality [(3.17)](#_bookmark27), which reads as

*𝛾*(Ψ(*𝑥*+) − Ψ(*𝑢*)) ≤ *𝐷* (*𝑢, 𝑥*) − *𝐷* (*𝑢, 𝑥*+) − (1 − *𝛾𝐿ℎ* )*𝐷* (*𝑥*+*, 𝑥*) ∀*𝑢* ∈ dom *ℎ* ∩ 𝖷*.* (3.21)

*ℎ ℎ 𝑓 ℎ*

The derivation of this inequality is analogous to inequality [(3.17)](#_bookmark27), re- placing the Lipschitz-gradient-based descent inequality [(3.15)](#_bookmark34) by the

relative smoothness inequality [(3.19)](#_bookmark38) with parameter *𝐿ℎ* . The contin-

*𝑓*

introduction of the *symmetry coefficient* of the DGF *ℎ* as uation of the proof differs then in one important aspect. It relies on the

*𝜈 ℎ { | 𝑥, 𝑢* 𝖷 𝖷 *, 𝑥 𝑢}.* (3.22)

( ) ∶= inf *𝐷ℎ*(*𝑥, 𝑢*) ( ) ∈ ◦ × ◦ ≠

*𝐷ℎ*(*𝑢, 𝑥*)

The symmetry coeﬃcient *𝜈*(*ℎ*) is confined to the interval [0,1], and

*𝜈*(*ℎ*) = 1 applies essentially only to the energy function *ℎ*(*𝑥*) = 1 *𝑥* 2.

made in [Bauschke et al. (2016)](#_bookmark94) is that the Lipschitz-gradient-based de- scent lemma has the equivalent, but insightful, expression

*( 𝐿𝑓 ‖𝑥‖*2 − *𝑓* (*𝑥*)*)* − *( 𝐿𝑓 ‖𝑢‖*2 − *𝑓* (*𝑢*)*)* ≥ *⟨𝐿 𝑢* − ∇*𝑓* (*𝑢*)*, 𝑥* − *𝑢⟩* ∀*𝑥, 𝑢* ∈ 𝖵*.*

*𝑓*

2

2

Choosing *𝛾*

*𝑘*+1

*𝛾*(Ψ(*𝑥*

) − Ψ(*𝑢*)) ≤ *𝐷* (*𝑢, 𝑥* ) − *𝐷* (*𝑢, 𝑥*

= 1+*𝜈*(*ℎ*)

*𝑓*

2*𝐿ℎ*

*, 𝑥*+ =

*ℎ*

*𝑥𝑘*+1

*𝑘*

*, 𝑥*

= *𝑥*

*ℎ*

*𝑘* gives

*𝑘*+1

) −

1 − *𝜈*(*ℎ*)

*𝑘*+1

2 *‖ ‖*

*𝑘*

*, 𝑥* )

This is just the gradient inequality for the convex function *𝑥* ↦ *𝐿𝑓* 2 −

*𝐷* (*𝑥*

2

*ℎ*

Setting *𝑢* = *𝑥𝑘* gives descent of the function value sequence (Ψ(*𝑥𝑘*)) ≥ .

*𝑓* (*𝑥*).

2 *‖𝑥‖*

Moreover, it immediately follows that

*𝑘* 0

**D****efinition 3.5.** The family of functions £(𝖷) is the class of DGFs *ℎ* ∈

Ψ(*𝑥𝑘*) − Ψ(*𝑢*) ≤ *𝑓 (𝐷* (*𝑢, 𝑥𝑘*−1) − *𝐷* (*𝑢, 𝑥𝑘*)*)*

0(𝖷) which are of *Legendre type*: *ℎ* essentially smooth and strictly con- 1 + *𝜈*(*ℎ*) *ℎ ℎ*

2*𝐿ℎ*

vex on int dom *ℎ* with cl(dom *ℎ*) = 𝖷.

In this section we work in a Bregman proximal setting with Legendre type distance generating function.

Summing from *𝑘* = 1*,* 2*,* … *, 𝑁* , the same argument as for the BPGM give

sublinear convergence of NoLips

2*𝐿ℎ*

**Assumption 4.** 𝖷 has nonempty interior and *ℎ* ∈ £(𝖷)*.*

Ψ(*𝑥𝑁* ) − Ψ(*𝑢*) ≤

*𝑓*

*𝑁* (1 + *𝜈*(*ℎ*))

*𝐷ℎ*(*𝑢, 𝑥*0)*.* (3.23)

**Remark 3.6.** For *ℎ* ∈ £(𝖷), it is true that 𝖷◦ *⊆* int(dom *ℎ*) ∩ 𝖷*.*

Based on the general intuition we have gained while working with a general proximal setup, a very tempting and natural generalization is the following.

[**Definition 3.6** (Relative Smoothness, (Bauschke et al., 2016;](#_bookmark94)

Comparing the constants in the complexity estimates of NoLips and BPGM we see that the relative eﬃciency of the two methods depends on

2*𝐿ℎ* ∕(1+*𝜈*(*ℎ*))

the relative condition number *𝑓* . Hence, even if the objective

*𝐿𝑓* ∕*𝛼*

function is globally Lipschitz smooth (i.e. admits a Lipschitz continu-

ous gradient), exploiting the idea of relative smoothness might lead to superior performance of NoLips.

[Van Nguyen, 2017))**.** The function *𝑓* is smooth relative to *ℎ* ∈ £(𝖷), if](#_bookmark94)

To establish global convergence of the trajectory (*𝑥𝑘*)

*𝑘*∈ℕ

, additional

there exists *𝐿ℎ >* 0 such that for any *𝑥, 𝑢* ∈ 𝖷◦

*𝑓*

*𝑓* (*𝑢*) ≤ *𝑓* (*𝑥*) + ∇*𝑓* (*𝑥*)*, 𝑢* − *𝑥* + *𝐿ℎ 𝐷* (*𝑢, 𝑥*)*.* (3.19)

*⟨ ⟩*

”reciprocity” conditions on the Bregman divergence must be imposed.

**Assumption 6.** The DGF *ℎ* ∈ £(𝖷) satisfies the Bregman reciprocity con-

*𝑓 ℎ*

Rearranging terms, a very concise and elegant way of writing the

dition: The level sets {*𝑢* ∈ 𝖷◦ *𝐷* (*𝑢, 𝑥*) ≤ *𝛽*} are bounded for all *𝛽* ∈ ℝ,

and *𝑥𝑘* → *𝑥* ∈ 𝖷◦ if and only if lim*𝑘*→∞ *𝐷ℎ*(*𝑥, 𝑥𝑘*) = 0*.*

*| ℎ*

relative smoothness condition is *𝐷𝐿ℎ ℎ 𝑓*

–

*𝑓*

(*𝑢, 𝑥*) ≥ 0 on 𝖷◦. This amounts

This assumption is necessary, as in some settings Bregman reci-

to saying that *𝐿ℎ ℎ* − *𝑓* is convex on 𝖷◦. Clearly, if *𝑓* and *ℎ* are twice

*𝑓* ◦ procity is violated. See Example 4.1 in [Doljansky and Teboulle (1998)](#_bookmark172) as

continuously differentiable on 𝖷 , the relative smoothness condition can be stated in terms of a positive semi-definitness condition on the set 𝖷◦ as

*𝐿ℎ* ∇2*ℎ*(*𝑥*) − ∇2*𝑓* (*𝑥*) ⪰ 0 ∀*𝑥* ∈ 𝖷◦*.* (3.20)

*𝑓*

Beside providing a non-Euclidean version of the descent lemma, the no- tion of relative smoothness allows us to rigorously apply gradient meth- ods to problems whose smooth part admits no global Lipschitz continu- ous gradient. This gains relevance in solving various classes of inverse problems under Poisson noise (see [Section 5.2](#_bookmark64) in [Bauschke et al. (2016)](#_bookmark94)), and optimal experimental design ([Lu et al., 2018](#_bookmark201)), a class of problems structurally equivalent to finding the minimum volume ellipsoid con- taining a list of vectors ([Boyd and Vandenberghe, 2004; Todd, 2016](#_bookmark127)).

**Assumption 5.** There exists a DGF *ℎ* ∈ £(𝖷) for which (*𝑓, ℎ*) is a rela-

a simple illustration. Under Bregman reciprocity, one can prove global convergence in the spirit of Opial’s lemma ([Opial, 1967](#_bookmark235)):

[*1*](#_bookmark7)*-*[*6*](#_bookmark37) *hold. Let* (*𝑥𝑘*)*𝑘*∈ℕ *be the sequence generated by BPGM with the relatively* **Theorem 3.7** ([Bauschke et al. (2016)](#_bookmark94), Theorem 2)**.** *Suppose Assumptions smooth pair 𝑓, ℎ with 𝛾 ,* 1+*𝜈*(*ℎ*) *and ℎ* 𝖷 *. Then, the sequence*

( ) ∈ (0 ) ∈ £( )

*𝐿ℎ*

*𝑓*

(*𝑥𝑘*)*𝑘*∈ℕ *converges to some solution 𝑥*∗ ∈ 𝖷∗*.*

Under additional assumption that *𝑓* is *𝜇ℎ* -relatively strongly convex ([Lu et al., 2018](#_bookmark201)) with *𝜇ℎ >* 0, i.e. (cf. [(3.20)](#_bookmark39))

*𝑓*

*𝑓*

∇2*𝑓* (*𝑥*) − *𝜇ℎ* ∇2*ℎ*(*𝑥*) ⪰ 0 ∀*𝑥* ∈ 𝖷◦*,* (3.24)

*𝑓*

it is possible to obtain linear convergence rate of BPGM, i.e.

≤

[Ψ(*𝑥𝑘*) − Ψmin(𝖷) 2*𝐿ℎ* exp(−*𝑘𝜇ℎ* ∕*𝐿ℎ* )*𝐷ℎ*(*𝑥*∗*, 𝑥*0). Stonyakin et al. (2020,](#_bookmark217)

[*𝑓 𝑓 𝑓*](#_bookmark217)

tively smooth pair.

assumption on the pair (*𝑓, ℎ*) (the so called NoLips algorithm of The complexity analysis of BPGM under a relative smoothness

[Bauschke et al. (2016)](#_bookmark94)), proceeds analogous to the previous analysis.

2 Variations on the same theme can be found in [Lu et al. (2018)](#_bookmark201) and further de- velopments can be found in [Bùi and Combettes (2021)](#_bookmark134); [Stonyakin et al. (2020)](#_bookmark217).

[2019) show how to adapt the method to cope with inexact oracles and](#_bookmark217) inexact Bregman proximal steps.

* 1. *Dual Averaging*

An alternative method called Dual Averaging (DA) was proposed in [Nesterov (2009)](#_bookmark221) and, on the contrary, is a primal-dual method making alternating updates in the space of gradients and in the space of iterates.

Below we give a self-contained complexity analysis of this scheme for

Rearranging and summing over *𝑘* = 1*,* 2*,* … *, 𝑁* , we get

non-smooth optimization. The following assumptions shall be in place:

**Assumption 7.** 𝖷 is a nonempty convex compact set.

**Assumption 8.** *𝑟* = 0 and *𝑓* ∈ Γ0(𝖵) with 𝖷 *⊂* dom(*𝑓* ). Furthermore,

′( *𝑘* ) 2 *.*

Let *ℎ* ∈  (𝖷) be a given DGF for the feasible set 𝖷 *⊆* 𝖵. Define the

*𝑓*

*𝑁*

*𝑘 𝐻*

*∑ 𝜆 ⟨𝑓* ′(*𝑥𝑘*)*, 𝑥𝑘* − *𝑥*0*⟩* ≤

*𝑘*=1

Observe

*𝛽*1

(*𝑦*1) − *𝐻*

*𝛽𝑁* +1

*𝑁*

(*𝑦𝑘*+1) +

*∑*

*𝑘*=1

2

*𝑘*

*𝜆*

2*𝛼𝛽𝑘*

*‖𝑓 𝑥 ‖*∗

sup

′(*𝑥*)

*‖*∗

*𝑥*∈𝖷 *‖𝑓*

≤ *𝑀*

for all *𝑓* ′(*𝑥*) ∈ *𝜕𝑓* (*𝑥*) and some *𝑀𝑓*

∈ (0*,* +∞).

*𝐻* (*𝑦*1) = *𝐻* (−*𝜆 𝑓* ′(*𝑥*0)) ≤ *𝐻* (0) + ∇*𝐻* (0)*,* −*𝜆 𝑓* ′(*𝑥*0) + 0 *𝑓* ′(*𝑥*0) 2

2

*𝜆*2

*𝛽 𝛽* 0 *𝛽 ⟨ 𝛽* 0 *⟩ ‖ ‖*

1

1

2

1

2

1

*𝛼𝛽*1

∗

0 *𝛼* = *𝜆*0

′(*𝑥*0) 2 ≤ *𝜆*0

′(*𝑥*0) 2 *.*

*ℎ*-center *𝑥* = argmin*𝑥*∈𝖷 *ℎ*(*𝑥*). Without loss of generality with assume

2*𝛼𝛽 ‖𝑓*

*‖*∗ 2*𝛼𝛽 ‖𝑓 ‖*∗

*ℎ*(*𝑥*0) = 0. Denote by

Θ*ℎ*(𝖷) ∶= max *ℎ*(*𝑝*)*.* (3.25)

*𝑘*

2*𝛼*

*𝛽𝑘*

∗

1

Therefore,

*𝑁*

0

*𝑁* 2

*𝑝*∈𝖷

*𝛽𝑁* +1

*∑ 𝜆 ⟨𝑓* ′(*𝑥𝑘*)*, 𝑥𝑘* − *𝑥*0*⟩* ≤ 1 *∑ 𝜆𝑘 ‖𝑓* ′(*𝑥𝑘*)*‖*2 − *𝐻*

(*𝑦𝑁*+1)*.* (3.30)

*function*

(∀*𝑦* ∈ 𝖵∗) ∶ *𝑔*(*𝑦*) = max *𝑦, 𝑥* − *𝑥*0 *,* (3.26)

*⟨ ⟩*

We emphasize that [Assumption 7](#_bookmark42) implies that Θ*ℎ*(𝖷) *<* ∞. Define the *gap*

*𝑘*=0

*𝑘*=0

Note that

*∑𝑁*

*𝜆*

*𝑓* ′( *𝑘* )*, 𝑥𝑘* − *𝑥*0

*𝜆*

*𝑓* ′(*𝑥𝑘*)*, 𝑥𝑘* − *𝑥*

+

*𝜆*

*𝑓* ′(*𝑥𝑘*)*, 𝑥* − *𝑥*0

*∑𝑁*

*∑𝑁*

*𝑥*∈𝖷

and

*𝑘*=0

*𝑘⟨ 𝑥*

*𝑘*=0

*∑𝑁*

=

=

*𝜆*

*𝑓* ′(*𝑥𝑘*)*, 𝑥𝑘* − *𝑥*

–

*𝑦𝑁*+1*, 𝑥* − *𝑥*0 *,*

for all *𝑥* ∈ 𝖷. This shows

*𝑘*=0

*⟩*

*𝑘⟨*

*⟩ 𝑘*=0

*𝑘⟨ ⟩*

*𝛽 𝑥*∈𝖷 *⟨ ⟩*

*𝐻* (*𝑦*) = max{ *𝑦, 𝑥* − *𝑥*0

– *𝛽ℎ*(*𝑥*)}*.* (3.27)

Note that *𝑔*(*𝑦*) ≥ 0 for all *𝑦* ∈ 𝖵∗, finite thanks to compactness of 𝖷.

*𝑘⟨*

*⟩ ⟨ ⟩*

Also, observe that for *𝛽*2

≥ *𝛽*1

*>* 0, it holds *𝐻𝛽*

(*𝑦*) ≤ *𝐻*

(*𝑦*) for all *𝑦* ∈ 𝖵∗. *𝑁*

*𝑁* 2

2 1 *∑ 𝜆 ⟨𝑓* ′(*𝑥𝑘*)*, 𝑥𝑘* − *𝑥⟩* ≤ 1 *∑ 𝜆𝑘 ‖𝑓* ′(*𝑥𝑘*)*‖*2 − *𝐻*

*⟨ ⟩*



Moreover, using the definition of the convex conjugate of the DGF

*ℎ* ∈ (𝖷), one sees that

*𝑘*=0

*𝑘*

2*𝛼*

*𝑘*=0

*𝛽*

∗

*𝛽𝑁* +1

*𝛽*

*𝑘*

(*𝑦𝑁*+1) + *𝑦𝑁*+1*, 𝑥* − *𝑥*0 (3.31)

*𝛼*

*∑*

*𝐻* (*𝑦*) = max{ *𝑦, 𝑥* − (*ℎ* + *𝛿* )(*𝑥*)} − *𝑦, 𝑥*0

for all *𝑥* ∈ 𝖷. Defining *𝜃𝑘*

∶= max

*𝑥*∈𝖷

*𝑘*

*𝑖*=0

*𝜆* ′( *𝑖* )*, 𝑥𝑖* − *𝑥* , [(3.31)](#_bookmark44) gives

*𝛽 𝑥*∈𝖵 *⟨ ⟩*

𝖷 *⟨ ⟩*

the estimate

= *𝛽*(*ℎ* + *𝛿* ) (*𝑦*∕*𝛽*) − *𝑦, 𝑥*

𝖷 *⟨ ⟩*

∗

0

for every *𝑦* ∈ 𝖵∗ and *𝛽 >* 0. Define the *mirror map*

*𝜃* ≤ 1 *∑ 𝜆𝑘 ‖𝑓* ′(*𝑥𝑘*)*‖*2 − *𝐻* (*𝑦𝑁*+1) + *𝑔*(*𝑦𝑁*+1)*.*

*𝑄𝛽* (*𝑦*) ∶= argmax { *𝑦, 𝑥* − *𝛽ℎ*(*𝑥*)}*.* (3.28)

*𝑖⟨𝑓 𝑥 ⟩*

*𝑁*

2

*𝑁*

2*𝛼 𝑘*=0 *𝛽𝑘*

∗

*𝛽𝑁* +1

*⟨*

*⟩*

*𝑥*∈𝖷

Then, from [(3.6)](#_bookmark21), one obtains the important identity

A simple application of the min-max inequality shows

*𝑔*(*𝑦*) = max *𝑦, 𝑥* − *𝑥*0

*⟨ ⟩*

*𝑥*∈𝖷

= max min{ *𝑦, 𝑥* − *𝑥*0

(∀*𝑦* ∈ 𝖵∗) ∶ ∇*𝐻𝛽* (*𝑦*) = *𝑄𝛽* (*𝑦*) − *𝑥*0*.*

*𝑥*∈𝖷 *𝛽*≥0 *⟨ ⟩ ℎ*

1 ≤ min *[(*max{*⟨𝑦, 𝑥* − *𝑥*0*⟩* − *𝛽ℎ*(*𝑥*)*)* + *𝛽*Θ*ℎ*(𝖷)*]*

+ *𝛽*(Θ (𝖷) − *ℎ*(*𝑥*))}

In particular, in view of [(3.6)](#_bookmark21), the function *𝐻𝛽* (*𝑦*) is seen to have a

-

*𝛽*≥0

*𝑥*∈𝖷

*𝛼𝛽*

Lipschitz continuous gradient. Given the current primal-dual pair (*𝑥, 𝑦*),

≤ *𝛽*Θ (𝖷

) + *𝐻𝛽* (*𝑦*)

DA performs a gradient step in the dual space 𝖵∗ to produce a new gradi- ent feedback point *𝑦*+ = *𝑦* − *𝜆𝑓* ′(*𝑥*), where *𝜆 >* 0 is a step size parameter.

*ℎ*

for all *𝑦* ∈ 𝖵∗ and *𝛽 >* 0. Hence, [(3.31)](#_bookmark44) gives

Taking this as a new signal, we update the primal state by applying the

mirror map *𝑥*+ = *𝑄𝛽* (*𝑦*+)*.*

*𝜃* ≤

1 *∑ 𝜆𝑖 ‖* ′

*𝑘* 2

Define Λ*𝑁* ∶= *∑𝑁*

*𝑘*=0

*𝑁*

*𝛽𝑁* +1 Θ*ℎ*(𝖷) + 2*𝛼*

*𝑁*

*𝑘*=0

*𝛽*

2

*𝑘*

*𝑓* (*𝑥* ) ∗

*𝜆𝑘* , and the ergodic average *𝑥̄𝑁* ∶= 1

*𝑁*

*‖ .*

Λ

*𝑁*

*𝑘*=0

*∑*

*𝜆𝑘 𝑥𝑘.*

### The Dual Averaging method (DA)

The subgradient inequality [(1.1)](#_bookmark8) applied to *𝑓* gives

**Input:** pick *𝑦*0 = 0*, 𝑥*0 = *𝑄𝛽* (0), nondecreasing learning sequence *𝑁 𝑁*

*𝑘*=0

*𝑘*

*𝑘*=0

*𝑘*

*𝑁*

≥ Λ

*𝑁*

(*𝑓* (*𝑥̄*

*𝑁*

) − Ψ

min

(𝖷)), and we conclude

(*𝛽* )

0

(*𝜆* )

*𝑘 𝑘*∈ℕ0 and non-increasing step-size sequence

**General step:** For *𝑘* = 0*,* 1*,* … do:

*𝑘 𝑘*∈ℕ0

*𝑁*

*𝑁*

*𝑁*

*∑ 𝜆 ⟨𝑓* ′(*𝑥𝑘*)*, 𝑥𝑘* − *𝑥⟩* ≥ *∑ 𝜆 𝑓* (*𝑥𝑘*) − Λ

Therefore, *𝜃*

*𝑓* (*𝑥*) ≥ Λ

(*𝑓* (*𝑥̄*

) − *𝑓* (*𝑥*))*.*

set *𝑥𝑘*+1 = *𝑄𝛽* (*𝑦𝑘*+1).

dual update *𝑦𝑘*+1 = *𝑦𝑘* − *𝜆𝑘 𝑓* ′(*𝑥𝑘*),

*𝑁* 2

*𝑘*+1

*‖*

*𝑁*

*𝑓*

*𝑥*

∗ *.*

Ψ(*𝑥̄*

# ) − Ψ

(𝖷) ≤ *𝛽𝑁* +1 Θ (𝖷) + 1 *∑ 𝜆𝑘 ‖*

′( *𝑘* ) 2

The function *𝑦* ↦ *𝐻* (*𝑦*) is convex and *√*continuously differentiable

We now assess the iteration complexity of DA, showing that it fea-

tures the same order convergence rate *𝑂*(1∕

*𝛽*

*𝑘*), just as BPGM and MD.

Let us now make the concrete choice of parameters *𝛽𝑘* = *𝛽 >* 0 and *𝜆𝑘* =

*√𝑘*+1 for all *𝑘* 0*.* Then,

≥

min

Λ

*𝑁*

*ℎ*

2*𝛼*Λ

*𝑁 𝑘*=0

*𝛽*

*𝑘*

1

with

*𝐻* (*𝑦* + *𝑤*) ≤ *𝐻* (*𝑦*) + ∇*𝐻* (*𝑦*)*, 𝑤* + 1

*𝑤* 2

∀*𝑦, 𝑤* ∈ 𝖵∗*.* (3.29)

*𝑁*

*𝑁 √*

*∑* 1

Λ =

∫

≥ *𝑁* +1 1

*√*

*𝑑𝑥* ≥ *√𝑁* + 1*,*

*𝛽 𝛽*

*⟨ 𝛽*

*⟩* 2*𝛼𝛽 ‖ ‖*∗

*𝑘*=0

*𝑘* + 1 0

*𝑥* + 1

Thanks to the monotonicity in the parameters, we get through some elementary manipulations the relation

*𝐻* (*𝑦𝑘*+1) ≤ *𝐻* (*𝑦𝑘* − *𝜆 𝑓* ′(*𝑥𝑘*))

*𝛽𝑘*+1 *𝛽𝑘*+1 *𝑘*

as well as

*∑ 𝜆𝑘* = 1 *∑*  1 ≤ 1 + ln(*𝑁* + 1) *.*

*𝑁*

2

*𝑁*

*𝑘*=0 *𝛽𝑘*

*𝛽 𝑘*=0 *𝑘* + 1

*𝛽*

≤ ( ∇ ( ) − ( ) + *𝜆*2 ( )

*𝑘*

*𝑦𝑘*) +

*𝑘*

′

*𝑘*

*𝑘*

′

*𝑘*

2

**Theorem 3.8.** *Suppose Assumptions* [*3*](#_bookmark13)*,* [*7*](#_bookmark42) *and Assumption* [*8*](#_bookmark43) *hold true. Let*

*𝛽𝑘*

*𝑓*

*𝐻*

+

*⟨ 𝐻𝛽𝑘 𝑦 ,*

*𝜆𝑘 𝑓 𝑥 ⟩*

2*𝛼𝛽 ‖𝑓 𝑥 ‖*∗

(*𝑥* )*𝑘 be generated by DA with parameters 𝛽𝑘* = *𝛽 >* 0 *and 𝜆𝑘* = *√𝑘*+1 *. Then,*

= *𝐻*

(*𝑦𝑘*) − *𝜆*

*𝜆*2

*𝑥𝑘* − *𝑥*0*, 𝑓* ′(*𝑥𝑘*) + *𝑘*

*𝑘*

1

*𝑓* ′(*𝑥𝑘*) 2 *.*

≤ *𝛽*Θ*ℎ*(𝖷)

*𝑀* 2 (1 + ln(*𝑁* + 1))

*𝛽𝑘*

*𝑘⟨*

*⟩* 2*𝛼𝛽𝑘 ‖ ‖*∗

Ψ(*𝑥̄𝑁* ) − Ψmin(𝖷)

*√𝑁* + 1

2*𝛼𝛽*

*√𝑁* + 1

(3.32)

fixed time window *𝑘* ∈ {0*,* 1*,* … *, 𝑁* }. Committing over this time interval A slightly better bound can be obtained if we decide to run DA over a on the constant parameter sequences *𝛽𝑘* = *𝛽 >* 0 and *𝜆𝑘* = *𝜆 >* 0 yields

Θ ( )

(𝖷) ≤ *𝛽*

*ℎ* 𝖷

+ 1 *𝜆*2

Assuming that the penalty function *ℎ* is of Legendre type, the primal

projection step is seen to be the regularized maximization step

*𝑥𝑘* = argmax{ *𝑦𝑘, 𝑢* − *𝛽 ℎ*(*𝑢*)} ⇔ *𝑦𝑘* = *𝛽* ∇*ℎ*(*𝑥𝑘*)*.*

*𝑢*∈𝖷

*⟨ ⟩ 𝑘 𝑘*

*𝑁* min

Ψ(*𝑥̄*

) − Ψ

2

(*𝑁* + 1)*𝜆*

2*𝛼 𝛽 𝑀𝑓 .*

*𝛽 √* 2*𝛼*Θ (𝖷)

Using the definition of the dual trajectory, we see that for all *𝑘* ≥ 0 the

primal-dual relation obeys:

Optimizing with respect to *𝜆 >* 0 gives the choice *𝜆* =

*𝑀𝑓*

*ℎ* , and

*𝑁*

0 = *𝜆𝑘* ∇*𝑓* (*𝑥𝑘*) + *𝛽𝑘*+1 ∇*ℎ*(*𝑥*

) − *𝛽𝑘* ∇*ℎ*(*𝑥𝑘*)*.*

*𝑘*+1

the complexity upper bound

+1

*√*  2Θ ( )

≤

*ℎ* 𝖷

Assuming that *𝛽𝑘*

≡ 1, this implies

Ψ(*𝑥̄𝑁* ) − Ψmin(𝖷)

*𝛼*(*𝑁* + 1) *𝑀𝑓 .*

*𝑢*∈𝖷

*⟨ 𝑘*

*⟩ ℎ 𝛿*𝖷 *𝑘*

*The eﬀectiveness of non-Euclidean setups* With the help of the explicit rate estimate [(3.32)](#_bookmark45) we are now in the position to evaluate the potential

*𝑥𝑘*+1 ∈ argmin{ *𝜆* ∇*𝑓* (*𝑥𝑘*)*, 𝑢* − *𝑥𝑘*

+ *𝐷* (*𝑢, 𝑥𝑘*)} = P

(*𝑥𝑘, 𝜆* ∇*𝑓* (*𝑥𝑘*))*.*

We will do so by focusing on the geometry 𝖷 = {*𝑥* ∈ ℝ*𝑛 𝑥𝑖* = 1}. eﬃciency gains we can make by adopting the non-Euclidean framework.

+

*𝑖*=1

*| ∑𝑛*

There are two natural projection frameworks for the unit simplex:

We have thus shown that DA and BPGM/MD agree if all parameters and initial conditions are chosen in the same way.

*3.4.2. Links to continuous-time dynamical systems*

The connection between numerical algorithms and continuous-time

* Consider the 𝓁 setup in which ⋅ = ⋅ and *ℎ*(*𝑥*) = 1 *𝑥* 2 −

dynamical systems for optimization is classical and well-documented

in the literature (see e.g. [Helmke and Moore (1996)](#_bookmark197) for a textbook

2

1 0

2*𝑛* . Then *𝑥*

*‖* 0*‖ ‖ ‖*2 2 *‖ ‖*2

reference). Here we describe an interesting link between dual av-

Θ*ℎ*(𝖷) = *𝑛*−1 . We denote by *𝑀𝑓,* ⋅

= (1∕*𝑛,* … *,* 1∕*𝑛*) and *ℎ*(*𝑥* ) = 0. The *ℎ*-diameter of 𝖷 is

the bound on the subgradients

2*𝑛*

*‖ ‖*2

[duced in](#_bookmark128) [Alvarez et al. (2004)](#_bookmark110)[;](#_bookmark128) [Attouch et al. (2004)](#_bookmark118)[; Attouch and](#_bookmark128)

of the function *𝑓* under the 𝓁2-norm. The corresponding complexity

eraging and a class or Riemannian gradient flows originally intro-

estimate is

[Teboulle (2004) and further studied in](#_bookmark128) [Bolte](#_bookmark120) [and Teboulle (2003).](#_bookmark128) A complexity analysis of discretized versions of these gradient flows

Complexity(𝖷*, ‖* ⋅ *‖*2) = *√* (*𝑛* − 1)∕*𝑛 𝑀*

*𝛼*(*𝑁* + 1)

*𝑓,‖*⋅*‖*2

has recently been obtained in [Bomze et al. (2019)](#_bookmark122). Our point of departure is the following continuous-time dynamical system based [on dual averaging, which has been introduced in Mertikopoulos and](#_bookmark204)

* A different sensible projection framework is obtained by consider the

*.*

= ⋅ with DGF *ℎ*(*𝑥*) = *∑𝑛 𝑥* ln(*𝑥* ) − ln(1∕*𝑛*). The *ℎ*-

𝓁1 -norm ⋅ 1 *𝑖 𝑖*

*𝑖*=1

*‖ ‖ ‖ ‖*

diameter is Θ*ℎ*(𝖷) = ln(*𝑛*). We let *𝑀𝑓,* ⋅ denote the bound on the

*‖ ‖*

[Staudigl (2018a) in the context of convex programming and in](#_bookmark204) [Mertikopoulos and Staudigl (2018b)](#_bookmark205) for general monotone variational inequality problems. The main ingredient of this dynamical system is a

subgradients of the function *𝑓*

∞

under the dual norm

𝓁∞

. The corre-

pair of primal-dual trajectories (*𝑥*(*𝑡*)*, 𝑦*(*𝑡*))*𝑡*≥0 evolving in continuous time

sponding complexity estimate is

*√* 2 ln(*𝑛*)

) =

according to the differential-projection system

*{𝑦*′(*𝑡*) ∶= *𝑑𝑦*(*𝑡*) = −*𝜆*(*𝑡*)∇*𝑓* (*𝑥*(*𝑡*))*,*

Complexity(𝖷*,*

*‖*

⋅ *‖*1

*𝛼*(*𝑁* + 1)

*𝑓,‖*⋅*‖*∞

*𝑑𝑡*

*𝑥*(*𝑡*) = *𝑄*1(*𝜂*(*𝑡*)*𝑦*(*𝑡*)) =∶ *𝑄*(*𝜂*(*𝑡*)*𝑦*(*𝑡*))*.*

(3.34)

To compare the complexity estimates implied by the two different Bregman setups, we compute the eﬃciency ratio

part DA, let us perform an Euler discretization of the dual trajectory

*𝑀*

*.*

Complexity(𝖷*,* ⋅ ) *√ ‖ ‖*

*𝑛* − 1 *𝑓,* ⋅ 2 *.*

*𝑀*

In this formulation, [Assumption 1](#_bookmark7)(a) is in place, in order to ensure that the dynamical system is well-posed, thanks to the Picard-Lindelöf theorem. To relate this scheme formally to its discrete-time counter-

Complexity(𝖷*,*

*‖ ‖*

for all ∈ *𝑛* , we see that 1 ≤ *‖ ‖*2

*𝑅* ∶= 2 =

If *𝑅 <* 1 then the 𝓁

setup. Since

mal space by applying the mirror map *𝑄*( 1 *𝑦𝑘*+1), where *𝛽*−1

2

*‖*

)

⋅ *‖*1

2*𝑛* ln(*𝑛*) *𝑀*

*𝑓,‖*⋅*‖*∞

by *𝑦𝑘*

– *𝑦*

*𝑘*−1

= −*𝜆𝑘* ∇*𝑓* (*𝑥𝑘*), and project the resulting point to the pri-

*𝑎* ∞ 2

≤ *𝑎*

setup is more eﬃcient than the 𝓁

*‖ ‖ ‖ ‖*

This implies

≤ *√𝑛‖𝑎‖*∞

*𝑀𝑓,* ⋅

*𝑎* ℝ

is the

1

*𝑀𝑓,* ⋅

*‖ ‖*∞

≤ *√𝑛*.

discrete-time learning rate appropriately sampled from the function *𝜂*(*𝑡*).

Legendre function *ℎ* ∈ (𝖷), so that As in [Section 3.4.1](#_bookmark46), let us assume that the mirror map is generated by a

£

*𝛽𝑘*+1

*𝑘*+1

*𝑛* − 1 2*𝑛* ln(*𝑛*)

*√*

≤ ≤ *𝑛* − 1 2 ln(*𝑛*)

*𝑥*(*𝑡*) = ∇*ℎ*∗(*𝜂*(*𝑡*)*𝑦*(*𝑡*))*.*

Let us further assume that *ℎ* is twice continuously differentiable and

The closer the eﬃciency ratio *𝑅* gets to the upper bound, the more fa- vorable the 𝓁1-setup would be compared to the standard 𝓁2-setup. This

*𝑅 √ .*

imentally verify the significant advantages of the 𝓁1 setup for large- is not an unrealistic situation in practice. [Ben-Tal et al. (2001)](#_bookmark109) exper-

dimensional simplex domains.

* + 1. *On the connection between Dual Averaging and Mirror Descent*

A deep and important connection between the Dual Averaging and Mirror Descent algorithms for convex non-smooth optimization has been observed in [Beck and Teboulle (2003)](#_bookmark101). To relate these iterates to BPGM,

[£](#_bookmark101)

we assume that *ℎ* ∈ (𝖷), in the sense of [Definition 3.5](#_bookmark36). Let us recall

that *ℎ* is essentially smooth if and only if its Fenchel conjugate *ℎ*∗ is es- sentially smooth. Moreover, ∇*ℎ* ∶ int(dom *ℎ*) → int(dom *ℎ*∗) is a bijection

with

(∇*ℎ*)−1 = ∇*ℎ*∗ and ∇*ℎ*∗(∇*ℎ*(*𝑥*)) = *𝑥,* ∇*ℎ*(*𝑥*) − *ℎ*(*𝑥*) (3.33)

*⟨ ⟩*

Since 𝖷 = cl(dom *ℎ*), it follows

dom *𝜕ℎ* = int(dom *ℎ*) = int(𝖷) with *𝜕ℎ*(*𝑥*) = {∇*ℎ*(*𝑥*)} ∀*𝑥* ∈ int(𝖷)*.*

*𝜂*(*𝑡*) ≡ 1. Differentiating the previous equation with respect to time *𝑡*

gives

*𝑥*′(*𝑡*) = ∇2 *ℎ*∗(*𝑦*(*𝑡*))*𝑦*′(*𝑡*) = −*𝜆*(*𝑡*)∇2 *ℎ*∗(*𝑦*(*𝑡*))∇*𝑓* (*𝑥*(*𝑡*))*.*

ing ∇*ℎ*∗(∇*ℎ*(*𝑥*)) = *𝑥* for all *𝑥* ∈ int dom *ℎ* (cf. [(3.33)](#_bookmark48)). Differentiating im- To make headway, recall the basic properties of Legendre function say- plicitly this identity, we obtain ∇2 *ℎ*∗(∇*ℎ*(*𝑥*)) Id, or

[≡](#_bookmark48)

∇2 *ℎ*∗(∇*ℎ*(*𝑥*)) = [∇2 *ℎ*(*𝑥*)]−1 =∶ *𝐻* (*𝑥*)−1 *.* (3.35)

As in [Section 3.4.1](#_bookmark46), it holds true that *𝑦*(*𝑡*) = ∇*ℎ*(*𝑥*(*𝑡*)) for all *𝑡* ≥ 0, we

therefore obtain the interesting characterization of the primal trajectory

as

*𝑥*′(*𝑡*) = −*𝜆*(*𝑡*)*𝐻* (*𝑥*(*𝑡*))−1 ∇*𝑓* (*𝑥*(*𝑡*))*.*

If 𝖷 is a smooth manifold, we can define a Riemannian metric

*𝑔* (*𝑢, 𝑣*) ∶= *𝐻* (*𝑥*)*𝑢, 𝑣* ∀(*𝑥, 𝑢, 𝑣*) ∈ 𝖷◦ × 𝖵 × 𝖵*.*

*𝑥 ⟨ ⟩*

The gradient of a smooth function *𝜙* with respect to the metric *𝑔* is then given by ∇*𝑔 𝜙*(*𝑥*) = *𝐻* (*𝑥*)−1 ∇*𝜙*(*𝑥*). Hence, the continuous-time version of

the dual averaging method gives rise the class of primal *Riemannian- Hessian gradient flows*

*𝑥*′(*𝑡*) + *𝜆*(*𝑡*)∇*𝑔 𝑓* (*𝑥*(*𝑡*)) = 0*, 𝑥*(0) ∈ 𝖷◦*.* (3.36)

for a given bounded linear operator **𝐀**. To streamline the presentation, we directly assume in this section that 𝖵 = 𝖵∗ = ℝ*𝑛*, and the underly- ing metric structure is generated by the Euclidean norm *𝑎 𝑎* =

This class of continuous-time dynamical systems gave rise to a vigorous

literature in connection with Nesterov’s optimal method, which we will thoroughly discuss in [Section 6](#_bookmark72). As an appetizer, consider the system of differential equations

*𝑦*′(*𝑡*) = −*𝜆*(*𝑡*)∇*𝑓* (*𝑥*(*𝑡*))*, 𝑥*′(*𝑡*) = *𝛾*(*𝑡*)[*𝑄*(*𝜂*(*𝑡*)*𝑦*(*𝑡*)) − *𝑥*(*𝑡*)]*.* (3.37)

problem can be equivalently written as

inf {Φ(*𝑥, 𝑧*) = *𝑔*(*𝑧*) + *𝑟*(*𝑥*) **𝐀***𝑥* − *𝑧* = 0*, 𝑥* ∈ 𝖷*, 𝑧* ∈ 𝖹}*,* (4.1)

≡

*𝑖*=1

*|*

*⟨𝑎, 𝑎⟩*1∕2 = *(∑𝑛*

*𝑎𝑖 )*1∕2 . Introducing the auxiliary variable *𝑧* = **𝐀***𝑥*, this

*‖ ‖ ‖ ‖*2

where 𝖷 = ℝ*𝑛* and 𝖹 = ℝ*𝑚*. We will call this the *primal* problem. The

Lagrangian associated to [(4.1)](#_bookmark49) is

*𝐿*(*𝑥, 𝑧, 𝑦*) = *𝑔*(*𝑧*) + *𝑟*(*𝑥*) + *𝑦,* **𝐀***𝑥* − *𝑧 ,*

Suppose that in [(3.37)](#_bookmark50) we take *𝑄*(*𝑦*) = *𝑦, 𝜂*(*𝑡*) = 1. This corresponds to the *⟨ ⟩*

Legendre function *ℎ*(*𝑥*) = 1

*𝑥* 2 + *𝛿* (*𝑥*) for a given closed convex set 𝖷.

2 *‖ ‖*2 𝖷

*𝑚*

where *𝑦* ∈ ℝ is the Lagrange multiplier associated with the linear con-

Under this specification, the dynamical system [(3.37)](#_bookmark50) becomes

*𝑦*′(*𝑡*) = −*𝜆*(*𝑡*)∇*𝑓* (*𝑥*(*𝑡*))*, 𝑥*′(*𝑡*) = *𝛾*(*𝑡*)[*𝑦*(*𝑡*) − *𝑥*(*𝑡*)]*.*

straint. The dual function is accordingly defined as

*𝑞*(*𝑦*) = inf *𝐿*(*𝑥, 𝑧, 𝑦*) = inf {*𝑔*(*𝑧*) − *𝑦, 𝑧* } + inf {*𝑟*(*𝑥*) + **𝐀***𝑥, 𝑦* }

Combining the primal and the dual trajectory, we easily derive a purely primal second-order in time dynamical system given by

*𝑥,𝑧*

*𝑧 ⟨ ⟩*

*𝑧 ⟨ ⟩ 𝑥 ⟨ ⟩*

*𝑥 ⟨ ⟩*

= − sup{ *𝑦, 𝑧* − *𝑔*(*𝑧*)} − sup{ − **𝐀***𝑥, 𝑦* − *𝑟*(*𝑥*)}

′′ ′

*( 𝛾*(*𝑡*)2 − *𝛾*′(*𝑡*) *)*

= −*𝑔*∗(*𝑦*) − *𝑟*∗(−**𝐀***⊤𝑦*)*.*

Setting *𝛾*(*𝑡*) = *𝛽*∕*𝑡* and *𝜆*(*𝑡*) = 1∕*𝛾*(*𝑡*) and rearranging gives

*𝑥* (*𝑡*) − *𝑥* (*𝑡*)

*𝛾*(*𝑡*)

+ *𝜆*(*𝑡*)∇*𝑓* (*𝑥*(*𝑡*)) = 0*.*

Hence, we can represent the dual problem as the minimization problem

*𝑥*′′(*𝑡*) +

*𝛽* + 1

*𝑡*

*𝑥*′(*𝑡*) + ∇*𝑓* (*𝑥*(*𝑡*)) = 0*,*

min{Ψ*̃* (*𝑦*) = *𝑔*∗(*𝑦*) + *𝑟*∗(−**𝐀***⊤𝑦*)} (4.2)

*𝑦*

method of Polyak [Polyak (1964)](#_bookmark240). For *𝛽* = 2 this gives the continuous- which corresponds to the continuous-time version of the Heavy-ball

time formulation of Nesterov’s accelerated scheme, as shown by [Su et al. (2016)](#_bookmark222).

More generally, suppose that *ℎ* is a twice continuously differentiable

≡

Legendre function and *𝜂*(*𝑡*) 1. Then a direct calculation shows that

*𝑥*′′(*𝑡*) + *(𝛾*(*𝑡*) − *𝛾*′(*𝑡*) *)𝑥*′(*𝑡*) + *𝛾*(*𝑡*)*𝜆*(*𝑡*)(∇2 *ℎ*)−1 (*𝑄*(*𝑦*(*𝑡*)))∇*𝑓* (*𝑥*(*𝑡*)) = 0*.*

*𝛾*(*𝑡*)

Using the identity [(3.35)](#_bookmark47), as well as *𝑥*′(*𝑡*) + *𝑥*(*𝑡*) = ∇*ℎ*∗(*𝑦*(*𝑡*)), it follows

*𝛾*(*𝑡*)

that

A classical implicit method for solving this minimization problem is the

*proximal point method*:

*𝑦𝑘*+1 ∈ argmin{Ψ*̃* (*𝑦*) + 1 *𝑦* − *𝑦𝑘* 2} (4.3)

*‖ ‖*

*𝑦* 2*𝑐*

where *𝑐 >* 0 is a regularization parameter controlling the effects of the quadratic penalty term. By Fermat’s optimality condition, the point *𝑦𝑘*+1

satisfied the monotone inclusion

0 ∈ *𝜕𝑔*∗(*𝑦𝑘*+1) − **𝐀***𝜕𝑟*∗(−**𝐀***⊤𝑦𝑘*+1) + 1 (*𝑦𝑘*+1 − *𝑦𝑘*)*.*

*𝑐*

This means that there exists *𝑥𝑘*+1 ∈ *𝜕𝑟*∗(−**𝐀***⊤𝑦𝑘*+1) and *𝑧𝑘*+1 ∈ *𝜕𝑔*∗(*𝑦𝑘*+1)

such that

∇2 *ℎ(𝑥*(*𝑡*) + *𝑥*′(*𝑡*) *)( 𝑥*′′(*𝑡*) + *(*1 − *𝛾*′(*𝑡*) *)𝑥*′(*𝑡*)*)* = −*𝜆*(*𝑡*)∇*𝑓* (*𝑥*(*𝑡*)) ⇔ *𝑑* ∇*ℎ(𝑥*(*𝑡*) + *𝑥*′(*𝑡*) *)*

+1 +1 1 +1

*𝛾*(*𝑡*)

*𝛾*(*𝑡*)

*𝛾*(*𝑡*)2

*𝑑𝑡*

*𝛾*(*𝑡*)

0 = *𝑧𝑘*

– **𝐀***𝑥𝑘*

+ (*𝑦𝑘*

*𝑐*

– *𝑦𝑘*)*.*

= −*𝜆*(*𝑡*)∇*𝑓* (*𝑥*(*𝑡*))*.*

This shows that for *𝜂* ≡ 1, the dynamic coincides with the Lagrangian

family of second-order systems constructed in [Wibisono et al. (2016)](#_bookmark243).

This means that the proximal point method can be implemented for the given instance as the implicit method

*𝑥𝑘*+1 ∈ argmin{*𝑟*(*𝑥*) + **𝐀***𝑥, 𝑦𝑘*+1 }*,*

*⟨*

These ideas are now investigated heavily when combined with numer- ical discretization schemes for dynamical system with the hope to get insights how to construct new and more eﬃcient algorithmic formula-

*𝑥*

*𝑧𝑘*+1 ∈ argmin{*𝑔*(*𝑧*) +

*⟨*

*𝑧*

– *𝑧, 𝑦𝑘*+1*⟩* }

*⟩*

tion of gradient-methods. This literature grew quite fastly over the last [years, and we mention (Attouch et al., 2020; 2018; Bah et al., 2019; Shi et al., 2019).](#_bookmark123)

### The Proximal Method of Multipliers and ADMM

In this section we turn our attention to a classical method for solv- ing linearly constrained optimization problems building on the classi- cal idea of the celebrated *method of multipliers*. An extremely powerful proponent of this class of algorithms is the *Alternating Direction Method of Multipliers* (ADMM), which has received enormous interest from dif- ferent directions, including PDEs ([Attouch et al., 2011; 2007](#_bookmark121)), mixed- [integer programming (](#_bookmark195)[Feizollahi et al., 2017](#_bookmark163)[), optimal control (Lin et al., 2012) and signal processing (](#_bookmark195)[Yang](#_bookmark247) [and Zhang, 2011; Yuan, 2012). The](#_bookmark195) very influential monograph ([Boyd et al., 2011](#_bookmark125)) contains over 180 refer- ences, reflecting the deep impact of alternating methods on optimization theory and its applications. Following the general spirit of this survey, we introduce alternating direction methods in a proximal framework,

*𝑦𝑘*+1 = *𝑦𝑘* + *𝑐*(**𝐀***𝑥𝑘*+1 − *𝑧𝑘*+1)*.*

erates *𝑥𝑘*+1*, 𝑧𝑘*+1*, 𝑦𝑘*+1 simultaneously. Of course, this does not give rise This defines a fully implicit iteration, which requires to compute the it-

to a practical algorithm. The main idea behind alternating methods is to organize the computations in a Gauss-Seidel kind of iterations in which the sequences are updated sequentially using the most recent informa-

tion available. To set the stage, observe that (*𝑥𝑘*+1*, 𝑧𝑘*+1) defined above

is the coordinate-wise minimum of the function

*𝐹* (*𝑥, 𝑧, 𝑦𝑘*) = *𝑔*(*𝑧*) + *𝑟*(*𝑥*) + *𝑐* **𝐀***𝑥* − *𝑧* + 1 *𝑦𝑘* 2*.*

*‖ ‖*

2 *𝑐*

In that sense, the proximal point method applied to the dual can be represented more compactly as

(*𝑥𝑘*+1*, 𝑧𝑘*+1) ∈ argmin *𝐹* (*𝑥, 𝑧, 𝑦𝑘*)*, 𝑦𝑘*+1 = *𝑦𝑘* + *𝑐*(**𝐀***𝑥𝑘*+1 − *𝑧𝑘*+1)*.*

(*𝑥,𝑧*)

This scheme is known as the *augmented Lagrangian method.*

Observe that minimizers of the function *𝐹* (*𝑥, 𝑧, 𝑦𝑘*) with respect to

(*𝑥, 𝑧*) agree with the minimizers of the function

[as pioneered by Rockafellar](#_bookmark212) [Rockafellar (1976a,b)](#_bookmark246)[, and due to Shefi and](#_bookmark212)

*𝐿* (*𝑥, 𝑧, 𝑦𝑘*) = *𝑔*(*𝑧*) + *𝑟*(*𝑥*) + *𝑐*

**𝐀***𝑥* − *𝑧* + 1 *𝑦𝑘* 2*,*

[Teboulle (2014). See also](#_bookmark212) [Banert](#_bookmark95) [et al. (2021) for some further important](#_bookmark212) *𝑐*

2 *‖ 𝑐 ‖*

elaborations.

To set the stage, consider the composite convex optimization prob- lem [(P)](#_bookmark6), in its special form [(2.7)](#_bookmark10):

Ψ(*𝑥*) = *𝑔*(**𝐀***𝑥*) + *𝑟*(*𝑥*)*,*

which is known as the *augmented Lagrangian* of problem [(4.1)](#_bookmark49). Using the augmented Lagrangian, an alternating minimization procedure build- ing on the proximal point idea gives rise to the celebrated *Alternating Direction of Method of Multipliers* (ADMM).

**Input:** pick (*𝑧*0*, 𝑦*0) ∈ 𝖹 × ℝ*𝑚* and penalty parameter *𝑐 >* 0; **The Alternating Direction of Method of Multipliers (ADMM) General step:** For *𝑘* = 0*,* 1*,* … do:

The coupling between primal and dual variables reads as

*𝑦𝑘*+1 = *𝑢𝑘*+1 − *𝑐𝑧𝑘*+1 *.*

Combining all these relations, we can write the dual minimization prob-

*𝑥𝑘*+1

*‖*

= argmin*𝑥*∈𝖷{*𝑟*(*𝑥*) +

**𝐀***𝑥* − *𝑧𝑘* +

1 *𝑦*

*𝑘* 2

} (4*.*4)

lem as

*𝑘*+1

*𝑐 𝑘*

1 *𝑘* 2

*𝑧𝑘*+1

= argmin

{*𝑔*(*𝑧*) +

2

*𝑐* **𝐀***𝑥*

*‖*

*𝑐*

*𝑘*+1

*𝑐*

– *𝑧* +

2

1 *𝑦𝑘* 2

} (4*.*5)

*𝑥* = argmin{*𝑟*(*𝑥*) + **𝐀***𝑥* − *𝑧*

*𝑥* 2

*‖*

*‖ ‖*

*𝑧*

2

+ *𝑦* }*,*

*𝑐*

*‖*2

*𝑐*

2

*𝑧*∈𝖹 2 *‖*

*𝑦𝑘*+1 = *𝑦𝑘* + *𝑐*(**𝐀***𝑥𝑘*+1 − *𝑧𝑘*+1)*.* (4*.*6)

*𝑐 ‖*2

*𝑧𝑘*+1 = argmin{*𝑔*(*𝑧*) + *𝑐* **𝐀***𝑥𝑘*+1 − *𝑧* + 1 *𝑦𝑘* 2}*,*

**Remark 4.1.** ADMM updates the decision variables in a sequential man-

trix **𝐀** is of special structure. In the context of the AC optimal power ner, and thus is not capable of featuring parallel updates unless the ma-

flow problem in electric power grid optimization ([Sun et al., 2013](#_bookmark224)) pro- vide such a modification of ADMM. Furthermore, the ADMM can be

form **𝐀**1*𝑥* + **𝐀**2*𝑧* = *𝑏*. For ease of exposition we stick to the simplified extended to consider formulations with general linear constraints of the

problem formulation above.

* 1. *The Douglas-Rachford algorithm and ADMM*

The Douglas-Rachford (DR) algorithm is a fundamental method to solve general monotone inclusion problems where the task is to find ze- [ros of the sum of two maximally monotone operators (see Bauschke and Combettes (2016) and](#_bookmark96) [Auslender](#_bookmark130) [and Teboulle (2006a)). To keep the fo-](#_bookmark96) cus on convex programming, we introduce this method for solving the

dual problem [(4.2)](#_bookmark51). To that end, let us define the matrix **𝐊** = −**𝐀***⊤*, so

that our aim is to solve the convex programming problem

*𝑦𝑘*+1 = *𝑦𝑘* + *𝑐*(**𝐀***𝑥𝑘*+1 − *𝑧𝑘*+1)

which is just the standard ADMM. By this we have recovered a classi- cal result on connection between the DR and ADMM algorithms due to [Gabay (1983)](#_bookmark173) and [Eckstein and Bertsekas (1992)](#_bookmark158).

* 1. *Proximal Variant of ADMM*

the term **𝐀***𝑥* in the update of *𝑥𝑘*+1. The presence of this factor makes One of the limitations of the ADMM comes from the presence of

it impossible to implement the algorithm in parallel, which makes it slightly unattractive for large-scale problems in distributed optimiza- tion. Moreover, due to the result of [Chen et al. (2016)](#_bookmark147) the conver- gence of ADMM for general linear constraints does not generalize to more than two blocks. Leaving parallelization issues aside, Shefi and Teboulle [Shefi and Teboulle (2014)](#_bookmark212) proposed an interesting extension of the ADMM by adding further quadratic penalty terms, which allows much flexibility by suitably choosing the norms employed in the algo-

rithm. Given some point (*𝑥𝑘, 𝑧𝑘, 𝑦𝑘*) ∈ 𝖷 × 𝖹 × ℝ*𝑚* and two positive def-

inite matrices **𝐌**1*,* **𝐌**2, we define the *proximal augmented Lagrangian* of

[(4.1)](#_bookmark49) as

*𝑃* (*𝑥, 𝑧, 𝑦*) = *𝐿* (*𝑥, 𝑧, 𝑦*) + 1 *𝑥* − *𝑥𝑘* 2 + 1 *𝑧* − *𝑧𝑘* 2 *.* (4.12)

min *𝑔*∗(*𝑧*) + *𝑟*∗(**𝐊***𝑧*)*.* (4.7) *𝑘*

*𝑧*

*𝑐* 2 *‖*

*‖***𝐌**1 2 *‖*

*‖***𝐌**2

Any solution *𝑧̄* ∈ dom(*𝑟*∗) satisfies the monotone inclusion

0 ∈ **𝐊***⊤𝜕𝑟*∗(**𝐊***𝑧̄*) + *𝜕𝑔*∗(*𝑧̄*)*.* (4.8)

The DR algorithm aims to determine such a point *𝑧̄* by iteratively con- structing a sequence {(*𝑢𝑘, 𝑣𝑘, 𝑦𝑘*)*, 𝑘* 0} determined by

≥

*𝑣𝑘*+1 = (Id +*𝑐***𝐊***⊤*◦*𝜕𝑟*∗◦**𝐊**)−1 (2*𝑦𝑘* − *𝑢𝑘*)*,*

*𝑢𝑘*+1 = *𝑣𝑘*+1 + *𝑢𝑘* − *𝑦𝑘 ,*

**𝐌**, which is a norm

if **𝐌** is positive definite.

2

**𝐌***𝑢*

=

*⟩*

Here, *‖𝑢‖***𝐌** *⟨𝑢,* is the semi-norm induced by

### The Alternating Direction proximal Method of Multipliers (AD-PMM)

**Input:** pick (*𝑥*0*, 𝑧*0*, 𝑦*0) ∈ 𝖷 × 𝖹 × ℝ*𝑚* and penalty parameter *𝑐 >* 0;

**General step:** For *𝑘* = 0*,* 1*,* … do:

*𝑦𝑘*+1 = (Id +*𝑐𝜕𝑔*∗)−1 (*𝑢𝑘*+1)*.*

*𝑧* − *𝑧𝑘* 2

} (4*.*14)

*𝑥𝑘*+1 = argmin

{*𝑟*(*𝑥*) + *𝑐*

**𝐀***𝑥* − *𝑧𝑘* + 1 *𝑦𝑘* 2 + 1

*𝑥* − *𝑥𝑘* 2

} (4*.*13)

*𝑥*∈𝖷 2 *‖*

To bring this into an equivalent form, let us focus on the definition of

{*𝑔*(*𝑧*) + *𝑐*

**𝐀***𝑥𝑘*+1 − *𝑧* + 1 *𝑦𝑘* 2 + 1

*𝑐 ‖*2 2 *‖*

*‖***𝐌**1

the *𝑦𝑘*+1 update, which reads as the inclusion

*𝑧𝑘*+1 = argmin

*𝑧*∈𝖹 2 *‖*

*𝑐 ‖*2 2 *‖*

*‖***𝐌**2

0 ∈ 1 (*𝑦𝑘*+1

– *𝑢*

*𝑘*+1

) + *𝜕𝑔*∗(*𝑦*

*𝑘*+1 )*.*

*𝑦𝑘*+1 = *𝑦𝑘* + *𝑐*(**𝐀***𝑥𝑘*+1 − *𝑧𝑘*+1) (4*.*15)*.*

*𝑐*

This is clearly recognizable as the first-order optimality condition of

the min {*𝑔*∗(*𝑦*) +  1 *𝑦* − *𝑢𝑘*+1 2}. Therefore, we can rewrite the above

*‖ ‖*

*𝑦* 2*𝑐* 2

iteration in terms of convex optimization subroutines as:

*𝑣𝑘*+1 = argmin{*𝑟*∗(**𝐊***𝑣*) + 1 *𝑣* − (2*𝑦𝑘* − *𝑢𝑘*) 2 }*,* (4.9)

*‖ ‖*2

*𝑣* 2*𝑐*

*𝑢𝑘*+1 = *𝑣𝑘*+1 + *𝑢𝑘* − *𝑤𝑘,* (4.10)

*𝑦𝑘*+1 = argmin{*𝑔*∗(*𝑦*) + 1 *𝑦* − *𝑢𝑘*+1 2}*.* (4.11)

*‖ ‖*2

*𝑦* 2*𝑐*

Via Fenchel-Rockafellar duality, the dual problem to [(4.9)](#_bookmark53) reads as

AD-PMM allows for various choices of the matrices **𝐌**1*,* **𝐌**2.

* With **𝐌**1 = **𝐌**2 = 0, we recover the classical ADMM. For any *𝑐 >* 0,

it is known ([Gabay, 1983; Glowinski and Tallec, 1989](#_bookmark173)) that conver-

gence in function values as well as global convergence to to dual multiplier are warranted. To ensure convergence of the primal se-

quence (*𝑥𝑘*)*𝑘*, one needs to assume that **𝐀** has full column rank.

* With the choice **𝐌**1 = *𝜇*1 **𝐈***𝑛,* **𝐌**2 = *𝜇*2 **𝐈***𝑚* with *𝜇*1 *, 𝜇*2 *>* 0, the AD-PMM

of ([Eckstein, 1994](#_bookmark157)) is recovered.

We give a brief analysis of the complexity of AD-PMM in the special

case of problem [(4.1)](#_bookmark49). Recall that a standing hypothesis in this survey

*𝑥𝑘*+1

= argmin{*𝑟*(*𝑥*) + *𝑐*

**𝐀***𝑥* +

1 (2*𝑦𝑘*

– *𝑢𝑘*)

2 }*,*

is that the smooth part *𝑓* of the composite convex programming prob-

*𝑥* 2 *𝑐* 2

*‖ ‖*

where the coupling between the primal and the dual variables is

*𝑢𝑘*+1 = *𝑦𝑘* + *𝑐***𝐀***𝑥𝑘*+1*.*

The dual to step [(4.11)](#_bookmark54) reads as

lem [(P)](#_bookmark6) admits a Lipschitz continuous gradient. Since *𝑓* (*𝑥*) = *𝑔*(**𝐀***𝑥*), the Lipschitz constant of ∇*𝑓* is determined by a corresponding Lipschitz as- sumption on ∇*𝑔*, with the constant henceforth denoted as *𝐿𝑔* , and a bound on spectrum of the matrix **𝐀**. To highlight the primal-dual na-

ture of the algorithm, a key element in the complexity analysis is the

bifunction

*𝑧𝑘*+1 = argmin{*𝑔*(*𝑧*) + *𝑐 𝑧* − 1 *𝑢𝑘*+1 2}*.* ∗

*𝑧* 2 *‖ 𝑐 ‖*2

*⟨ ⟩*

*𝑆*(*𝑥, 𝑦*) = *𝑟*(*𝑥*) − *𝑔* (*𝑦*) + *𝑦,* **𝐀***𝑥* = *𝐿*(*𝑥,* 0*, 𝑦*)*.*

Our derivation of an iteration complexity estimate of AD-PMM proceeds in two steps. First, we present an interesting “Meta-Theorem”, due to [Shefi and Teboulle (2014)](#_bookmark212), and stated here as [Proposition 4.2](#_bookmark56). It for-

mulates general convergence guarantees for any primal-dual algorithms

Dividing both sides by *𝑁* and using the convexity of the Lagrangian with respect to (*𝑥, 𝑧*) and the linearity in *𝑦*, we easily get

*𝐿*(*𝑥̄ , 𝑧̄ , 𝑦*) − *𝐿*(*𝑥, 𝑧, 𝑦̄* ) ≤ 1 *(𝐶*(*𝑥, 𝑧*) + 1 *‖𝑦* − *𝑦*0*‖*2 *)*

*∑*

*∑*

*∑*

*𝑁*

*𝑁*

*𝑁*

2*𝑁*

*𝑐*

2

satisfying a specific per-iteration bound. We then apply this general re-

sult to AD-PMM, by verifying that this scheme actually satisfies these

in terms of the ergodic average

mentioned per-iteration bounds.

We start with an auxiliary technical fact.

*𝑥̄𝑁*

## = 1

*𝑁*

*𝑁* −1

*𝑥𝑘, 𝑦̄𝑁*

*𝑘*=0

## = 1

*𝑁*

*𝑁* −1

*𝑦𝑘, 𝑧̄𝑁*

*𝑘*=0

## = 1

*𝑁*

*𝑁* −1

*𝑧𝑘,*

*𝑘*=0

**Lemma 4.1.** *Let ℎ* ∶ ℝ*𝑛* → ℝ *be a proper convex and 𝐿ℎ -Lipschitz contin-*

and the constant *𝐶*(*𝑥, 𝑧*) = *𝑐* **𝐀***𝑥* − *𝑧*0 2 +

*𝑥* − *𝑥*0 2 + *𝑧* − *𝑧*0 2 .

*uous. Then, for any 𝜉* ∈ ℝ*𝑛 we have*

[*‖ ‖*](#_bookmark56) *‖ ‖***𝐌**1 *‖ ‖***𝐌**2

*ℎ*(*𝜉*) ≤ max{ *𝜉, 𝑢* − *ℎ*∗(*𝑢*) ∶ *𝑢* ≤ *𝐿* }*.* (4.16)

*⟨ ⟩ ‖ ‖*2 *ℎ*

**Proof.** Since *ℎ* is convex and continuous, it agrees with its biconjugate:

*ℎ*∗∗ = *ℎ*. By Corollary 13.3.3 in [Rockafellar (1970)](#_bookmark245), dom *ℎ*∗ is bounded

averages (*𝑥̄𝑘 , 𝑧̄𝑘 , 𝑦̄𝑘* ) generated by AD-PMM, and derive a *𝑂*(1∕*𝑁* ) Therefore, we can apply [Proposition 4.2](#_bookmark56) to the sequence of ergodic

convergence rate in terms of the function value.

* 1. *Relation to the Chambolle-Pock primal-dual splitting*

with dom *ℎ*∗ *⊆* {*𝑢* ∶

gives

≤ *𝐿* }. Hence, the definition of the conjugate

*‖𝑢‖*2

*ℎ*

*⟩*

*‖ ‖*2 *ℎ ⟨ ⟩*

≤

In this subsection we discuss the relation between ADMM and the

*ℎ*(*𝜉*) = sup { *𝑢, 𝜉*

*⟨*

*𝑢*∈dom *ℎ*∗

□

– *ℎ*∗(*𝑢*)} ≤ max { *𝜉, 𝑢* − *ℎ*∗(*𝑢*)}*.*

*𝑢*∶ *𝑢 𝐿*

celebrated Chambolle-Pock (a.k.a Primal-Dual Hybrid Gradient) method ([Chambolle and Pock, 2011](#_bookmark146)), designed for problems in the form [(2.8)](#_bookmark11).

**Proposition 4.2.** *Let* (*𝑥*∗*, 𝑦*∗*, 𝑧*∗) *be a saddle point for 𝐿. Let*

≥

{(*𝑥𝑘, 𝑦𝑘, 𝑧𝑘*); *𝑘* 0} *be a sequence generated by some algorithm for which the following estimate holds for any 𝑦* ∈ ℝ*𝑚:*

### The Chambolle-Pock primal-dual algorithm (CP)

**Input:** pick (*𝑥*0*, 𝑦*0*, 𝑝*0) ∈ ℝ*𝑛* × ℝ*𝑚* × ℝ*𝑚* and *𝑐, 𝜏 >* 0*, 𝜃* ∈ [0*,* 1];

**General step:** For *𝑘* = 0*,* 1*,* … do:

1

*‖𝑥 𝑥 ‖*

2

*𝑦* − (*𝑦𝑘* + *𝑐***𝐀***𝑥𝑘*+1) 2} (4*.*20)

( *𝑘 𝑘* ) − Ψ( ∗) ≤ 1 *[* ( ∗ ∗) + 1

*‖*

*𝐿 𝑥 , 𝑧 , 𝑦*

*𝑥*

2*𝑘 𝐶 𝑥 , 𝑧*

*𝑐*

*𝑦* − *𝑦*

2

(4.17)

*𝜏*

*𝑦𝑘*+1 = argmin {*𝑔*∗(*𝑦*) + 1

0 *‖*2 *]*

*𝑥𝑘*+1 = argmin*𝑥*{*𝑟*(*𝑥*) + 2 − ( *𝑘* − *𝜏***𝐀***⊤𝑝𝑘*) 2

(4*.*19)

*for some constant 𝐶*(*𝑥*∗*, 𝑧*∗) *>* 0*. Then*

*𝑦* 2*𝑐 ‖ ‖*2

Ψ(*𝑥𝑘*) − Ψ(*𝑥*∗) ≤

*𝐶*1 (*𝑥*∗*, 𝑧*∗*, 𝐿𝑔* )

2*𝑘 .*

*𝑝𝑘*+1 = *𝑦𝑘*+1 + *𝜃*(*𝑦𝑘*+1 − *𝑦𝑘*) (4*.*21)*.*

*where 𝐶* (*𝑥*∗*, 𝑧*∗*, 𝐿* ) = *𝐶*(*𝑥*∗*, 𝑧*∗) + 2 (*𝐿*2 + *𝑦*0 2)*.*

1 *𝑔*

*𝑐 𝑔*

*‖ ‖*2

For later references it is instructive to write this algorithm slightly

**Proof.** Thanks to the Fenchel inequality

*𝐿*(*𝑥, 𝑧, 𝑦*) − *𝑆*(*𝑥, 𝑦*) = *𝑔*(*𝑧*) + *𝑔*∗(*𝑦*) − *𝑦, 𝑧* ≥ 0*.*

*⟨*

*⟩*

of the step *𝑥𝑘*+1, we see differently in operator-theoretic notation. From the optimality condition

0 ∈ *𝜕𝑟*(*𝑥𝑘*+1) + 1 (*𝑥𝑘*+1 − *𝑤𝑘*) ⇔ 0 ∈ (Id +*𝜏𝜕𝑟*)(*𝑥𝑘*+1) − *𝑤𝑘*

By the definition of the convex conjugate

Ψ( ∗

*𝑥*) = *𝑔*(**𝐀***𝑥*) + *𝑟*(*𝑥*) = sup{*𝑟*(*𝑥*) + *𝑦,* **𝐀***𝑥* − *𝑔* (*𝑦*)} = sup *𝑆*(*𝑥, 𝑦*)*.*

*⟨ ⟩*

*𝑦 𝑦*

Now, since *𝑔* is convex and continuous on ℝ*𝑚*, we know *𝑔* = *𝑔*∗∗, and we

can apply [Lemma 4.1](#_bookmark55) to obtain the string of inequalities:

*𝜏*

where *𝑤𝑘* = *𝑥𝑘* − *𝜏***𝐀***⊤𝑝𝑘*. Hence, we can give an explicit expression of

the update as

*𝑥𝑘*+1 = (Id +*𝜏𝜕𝑟*)−1 (*𝑤𝑘*) = (Id +*𝜏𝜕𝑟*)−1 (*𝑥𝑘* − *𝜏***𝐀***⊤𝑝𝑘*)*.*

Similarly, we can write the update *𝑦𝑘*+1 explicitly as

Ψ(*𝑥𝑘* ) − Ψ(*𝑥*∗ ) = sup{*𝑆*(*𝑥𝑘 , 𝑦*) − Ψ(*𝑥*∗ )}≤ sup

{*𝑆*(*𝑥𝑘 , 𝑦*) − Ψ(*𝑥*∗ )} ≤ sup {*𝐿*(*𝑥𝑘 , 𝑧𝑘 , 𝑦*)−Ψ(*𝑥*∗ )} *𝑦𝑘*+1 = (Id +*𝑐𝜕𝑔*∗)−1 (*𝑦𝑘* + *𝑐***𝐀***𝑥𝑘*+1 )*.*

*𝑦 𝑦*∶ *𝑦* ≤*𝐿𝑔*

*‖ ‖*2

*‖ ‖*2

≤ sup

*{ (*

*‖ ‖*2

1

*𝐶*(*𝑥*∗ *, 𝑧*∗ ) + 1 *𝑦* − *𝑦*0 2

*𝑐 ‖*

*𝑦*∶ *𝑦* ≤*𝐿*

*𝑔*

2*𝑘*

*𝑦*∶ *𝑦* ≤*𝐿𝑔*

*)} [ ]*

≤ 1

*𝐶*(*𝑥*∗ *, 𝑧*∗ ) + 2 (*𝐿* + *𝑦*0 2 ) *.*

When *𝜃* = 0 we obtain the classical Arrow-Hurwicz primal-dual algo-

*‖*2

*‖*

*‖*2

= 1 the last line in CP becomes

2*𝑘*

*𝑐*

*𝑔*

rithm ([Arrow et al., 1958](#_bookmark116)). For

*𝜃*

□

To apply this Meta-Theorem, we need to verify that AD-PMM satis- fies the condition [(4.17)](#_bookmark57). To make progress towards that end, Lemma

4.2 in [Shefi and Teboulle (2014)](#_bookmark212) proves that

*𝐿*(*𝑥𝑘*+1*, 𝑧𝑘*+1*, 𝑦*) − *𝐿*(*𝑥, 𝑧, 𝑦𝑘*+1) ≤ *𝑇* (*𝑥, 𝑧, 𝑥𝑘*+1) + *𝑅* (*𝑥, 𝑦, 𝑧*) (4.18)

*𝑘 𝑘*

for all (*𝑥, 𝑧, 𝑦*) ∈ 𝖷 × 𝖹 × ℝ*𝑚* and some explicitly given functions *𝑇𝑘* and

*𝑅𝑘* . Furthermore, it is shown that

*𝑝𝑘*+1 = 2*𝑦𝑘*+1 − *𝑦𝑘*, which corresponds to a simple linear extrapolation

[Pock (2011) provide a *𝑂*(1∕*𝑁* ) non-asymptotic convergence guarantees based on the current and previous iterates. In this case, Chambolle and](#_bookmark146)

in terms of the primal-dual gap function of the corresponding saddle- point problem. The CP primal-dual splitting method has been of im- mense importance in imaging and signal processing and constitutes nowadays a standard method for tackling large-scale instances in these

application domains. Interestingly, if *𝜃* = 1, CP is a special case of the

proximal version of ADMM (AD-PMM). To establish this connection, let

1 *⊤*

*𝑇* (*𝑥, 𝑧, 𝑥𝑘*+1) ≤ *𝑐 (‖***𝐀***𝑥* − *𝑧𝑘‖*2 − *‖***𝐀***𝑥* − *𝑧𝑘*+1*‖*2 + *𝑐 ‖***𝐀***𝑥𝑘*+1 − *𝑧𝑘*+1*‖*2 *),* and

*𝑘*

2

2

2

2

2

tions, we arrive at the update formula for *𝑥𝑘*+1 in AD-PMM (4.13) as

*𝑥*

*𝑥* − (*𝑥*

– *𝜏***𝐀** (*𝑦*

+ *𝑐*(**𝐀***𝑥*

– *𝑧* ))) 2

us set **𝐌**1 = *𝑐* Id −*𝑐***𝐀 𝐀** and **𝐌**2 = 0. After some elementary manipula-

*𝑅* (*𝑥, 𝑧, 𝑦*) ≤ 1 *(*Δ (*𝑥,* **𝐌** ) + Δ (*𝑧,* **𝐌** ) + 1 Δ (*𝑦,* Id)*)* − *𝑐 ‖***𝐀***𝑥𝑘*+1 − *𝑧𝑘*+1*‖*2 *,*

*𝑘*

2

*𝑘*

1

*𝑘*

2

*𝑐*

*𝑘*

2

2

where for any point *𝑧* and positive semi-definite matrix **𝐌**,

*𝑘*+1

*𝑥*

1 *𝑘*

*⊤ 𝑘*

*𝑘 𝑘*

2 }*.*

Δ (*𝑧,* **𝐌**) = 1

= argmin{*𝑟*(*𝑥*) + 2*𝜏*

*𝑧* − *𝑧𝑘* 2 − 1

*𝑧* − *𝑧𝑘*+1 2 *.*

lently as

*𝑘* 2 *‖ ‖***𝐌** 2 *‖ ‖***𝐌**

*‖*

*‖*

Introducing the variable *𝑝𝑘* = *𝑦𝑘* + *𝑐*(**𝐀***𝑥𝑘* − *𝑧𝑘*), the above reads equiva-

*𝑥𝑘*+1 = argmin{*𝑟*(*𝑥*) + 1

*𝑥* − (*𝑥𝑘* − *𝜏***𝐀***⊤𝑝𝑘*) 2 } = Prox

(*𝑥𝑘* − *𝜏***𝐀***⊤𝑝𝑘*)*.*

Using these bounds and summing inequality [(4.18)](#_bookmark58) over *𝑘* = 0*,* 1*,* … *, 𝑁* −

1, we get

*𝑥* 2*𝜏 ‖*

*‖*2 *𝜏𝑟*

*𝑁* −1

*∑*

For **𝐌**2 = 0, the second update step in AD-PMM (4.14) reads as

[*𝐿*(*𝑥*

*𝑘*+1

*, 𝑧𝑘*+1

*, 𝑦*) − *𝐿*(*𝑥, 𝑧, 𝑦*

*𝑘*+1

)] ≤

1 *(𝑐* **𝐀***𝑥* − *𝑧*0 2

+ *𝑥* − *𝑥*0 2

+ *𝑧* − *𝑧*0 2

+ 1 *𝑦* − *𝑦*

0 2 *)*

*𝑧𝑘*+1

= (Id +

1 *𝜕𝑔*)

−1 *(*

**𝐀***𝑥*

*𝑘*+1 +

1 *𝑦*

*𝑘)*

= Prox 1

*(* 1 (*𝑐***𝐀***𝑥*

*𝑘*+1

+ *𝑦𝑘*)*).*

*𝑘*=0

2 *‖ ‖*2 *‖*

*‖***𝐌**1 *‖*

*‖***𝐌**2

*𝑐 ‖ ‖*2

*𝑐 𝑐*

*𝑐 𝑔 𝑐*

Moreau’s identity [Bauschke and Combettes (2016](#_bookmark96), Proposition 23.18) states that

*𝑐* Prox 1 *𝑔* (*𝑢*∕*𝑐*) + Prox*𝑐𝑔*∗ (*𝑢*) = *𝑢* ∀*𝑢* ∈ 𝖵*.* (4.22)

*𝑐*

The practical application of an LO requires to make a selection from the set of solutions of the defining linear minimization problem. The precise definition of such a selection mechanism is not of any impor-

tance, and thus we are just concerned with any answer (*𝑦*) revealed

𝖷

G

Applying this fundamental identity, we see by the oracle.

*𝑐𝑧𝑘*+1 + Prox*𝑐𝑔*∗ (*𝑦𝑘* + *𝑐***𝐀***𝑥𝑘*+1) = *𝑦𝑘* + *𝑐***𝐀***𝑥𝑘*+1*.*

The second summand is just the *𝑦𝑘*+1-update in the CP algorithm, so that

we deduce

*𝑐𝑧𝑘*+1 + *𝑦𝑘*+1 = *𝑦𝑘* + *𝑐***𝐀***𝑥𝑘*+1 ⇔ *𝑦𝑘*+1 = *𝑦𝑘* + *𝑐*(**𝐀***𝑥𝑘*+1 − *𝑧𝑘*+1)*.*

Consequently,

*𝑝𝑘*+1 = *𝑦𝑘*+1 + *𝑐*(**𝐀***𝑥𝑘*+1 − *𝑧𝑘*+1) = 2*𝑦𝑘*+1 − *𝑦𝑘,*

and hence we recover the three-step iteration defining CP:

The information-theoretic assumption that the optimizer can only query a linear minimization oracle is clearly the main difference be- tween CG and other gradient-based methods discussed in [Section 3](#_bookmark12). For instance, the dual averaging algorithm solves at each iteration a strongly convex subproblem of the form

min{ *𝑦, 𝑢* + *ℎ*(*𝑢*)}*,* (5.3)

*𝑢*∈𝖷 *⟨ ⟩*

where *ℎ* ∈  (𝖷), whereas CG solves a single linear minimization prob-

*𝛼*

lem at each iteration. This difference in the updating mechanism yields

*𝑥𝑘*+1

= argmin{*𝑟*(*𝑥*) + 1 *𝑥* − (*𝑥𝑘* − *𝜏***𝐀***⊤𝑝𝑘*) 2 }

*𝑥* 2*𝜏*

*‖ ‖*2

the following potential advantages of the CG method.

1. Low iteration costs: In many cases it is much easier to construct an

*𝑦𝑘*+1 = argmin{*𝑔*∗(*𝑦*) + 1

*𝑦* − (*𝑦𝑘* + *𝑐***𝐀***𝑥𝑘*+1) 2 }

LO rather than solving the non-linear subproblem [(5.3)](#_bookmark59). We empha-

*𝑦*

*𝑝𝑘*+1 = 2*𝑦𝑘*+1 − *𝑦𝑘 .*

2*𝑐 ‖ ‖*2

ture of the objective function *𝑓* , but rather on the geometry of the size that this potential benefit of CG does not depend on the struc-

Given the above derivations, we can summarize this subsection by the following interesting observation.

**Proposition 4.3** (Proposition 3.1, [Shefi and Teboulle (2014)](#_bookmark212))**.** *Let*

(*𝑥𝑘, 𝑦𝑘, 𝑝𝑘*) *be a sequence generated by CP with 𝜃* = 1*. Then, the 𝑦𝑘*+1*-update*

*(4.20) is equivalent to*

feasible set 𝖷. To illustrate this point, consider the spectrahedron

𝖷 = {**𝐗** ∈ ℝ*𝑛*×*𝑛* **𝐗** ⪰ 0*,* tr(**𝐗**) 1}. Computing the orthogonal projec- tion of some symmetric matrix **𝐘** onto the spectrahedron requires first to compute the full spectral decomposition **𝐘** = **𝐔𝐃𝐔***⊤* , and then for the diagonal matrix **𝐃** computing the projection of its diagonal el-

sym *|*

≤

ements onto the simplex. The resulting projection is therefore given

*𝑧𝑘*+1 = argmin{*𝑔*(*𝑧*) + *𝑐*

**𝐀***𝑥𝑘*+1 − *𝑧* + 1 *𝑦𝑘* 2}*,* by

*𝑧* 2 *‖*

*𝑐 ‖*2

*𝑃*𝖷(**𝐘**) = **𝐔** Diag(*𝑃*

Δ*𝑛*

(diag(**𝐃**)))**𝐔***⊤.*

*𝑦𝑘*+1 = *𝑦𝑘* + *𝑐*(**𝐀***𝑥𝑘*+1 − *𝑧𝑘*+1)

*which corresponds to the primal 𝑧𝑘*+1*-minimization step (4.14) with* **𝐌**2 = 0*,*

In contrast, computing a linear oracle over 𝖷 for the symmetric

matrix **𝐘** involves finding the eigenvector of **𝐘** corresponding to

𝖷

*and to the dual multiplier update for 𝑦𝑘*+1 *(4.15) of AD-PMM, respectively.*

the minimal eigenvalue, that is G (**𝐘**) = *𝑢𝑢⊤*, where *𝑢⊤***𝐘***𝑢* = *𝜆*

min

(**𝐘**).

*Moreover, the minimization step with respect to 𝑥 in the CP algorithm given*

*in (4.19) together with (4.15) reduces to (4.13) of AD-PMM with* **𝐌**1 =

*𝜏* Id −*𝑐***𝐀***⊤***𝐀***.*

### The Conditional Gradient Method

The eﬃciency of the Bregman proximal gradient method stands and falls with the relative ease of evaluating the Bregman proximal operator [(3.7)](#_bookmark26). In this section, we present a class of first-order methods which gain relevance in large-scale problems for which the computation of the projection-like operators is a significant computational bottleneck. We describe *conditional gradient* (CG) methods, a family of methods which, originating in the 1960’s, have received much attention in both machine learning and optimization in the last 20 years. CG is designed to solve convex programming problems over compact convex sets. Therefore, we assume in this section that the feasible set 𝖷 is a compact convex set.

**Assumption 9.** The set 𝖷 is a compact convex subset in a finite- dimensional real vector space 𝖵.

* 1. *Classical Conditional gradient*

CG, also known as the *Frank-Wolfe method*, was independently de- veloped by Frank and Wolfe [Frank and Wolfe (1956)](#_bookmark167) for linearly con- [strained quadratic problems, and by Levitin and Polyak Levitin and Polyak (1966) for general smooth convex optimization problems over](#_bookmark192) compact domains:

*|*

This operation can be typically done using numerical linear algebra techniques such as Power, Lanczos or Kaczmarz, and randomized versions thereof (see [Kuczyński and Woźniakowski (1992)](#_bookmark184) for gen- eral complexity results). For large-scale problems, computing such a leading eigenvector to a predefined accuracy is much more eﬃcient than a full spectral decomposition.

1. Simplicity: The definition of an LO does not rely on a specific DGF



*ℎ* ∈ (𝖷) and makes the update aﬃne invariant.

*𝛼*

1. Structural properties of the updates: When the feasible set 𝖷 can be

represented as the convex hull of a countable set of atoms (”gener- ators”), then CG often leads to simple updates, activating only few atoms at each iteration. In particular, in the case of the spectrahe- dron, the LO returns a matrix of rank one, which allows for sparsity preserving iterates.

the LO at a given gradient feedback *𝑦* = ∇*𝑓* (*𝑥*), and returns the target The classical form of CG takes the answer obtained from querying

vector

*𝑝*(*𝑥*) = G (∇*𝑓* (*𝑥*)) ∀*𝑥* ∈ 𝖷*.* (5.4)

𝖷

It proposes then to move in the direction *𝑝*(*𝑥*) − *𝑥*. As in every opti-

rules to guarantee reasonable numerical performance. Letting *𝑥𝑘*−1 and mization routine, a key question is how to design eﬃcient step-size

*𝑝𝑘* = *𝑝*(*𝑥𝑘*−1) be a current position of the method together with its im-

plied target vector, the following policies are standard choices:

Standard: *𝛾* = 1 *,* (5.5)

*𝑘*

Ψmin

(𝖷) ∶= min{*𝑓* (*𝑥*) *𝑥* ∈ 𝖷}*.* (5.1)

2 + *𝑘*

CG attempts to solve problem [(5.1)](#_bookmark62) by sequentially calling a *linear oracle*

(LO), a fundamental notion we introduce next.

Exact line search: *𝛾𝑘* ∈ argmin *𝑓* (*𝑥𝑘*−1 + *𝑡*(*𝑝𝑘* − *𝑥𝑘*−1))*,* (5.6)

*𝑡*∈(0*,*1]

**Definition 5.1.** The Operator G ∶ 𝖵∗ → 𝖷 is a *linear oracle* (LO) over

∇

set

𝖷 *⟨ ⟩*

*‖*

*𝑘*

*𝑓 ‖𝑥*

𝖷 if for any vector

*𝑦* ∈

𝖵∗

𝖷

we have that

G (*𝑦*) ∈ argmin *𝑦, 𝑠 .* (5.2)

*𝑠*∈𝖷

Adaptive: *𝛾*

= min *{ ⟨*

*𝑓* (*𝑥𝑘*−1)*, 𝑥𝑘*−1 − *𝑝𝑘*

*𝐿*

*𝑘*−1 −

*𝑝𝑘* 2

*⟩ ,* 1

*}.* (5.7)

Exact line search is conceptually attractive, but can be costly in large- scale applications when computing the function value is computation-

Note that when choosing *ℎ* to be the squared Euclidean norm *ℎ*(*𝑥*) =

1 *𝑥* 2 and *𝐿ℎ* = *𝐿* , then Assumption [10](#_bookmark68) is equivalent to the Lipschitz

2 *‖ ‖*

*𝑓 𝑓*

ally expensive. To understand the construction of the adaptive step-size scheme, it is instructive to introduce a primal gap (merit) function to the problem defined as

𝚎(*𝑥*) ∶= sup ∇*𝑓* (*𝑥*)*, 𝑥* − *𝑢 .* (5.8)

*⟨ ⟩*

*𝑢*∈𝖷

[This merit function is just the gap program (see e.g. Facchinei and Pang (2003)) associated to the monotone variational inequality](#_bookmark161) [(2.6)](#_bookmark17) [in](#_bookmark161) which the non-smooth part is trivial. In terms of this merit function, the descent lemma [(3.15)](#_bookmark34) yields immediately

gradient assumption, where Ω*ℎ*(𝖷) is the diameter of set 𝖷. On the other

hand, choosing *ℎ*(*𝑥*) = *𝑓* (*𝑥*) and *𝐿ℎ* = *𝐿𝑓* , we essentially retrieve the fi-

*𝑓*

nite curvature assumption used by Jaggi [Jaggi (2013)](#_bookmark200).

**Remark 5.1.** It is clear that the finite curvature assumption [(5.10)](#_bookmark69) is not compatible with the DGF to be essentially smooth on 𝖷. We are therefore forced to work with non-steep distance-generating functions.

The analysis of CG under a relative smoothness condition and [Assumption 10](#_bookmark68) runs in the same way as for the classical CG. However, the adaptive step-size is reformulated as

*⟨*

*𝑓* (*𝑥* + *𝑡*(*𝑝*(*𝑥*) − *𝑥*)) ≤

( ) +

∇ ( )

( ) −

+ *𝐿𝑓 𝑡*2

( ) − 2

*{ 𝑘*−1

*𝑘*−1

*𝑘 }*

*𝑓 𝑥 𝑡⟨ 𝑓 𝑥 , 𝑝 𝑥*

*𝑘*

*𝑥⟩* 2 *‖𝑝 𝑥 𝑥‖*

*𝛾* = min

∇*𝑓* (*𝑥*

)*, 𝑥*

– *𝑝*

*,* 1 *.*

*⟩*

= *𝑓* (*𝑥*) − *𝑡*𝚎(*𝑥*) + *𝐿𝑓 𝑡*

2

*𝐿ℎ* Ω2 (𝖷)

*𝑝*(*𝑥*) − *𝑥* 2 = *𝑓* (*𝑥*) − *𝜂* (*𝑡*)*,*

*𝑓 ℎ*

*𝐿 𝑡*2

2 *‖ ‖ 𝑥*

This can be easily seen by replacing the upper model function *𝑓* (*𝑥*) −

*𝑡*𝚎(*𝑥*) + *𝐿ℎ 𝐷ℎ* (*𝑥* + *𝑡*(*𝑝* − *𝑥*)*, 𝑥*), with its more conservative bound *𝑓* (*𝑥*) −

where *𝜂* (*𝑡*) ∶= *𝑡*𝚎(*𝑥*) − *𝑓 𝑝*(*𝑥*) − *𝑥* 2. Optimizing this function with re- *𝑓*

*𝑥* 2 *‖ ‖*

spect to *𝑡* ∈ [0*,* 1] yields the largest-possible per-iteration decrease and

*𝑡*𝚎(*𝑥*) + *𝑓* Ω (𝖷). Of course, in the case of the Euclidean norm this re-

2

*ℎ*

*𝐿ℎ 𝑡*2 2

returns the adaptive step-size rule in [(5.7)](#_bookmark63). Once the optimizer decided upon the specific step-size policy, the classical CG picks one of the step sizes [(5.5), (5.6)](#_bookmark61), or [(5.7)](#_bookmark63), and performs the update

*𝑥𝑘* = *𝑥𝑘*−1 + *𝛾𝑘* (*𝑝*(*𝑥𝑘*) − *𝑥𝑘*−1)*.*

### The classical conditional gradient (CG)

**Input:** A linear oracle , a starting point *𝑥*0 ∈ 𝖷. **Output:** A solution *𝑥* such that Ψ(*𝑥*) − Ψmin(𝖷) *< 𝜀*. **General step:** For *𝑘* = 1*,* 2*,* …

𝖷

G

G

Compute *𝑝𝑘* = (∇*𝑓* (*𝑥𝑘*−1));

*𝑋*

sults in a smaller step-size than the adaptive step, which hints towards a deterioration of performance. Nevertheless, this trick allows us to han- dle convex programming problems outside the Lipschitz smooth case, [which is not uncommon in various applications (Bian and Chen, 2015; Bian et al., 2015; Haeser et al., 2018).](#_bookmark117)

* 1. *Generalized Conditional Gradient*

Introduced by Bach [Bach (2015)](#_bookmark135) and [Nesterov (2018a)](#_bookmark227), the general- ized conditional gradient (GCG) method, is targeted to solve our master problem [(P)](#_bookmark6) over a compact set 𝖷. To handle the composite case, we need to modify our definition of a linear oracle accordingly.

Choose a step-size *𝛾𝑘* either by (5.5), (5.6), (5.7);

Update *𝑥𝑘* = *𝑥𝑘*−1 + *𝛾𝑘* (*𝑝𝑘* − *𝑥𝑘*−1);

**Definition 5.2.** Operator G

𝖷*,𝑟*

∶ 𝖵∗ → 𝖷 is a *generalized linear oracle*

∗

Compute 𝚎*𝑘* = 𝚎(*𝑥𝑘*−1).

If 𝚎*𝑘 < 𝜀* return *𝑥𝑘*.

(GLO) over set 𝖷 with respect to function *𝑟* if for any vector *𝑦* ∈ 𝖵 we

have that

G (*𝑦*) ∈ argmin *𝑦, 𝑥* + *𝑟*(*𝑥*)*.*

𝖷*,𝑟*

*𝑥*∈𝖷 *⟨ ⟩*

The convergence properties of classical CG under either of the step- size variants above is well documented in the literature (see e.g. the recent text by [Lan (2020)](#_bookmark188), or [Jaggi (2013)](#_bookmark200)). We will obtain a full con-

vergence and complexity theory under our more general analysis of the

Besides this more demanding oracle assumption, the resulting gener- alized conditional gradient method is formally identical to the classical CG. In particular, we can consider the target vector

generalized CG scheme.

*𝑝*(*𝑥*) = G

𝖷*,𝑟*

(∇*𝑓* (*𝑥*)) ∀*𝑥* ∈ 𝖷 (5.11)

* + 1. *Relative smoothness*

The basic ingredient in proving convergence and complexity results on the classical CG is the fundamental inequality

*𝑓* (*𝑥* + *𝑡*(*𝑝*(*𝑥*) − *𝑥*)) ≤ (*𝑥*) − *𝑡*𝚎(*𝑥*) + *𝐿𝑓 𝑡 𝑝*(*𝑥*) − *𝑥* 2*.*

2

*𝑓 ‖* [*‖*](#_bookmark35)

2

and the same three step size policies as in the classical CG, with the standard step size remaining the same and the obvious modifications for the two other step size policies:

Exact line search: *𝛾𝑘* ∈ argmin Ψ(*𝑥𝑘*−1 + *𝑡*(*𝑝𝑘* − *𝑥𝑘*−1))*,* (5.12)

*𝑡*∈[0*,*1]

*{ 𝑟*(*𝑥𝑘*−1) − *𝑟*(*𝑝𝑘*) + ∇*𝑓* (*𝑥𝑘*−1)*, 𝑥𝑘*−1 − *𝑝𝑘 }*

Based on the relative smoothness analysis in [Section 3.3.3](#_bookmark35), it seems to be intuitively clear that we could easily prove also convergence of CG when instead of the restrictive Lipschitz gradient assumption we make

Adaptive: *𝛾𝑘* = min

*𝐿𝑓 ‖𝑥𝑘⟨*−1 − *𝑝𝑘 ‖*2

*⟩ ,* 1

*.*

(5.13)

a relative smoothness assumption in terms of the pair (*𝑓, ℎ*) for some DGF *ℎ* ∈ *𝛼*(𝖷). Indeed, if we are able to estimate a scalar *𝐿ℎ >* 0 such



*𝑓*

that *𝐿ℎ ℎ*(*𝑥*) − *𝑓* (*𝑥*) is convex on 𝖷, then the modified Descent Lemma

*𝑓*

[(3.19)](#_bookmark38) yields the overestimation

*𝑓* (*𝑥* + *𝑡*(*𝑝* − *𝑥*)) ≤ *𝑓* (*𝑥*) − *𝑡*𝚎(*𝑥*) + *𝐿ℎ 𝐷* (*𝑥* + *𝑡*(*𝑝* − *𝑥*)*, 𝑥*)*.* (5.9) Instead of requiring that *𝑓* has a Lipschitz continuous gradient over the convex compact set 𝖷, let us alternatively require the following:

**Assumption 10.** There exists a DGF *ℎ* ∈  (𝖷) and a constant *𝐿ℎ >* 0,

*𝑓 ℎ*

The adaptive step size variant is derived from an augmented merit func- tion, taking into consideration the non-smooth composite nature of the underlying optimization problem. Indeed, as again can be learned from the basic theory of variational inequalities (see [Nesterov (2007)](#_bookmark219)), the natural merit function for the composite model problem [(P)](#_bookmark6) is the non- smooth function

𝚎(*𝑥*) = sup Γ(*𝑥, 𝑢*)*,* where Γ(*𝑥, 𝑢*) ∶= *𝑟*(*𝑥*) − *𝑟*(*𝑢*) + ∇*𝑓* (*𝑥*)*, 𝑥* − *𝑢 .* (5.14)

*⟨ ⟩*

*𝑢*∈𝖷

By definition, we see that 𝚎(*𝑥*) ≥ 0 for all *𝑥* ∈ 𝖷, with equality if and

*𝛼 𝑓* ∗

such that *𝐿ℎ ℎ* − *𝑓* is convex on 𝖷, and *ℎ* has a finite curvature on 𝖷,

only if *𝑥* ∈ 𝖷 *.* These basic properties justify our terminology, calling

*𝑓*

that is,

2*𝐷* (*𝑡𝑢* + (1 − *𝑡*)*𝑥, 𝑥*)

𝚎(*𝑥*) a merit function. Of course, 𝚎(⋅) is also easily seen to be convex. Furthermore, using the convexity of *𝑓* , one first sees that

Ω2 (𝖷) ∶= sup *ℎ*

*<* ∞*.* (5.10)

∇ ( ) −

≥ ( ) − ( )

*ℎ 𝑥,𝑢*∈𝖷*,𝑡*∈[0*,*1]

*𝑡*2

*⟨ 𝑓 𝑥 , 𝑥*

*𝑢⟩*

*𝑓 𝑥 𝑓 𝑢 ,*

so that for all *𝑥, 𝑢* ∈ dom(*𝑟*),

Γ(*𝑥, 𝑢*) ≥ *𝑟*(*𝑥*) − *𝑟*(*𝑢*) + *𝑓* (*𝑥*) − *𝑓* (*𝑢*) = Ψ(*𝑥*) − Ψ(*𝑢*)*.*

From here, one immediately arrives at the relation

where the second inequality follows from *𝑠𝐾 <* min{*𝐿𝑓* Ω2 *, 𝑠*0}, the third inequality follows from *𝑎* being a monotonic function in *𝑎* 0 for

any *𝑘* ≥ *𝐾* + 1, and the last inequality follows from *𝐾* ≤ max *{*2*,*  *𝑠*0 *}*.

*𝑎*+(*𝑘*−*𝐾*)

≥

*𝐿𝑓* Ω2

𝚎(*𝑥*) ≥ Ψ(*𝑥*) − Ψmin(𝖷)*.* (5.15) Combining these two results, we have that

*𝑠𝑘* ≤

Clearly, with *𝑟* = 0, the above specification yields the classical CG. 2 max{*𝑠*0 *, 𝐿𝑓* Ω2 }

*.*

* + 1. *Basic Complexity Properties of GCG 𝑘*

We now turn to prove that the GCG method with one of the above □

mentioned step-sizes converges at a rate of *𝑂*( 1 ). We will derive this rate

*𝑘*

under the standard Lipschitz smoothness assumption on *𝑓* . This gives us

assumed convexity of the non-smooth function *𝑟*(⋅), we readily obtain access to the classical descent lemma [(3.15)](#_bookmark34). Combining this with the

*𝑡*2 *𝐿*

* + 1. *Alternative assumptions and step-sizes*

A key takeaway from the analysis of the generalized conditional gra- dient is that one needs to have a bound on the quadratic term of the upper model

Ψ(*𝑥𝑘*−1 + *𝑡*(*𝑝𝑘* − *𝑥𝑘*−1)) ≤ *𝑓* (*𝑥𝑘*−1) − *𝑡* ∇*𝑓* (*𝑥𝑘*−1)*, 𝑝𝑘* − *𝑥𝑘*−1 +

*𝑓 𝑝𝑘* − *𝑥𝑘*−1 2

*𝑘*−*⟨*1 *𝑘*

*⟩* 2 *‖ ‖*

*𝑡* ↦ *𝑄*(*𝑥, 𝑝*(*𝑥*)*, 𝑡, 𝐿* ) ∶= Ψ(*𝑥*) − *𝑡*𝚎(*𝑥*) +

*𝐿𝑓 𝑡*2

*𝑝*(*𝑥*) − *𝑥* 2*.*

+ (1 − *𝑡*)*𝑟*(*𝑥* ) + *𝑡𝑟*(*𝑝* )

*𝑡*2 *𝐿𝑓*

*𝑓* 2 *‖ ‖*

= Ψ(*𝑥𝑘*−1) − *𝑡*𝚎(*𝑥𝑘*−1) +

*𝑝𝑘* − *𝑥𝑘*−1 2 *.*

Such a bound was given to us essentially for free under the compactness

2

*‖ ‖*

Based on this fundamental inequality of the per-iteration decrease, we

can deduce the iteration complexity via an induction argument. First, one observes that for each of the three introduced step-size rules (stan- dard, line search and adaptive), one obtains a recursion of the form

Ψ(*𝑥𝑘*−1 + *𝛾* (*𝑝𝑘* − *𝑥𝑘*−1)) ≤ Ψ(*𝑥𝑘*−1) − *𝛾* 𝚎(*𝑥𝑘*−1) + *𝑘 𝑝𝑘* − *𝑥𝑘* 2*.*

*𝐿𝑓 𝛾*2

on the smooth part *𝑓* . The resulting complexity constant is then de- assumption of the domain 𝖷, and the Lipschitz-smoothness assumption termined by *𝐿𝑓* Ω2 . Moreover, this constant will be involved in lower

bounds of the adaptive step-size rule [(5.13)](#_bookmark67). However, such a constant may not be known, or may be expensive to compute. Moreover, a global

estimate of this constant is not actually needed for obtaining an upper

*𝑘 𝑘*

When denoting *𝑠𝑘* ∶= Ψ(*𝑥𝑘*) − Ψmin(𝖷), 𝚎*𝑘* = 𝚎(*𝑥𝑘*−1) and Ω2 ≡

2 *‖ ‖*

bound. To see this, we proceed formally as follows. Consider an alter-

native quadratic function of the form

Ω2 (𝖷) = max *𝑥* − *𝑢* 2, this gives us 2

2 *‖*⋅*‖*

1 2

*𝑥,𝑢*∈𝖷 *‖ ‖*

*𝑄*(*𝑥, 𝑝, 𝑡, 𝑀* ) ∶= Ψ(*𝑥*) − *𝑡*𝚎(*𝑥*) + *𝑡 𝑀 𝑞*(*𝑝, 𝑥*)*,*

2

*𝐿 𝛾*2

≤ *𝑓*

*𝑠𝑘 𝑠𝑘*−1 − *𝛾* 𝚎*𝑘* + *𝑘* Ω2 *.*

where *𝑞*(*𝑝, 𝑥*) is a positive function bounded by some constant *𝐶*, and

*𝑘* 2

choose *𝛾*(*𝑥, 𝑀* ) ∶= min{1*,*  𝚎(*𝑥*) }, for *𝑝*(*𝑥*) = G (∇*𝑓* (*𝑥*)). Let *𝑀 >* 0

Applying to this recursion Lemma 13.13 in [Beck (2017)](#_bookmark97), we deduce the

*𝑀𝑞*(*𝑝*(*𝑥*)*,𝑥*)

𝖷*,𝑟*

next iteration complexity result for GCG.

**Theorem 5.3.** *Consider algorithm GCG with one of the step size rules: stan- dard* [*(5.5)*](#_bookmark61)*, line search* [*(5.12)*](#_bookmark66)*, or adaptive* [*(5.13)*](#_bookmark67)*. Then*

be a constant such that the point obtained by using this step-size is upper bounded by the corresponding quadratic function, *i.e.*,

Ψ((1 − *𝛾*(*𝑥, 𝑀* ))*𝑥* + *𝛾*(*𝑥, 𝑀* )*𝑝*(*𝑥*)) ≤ *𝑄*(*𝑥, 𝑝*(*𝑥*)*, 𝛾*(*𝑥, 𝑀* )*, 𝑀* ) *<* Ψ(*𝑥*)*.* (5.16)

Thus applying the update *𝑥*+ ∶= (1 − *𝛾*(*𝑥, 𝑀* ))*𝑥* + *𝛾*(*𝑥, 𝑀* )*𝑝*(*𝑥*), we obtain

Ψ(*𝑥𝑘*) − Ψmin

(𝖷) ≤ 2 max{Ψ(*𝑥*0) − Ψmin(𝖷)*, 𝐿𝑓* Ω2 }

*𝑘*

∀*𝑘*

≥ 1*.*

Ψ(*𝑥*+) − Ψmin(𝖷) ≤ Ψ(*𝑥*) − Ψmin(𝖷) − 1 𝚎(*𝑥*) ≤ 1 (Ψ(*𝑥*) − Ψmin(𝖷))

**Proof.** We give a self-contained proof of this result for the adaptive step-size policy [(5.13)](#_bookmark67).

–

≥

If *𝛾𝑘* = 1, the per-iteration progress is easily seen that 𝚎*𝑘 𝐿𝑓 ‖𝑝𝑘*

– *𝑘* + *𝐿𝑓*

*𝑝𝑘* − *𝑥𝑘*−1 2 ≤ −𝚎*𝑘*

*‖*

2 *‖*

*‖*

2 and thus

*𝑥𝑘*−1 2

which implies

𝚎

2 2

if *𝛾*(*𝑥, 𝑀* ) = 1, and

Ψ(*𝑥*+) − Ψmin(𝖷) ≤ Ψ(*𝑥*) − Ψmin(𝖷) − 1 𝚎(*𝑥*)2 ≤ Ψ(*𝑥*) − Ψmin(𝖷)

2*𝑀𝑞*(*𝑝*(*𝑥*)*, 𝑥*)

– 1

*𝑀𝐶*

(Ψ(*𝑥*) − Ψmin(𝖷))2

*𝑠𝑘* ≤ *𝑠𝑘*−1 − 𝚎*𝑘* + *𝐿𝑓*

*𝑝𝑘* − *𝑥𝑘*−1 2 ≤ *𝑠𝑘*−1 − 1 𝚎*𝑘* ≤ *𝑠𝑘*−1 − 1 *𝑠𝑘*−1 = 1 *𝑠𝑘*−1

𝚎(*𝑥*) *𝑘*

2 *‖ ‖* 2

2

where for the last inequality we use [(5.15)](#_bookmark70). For *𝛾𝑘* =

way, we get the familiar recursion

2 2 if *𝛾*(*𝑥, 𝑀* ) = *𝑀𝑞*(*𝑝*(*𝑥*)*,𝑥*) . If (*𝑥* )*𝑘*≥0 is the trajectory defined in this specific

*⟨ 𝑘*−1 −*𝑝𝑘 ‖*2

*‖*

*𝑟*(*𝑥𝑘* )−*𝑟*(*𝑝𝑘* )+ ∇*𝑓* (*𝑥𝑘*−1 )*,𝑥𝑘*−1 −*𝑝𝑘*

*𝐿𝑓 𝑥*

veals

𝚎*𝑘*

*𝐿𝑓 ‖𝑝*

2

2*𝑀𝑘𝐶*

*⟩* = *𝑘* −*𝑥𝑘*−1 2 , a simple computation re-

*𝑠𝑘* ≤ min{ 1 *𝑠𝑘*−1*, 𝑠𝑘*−1 − 1

(*𝑠𝑘*−1)2 }

*𝑠𝑘* ≤ *𝑘*−1 (𝚎*𝑘* )2

*‖*

*𝑠* −

≤ *𝑘*−1 (𝚎*𝑘*)2

≤ *𝑘*−1 − (*𝑠𝑘*−1)2

in terms of the approximation error *𝑠𝑘* ∶= Ψ(*𝑥𝑘*) − Ψmin(𝖷), and the local

2*𝐿*

*𝑓 ‖*

*𝑝𝑘* − *𝑥𝑘*−1 2

*𝑠*

2*𝐿𝑓* Ω2

*𝑠* −

*‖*

*𝑓*

2*𝐿* Ω2 *.*

estimates (*𝑀𝑘* )*𝑘*≥0. Thus, as we are able to bound *𝑀𝑘* from above for all

iterations of the algorithm, the same convergence as for GCG can be

achieved.

*𝑘 {* 1 *𝑘*−1

Summarizing these two cases, we see

*𝑠*

≤ max

2

*𝑠*

Lipschitz smooth objective functions, we can try to determine *𝑀*

via a

*𝑘*−1

(*𝑠𝑘*−1 )2 *}*

Based on this observation, and knowing that *𝑀𝑘* must be bounded for

*𝑘*

Thus, the convergence is split into two periods, which are split by *𝐾* ∶= log *⌈*  *𝑠*0 *⌉* . If *𝑘* ≤ *𝐾* then *𝑠𝑘*−1 ≥ *𝐿* Ω2 and thus *𝑠𝑘* ≤ 1 *𝑠𝑘*−1,

*( )*

*, 𝑠*

–

2*𝐿𝑓* Ω2

*.*

2

min{*𝐿𝑓* Ω2 *,𝑠*0 }

*𝑓*

2

level set {*𝑥* ∈ 𝖷 Ψ(*𝑥*) ≤ Ψ(*𝑥*0)}. Hence, it is suﬃcient for *𝑄*(*𝑥𝑘, 𝑝𝑘, 𝑡, 𝑀* ) to

struction, the resulting iterates *𝑥𝑘* will induce monotonically decreasing backtracking procedure, as suggested in [Pedregosa et al. (2020)](#_bookmark238). By con- function values so that the whole trajectory *𝑥𝑘* will be contained in the

which implies

*𝑠𝑘* ≤ 2−*𝑘𝑠*0*, 𝑘* ∈ {0*,* 1*,* … *, 𝐾*}*.*

However, if *𝑘 > 𝐾* then *𝑠𝑘*−1 *<* min{*𝐿* Ω2 *, 𝑠*0} and *𝑠𝑘* ≤ *𝑠𝑘*−1 −

*𝑡*

Ψ(*𝑥* ) ≤ Ψ(*𝑥*0). Thus, the Lipschitz continuity (or curvature) can be as-

be an upper bou*|*nd on Ψ(*𝑥𝑡* ) for any point *𝑥𝑡* = (1 − *𝑡*)*𝑥𝑘*−1 + *𝑡𝑝𝑘* such that sumed only on the appropriate level set and there is no need to insist on

1 *𝑓* global Lipschitz smoothness on the entire set 𝖷. This insight enabled, for

*𝑠𝑘*−1 2, which by induction (see for example [Dunn (1979](#_bookmark180),

2 ( )

2*𝐿𝑓* Ω

Lemma 5.1)) implies that

*𝑠𝑘* ≤ *𝑠𝐾* ≤ 2*𝐿𝑓* Ω2 ≤ max{*𝐾,* 2}*𝐿𝑓* Ω ≤ 2 max{*𝑠 , 𝐿𝑓* Ω } ≥ + 1

*,*

2 0 2

*, 𝑘 𝐾*

example, proving the *𝑂*(1∕*𝑘*) convergence rate of CG with adaptive and

exact step-size rules when applied to self-concordant functions, which

[are not necessarily Lipschitz smooth on the predefined set 𝖷 (Carderera](#_bookmark140)

*𝑠𝐾 𝑘 𝐾*

1 + ( − )

2*𝐿𝑓* Ω2

2 + (*𝑘* − *𝐾*) *𝑘 𝑘*

[et al., 2021; Dvurechensky et al., 2020a; 2020; Zhao and Freund, 2020).](#_bookmark140)

However, this observation need not apply to the standard step size rule

AW-CG produces *𝑝𝑘* ∈ A and *𝑢𝑘* ∈ *𝑆 𝑘*, and the away step maximal step

[(5.5)](#_bookmark61), since the standard step-size choice does not guarantee that all the

iterates remain in the appropriate level set.

size is respecified as *𝛾*max

*𝜆𝑢𝑘*

1−*𝜆𝑢𝑘*

=

. This implies, that using the maximal

To conclude, we reiterate that the step-size choices analyzed here are the most common, but there may be many more choices of step- [size which provide similar guarantees. For example, Freund and Gri- gas (2016) suggests new step-size rules based on an alternative analysis](#_bookmark168) of the CG method that utilizes an updated duality gap. ([Nesterov, 2018a](#_bookmark227)) discusses recursive step-size rules, and in [Dvurechensky et al. (2020a)](#_bookmark153); [Odor et al. (2016)](#_bookmark234) new step-size rules are suggested based on additional assumptions on the problem structure.

* 1. *Variants of CG*

One of the main drawbacks of CG method is that, in general, it comes with worse complexity bounds than BPGM for strongly convex [functions. Indeed, it was shown as early as in 1968 by Canon and Cul-](#_bookmark139)

[lum (1968) (see also](#_bookmark139) [Lan](#_bookmark186) [(2013, 2020)) that the rate of *𝑂*( 1 ) is in fact](#_bookmark139)

ary of 𝖷. Thus, when *𝑓* is strongly convex, Jaggi and Lacoste-Julian away-step step-size will not necessarily result on a point on the bound-

[Lacoste-Julien and Jaggi (2015)](#_bookmark187) show a linear convergence of AW-CG with a rate that only depends on the geometry of set 𝖷, which is cap- tured by the *pyramidal width* parameter. The Pairwise variant of AW-CG, which is also presented and analyzed in [Lacoste-Julien and Jaggi (2015)](#_bookmark187),

takes *𝑑𝑘* = *𝑢𝑘* − *𝑝𝑘* and *𝛾*max = *𝜆𝑢𝑘* , and has similar analysis.

vergence results of AS-CG to functions of the form *𝑓* (*𝑥*) = *𝑔*(**𝐀***𝑥*) + *𝑏, 𝑥* In [Beck and Shtern (2017)](#_bookmark99), Beck and Shtern extend the linear con- where *𝑔* is a strongly convex function. The linear rate depends on a

*⟨ ⟩*

the geometry of 𝖷 as well as matrix **𝐀**. It is also worth mentioning, a parameter based on the Hoffman constant, which captures both on

stream of work which shows linear convergence of AS-CG where the strong convexity assumption is replaced by the assumption that suﬃ- cient second order optimality conditions, known as Robinson conditions

tight, even when the function *𝑓*

[*𝑘*](#_bookmark139)

is strongly convex. This slow conver-

([Robinson, 1982](#_bookmark244)), are satisfied (see for example [Damla et al. (2008)](#_bookmark160)).

ferent extreme points in 𝖷. In the smooth case, where *𝑟* = 0, and the gence is due to the well-documented zig-zagging effect between dif- objective function *𝑓* and the feasible set 𝖷 are both strongly convex, only a rate of *𝑂*(  1 ) can be shown ([Garber and Hazan, 2015](#_bookmark172)), whereas

*𝑘*2

1

*5.3.2. Fully-corrective CG*

The Fully-corrective variant of CG (FC-CG) also involves polyhedral

𝖷, and aims to reduce the number of calls to the linear oracle, by re- placing them with a more accurate minimization over a convex-hull of

([Nesterov, 2018a](#_bookmark227)) showed an accelerated *𝑂*( *𝑘*2 ) rate of convergence

for GCG with strongly convex *𝑟* (*𝜇 >* 0). Linear convergence of the CG

method can only be proved under additional assumptions regarding the

[problem structure or location of the optimal solution (see e.g. Beck and](#_bookmark102) [Teboulle](#_bookmark193) [(2004);](#_bookmark102) [Dunn](#_bookmark180) [(1979);](#_bookmark102) [Epelman](#_bookmark159) [and Freund (2000); Guélat and](#_bookmark102) [Marcotte (1986);](#_bookmark193) [Levitin](#_bookmark192) [and Polyak (1966)).](#_bookmark193)

Departing from these somewhat negative results, variants of the clas- sical CG were suggested in order to obtain the desired linear conver-

gence in the case of strongly convex function *𝑓* . We will discuss four of

these variants: Away-step CG, Fully-corrective CG, CG based on a local

linear optimization oracle (LLOO), and CG with sliding.

* + 1. *Away-step CG*

The away-step variation of CG (AW-CG), first suggested by

two calls of the LO at each iteration. The first call generates *𝑝𝑘* = [Wolfe (1970)](#_bookmark246), treats the case where 𝖷 is a polyhedron. It requires

G

𝖷(∇*𝑓* (*𝑥𝑘*−1)), defined in the original CG algorithm, while the second

G

call generates an additional vector *𝑢𝑘* = (−∇*𝑓* (*𝑥𝑘*−1)). The two vec-

some subset A*𝑘 ⊆* A. The heart of the method is a correction routine, which updates the correction atoms A*𝑘* and iterate *𝑥𝑘*, and satisfy the

following:

*𝑆 𝑘 ⊆* A*𝑘*

*𝑓* (*𝑥𝑘*) ≤ min *𝑓* ((1 − *𝑡*)*𝑥𝑘*−1 + *𝑡𝑝𝑘*)

*𝑡*∈[0*,*1]

≥

*𝜖* max ∇*𝑓* (*𝑥𝑘*)*, 𝑠* − *𝑥𝑘*

*⟨ ⟩*

*𝑠*∈*𝑆 𝑘*

where *𝑝𝑘* = G (∇*𝑓* (*𝑥𝑘*−1)), and *𝜖* is a given accuracy parameter. The FC-

𝖷

CG was known by various names depending on the updating scheme of

A

*𝑘* and *𝑥𝑘* ([Holloway, 1974; Von Hohenbalken, 1977](#_bookmark198)), and was uni-

[fied and analyzed to show linear convergence in Lacoste-Julien and](#_bookmark187)

[Jaggi (2015). The convergence analysis of FC-CG is similar to that of](#_bookmark187) AW-CG, and is based on the correction routine guaranteeing that the forward step is larger than the away-step computed in the previous it- eration.

In order to apply FC-CG one must choose a correction routine, and

𝖷

tors *𝑝𝑘* and *𝑢𝑘* define the *forward direction 𝑑𝑘*

*𝐹 𝑊*

= *𝑝𝑘* − *𝑥𝑘*−1 and the *away*

the linear convergence analysis does not take into account the computa-

*direction 𝑑𝑘* = *𝑥𝑘*−1 − *𝑢𝑘*, respectively. By construction, both of this di-

*𝐴*

A

iteration *𝑘* is obtained by rections are non-ascent directions. The effectively chosen direction at

} *⟨*

*⟩*

AS-CG on the subset *𝑘* = *𝑆 𝑘*−1 ∪ {*𝑝𝑘*} until the conditions are satisfied. tional cost of this routine. One choice of a correction routine is to apply This correction routine is wise only if eﬃcient linear oracles A*𝑘* can

*𝑑𝑘* = argmax

G

– ∇*𝑓* (*𝑥𝑘*−1)*, 𝑑 ,*

be constructed for all *𝑘* such that their low computational cost balances

*𝑑 𝑑𝑘*

∈{

*𝐹𝑊*

*,𝑑𝑘*

*𝐴*

the routine’s iteration complexity.

ensuring the chosen direction is a descent direction for non optimal *𝑥𝑘*−1,

with a corresponding updating step

*𝑥𝑘* = *𝑥𝑘*−1 + *𝛾𝑘 𝑑𝑘.*

Here, the choice of the step-size *𝛾𝑘* will also depend on the direc-

*5.3.3. Enhanced LO based CG*

A variant of CG which is based on an enhanced linear minimization oracle, was suggested by Garber and Hazan [Garber and Hazan (2016)](#_bookmark174).

G

GIn this variant, the linear oracle (*𝑐*) is replaced by a *local oracle*

𝖷*,𝜌*

(*𝑐, 𝑥, 𝛿*) with some constant *𝜌* ≥

𝖷

1, which takes an additional radius

[tion chosen. The first analysis of this algorithm by Guélat and Mar-](#_bookmark193)

input *𝛿* and returns a point *𝑝* ∈ 𝖷 satisfying

over *𝛾* ∈ [0*, 𝛾* ], where *𝛾* ∶= max{*𝑡* 0 ∶ *𝑥𝑘*−1 + *𝑡𝑑𝑘* ∈ 𝖷}. Under [cotte (1986) assumes that the step-size is chosen using exact line search](#_bookmark193)

*𝑘* max max

≥

*‖𝑝*

– *𝑥*

≤ *𝜌𝛿*

≤

*‖*

min

*𝑢, 𝑐 .*

convex *𝑓* . However, this rate estimate depends on the distance between this step-size choice, they prove linear convergence of CG for strongly

*⟨𝑝, 𝑐⟩*

*𝑢*∈𝖷∶*‖𝑢*−*𝑥‖*≤*𝛿 ⟨ ⟩*

the optimal solution and the boundary of set *𝑇 ⊂* 𝖷, which is the minimal Thus, the only deviation from the CG algorithm is that *𝑝𝑘* is obtained by

face of 𝖷 containing the optimal solution. This result was later extended applying G (∇*𝑓* (*𝑥𝑘*)*, 𝑥𝑘, 𝛿* ) for a suitably chosen sequence (*𝛿* ) . The

*𝑋,𝜌 𝑘 𝑘 𝑘*

in [Lacoste-Julien and Jaggi (2015)](#_bookmark187), with a slight variation on the original algorithm. In this variation, the set 𝖷 is represented as the convex hull of a finite set of atoms (not necessarily containing only its vertices), and a representation of the current iterate as a convex combination of these

A

atoms is maintained throughout the algorithm, *i.e.*, *𝑥𝑘* = *∑𝑆𝑘 𝜆𝑘𝑎* where

*𝑆 𝑘* = {*𝑎* ∈ A ∶ *𝜆𝑘 >* 0} is defined as the set of active atoms. Thus, the

*𝑎*

linear convergence for the case where the smooth part *𝑓* is strongly con-

vex, is obtained by a specific update of *𝛿𝑘* at each step of the algorithm. This update depends on the Lipschitz constant *𝐿𝑓* , the strong convex- ity constant of *𝑓* , and the parameter *𝜌*. Moreover, despite the fact that

LLOO-CG can theoretically be applied to any set 𝖷, constructing a gen-

eral LLOO is challenging. In [Garber and Hazan (2016)](#_bookmark174), the authors sug-

*𝑎*

**The procedure CndG**(*𝑔, 𝑢, 𝛽, 𝜂*)

geometric properties the polytope which may generally not tractably computed. Thus, while the strong convexity and geometric properties

eralize it for convex polytopes with *𝜌* =

*𝑛𝜌̃* where *𝜌̃* depends on some

gest an LLOO with *𝜌* = *√𝑛* when the set *√*𝖷 is the unit simplex, and gen-

**Input:** *𝑢*1 = *𝑢, 𝑡* = 1.

**Output:** point *𝑢*+ = CndG(*𝑔, 𝑢, 𝛽, 𝜂*)*.*

**General step:** Let *𝑣* = argmax *𝑔* + *𝛽*(*𝑢* − *𝑢*)*, 𝑢* − *𝑥*

≤

If *𝑉𝑔,𝑢,𝛽* (*𝑢𝑡* ) =

*𝑔* + *𝛽*(*𝑢𝑡* − *𝑢*)*, 𝑢𝑡* − *𝑣𝑡*

*𝜂*, set *𝑢*+

*𝑢𝑡* ;

else, set *𝑢𝑡*+1 = (1 − *𝛼𝑡* )*𝑢𝑡* + *𝛼𝑡 𝑣𝑡* , where

1*,*  *𝑡 𝑡 𝑡*

of the problem are only used for the analysis of the AW-CG and FC-CG,

the associated parameters are explicitly used in the execution of LLOO-

*⟨ 𝑡 𝑥*∈𝖷 *⟨ ⟩ 𝑡*

*𝑡* = *⟩*

geometric parameters renders the LLOO-CG less applicable in practice.

CG. The diﬃculty of accurately estimating the strong convexity and the

*{ ⟨𝛽*(*𝑢* − *𝑢* ) − *𝑔, 𝑣* − *𝑢 ⟩}*

– *𝑢*

2

*𝛼𝑡* = min

*.*

*𝛽‖𝑣𝑡 𝑡 ‖*

*5.3.4. CG with gradient sliding*

Set *𝑡* ← *𝑡* + 1. Repeat General step.

Each iteration of CG requires one call to the linear minimization or- acle and one gradient evaluation. Coupled with our knowledge about

the iteration complexity of CG, this fact implies that CG requires *𝑂*(1∕*𝜀*)

*where* Ω ≡ Ω 1

2 (𝖷)*. The number of calls of the linear minimization oracle*

gradient evaluations of the objective function. This is suboptimal, when

*√*

compared with the *𝑂*(1∕

*𝜀*) gradient evaluations for smooth convex op-

2 *‖*⋅*‖*

linear minimization oracle, the order estimate *𝑂*(1∕*𝜀*) for the number timization, as we will see in [Section 6](#_bookmark72). While it is known that within the

*. In particular, if the parameter sequences in S-CG are*

*𝜂𝑘*

*⌉*

*is bounded by*

6*𝛽𝑘* Ω2

*⌈*

*chosen as*

3*𝐿 𝐿* Ω2

*𝑓* 3

of calls of the LO is unimprovable, in this section we review a method based on the linear minimization oracle which can skip the computa- tion of gradients from time to time. This improves the complexity of

*𝛽𝑘* = *𝑘* + 1 *, 𝛾𝑘* =

*then*

*, 𝜂* = *𝑓 ,*

*𝑘* + 2 *𝑘 𝑘*(*𝑘* + 1)

LO-based methods and leads us to the *conditional gradient sliding* (S-CG) algorithm introduced by Lan and Zhou [Lan and Zhou (2016)](#_bookmark189). S-CG is a numerical optimization method which runs in epochs and overall con- tains some similarities with accelerated methods, to be thoroughly sur-

*𝑓* (*𝑦𝑘*) − *𝑓* (*𝑢*) ≤ 15*𝐿𝑓* Ω2

2(*𝑘* + 1)(*𝑘* + 2)

*.*

*As a consequence, the total number of calls of the function gradients and the*

*𝐿* Ω2 2

convex programming problem for which *𝑟* = 0. veyed in [Section 6](#_bookmark72). S-CG has been described in the context of the smooth

*LO oracle is bounded by 𝑂*

### Accelerated Methods

*𝑓*

*𝜀*

*(√ )*

*, and 𝑂*(*𝐿𝑓* Ω ∕*𝜀*)*, respectively.*

**Input:** A linear oracle a starting point *𝑥*0 ∈ 𝖷. **The conditional gradient sliding methods (S-CG)** (*𝛽𝑘* )*𝑘,* (*𝛾𝑘* )*𝑘* parameter sequence such that

𝖷

G

*𝛾*1 = 1*, 𝐿𝑓 𝛾𝑘* ≤ *𝛽𝑘 ,*

*𝛽𝑘 𝛾𝑘* ≥ *𝛽𝑘*−1 *𝛾𝑘*−1

Γ*𝑘* Γ*𝑘*−1

*,*

where

In previous sections we focused on simple first-order methods with sublinear convergence guarantees in the convex case, and linear con- vergence in the strongly convex case. Towards the end of the discussion in [Section 3](#_bookmark12), we pointed out the possibility to accelerate simple itera- tive schemes via suitably defined extrapolation steps. In this last section of the survey, we are focusing on such *accelerated methods*. The idea of acceleration dates back to 1980’s. The rationale for this research direc- tion is the desire to understand the computational boundaries of solv-

Γ = *{*1 if *𝑘* = 1*,*

(5*.*17)

ing optimization problems. Of particular interest has been the uncon-

*𝑘* Γ

*𝑘*

*𝑘*−1

(1 − *𝛾* ) if *𝑘* ≥ 2*.*

strained smooth, and strongly convex optimization problem. This would

be covered by our generic model [(P)](#_bookmark6) by setting *𝑟* = 0*,* 𝖷 = 𝖵 = ℝ*𝑛* and *𝑓*

**General step:** For *𝑘* = 1*,* 2*,* …

Compute

*𝑧𝑘* = (1 − *𝛾𝑘* )*𝑦𝑘*−1 + *𝛾𝑘 𝑥𝑘*−1*,*

*𝑥𝑘* = CndG(∇*𝑓* (*𝑧𝑘*)*, 𝑥𝑘*−1*, 𝛽𝑘 , 𝜂𝑘* )*,*

*𝑦𝑘* = (1 − *𝛾𝑘* )*𝑦𝑘*−1 + *𝛾𝑘 𝑥𝑘.*

tially updated sequences. The update of the sequence (*𝑥𝑘*) is stated in Similarly to accelerated methods, S-CG keeps track of three sequen-

terms of a procedure CndG, which describes an inner loop of condi- tional gradient steps. This subroutine aims at approximately solving for

the proximal step

strongly convex with parameter *𝜇𝑓 >* 0 and *𝐿𝑓* -smooth. The standard

approach to quantify the computational hardness of optimization prob-

point *𝑥*, the oracle reports the corresponding function value *𝑓* (*𝑥*), and in lems is through the *oracle model* of optimization; Upon receiving a query first-order models, the function gradient ∇*𝑓* (*𝑥*) as well. In their seminal

mization algorithm, there exists an *𝐿𝑓* -smooth (with some *𝐿𝑓 >* 0) and work, [Nemirovski and Yudin (1983)](#_bookmark213) showed that for any first-oder opti- convex function *𝑓* ∶ ℝ*𝑛* → ℝ such that the number of queries required to obtain an *𝜀*-optimal solution *𝑥*∗ which satisfies

*𝑓* (*𝑥*∗) *<* min *𝑓* (*𝑥*) + *𝜀,*

*𝑥*

is at least of the order of min{*𝑛, √𝐿𝑓* ∕*𝜇𝑓* } ln(1∕*𝜀*) if *𝜇𝑓 >* 0

and min{*𝑛* ln(1∕*𝜀*)*,*

*𝐿𝑓* ∕*𝜀*}, if *𝜇𝑓* = 0. This bound, obtained by

*√*

min *𝑓* (*𝑧𝑘*) + ∇*𝑓* (*𝑧𝑘*)*, 𝑥* − *𝑧𝑘*

+ *𝛽𝑘*

*𝑥* − *𝑥𝑘*−1 2

information-theoretical arguments, turned out to be tight. Nemirovski [Nemirovski (1982)](#_bookmark210) proposed a method achieving the optimal rate

*𝑥*∈*𝑋 ⟨*

*⟩* 2 *‖ ‖*

*𝑂*(1∕*𝑘*2) via a combination of standard gradient steps with the

up to an accuracy of *𝜂𝑘* . As will become clear later, the S-CG can thus

be thought of as an approximate version of the accelerated scheme pre-

sented in [Section 6.1](#_bookmark73).

The main performance guarantee of the algorithm S-CG is summa- rized in the following theorem:

**Theorem 5.4.** *For all 𝑘* ≥ 1 *and 𝑢* ∈ 𝖷*, we have*

classical center of gravity method, which required additional small- [dimensional minimization, see also a recent paper (Nesterov et al., 2020).](#_bookmark229) [Nesterov](#_bookmark216) [(1983) proposed an optimal method with explicit](#_bookmark229) step-sizes, which is nowadays known as Nesterov’s accelerated gradient method.

* 1. *Accelerated Gradient Method*

*𝑓* (*𝑦𝑘*) − *𝑓* (*𝑢*) ≤ *𝛽𝛾𝑘* Ω + Γ *∑ 𝜂𝑖 𝛾𝑖 ,* (5.18)

*𝑘*

2

In this section we consider one of the multiple variants of an Acceler-

*𝑘*

2

*𝑖*=1

Γ*𝑖*

ated Gradient Method. This variant is close to the accelerated proximal

method in [Tseng (2008)](#_bookmark236), which has been very influential to the field. An- other very influential version of the accelerated method, especially in applications, is the FISTA algorithm ([Beck and Teboulle, 2009a](#_bookmark105)), which is excellently described in [Beck (2017)](#_bookmark97). A recent review on accelerated

methods is [d’Aspremont et al., 2021](#_bookmark165). The version we present here is in-

where in the first inequality used the convexity of *𝑟*, and in the second inequality we used the convexity of *𝑓* . Now we plug [(6.2)](#_bookmark77) and [(6.3)](#_bookmark80) into

[(6.1)](#_bookmark76) to obtain

Ψ(*𝑥𝑘*+1) ≤ *𝐴𝑘* Ψ(*𝑥𝑘*) + *𝛼𝑘*+1 *(𝑓* (*𝑦𝑘*+1) + *⟨*∇*𝑓* (*𝑦𝑘*+1)*, 𝑢𝑘*+1 − *𝑦𝑘*+1*⟩* + *𝑟*(*𝑢𝑘*+1)*)*

*𝐴𝑘*+1

*𝐴𝑘*+1

[spired by the *Method of Similar Triangles* (Gasnikov and Nesterov, 2018; Nesterov, 2018b). For an illustration see](#_bookmark179) [Figure](#_bookmark86) [1](#_bookmark179)

Accelerated schemes generically produce three sequences (*𝑢𝑘, 𝑥𝑘, 𝑦𝑘*),

+ 1 *𝐷* (*𝑢𝑘*+1*, 𝑢𝑘*)

*𝐴𝑘*+1 *ℎ*

= *𝐴𝑘* Ψ(*𝑥𝑘*) + 1 *[𝛼 (𝑓* (*𝑦𝑘*+1) + *⟨*∇*𝑓* (*𝑦𝑘*+1)*, 𝑢𝑘*+1 − *𝑦𝑘*+1*⟩* + *𝑟*(*𝑢𝑘*+1)*)*

*]*

which are iteratively constructed via specifically designed inertial, re-

*𝐴*

*𝑘*+1

*𝐴*

*𝑘*+1

*𝑘*+1

ations are governed by control sequences (*𝐴𝑘* )*𝑘* and (*𝛼𝑘* )*𝑘*. laxation and gradient steps. The relative magnitude of inertia and relax-

The version we present below, is very flexible and allows one to obtain accelerated methods for many settings. As a particular example, below in [Section 6.4](#_bookmark88), we show how a slight modification of this method

+*𝐷ℎ* (*𝑢𝑘*+1*, 𝑢𝑘*) *.* (6.4)

Given the definition of *𝑢𝑘*+1 as a Prox-Mapping, we can apply [(3.12)](#_bookmark31) by substituting *𝑥*+ = *𝑢𝑘*+1, *𝑥* = *𝑢𝑘*, *𝛾* = *𝛼𝑘*+1 . In this way, we obtain, for any

*𝑢* ∈ 𝖷,

*⟨ ⟩*

allows one to obtain universal accelerated gradient method.

Ψ(*𝑥𝑘*+1) ≤ *𝐴𝑘* Ψ(*𝑥𝑘*) + 1 *(𝛼*

(*𝑓* (*𝑦𝑘*+1) + ∇*𝑓* (*𝑦𝑘*+1)*, 𝑢𝑘*+1 − *𝑦𝑘*+1 + *𝑟*(*𝑢𝑘*+1))

Our aim is to solve the composite model problem [(P)](#_bookmark6) within a general Bregman proximal setup, formulated in [Section 3.2](#_bookmark20). We are given a DGF



*ℎ* ∈ 1(𝖷). The scaling of the strong convexity parameter to the value

*𝐴𝑘*+1

+*𝐷ℎ* (*𝑢𝑘*+1*, 𝑢𝑘*)*)*

*⟨ ⟩*

*𝐴𝑘*+1

*𝑘*+1

1. actually is without loss of generality, modulo a constant rescaling of

(3≤*.*12) *𝐴𝑘* Ψ(*𝑥𝑘*) + 1 *(𝛼* (*𝑓* (*𝑦𝑘*+1) + ∇*𝑓* (*𝑦𝑘*+1)*, 𝑢* − *𝑦𝑘*+1 + *𝑟*(*𝑢*))

+*𝐷ℎ* (*𝑢, 𝑢𝑘*) − *𝐷ℎ* (*𝑢, 𝑢𝑘*+1)

*)*

the employed DGF

*𝐴𝑘*+1

*𝐴𝑘*+1

*𝑘*+1

≤ *𝐴𝑘* Ψ(*𝑥𝑘*) + *𝛼𝑘*+1 (*𝑓* (*𝑢*) + *𝑟*(*𝑢*)) + 1 *𝐷* (*𝑢, 𝑢𝑘*) − 1 *𝐷* (*𝑢, 𝑢𝑘*+1)

### The Accelerated Bregman Proximal Gradient Method

*𝐴𝑘*+1

*𝐴𝑘*+1

*𝐴𝑘*+1 *ℎ*

*𝐴𝑘*+1 *ℎ*

### (A-BPGM)

0 0 0

= *𝐴𝑘* Ψ(*𝑥𝑘*) + *𝛼𝑘*+1 Ψ(*𝑢*) + 1 *𝐷* (*𝑢, 𝑢𝑘*) − 1 *𝐷* (*𝑢, 𝑢𝑘*+1)*,* (6.5)

**Input:** pick *𝑥*

= *𝑢*

= *𝑦*

∈ dom(*𝑟*) ∩ 𝖷◦, set *𝐴*0 = 0

*𝐴𝑘*+1

*𝐴𝑘*+1

*𝐴𝑘*+1 *ℎ*

*𝐴𝑘*+1 *ℎ*

**General step:** For *𝑘* = 0*,* 1*,* … do:

2 where we also used convexity of *𝑓* . Multiplying both sides of the last

Find *𝛼𝑘*+1 from quadratic equation *𝐴𝑘* + *𝛼𝑘*+1 = *𝐿𝑓 𝛼𝑘*+1 . Set

inequality by *𝐴*

*𝑘*+1

, summing these inequalities from *𝑘* = 0 to *𝑘* = *𝑁* − 1,

*𝐴𝑘*+1 = *𝐴𝑘* + *𝛼𝑘*+1 .

and using that *𝐴*

– *𝐴*

= *∑𝑁* −1 *𝛼*

, we obtain

Set *𝑦𝑘*+1 = *𝛼𝑘*+1 *𝑢𝑘* + *𝐴𝑘 𝑥𝑘*.

*𝐴* Ψ(*𝑥𝑁* ) ≤ *𝐴* Ψ(*𝑥*0) + (*𝐴*

– *𝐴* )Ψ(*𝑢*) + *𝐷* (*𝑢, 𝑢*0) − *𝐷* (*𝑢, 𝑢𝑁* )*.*

*𝐴𝑘*+1

*𝑁*

0

*𝑘*=0

*𝑘*+1

*𝐴𝑘*+1

*𝑁* 0

*𝑁* 0 *ℎ ℎ*

(6.6)

Set

Since *𝐴* = 0, we can choose *𝑢* = *𝑥*∗ ∈ argmin{*𝐷* (*𝑢, 𝑢*0) *𝑢* ∈ 𝖷∗} *⊆* 𝖷∗ and

*𝑢𝑘*+1 = P *ℎ*

*𝛼𝑘*+1

, so that, for all *𝑁*

,

(*𝑢𝑘, 𝛼*

∇*𝑓* (*𝑦𝑘*+1))

( ∗ 0 ) ≥ 0

≥ 1 *ℎ* *|*

= argmin*𝑥*∈𝖷*{𝛼𝑘*+1 *(𝑓* (*𝑦𝑘*+1)+*⟨*∇*𝑓* (*𝑦𝑘*+1)*, 𝑥* − *𝑦𝑘*+1*⟩*+*𝑟*(*𝑥*)*)*+*𝐷ℎ* (*𝑥, 𝑢𝑘*)*}.*

*𝑟*

*𝑘*+1

*𝐷ℎ 𝑥 , 𝑢𝑁*

Ψ(*𝑥𝑁* ) − Ψmin(𝖷) ≤

*𝐷ℎ* (*𝑥*∗*, 𝑢*0)

*, 𝐷* (*𝑥*∗*, 𝑢𝑁* ) ≤ *𝐷* (*𝑥*∗*, 𝑢*0)*.* (6.7)

Set

=

+

*𝑥𝑘*+1

*𝛼𝑘*+1

*𝐴*

*𝑢𝑘*+1

*𝐴𝑘*

*𝐴*

*𝑥𝑘*.

*𝐴𝑁 ℎ ℎ*

*𝑘*+1 *𝑘*+1

We start the analysis A-BPGM applying the descent Lemma property

tween the iterates {*𝑢𝑁* }*𝑁*≥0 and the solution *𝑥*∗ is bounded by the Breg- So, we see from the second inequality that the Bregman distance be-

man distance between the starting point and the solution *𝑥*∗. Then,

from the inequality *𝐷* (*𝑥*∗*, 𝑢𝑁* ) ≥ 1 *𝑥*∗ − *𝑢𝑁* 2 it follows that *𝑥*∗ − *𝑢𝑁*

is bounded for any *𝑁* , which leads to the existence of a subsequence

converging to *𝑥*∗ by the continuity of Ψ. To obtain the convergence rate

[(3.15)](#_bookmark34) which holds for any two points due to *𝐿𝑓* -smoothness:

*ℎ* 2 *‖ ‖ ‖ ‖*

*⟨ ⟩* 2 *‖*

Ψ(*𝑥𝑘*+1) = *𝑓* (*𝑥𝑘*+1) + *𝑟*(*𝑥𝑘*+1) ≤ *𝑓* (*𝑦𝑘*+1) + ∇*𝑓* (*𝑦𝑘*+1)*, 𝑥𝑘*+1 − *𝑦𝑘*+1 + *𝐿𝑓*

*𝑥𝑘*+1 − *𝑦𝑘*+1 2 + *𝑟*(*𝑥𝑘*+1 )*.*

*‖* (6.1)

in terms of the objective residual it remains to estimate the sequence

Let us next consider the squared norm term. Using the definition of

*𝐴𝑁* from below.

2

*𝑥𝑘*+1

*, 𝑦*

*𝑘*+1

and the quadratic equation for *𝛼𝑘*+1 given in the listing of A-

We prove by induction that *𝐴* ≥ (*𝑘*+1) . For *𝑘* = 1 this inequality

*𝑓*

*𝑘* 4*𝐿*

BPGM, as well as the 1-strong convexity of the Bregman divergence, i.e. [(3.5)](#_bookmark19), we obtain

*𝐿𝑓* *‖𝑥𝑘*+1 − *𝑦𝑘*+1 *‖*2 = *𝐿𝑓 ‖ 𝛼𝑘*+1 *𝑢𝑘*+1 + *𝐴𝑘 𝑥𝑘* − *( 𝛼𝑘*+1 *𝑢𝑘* + *𝐴𝑘 𝑥𝑘 )‖*2

*𝑓*

– *𝑢*

*𝐷* (*𝑢*

= 1 +

1 + *𝐴𝑘* ≥ 1

+

*𝐴𝑘* ≥ 1 + *𝑘* + 1 = *𝑘* + 2 *.*

holds as equality since *𝐴*0 = 0, and, hence, *𝐴*1 = *𝛼*1 = 1 . Let us prove

the induction step. From the quadratic equation *𝐴𝑘* + *𝛼𝑘*+1 = *𝐿𝑓 𝛼*2 , we

*𝐿𝑓*

*𝑘*+1

have

*√*

2 2 *𝐴𝑘*+1

*𝐿𝑓 𝛼*2

*𝛼*

*𝐴𝑘*+1

*𝐴𝑘*+1

*𝐴𝑘*+1 *√*

2

*𝑘*+1

2*𝐴*

=

*𝑘*+1

*𝑘* 2

1

*𝑘*+1

*𝑘* 2 ≤ 1

*𝑘*+1 *, 𝑢𝑘* )*.*

(6.2)

*𝑘*+1

*‖𝑢*

*‖* 2*𝐴𝑘*+1 *‖*

*‖ 𝐴𝑘*+1 *ℎ*

*𝑘*+1

2*𝐿𝑓*

4*𝐿*2

*𝐿𝑓*

2*𝐿*

*𝐿𝑓*

2*𝐿𝑓*

2*𝐿𝑓*

2*𝐿𝑓*

Next, we consider the remaining terms in the r.h.s. of [(6.1)](#_bookmark76). Substituting

– *𝑢*

=

*𝑢*

*𝑥𝑘*+1 and using *𝐴𝑘*+1 = *𝐴𝑘* + *𝛼𝑘*+1 , we obtain

(6.8)

*𝑓* (*𝑦*

*𝛼𝑘*+1 + *𝐴𝑘*

*⟩*

*𝑘*+1

) + ∇*𝑓* (*𝑦*

*(*

*⟨*

*𝛼𝑘*+1 + *𝐴𝑘*

*𝑘*+1

)*, 𝑥*

*𝑘*+1

– *𝑦*

*)*

*𝑓* (*𝑦𝑘*+1) + ∇*𝑓* (*𝑦𝑘*+1)*, 𝛼𝑘*+1 *𝑢𝑘*+1 + *𝐴𝑘 𝑥𝑘* −

*𝑘*+1

+ *𝑟*(*𝑥*

*𝑘*+1 )

*𝐴𝑘*+1

*( )*

*𝑦𝑘*+1

= *𝐴𝑘*

+ *𝛼𝑘*+1 ≥

(*𝑘* + 1)2

4*𝐿𝑓*

+ *𝑘* + 2 = 2*𝐿𝑓*

*𝑘*2 + 2*𝑘* + 1 + 2*𝑘* + 4 ≥

4*𝐿𝑓*

(*𝑘* + 2)2

4*𝐿𝑓*

*.*

*𝐴𝑘*+1 *𝐴𝑘*+1

=

+ *𝑟( 𝛼𝑘*+1 *𝑢𝑘*+1 + *𝐴𝑘 𝑥𝑘 )*

*𝐴𝑘*+1

*𝐴𝑘*+1

*⟨ 𝐴𝑘*+1

*𝐴𝑘*+1

*𝐴𝑘*+1

*𝐴𝑘*+1 *⟩*

(6.9)

Thus, combining [(6.9)](#_bookmark78) with [(6.7)](#_bookmark75), we obtain that the A-BPGM has opti-

mal convergence rate:

≤ *𝐴𝑘 𝑓* (*𝑦𝑘*+1) + ∇*𝑓* (*𝑦𝑘*+1)*, 𝑥𝑘* − *𝑦𝑘*+1 + *𝑟*(*𝑥𝑘*)

*( ⟨ ⟩ )*

*𝐴𝑘*+1

+ *𝛼𝑘*+1 *(𝑓* (*𝑦𝑘*+1) + *⟨*∇*𝑓* (*𝑦𝑘*+1)*, 𝑢𝑘*+1 − *𝑦𝑘*+1*⟩* + *𝑟*(*𝑢𝑘*+1)*)*

Ψ(*𝑥𝑁* ) − Ψ

min

(𝖷) ≤

4*𝐿𝑓 𝐷ℎ* (*𝑥*∗*, 𝑢*0) (*𝑁* + 1)2

*.* (6.10)

*𝐴𝑘*+1

≤ *𝐴𝑘* *(𝑓* (*𝑥𝑘*) + *𝑟*(*𝑥𝑘*)*)* + *𝛼𝑘*+1 *(𝑓* (*𝑦𝑘*+1) + *⟨*∇*𝑓* (*𝑦𝑘*+1)*, 𝑢𝑘*+1 − *𝑦𝑘*+1*⟩* + *𝑟*(*𝑢𝑘*+1)*)*

*𝐴𝑘*+1

*𝐴𝑘*+1

*Closing Remarks* Mainly driven by applications in imaging and ma- chine learning, the research on acceleration techniques has been very

= *𝐴𝑘* Ψ(*𝑥𝑘*) + *𝛼𝑘*+1 *(𝑓* (*𝑦𝑘*+1) + *⟨*∇*𝑓* (*𝑦𝑘*+1)*, 𝑢𝑘*+1 − *𝑦𝑘*+1*⟩* + *𝑟*(*𝑢𝑘*+1)*),* (6.3)

productive in the last 20 years. During this time span it received exten-

*𝐴𝑘*+1

*𝐴𝑘*+1

sions to composite optimization ([Beck and Teboulle, 2009a; Nesterov,](#_bookmark105)

[2013](#_bookmark105)), general proximal setups ([Nesterov, 2005b; 2018b](#_bookmark218)), stochastic [optimization problems (Dvurechensky et al., 2018a; Dvurechensky and](#_bookmark143)

[Gasnikov, 2016; Ghadimi and Lan, 2012; 2013; Lan, 2012), optimiza-](#_bookmark143)

some fixed point and *𝑥* is such that *𝑥* − *𝑥*∗ 2 ≤ *𝑅*2, then

*( 𝑥* − *𝑥*∗ *)* Ω

*ℎ*

*𝑅*

≤

2

*,* (6.12)

*‖ ‖*

[tion with inexact oracle (Cohen et al., 2018; d’Aspremont, 2008; De- volder et al., 2014; Dvurechensky and Gasnikov, 2016; Stonyakin et al.,](#_bookmark152) [2020](#_bookmark106)[), variance reduction methods (Allen-Zhu, 2017; Frostig et al.,](#_bookmark152) [2015; Lan and Zhou, 2017; Lin et al., 2015; Zhang and Xiao, 2015),](#_bookmark106)

where Ω is some known number. For example, in the Euclidean setup Ω = 1, and other examples are given in [Juditsky and Nesterov (2014](#_bookmark206), Section 2.3), where typically Ω = *𝑂*(ln *𝑛*).

[random coordinate descent (Fercoq and Richtárik, 2015; Lee and Sid-](#_bookmark166)

[ford, 2013; Lin et al., 2014; Nesterov, 2012; Nesterov and Stich, 2017; Shalev-Shwartz and Zhang, 2014) and other randomized methods such](#_bookmark166) [as randomized derivative-free methods (Dvurechensky et al., 2017; Gor-](#_bookmark151)

### The Restarted Accelerated Bregman Proximal Gradient Method (R-A-BPGM)

≤

*( ) ‖*

**Input:** *𝑧*0 ∈ dom(*𝑟*) ∩ 𝖷◦ such that *𝑧*0 − *𝑥*∗ 2 2 Ω*, 𝐿 , 𝜇*.

[bunov et al., 2018; Nesterov and Spokoiny, 2017; Vorontsova et al.,](#_bookmark151) [2019b) and randomized directional search (Dvurechensky et al., 2017;](#_bookmark151)

[2021; Vorontsova et al., 2019a), second-order methods (Monteiro and](#_bookmark151)

.

**General step:** For *𝑝* = 0*,* 1*,* … do:

2 *𝑥*−*𝑧𝑝*

Set *ℎ𝑝* (*𝑥*) = *𝑅𝑝ℎ*

*𝑅*

, where *𝑅𝑝* ∶= *𝑅𝑝*−1 ∕2 = *𝑅*0 ⋅ 2

*‖ 𝑅*0 , *𝑓*

−*𝑝*

[Svaiter,](#_bookmark91) [2013; Nesterov, 2008), and even high-order methods (Baes,](#_bookmark206)

[2009; Gasnikov et al., 2019; Nesterov, 2019). Under additional assump-](#_bookmark91) tions on the Bregman divergence it is possible to propose accelerated

*𝑝*

*⌈ √* Ω*𝐿 ⌉*

Make *𝑁* =

2

*𝑓*

*𝜇*

– 1 steps of A-BPGM with starting

Bregman proximal gradient method in the setting of relative smooth- [ness (](#_bookmark152)[Hanzely et al., 2021](#_bookmark196)[) and relative strong convexity (Dvurechensky et al., 2021; Hendrikx et al., 2020) (see](#_bookmark152) [Section](#_bookmark35) [3.3.3 for the definition of](#_bookmark152) relative smoothness). Yet, the negative result of ([Dragomir et al., 2021](#_bookmark176)) suggest that, in general, the acceleration in the relative smoothness set- ting is not possible.

As it was mentioned above, accelerated gradient method in the form of A-BPGM can serve as a template for many acceleration techniques.

The examples of accelerated methods which have a close form include

point *𝑥*0 = *𝑧𝑝* and proximal setup given by DFG *ℎ𝑝* (*𝑥*)

Set *𝑧𝑝*+1 = *𝑥𝑁* .

We next use the above assumptions to show the accelerated loga-

imal steps to find a point *𝑥̂* such that *𝑓* (*𝑥̂*) − *𝑓* (*𝑥*∗) *𝜀* is proportional to rithmic complexity of R-A-BPGM, i.e. that the number of Bregman prox-

≤

*√*

*𝐿𝑓* ∕*𝜇* log2(1∕*𝜀*) instead of (*𝐿𝑓* ∕*𝜇*) log2(1∕*𝜀*) for the BPGM under the er-

ror bound condition. The idea of the proof is to show by induction that,

≥ ≤

for all *𝑝* 0, *𝑧𝑝* − *𝑥*∗ 2 *𝑅*2. For *𝑝* = 0 this holds by the assumption on

and *𝑅*0. So, next we prove an induction step from *𝑝* − 1 to *𝑝*. Using the

[primal-dual accelerated methods (Dvurechensky et al., 2018b; Lin et al.,](#_bookmark149) 0

[2019; Tseng, 2008), random coordinate descent and other random-](#_bookmark149)

*𝑧*

*‖ ‖ 𝑝*

[ized algorithms (Diakonikolas and Orecchia, 2018; Dvurechensky et al., 2017; Fercoq and Richtárik, 2015), methods for stochastic optimization](#_bookmark171)

definition of *ℎ𝑝*−1 , assumptions about *ℎ*, and the inductive assumption,

we have

([Dvurechensky et al., 2018a; Lan, 2012](#_bookmark143)), methods with inexact oracle [(](#_bookmark180)[Cohen et al., 2018](#_bookmark152)[) and inexact model of the objective (Gasnikov and Tyurin, 2019; Stonyakin et al., 2020). Moreover, only using this one-](#_bookmark180)

*𝑝*−1

*𝐷 𝑥*∗*, 𝑧*

*𝑝*−1

*ℎ* (

*𝑝*−1

) ≤ *ℎ*

*𝑝*−1

(*𝑥*∗) = *𝑅*2

*𝑧𝑝*−1 − *𝑥*∗ (6≤*.*12) Ω*𝑅𝑝*−1

*𝑅𝑝*−1 2

2

*ℎ*

*( )*

*.* (6.13)

projection version it was possible to obtain accelerated gradient meth- ods with inexact model of the objective ([Gasnikov and Tyurin, 2019](#_bookmark180)),

our choice of the number of steps *𝑁* , we obtain Thus, applying the error bound condition [(6.11)](#_bookmark81), the bound [(6.10)](#_bookmark79) and

accelerated decentralized distributed algorithms for stochastic convex *𝜇*

(6*.*11)

(6*.*10) *𝐿𝑓 𝐷ℎ*

(*𝑥*∗ *, 𝑧𝑝*−1 ) (6*.*13) *𝐿𝑓* Ω*𝑅*2

*𝑝* − ∗ 2

*‖*

≤ Ψ(*𝑧𝑝*) − Ψ

(𝖷) = Ψ(*𝑥𝑁* ) − Ψ

(𝖷) ≤

*𝑝*−1

≤ *𝑝*−1

[optimization (Dvinskikh et al., 2019; Gorbunov et al., 2019; Rogozin et al., 2021), and accelerated method for stochastic optimization with](#_bookmark139) heavy-tailed noise ([Gorbunov et al., 2020; 2021](#_bookmark185)). The key to the last two

1. *‖𝑧*

*𝑥*

2

≤ *𝜇𝑅*

*𝑝*−1 =

8

min

*𝜇𝑅*2

2 *.*

*𝑝*

min

(*𝑁* + 1)2

2(*𝑁* + 1)2

results is the proof that the sequence generated by the one-projection ac-

So, we obtain that

*𝑝* − ∗ ≤

= *𝑅*

⋅ 2−*𝑝* and Ψ(*𝑧𝑝*) − Ψ

(𝖷) ≤

celerated gradient method is bounded with large probability, which, to *𝜇* *𝑅*2 ⋅2−2*𝑝*

*‖𝑧*

*𝑥 ‖*

*𝑅𝑝* 0

min

our knowledge, is not possible to prove for other types of accelerated methods applied to stochastic optimization problems.

02 . To estimate the total number of basic steps of A-BPGM to

achieve Ψ(*𝑧𝑝*) − Ψmin(𝖷) *𝜀*, we need to multiply the suﬃcient num-

≤

ber of restarts *𝑝̂* = *⌈* 1 log2 *⌉* by the number of A-BPGM steps *𝑁* in

*𝜇𝑅*2

2

0

2*𝜀*

* 1. *Linear Convergence*

*𝜇𝑅 )*

2

Under additional assumptions, we can use the scheme A-BPGM to obtain a linear convergence rate, or, in other words, logarithmic in the

that Ψ(*𝑥*) satisfies a quadratic error bound condition for some *𝜇 >* 0: desired accuracy complexity bound. One such possible assumption is

each restart. This leads to the complexity estimate *𝑂(√* Ω*𝐿𝑓* log2 0 which is optimal ([Nemirovski and Yudin, 1983; Nesterov, 2018b](#_bookmark213)) for first-order methods applied to smooth strongly convex optimization problems.

*Closing Remarks* The restart technique which we used above was

*𝜇*

*𝜀*

Ψ(*𝑥*) − Ψ

*‖*

min

(𝖷) ≥ *𝜇 𝑥*

2

*‖*

– *𝑥*∗ 2*.* (6.11)

extended in the past 20 years to many settings including problems [with non-quadratic error bound condition (Juditsky and Nesterov,](#_bookmark206)

This is a weaker assumption than the assumption that Ψ(*𝑥*) is *𝜇*-strongly convex with *𝜇 >* 0. For a review of different additional conditions

which allow to obtain linear convergence rate we refer the reader to [Bolte et al. (2017)](#_bookmark119); [Necoara et al. (2019)](#_bookmark209). The linear convergence rate can be obtained under quadratic error bound condition by a widely used restart technique, which dates back to [Nemirovskii and Nesterov (1985)](#_bookmark214); [Nesterov (1983)](#_bookmark216).

sumptions. First, without loss of generality, we assume that 0 ∈ 𝖷, To apply the restart technique, we make several additional as- 0 = arg min*𝑥*∈𝖷 *ℎ*(*𝑥*) and *ℎ*(0) = 0. Second, we assume that we are given

≤

a starting point *𝑥*0 ∈ 𝖷 and a number *𝑅 >* 0 such that *𝑥*0 − *𝑥*∗ 2 *𝑅*2.

*ℎ* is bounded on the unit ball

([Juditsky and Nesterov, 2014](#_bookmark206)) in the following sense. Assume that *𝑥*∗ is

[2014; Roulet and d’Aspremont, 2017), stochastic optimization prob-](#_bookmark206) [lems (Bayandina et al., 2018; Dvurechensky and Gasnikov, 2016; Gas- nikov and Dvurechensky, 2016; Ghadimi and Lan, 2013; Juditsky and](#_bookmark100) [Nesterov,](#_bookmark145) [2014), methods with inexact oracle (Dvurechensky and Gas-](#_bookmark100) [nikov, 2016; Gasnikov and Dvurechensky, 2016), randomized methods](#_bookmark145) ([Allen-Zhu and Hazan, 2016; Fercoq and Qu, 2020](#_bookmark107)), conditional gra- dient ([Kerdreux et al., 2019; Lan, 2013](#_bookmark182)), variational inequalities and saddle-point problems ([Stonyakin et al., 2018; 2020](#_bookmark216)), methods for con- strained optimization problems ([Bayandina et al., 2018](#_bookmark100)). There are ver- sions of this technique even for discrete and submodular optimization

problems ([Pokutta, 2020](#_bookmark239)).

0

Finally, we make the assumption that

*‖ ‖* 0

A possible drawback of the restart scheme is that one has to

0 *‖ ‖*

know an estimate *𝑅*

for

*𝑧*0 − *𝑥*∗ . It is possible to avoid this by

directly incorporating the parameter *𝜇* into the steps of A-BPGM,

see e.g. [d’Aspremont et al. (2021)](#_bookmark165); [Devolder (2013)](#_bookmark169); [Lan (2020)](#_bookmark188);

[Nesterov (2018b)](#_bookmark228); [Stonyakin et al. (2020)](#_bookmark217). Yet, in this case, a stronger

Here the adjoint operator **𝐀**∗ is defined by equality **𝐀***𝑥, 𝑤* 𝖤 =

**𝐀**∗ *𝑤, 𝑥* 𝖵 and the norm of the operator 𝖵*,*𝖤 is defined by **𝐀** = max { **𝐀***𝑥, 𝑤* ∶ *𝑥* = 1*, 𝑤* = 1}. Since 𝖶 is bounded, Ψ (*𝑥*) is a

*⟨*

**𝐀**

𝖵*,*𝖤

*𝑥,𝑤 ⟨*

uniform approximation for the function Ψ, namely, for all *𝑥* ∈ 𝖷,

*⟩*

*⟩ ‖ ‖ ⟨ ‖ ‖ ⟩*

*‖ ‖*𝖵 *‖ ‖*𝖤 *𝜏*

assumption that Ψ(*𝑥*) is strongly convex or relatively strongly con-

vex ([Lu et al., 2018](#_bookmark201)) is used. The second drawback of the restart

Ψ (*𝑥*) ≤ Ψ(*𝑥*) ≤ Ψ (*𝑥*) + *𝜏*Θ (𝖶)*,* (6.17)

technique and direct incorporation of *𝜇* into the steps, is that both

require to know the value of the parameter *𝜇*. This is in contrast

to non-accelerated BPGM, which using the same step-size as in the

*𝜏*

where

Θ*ℎ*

*𝑤*

Then, the idea is to choose *𝜏* suﬃciently small and apply accelerated

gradient method to minimize Ψ (*𝑥*) on 𝖷 with a DGF *ℎ*

*𝜏*

(𝖶) ∶= max{*ℎ𝑤*

*ℎ𝑤*

(*𝑧*) *𝑧* ∈ 𝖶}, assumed to be a finite number.

*|*

∈  (𝖷) with

[*(*](#_bookmark217)2 [*)*](#_bookmark217)

non-strongly convex case automatically has linear convergence rate

[and complexity *𝑂*](#_bookmark217)

[*𝐿𝑓* log](#_bookmark217)

*𝜇𝑅*[0](#_bookmark217)

[, see e.g. Stonyakin et al. (2020,](#_bookmark217)

a nonrestrictive assumption that min*𝑢*∈𝖷 *ℎ𝑥* (*𝑢*) = 0. Doing this, and as-

[2](#_bookmark217)

*𝜏 𝑥* 1

[*𝜇*](#_bookmark217)

[*𝜀*](#_bookmark217)

[only rough estimates of the parameter *𝜇* are proposed in Fercoq and](#_bookmark164) 2019). Several recipes on how to restart accelerated methods with

0 ≤ Ψ(*𝑥𝑁* ) − Ψ

*ℎ𝑥*

(𝖷) ≤ Ψ (*𝑥𝑁* ) + *𝜏*Θ

suming that Θ

*𝑥*

(𝖶) − Ψ (*𝑥*∗) ≤ Ψ (*𝑥𝑁* ) + *𝜏*Θ

(𝖷) = max{*ℎ* (*𝑢*) *𝑢* ∈ 𝖷} *<* ∞, we can apply the result

[(6.10)](#_bookmark79) to Ψ*𝜏* (*𝑥*) and, using [(6.17)](#_bookmark82), to obtain

*|*

(𝖶) − Ψ (*𝑥*∗) ≤ *𝜏*Θ

(𝖶)

[Qu (2020) and a parameter-free accelerated method is proposed in](#_bookmark164) [Carderera et al. (2021)](#_bookmark142); [Nesterov (2013)](#_bookmark223).

min

*𝜏*

+ 4*𝐿𝜏* Θ*ℎ𝑥* (𝖷) (*𝑁* + 1)2

*ℎ𝑤*

4 **𝐀** 2 Θ

*𝜏*(*𝑁* + 1)2

*𝜏*

(𝖷)

*𝜏*

4*𝐿* Θ

(𝖷)

*ℎ𝑤*

*𝜏 𝜏*

*ℎ𝑤*

= *𝜏*Θ (𝖶) +

*6.3. Smooth minimization of non-smooth functions*

*ℎ𝑤*

*‖ ‖*𝖵*,*𝖤

*ℎ𝑥*

+ *𝑓*

*ℎ𝑥 .*

An important observation made during the last 20 years of devel-

(*𝑁* + 1)2

opment of first-order methods for convex programming is that there is

Choosing *𝜏* to minimize the r.h.s., i.e. *𝜏* = 2*‖***𝐀***‖*𝖵*,*𝖤 *√* Θ*ℎ𝑥* (𝖷) , we obtain

a large gap between the optimal convergence rate for black-box non-

*√*

*𝑁* +1

Θ*ℎ𝑤* (𝖶)

smooth optimization problems, i.e. *𝑂*(1∕

*𝑁* ) and the optimal conver-

≤

gence rate for black-box smooth optimization problems, i.e. *𝑂*(1∕

*𝑁*

2 ).

0

≤ *‖ ‖*

# *√*Θ (𝖷)Θ

(𝖶)

Certainly, there arises the need to understand how this significant gap can be reduced. An important step towards that direction is *Nesterov’s smoothing technique* ([Nesterov, 2005b](#_bookmark218)). To motivate this approach, let us make the following thought experiment. Assume that we mini-

4 **𝐀**

𝖵*,*𝖤

*ℎ𝑥*

*ℎ𝑤*

4*𝐿𝑓* Θ*ℎ* (𝖷)

Ψ(*𝑥𝑁* ) − Ψmin(𝖷)

mize a smooth function by *𝑁* steps of A-BPGM, i.e. solve problem

(6.18)

A more careful analysis in the proof of [Nesterov (2005b](#_bookmark218), Theorem 3), allows also to obtain an approximate solution to the conjugate problem

*𝑁* + 1

+ *𝑥 .*

(*𝑁* + 1)2

max{*𝜓* (*𝑤*) ∶= −*𝜅*(*𝑤*) + min ( **𝐀***𝑥, 𝑤* + *𝑓* (*𝑥*))}*.* (6.19)

[(P)](#_bookmark6) with *𝑟* = 0. Then at each iteration we observe first-order information

(*𝑓* (*𝑦𝑘*+1)*,* ∇*𝑓* (*𝑦𝑘*+1)) and can construct a *non-smooth* piecewise linear ap-

*𝑤*∈𝖶

*𝑥*∈𝖷 *⟨ ⟩*

proximation of *𝑓* as *𝑔*(*𝑥*) = max

{*𝑓* (*𝑦𝑘*+1) + ∇*𝑓* (*𝑦𝑘*+1)*, 𝑥* − *𝑦𝑘*+1 }. If

In each iteration of A-BPGM, the optimizer needs to calculate ∇Ψ*𝜏* (*𝑦𝑘*+1),

*𝑘*=1*,...,𝑁 ⟨*

*⟩* which requires to calculate *𝑤̂* (*𝑦𝑘*+1). This information is aggregated to

we now make *𝑁* steps of A-BPGM with the same starting point to min-

*∑*

imize *𝑔*(*𝑥*), and choose the appropriate subgradients of *𝑔*(⋅), the steps

*𝜏*

obtain the vector *𝑤̂𝑁* = *𝑁*−1 *𝛼𝑘*+1 *𝑤̂* (*𝑦𝑘*+1) and is used to obtain the

will be absolutely the same as when we minimized *𝑓* (*𝑥*), and we will be able to minimize a *non-smooth* function *𝑔* with much faster rate 1∕*𝑁* 2 than the lower bound 1∕ *𝑁* . This leads to a way of trying to find a

*√*

*𝑘*=0

following primal-dual result

0 ≤ Ψ( ) − Ψ ( ) ≤ Ψ( ) − (

*𝐴𝑘*+1 *𝜏*

) ≤ 4*‖***𝐀***‖*𝖵*,*𝖤 *√*Θ*ℎ𝑥* (𝖷)Θ*ℎ𝑤* (𝖶) + 4*𝐿𝑓* Θ*ℎ* (𝖷)

suﬃciently wide class of non-smooth functions which can be eﬃciently minimized by A-BPGM. To do so in a systematic way, we have to leave the pure black-box model of convex programming.

*𝑥𝑁*

min 𝖷

*𝑥𝑁*

*𝜓 𝑤̂𝑁*

*𝑁* + 1

*𝑥*

(*𝑁* + 1)2

*.*

(6.20)

Consider the model problem [(P)](#_bookmark6), with the added assumption that the non-smooth part admits a Fenchel representation of the form

*𝑟*(*𝑥*) = max{ **𝐀***𝑥, 𝑤* − *𝜅*(*𝑤*)}*.* (6.14)

*⟨ ⟩*

*𝑤*∈𝖶

Here, 𝖶 *⊆* 𝖤 is a compact convex subset of a finite-dimensional real vec- tor space 𝖤, and *𝜅* ∶ 𝖶 → ℝ is a continuous convex function on 𝖶. **𝐀** is a linear operator from 𝖵 to 𝖤∗. This additional structure of the problem

gives rise to a min-max formulation of [(P)](#_bookmark6), given by

min max{*𝑓* (*𝑥*) + **𝐀***𝑥, 𝑤* − *𝜅*(*𝑤*)}*.* (6.15)

*⟨ ⟩*

to obtain convergence rate *𝑂*(1∕*𝑁* ) for non-smooth optimization, which In both cases using the special structure of the problem it is possible is better than the lower bound *𝑂*(1∕ *𝑁* ) for general non-smooth opti-

mization problems.

*√*

We illustrate the smoothing technique by two examples of piecewise- linear minimization; see also [Figure 2](#_bookmark87)

fit of some signal *𝑏* ∈ 𝖤, given linear observations **𝐀***𝑥*. where **𝐀** ∶ 𝖵 → 𝖤 **Example 6.1** (Uniform fit)**.** Consider the problem of finding a uniform

is a bounded linear operator. This problem amount to minimize the non-

*𝑥*∈𝖷 *𝑤*∈𝖶

tion *𝑟* can be well approximated by a class of smooth convex functions, The main idea of Nesterov is based on the observation that the func- defined as follows. Let *ℎ𝑤* ∈ 1(𝖶) with a nonrestrictive assumption



**𝐀***𝑥* − *𝑏* . Of course, this problem can be equivalently

parameter vector *𝑥* is large, such a direct approach could turn out to be formulated as an LP, however in case where the dimensionality of the

smooth function *‖ ‖*∞

not very practical. Adopting the just introduced smoothing technology, the representation [(6.15)](#_bookmark84) can be obtained using the definition of the dual

resentation is obtained using the unit simplex 𝖶 = {*𝑤* ∈ ℝ2*𝑚*

1}, matrix **𝐀***̂* = [**𝐀**; −**𝐀**], and vector *𝑏̂* = [*𝑏*; −*𝑏*]. For the set 𝖶, a natural

that min*𝑧*∈𝖶 *ℎ𝑤* (*𝑧*) = 0. For given *𝜏 >* 0, define the function

norm ⋅

, i.e.

**𝐀***𝑥* − *𝑏*

= max

**𝐀***𝑥* − *𝑏, 𝑤* . Yet, a better rep-

*‖ ‖*1 *‖ ‖*∞

*𝑤*∶*‖𝑤‖*1 ≤1 *⟨*

*⟩ ∑*2

Ψ (*𝑥*) ∶= *𝑓* (*𝑥*) + max{ **𝐀***𝑥, 𝑤* − *𝜅*(*𝑤*) − *𝜏ℎ* (*𝑤*)}*.* (6.16)

*𝜏 ⟨ ⟩ 𝑤*

*𝑤*∈𝖶

Since *ℎ𝑤* is 1-strongly convex on the compact convex set 𝖶, the inner

solution. We denote by *𝑤̂𝜏* (*𝑥*) this optimal solution for a fixed *𝑥*. The maximization problem is strongly concave and hence admits a unique

entropy *ℎ𝑤* (*𝑤*) = ln 2*𝑚* +

*𝑚*

*𝑖*=1

=

Bregman setup is the no*∑*rm *‖𝑤‖*𝖤

*‖𝑤‖*1

+ *𝑖*=1 *𝑖*

and the Boltzmann-Shannon

*𝑚 𝑤* =

*𝑤𝑖* ln *𝑤𝑖* . This gives

*|*

main technical lemma needed for the analysis is the following.

Ψ (*𝑥*) = max{ **𝐀***̂ 𝑥* − *𝑏̂, 𝑤* − *𝜏ℎ* (*𝑤*)} = *𝜏* ln

*(*  1

*𝑚*

exp

*∑*

*(*  *⟨𝑎𝑖 , 𝑥⟩* − *𝑏𝑖 )*

**Proposition 6.1** ([Nesterov (2005b)](#_bookmark218))**.** *The function* Ψ*𝜏* (*𝑥*) *is well defined, convex and continuously diﬀerentiable at any 𝑥* ∈ 𝖷 *with* ∇Ψ*𝜏* (*𝑥*) = ∇*𝑓* (*𝑥*) +

**𝐀**∗ *𝑤̂𝜏* (*𝑥*)*. Moreover,* ∇Ψ*𝜏* (*𝑥*) *is Lipschitz continuous with constant 𝐿𝜏* = *𝐿𝑓* +

**𝐀** 2

*‖ ‖*𝖵*,*𝖤

*.*

*𝜏*

*𝜏 𝑤*∈𝖶 *𝑤*

+ exp *(*− *⟨𝑎𝑖 , 𝑥⟩* − *𝑏𝑖 )),*

*𝜏*

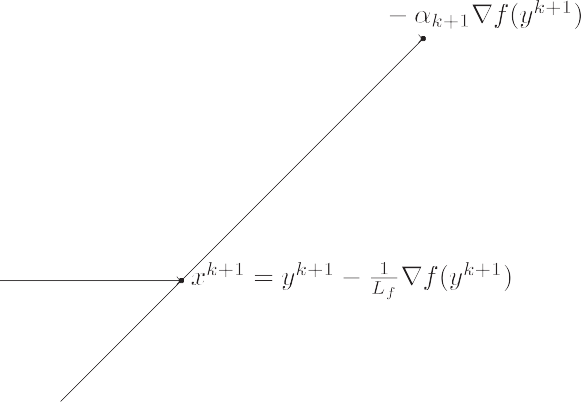
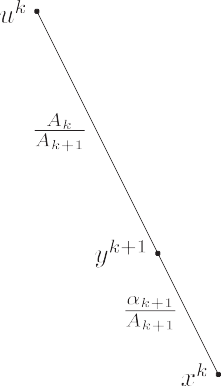
*⟨ ⟩*

which is recognized as a softmax function.

2*𝑚*

*𝑖*=1 *𝜏*

**Fig. 1.** Illustration of the three sequences of the A-BPGM in the uncon-



strained case 𝖷 = ℝ*𝑛*, *𝑟* = 0, *ℎ* = 1 *𝑥* 2. In this simple case it is easy to

∇*𝑓* (*𝑦𝑘*+1) *‖ ‖*

see that *𝑢𝑘*+1 = *𝑢𝑘* − *𝛼*

*𝑘*+1

2 2

, and the sequence

*𝑢𝑘* accumulates the

previous gradient, while helping to keep momentum. Also by the similar-

ity of the triangles, *𝑥𝑘*+1 = *𝑦𝑘*+1 − *𝛼* ∇*𝑓* (*𝑦𝑘*+1) ⋅ *𝛼𝑘*+1 = *𝑦𝑘*+1 − 1 ∇*𝑓* (*𝑦𝑘*+1),

*𝑘*+1

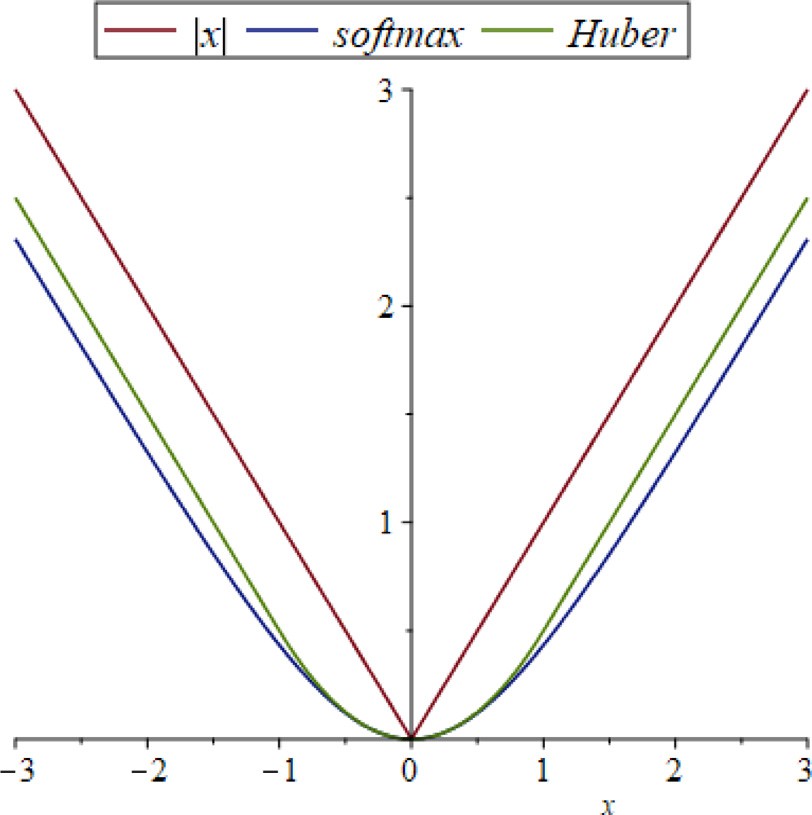
*𝐴𝑘*+1

*𝐿𝑓*

i.e. *𝑦𝑘* is the sequence obtained by gradient descent steps. Finally, the se-

quence *𝑥𝑘* is a convex combination of the momentum step and the gra-

dient step. The illustration is inspired by personal communication with [Yu. Nesterov on the Method of Similar Triangles (Gasnikov and Nesterov, 2018; Nesterov, 2018b).](#_bookmark179)



**Fig. 2.** Absolute value function *𝑥* , its softmax smoothing and Huber smooth- ing, both with *𝜏* = 1.

*| |*

**Example 6.2 (.** 𝓁1**-fit)** In compressed sensing [Candes and Tao (2007)](#_bookmark136);

minimize the 𝓁1 norm of the residual vector **𝐀***𝑥* − *𝑏* over a given closed [Candes et al. (2006)](#_bookmark137); [Donoho (2006)](#_bookmark175) one encounters the problem to

convex set 𝖷. While it is well-known that this problem can in principle again be reformulated as an LP, the typical high-dimensionality of such problems makes this direct approach often not practicable. Adopting the

≤

smoothing technology, it is natural to choose 𝖶 = {*𝑤* ∈ ℝ*𝑚 𝑤* 1}

the above rate estimate with the one reported in [Nemirovski (2004)](#_bookmark211), one observes that the bound in [Nemirovski (2004)](#_bookmark211) has a similar to [(6.20)](#_bookmark83) structure, yet with the second term being non-accelerated,

i.e. proportional to 1∕*𝑁* . This approach was generalized to obtain

an accelerated method for a special class of variational inequali-

ties in [Chen et al. (2017)](#_bookmark150), where an optimal iteration complexity

*𝑂*(*𝐿*∕ *𝜀*) to reach an *𝜀*-close solution is reported. In the original

*√*

paper ([Nesterov, 2005b](#_bookmark218)), the smoothing parameter is fixed and re-

quires to know the parameters of the problem in advance. This has been improved in [Nesterov (2005a)](#_bookmark217), where an adaptive version of the smoothing techniques is proposed. This framework was extended [in](#_bookmark232) [Alacaoglu et al. (2017)](#_bookmark103)[;](#_bookmark232) [Tran-Dinh et al. (2020)](#_bookmark231)[; Tran-Dinh and Cevher (2014);](#_bookmark232) [Tran-Dinh](#_bookmark233) [et al. (2018) for structured composite opti-](#_bookmark232) mization problems in the form [(2.7)](#_bookmark10) and a related primal-dual represen- tation [(2.8)](#_bookmark11). A related line of works studies minimization of strongly con- vex functions under linear constraints. Similarly to [(6.16)](#_bookmark85) the objective in the Lagrange dual problem has Lipschitz gradient, yet the challenge is that the feasible set in the dual problem is not bounded. Despite that [it is possible to obtain accelerated primal-dual methods (Anikin et al., 2017; Chernov et al., 2016; Dvurechensky et al., 2016; 2018b; Guminov et al., 2021; 2019; Ivanova et al., 2020; Kroshnin et al., 2019; Nesterov et al., 2020; Tran-Dinh and Cevher, 2014; Tran-Dinh et al., 2018). In](#_bookmark115) particular, this allows to obtain improved complexity bounds for differ- [ent types of optimal transport problems (Dvurechensky et al., 2018a; 2018b; Guminov et al., 2021; Kroshnin et al., 2019; Lin et al., 2020; 2019; 2019; Tupitsa et al., 2020; Uribe et al., 2018).](#_bookmark143)

* 1. *Universal Accelerated Method*

1 *∑𝑚 ‖ ‖* 2

*|‖ ‖*∞

As it was discussed in the previous subsection, there is a gap in the

and *ℎ𝑤* (*𝑤*) = 2

Ψ (*𝑥*) = max{ **𝐀***𝑥* − *𝑏, 𝑤* − *𝜏ℎ* (*𝑤*)} =

*𝑤*∈𝖶 *⟨*

*⟩*

*𝜏*

*𝑖*=1

*𝑎𝑖* 𝖤*,*∗ *𝑤𝑖* , which gives

*𝑚*

*∑*

two classes which allows to obtain uniformly optimal complexity bounds

*𝑤*

*𝑎*

*( |⟨𝑎 , 𝑥⟩* − *𝑏 |)*

convergence rate between the class of non-smooth convex optimization problems and the class of smooth convex optimization problems. In this subsection, we present a unifying framework ([Nesterov, 2015](#_bookmark225)) for these

where *𝜓* (*𝑡*) is the Huber function equal to *𝑡*2∕(2*𝜏*) for 0 ≤ *𝑡* ≤ *𝜏* and *𝑡* −

*𝜓*

*𝑖 𝑖*

*,*

*𝑖*

=1 *‖ 𝑖 ‖𝑥,*∗

*𝜏*

*‖𝑎𝑖 ‖*𝖤*,*∗

*𝜏*∕2 if *𝑡* ≥ *𝜏*.

*𝜏*

For the particular case of smoothing the absolute value function *|𝑥|*,

for both classes by a single method without the need to know whether the objective is smooth or non-smooth.

Consider the Problem [(P)](#_bookmark6) with *𝑓* which belongs to the class of func-

tions with Hölder-continuous subgradients, i.e. for some *𝐿 >* 0 and

≤ *𝐿*

ing and Huber smoothing, both with *𝜏* = 1. Potentially, other ways [Figure 2](#_bookmark87) gives the plot of the original function, its softmax smooth-

[of smoothing a non-smooth function can be applied, see Beck and Teboulle (2012) for a general framework.](#_bookmark108)

*Closing Remarks* Let us make several remarks on the related literature. A close approach is proposed in [Nemirovski (2004)](#_bookmark211), where the problem [(6.15)](#_bookmark84) is considered directly as a min-max saddle-point problem. These classes of equilibrium problems are typically solved via tools from mono- tone variational inequalities, whose performance is typically worse than the performance of optimization algorithms. In particular, contrasting

*𝜈*

*𝜈*

*𝜈* ∈ [0*,* 1] it holds that

*𝑓* (*𝑥*) − *𝑓* (*𝑦*)

*𝑥* − *𝑦*

for all *𝑥, 𝑦* ∈ dom *𝑓*

′

′

′ [*‖*](#_bookmark9)′ *‖ ‖ ‖*

∗ *𝜈*

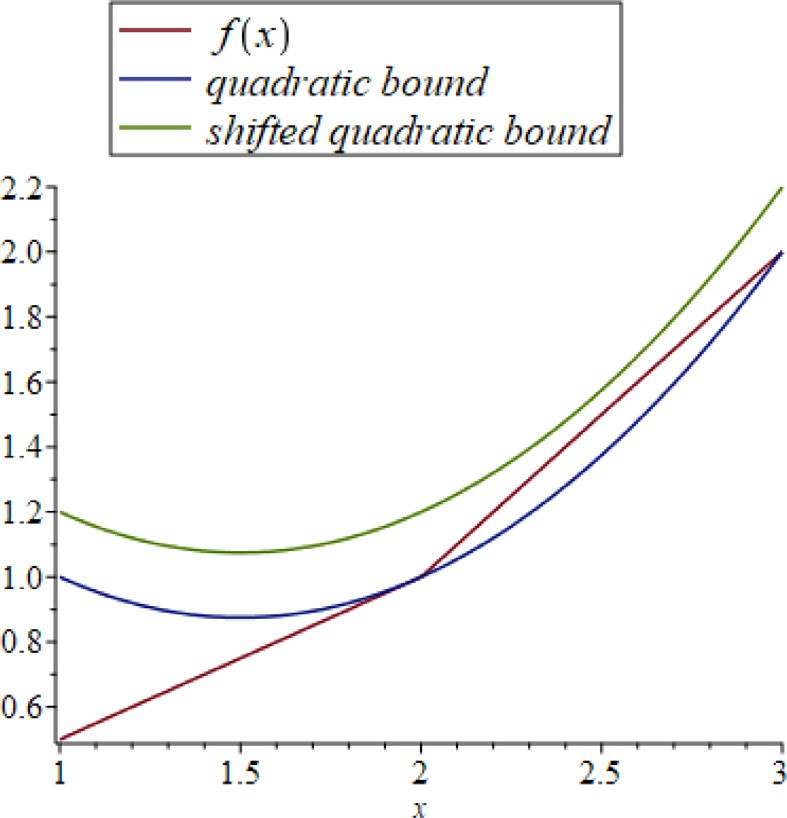
and all *𝑓* (*𝑥*) ∈ *𝜕𝑓* (*𝑥*) and *𝑓* (*𝑦*) ∈ *𝜕𝑓* (*𝑦*). If *𝜈* = 1, we recover the *𝐿𝑓* - smoothness condition [(2.1)](#_bookmark9). If *𝜈* = 0 we have that *𝑓* has bounded varia-

tion of the subgradient, which is essentially equivalent to the bounded [subgradient](#_bookmark170) [Assumption 8](#_bookmark43)[. The main observation (Devolder et al., 2014; Nesterov, 2015) is that this Hölder condition allows to prove an inex-](#_bookmark170) act version of the ”descent Lemma” inequality [(3.15)](#_bookmark34). More precisely

[Nesterov (2015](#_bookmark225), Lemma 2), for any *𝑥, 𝑦* ∈ dom *𝑓* and any *𝛿 >* 0,

*𝑓* (*𝑦*) ≤ *𝑓* (*𝑥*) + *𝑓* ′(*𝑥*)*, 𝑦* − *𝑥* + *𝐿𝜈 𝑦* − *𝑥* 1+*𝜈* ≤ *𝑓* (*𝑥*) + *𝑓* ′(*𝑥*)*, 𝑦* − *𝑥* + *𝐿 𝑦* − *𝑥* 2 + *𝛿,*

*⟨ ⟩* 1 + *𝜈 ‖ ‖ ⟨ ⟩* 2 *‖ ‖*

the same steps as the derivation of the convergence rate for A-BPGM in [Section 6.1](#_bookmark73). The first thing which is changed is [equation (6.1)](#_bookmark76), where now the inexact descent Lemma is used instead of the exact one. The

only difference is that *𝐿𝑓* is changed to its local approximation *𝐿𝑘*+1 and

add the error term *𝜀𝛼𝑘*+1 appears in the r.h.s. In [(6.2)](#_bookmark77) the new quadratic

2*𝐴𝑘*+1

equation with *𝐿𝑘*+1 is used and the inequality remains the same. This

term *𝜀𝛼𝑘*+1 in the r.h.s. Finally, this leads to the bound eventually leads to [(6.5)](#_bookmark74) with the only change being an additive error

2*𝐴𝑘*+1

Ψ(*𝑥𝑁* ) − Ψmin

(𝖷) ≤ *𝐷ℎ* (*𝑢*∗*, 𝑢*0) + *𝜀*

*𝐴𝑁* 2

*.*

After some algebraic manipulation, [Nesterov (2015](#_bookmark225), p.397) obtains an

inequality *𝐴* ≥ *𝑁* 1+*𝜈 𝜀* 1+*𝜈* . Substituting, we obtain

1+3*𝜈* 1−*𝜈*

*𝑁* 2+4*𝜈*  2

2 1+*𝜈 𝐿* 1+*𝜈*

*𝜈*

2+4*𝜈*  2

*𝑁* ≤ *ℎ 𝜈*

2 1+*𝜈 𝐷* (*𝑢*∗*, 𝑢*0)*𝐿* 1+*𝜈*

Ψ(*𝑥* ) − Ψmin(𝖷) +

*𝜀 .*

1+3*𝜈*

1−*𝜈* 2

**Fig. 3.** Non-smooth function *𝑓* (*𝑥*) = max{*𝑥* − 1*, 𝑥*∕2}, a quadratic function con-

*𝑁* 1+*𝜈 𝜀* 1+*𝜈*

Since the method does not require to know *𝜈* and *𝐿𝜈* , the iteration com- plexity to achieve accuracy *𝜀* is

*)*

structed using the first-order information at the point *𝑥* = 2, and a shifted

quadratic function constructed using the first-order information at the point

*𝑥* = 2. As one can see, adding a shift allows to obtain an upper quadratic bound

= inf *( 𝐿𝜈*

*⎜𝜈*∈[0*,*1] *𝜀*

*⎝ ⎠*

*𝑁 𝑂⎛*

2 1+3*𝜈 (*

1+*𝜈*

*𝐷ℎ* (*𝑢*∗ *, 𝑢*0 ) 1+3*𝜈*

*) ⎞*

*.*

*⎟*

for the objective, which is then minimized to obtain a new test point.

where

(6.21)

It is easy to see that the oracle complexity, i.e. the number of proximal

calls for each *𝑘* is 2(*𝑖𝑘* + 1). Further, *𝐿𝑘*+1 = 2*𝑖𝑘* −1 *𝐿𝑘* , which means that operations, is approximately the same. Indeed, the number of oracle the total number of the oracle calls up to iteration *𝑁* is *𝑁*−1 2(*𝑖* + 1) = *∑𝑁* −1 2(2 log2 *𝐿𝑘*+1 ) = 4*𝑁* + 2 log2 *𝐿𝑁* , i.e. is, up to a logarithmic

*𝑘*=0

*∑ 𝑘*

*𝑘*=0

*𝐿𝑘*

*𝐿*0

*𝐿* ≥

( ) ∶= 1 − *𝜈* 1 1+*𝜈*

1 + *𝜈 𝛿*

*𝐿 𝛿 ( )*

1−*𝜈*

2

1+*𝜈*

*𝜈*

*𝐿*

(6.22)

term, four times larger than *𝑁* . The obtained oracle complexity coin-

[cides up to a constant factor with the lower bound (Nemirovski and](#_bookmark213)

with the convention that 00 = 1. We illustrate this by [Figure 3](#_bookmark89) where we plot a quadratic bound in the r.h.s. of [(6.21)](#_bookmark90) with *𝛿* = 0 and a

shifted quadratic bound in the r.h.s. of [(6.21)](#_bookmark90) with some *𝛿 >* 0. The first

tions with Hölder-continuous gradients. In the particular case *𝜈* = 0, we [Yudin, 1983) for first-order methods applied to minimization of func-](#_bookmark213)

*𝐿*2 *𝐷ℎ* (*𝑢*∗ *,𝑢*0 )

*( )*

obtain the complexity *𝑂*

0

2

, which corresponds to the conver-

quadratic bound can not be an upper bound for *𝑓* (*𝑦*) for any *𝐿 >* 0, and

the positive shift allows to construct an upper bound. Thus, it is suf-

ficient to equip the A-BPGM with a backtracking line-search to obtain

a universal method. The resulting algorithm is listed below as Univer-

*𝜀*

gence rate 1∕ *𝑘*, which is typical for general non-smooth minimization. In the opposite case of smooth minimization corresponding to *𝜈* = 1, we

*√*

*(√ )*

*𝐿 𝐷* (*𝑢*∗ *,𝑢*0 )

obtain the complexity *𝑂*

1 *ℎ*

*𝜀*

, which corresponds to the op-

sal Accelerated Bregman Proximal Gradient Method (U-A-BPGM). A key step is the potentially non-monotone adjustment of the local Lipschitz

gradient estimate *𝐿𝑘* .

timal convergence rate 1∕*𝑘*2. The same idea can be used to obtain uni-

versal version of BPGM ([Nesterov, 2015](#_bookmark225)). One can also use the strong

convexity assumption to obtain faster convergence rate of U-A-BPGM

[either by restarts (Kamzolov et al., 2020; Roulet and d’Aspremont,](#_bookmark181)

### The Universal Accelerated Bregman Proximal Gradient Method (U-A-BPGM)

**Input:** Pick *𝑥*0 = *𝑢*0 = *𝑦*0 ∈ dom(*𝑟*) ∩ 𝖷◦, *𝜀 >* 0, 0 *< 𝐿*0 *< 𝐿*(*𝜀*∕2), set

*𝐴*0 = 0

**General step:** For *𝑘* = 0*,* 1*,* … do:

≥

Find the smallest integer *𝑖* 0 such that if one defines *𝛼*

[2017), or by incorporating the strong convexity parameter in the steps](#_bookmark181) ([Stonyakin et al., 2020](#_bookmark217)). The same backtracking line-search can be ap-

plied in a much simpler way if one knows that *𝑓* is *𝐿𝑓* -smooth with

[tice caused by a pessimistic estimate for *𝐿𝑓* (Dvinskikh et al., 2020;](#_bookmark141) some unknown Lipschitz constant or to achieve acceleration in prac-

[Dvurechensky et al., 2016; 2018b; Malitsky and Pock, 2018; Nesterov,](#_bookmark141)

*𝑘*

*𝑖* −1 2

*𝑘*+1

[2013; Tran-Dinh et al., 2018). The idea is to use the standard exact ”de-](#_bookmark141)

from the quadratic equation *𝐴𝑘* + *𝛼𝑘*+1 = 2 *𝑘*

*𝐴𝑘*+1 = *𝐴𝑘* + *𝛼𝑘*+1 ,

sets *𝑦𝑘*+1 = *𝛼𝑘*+1 *𝑢𝑘* + *𝐴𝑘 𝑥𝑘*,

*𝐿𝑘𝛼𝑘*+1 , sets

scent Lemma” inequality in each step of the accelerated method.

The idea of universal methods turned out to be very productive and

sets *𝑢𝑘*+1 =

*𝐴𝑘*+1

*𝐴𝑘*+1

several extensions have been proposed in the literature, including uni- versal primal-dual methods ([Baimurzina et al., 2019](#_bookmark92)), universal method

argmin*𝑥*∈𝖷 *𝛼𝑘*+1 *𝑓* (*𝑦𝑘*+1) + *𝑓* ′(*𝑦𝑘*+1)*, 𝑥* − *𝑦𝑘*+1 + *𝑟*(*𝑥*) + *𝐷ℎ* (*𝑥, 𝑢𝑘*) *,*

sets *𝑥𝑘*+1 = *𝛼𝑘*+1 *𝑢𝑘*+1 + *𝐴𝑘 𝑥𝑘* , then it holds that *𝑓* (*𝑥𝑘*+1) ≤

*{ ( ⟨ ⟩ ) }*

*𝐴𝑘*+1

*𝐴𝑘*+1

for convex and non-convex optimization ([Ghadimi et al., 2019](#_bookmark183)), and a universal primal-dual hybrid of accelerated gradient method with con-

jugate gradient method using additional one-dimensional minimization

*𝑓* (*𝑦𝑘*+1) + *𝑓* ′(*𝑦𝑘*+1)*, 𝑥𝑘*+1 − *𝑦𝑘*+1

+ 2*𝑖𝑘* −1 *𝐿𝑘*

*𝑥𝑘*+1 − *𝑦𝑘*+1 2 + *𝜀𝛼𝑘*+1 .

([Nesterov et al., 2020](#_bookmark229)). The above-described method is not the only way

2

*⟨ ⟩ ‖*

Set *𝐿𝑘*+1 = 2*𝑖𝑘* −1 *𝐿𝑘* and go to the next iterate *𝑘*.

We first observe that for suﬃciently large *𝑖* , 2*𝑖* −1 *𝐿*

*𝑘*

*𝑘*

*𝑘*

*‖* 2*𝐴𝑘*+1

≥ *𝐿( 𝜀𝛼𝑘*+1 *)*, see

2*𝐴*

*𝑘*+1

to obtain adaptive and universal methods for smooth and non-smooth optimization problems. An alternative way which uses the norm of the current (sub)gradient to define the step-size was initiated probably by [Polyak (1987)](#_bookmark241) and became very popular in stochastic optimization for

machine learning after the paper [Duchi et al., 2011](#_bookmark177). On this avenue

[Nesterov (2015](#_bookmark225), p.396). This means that the process of finding *𝑖𝑘* is finite

since the condition which is checked for each *𝑖𝑘* is essentially [(6.21)](#_bookmark90) with

*𝛿 𝜀𝛼𝑘*+1 . The proof of convergence of (U-A-BPGM) follows essentially

=

2*𝐴𝑘*+1

it was possible to obtain for *𝜈* ∈ {0*,* 1} universal accelerated optimiza-

tion method ([Levy et al., 2018](#_bookmark194)) and universal methods for variational

inequalities and saddle-point problems ([Bach and Levy, 2019](#_bookmark138)).

* 1. *Connection between Accelerated method and Conditional Gradient*

In this subsection we describe how a variant of conditional gradient method can be obtained as a particular case of A-BPGM with inexact

means that the change of [(3.12)](#_bookmark31) to [(3.14)](#_bookmark33) with Δ = Ω2 leads to an addi- tive term Ω2 in the r.h.s. of [(6.5)](#_bookmark74):

*𝐴𝑘*+1

Ψ(*𝑥𝑘*+1) ≤ *𝐴𝑘* Ψ(*𝑥𝑘*) + *𝛼𝑘*+1 Ψ(*𝑢*) + 1 *𝐷* (*𝑢, 𝑢𝑘*) − 1 *𝐷* (*𝑢, 𝑢𝑘*+1) + Ω2 *, 𝑢* ∈ 𝖷*.*

*ℎ*

*ℎ*

Bregman Proximal step. We assume that *𝑓* is *𝐿𝑓* -smooth and for sim-

*𝐴𝑘*+1

*𝐴𝑘*+1

*𝐴𝑘*+1

*𝐴𝑘*+1

*𝐴𝑘*+1

(6.24)

plicity choose *ℎ* to be the squared Euclidean norm *ℎ*(*𝑥*) = 1 *𝑥* 2. Since

*‖ ‖*

2

*∑*

we consider a conditional gradient method, it is natural to assume that

≡

Multiplying both sides of the last inequality by *𝐴𝑘*+1 , summing these in-

the set 𝖷 is bounded with max*𝑥,𝑢*∈𝖷

*𝑘*=0

Ψ(*𝑥𝑁* ) ≤ *𝐴* Ψ(*𝑥*0) + (*𝐴*

*𝐷* (*𝑥, 𝑢*) ≤ Ω2 1 2

2 (𝖷). We fol-

equalities from *𝑘* = 0 to *𝑘* = *𝑁* − 1, and using that *𝐴𝑁* − *𝐴*0 =

*𝑁* −1 *𝛼𝑘*+1 ,

2 *‖*⋅*‖*

*ℎ*

low the idea of [Ben-Tal and Nemirovski, 2020](#_bookmark111), where the main obser-

2

Ω

2 1

we obtain

vation of is that the Prox-Mapping in A-BPGM can be calculated inex- *𝑁*

*𝐴*

actly by applying the generalized linear oracle given in [Definition 5.2](#_bookmark65).

0 *𝑁* 0 *ℎ*

*ℎ*

(6.25)

*|*

The idea is very similar to the conditional gradient sliding described in [Section 5.3.4](#_bookmark71) with the difference that here we implement an approx- imate Bregman Proximal step using only *one* step of the generalized

≥

– *𝐴* )Ψ(*𝑢*) + *𝐷* (*𝑢, 𝑢*0) − *𝐷* (*𝑢, 𝑢𝑁* ) + *𝑁* Ω2 *.*

Since *𝐴*0 = 0, we can choose *𝑢* = *𝑥*∗ ∈ argmin{*𝐷ℎ* (*𝑢, 𝑢*0) *𝑢* ∈ 𝖷∗}, so that, for all *𝑁* 1,

conditional gradient method. The resulting algorithm is listed below

Ψ(*𝑥𝑁* ) − Ψ

(𝖷) ≤ *𝐷ℎ* (*𝑥*∗*, 𝑢*0) + *𝑁* Ω2 ≤ Ω2

+ *𝑁* Ω2

with the only difference with A-BPGM being the change of the Breg-

P

man Proximal step *𝑢𝑘*+1 = ( *𝑘* ∇*𝑓* (*𝑦𝑘*+1)) to the step *𝑢𝑘*+1 =

G

min

*𝐴𝑁 𝐴𝑁*

2*𝐴 𝐴 ,*

𝖷*,𝛼*

≥

*𝑘*+1

*𝛼𝑘*+1 *𝑟 𝑢 , 𝛼𝑘*+1

*𝑟*(*𝛼𝑘*+1 ∇*𝑓* (*𝑦𝑘*+1)) given by generalized linear oracle.

which, given the lower bound *𝐴𝑁*

(*𝑁* +1)2 leads to the final result for

4*𝐿𝑓*

*𝑁 𝑁*

the convergence rate of this inexact A-BPGM implemented via general- ized linear oracle:

### Conditional Gradient Method by A-BPGM with Approximate Bregman Proximal Step

**Input:** pick *𝑥*0 = *𝑢*0 = *𝑦*0 ∈ dom(*𝑟*) ∩ 𝖷◦, set *𝐴*0 = 0

Ψ(*𝑥𝑁* ) − Ψmin

(𝖷) ≤ 2*𝐿𝑓* Ω2

(*𝑁* + 1)2

+ 4*𝐿𝑓* Ω2

*𝑁* + 1

*.*

**General step:** For *𝑘* = 0*,* 1*,* … do:

Find *𝛼𝑘*+1 from quadratic equation *𝐴𝑘* + *𝛼𝑘*+1 = *𝐿𝑓 𝛼*2

. Set

Thus, we obtain a variant of conditional gradient method with the

same convergence rate 1∕*𝑁* as for the standard conditional gradient

*𝐴𝑘*+1

= *𝐴𝑘*

+ *𝛼*

*𝑘*+1 .

*𝑘*+1

method. Using the same approach, but with U-A-BPGM as the basis

Set *𝑦𝑘*+1 = *𝛼𝑘*+1 *𝑢𝑘* + *𝐴𝑘 𝑥𝑘*.

method, one can obtain a universal version of conditional gradient

*𝐴𝑘*+1

*𝐴𝑘*+1

method ([Stonyakin et al., 2020](#_bookmark217)) for minimizing objectives with Hölder-

Set (Approximate Bregman proximal step by generalized linear oracle)

*{ ( )}*

*𝑢*G*𝑘*+1 = argmin*𝑥*∈𝖷 *𝛼𝑘*+1 *𝑓* (*𝑦𝑘*+1) + *⟨*∇*𝑓* (*𝑦𝑘*+1)*, 𝑥* − *𝑦𝑘*+1*⟩* + *𝑟*(*𝑥*) =

*𝑟*(*𝛼𝑘*+1 ∇*𝑓* (*𝑦𝑘*+1)).

continuous gradient. The bounds in this case a similar to the ones ob- tained in a more direct universal method in [Nesterov (2018a)](#_bookmark227). Similar

bounds were also recently obtained in [Zhao and Freund (2020)](#_bookmark249).

Set *𝑥𝑘*+1 = *𝛼𝑘*+1 *𝑢𝑘*+1 + *𝐴𝑘 𝑥𝑘* .

𝖷*,𝛼*

*𝑘*+1

*𝐴𝑘*+1

*𝐴𝑘*+1

### Conclusion

A-BPGM is in one simple change of the step for *𝑢𝑘*+1, to obtain the con- Since the difference between such conditional gradient method and

vergence rate of the former, it is suﬃcient to track, what changes such approximate Bregman Proximal step entails in the convergence rate proof for A-BPGM. In other words, we need to understand what hap- pens with the proof for A-BPGM if the Bregman Proximal step is made inexactly by applying the generalized linear oracle. The first important difference is that we need an inexact version of inequality [(3.12)](#_bookmark31), which was used in the convergence proof of A-BPGM and which the result of the exact Bregman Proximal step. To obtain its inexact version, let us denote

*( ⟨ ⟩ )*

*𝜑*(*𝑥*) = *𝛼𝑘*+1 *𝑓* (*𝑦𝑘*+1) + ∇*𝑓* (*𝑦𝑘*+1)*, 𝑥* − *𝑦𝑘*+1 + *𝑟*(*𝑥*) *.*

set 𝖷 to obtain *𝑢𝑘*+1. Thus, by the optimality condition, we have that Then generalized linear oracle actually minimizes this function on the there exists *𝜉* ∈ *𝜕𝜑*(*𝑢𝑘*+1) such that *𝜉, 𝑢𝑘*+1 − *𝑥* 0 for all *𝑥* ∈ 𝖷. Now

*⟨ ⟩*

≤

we remind that the Bregman Proximal step in A-BPGM minimizes

*𝜑*(*𝑥*) + *𝐷ℎ* (*𝑥, 𝑢𝑘*). These observations allow to estimate the inexactness

of the Bregman Proximal step implemented via generalized linear ora-

We close this survey, with a very important fact which Nesterov writes in the introduction of his important textbook ([Nesterov, 2018b](#_bookmark228)): *in general, optimization problems are unsolvable.* Convex programming stands out from this general fact, since it describes a significantly large class of model problems, with important practical applications, for which general solution techniques have been developed within the mathematical framework of interior-point techniques. However, mod- ern optimization problems are large-scale in nature, which renders these polynomial time methods impractical. First-order methods have become the gold standard in balancing cheap iterations with low solution accu- racy, and many theoretical and practical advances having been made in the last 20 years.

Despite the fact that convex optimization is approaching the state

of being a primitive similar to linear algebra techniques, we foresee that the development of first-order methods has not come to a halt yet. In connection with stochastic inputs, the combination of acceler- ation techniques with other performance boosting tricks, like variance reduction, incremental techniques, as well as distributed optimization, still promises to produce some new innovations. On the other hand,

cle. Indeed, for *𝑢𝑘*+1 = G

𝖷*,𝛼𝑘*+1 *𝑟*

(*𝛼𝑘*+1

∇*𝑓* (*𝑦𝑘*+1))

there is also still much room for improvement of algorithms for opti- mization problems which do not admit a prox-friendly geometry. Dis-

+ ∇*ℎ*(*𝑢𝑘*+1) − ∇*ℎ*(*𝑢𝑘*)*, 𝑢𝑘*+1 − *𝑥* ≤ ∇ (*𝑢𝑘*+1) − ∇*ℎ*(*𝑢𝑘*)*, 𝑢𝑘*+1 − *𝑥*

*⟨ ℎ*

*⟩*

*⟩*

*⟨𝜉*

= −*𝐷* (*𝑥, 𝑢𝑘*) + *𝐷* (*𝑥, 𝑢𝑘*+1) + *𝐷* (*𝑢𝑘*+1*, 𝑢𝑘*) ≤ Ω2 *,* (6.23)

tributed optimization, in particular in the context of federated learning

*ℎ ℎ ℎ*

max *𝐷* (*𝑥, 𝑢*) Ω2 . This inequality provides inexact version of the where we used three-point identity in [Lemma 3.3](#_bookmark30) and that optimality condition [(3.10)](#_bookmark28) in the problem min*𝑥*∈𝖷{*𝜑*(*𝑥*) + *𝐷ℎ* (*𝑥, 𝑢𝑘*)}, i.e. [(3.13)](#_bookmark32) with Δ = Ω2 . This in order leads to [(3.14)](#_bookmark33) with Δ = Ω2 , which is

*𝑥,𝑢*∈𝖷 *ℎ* 2

≤

the desired inexact version of [(3.12)](#_bookmark31).

Let us now see, how this affects the convergence rate proof of A- BPGM. Inequality [(3.12)](#_bookmark31) was used in the analysis only in [(6.5)](#_bookmark74). This

is now a very active area of research, see [Kairouz et al. (2021)](#_bookmark208) for a

recent review of federated learning and ([Gorbunov et al., d](#_bookmark190)) for a re- cent review of distributed optimization. Another important focus in the research in optimization methods is now on numerical methods for non- convex optimization motivated by training of deep neural networks, see [Danilova et al. (2020)](#_bookmark162); [Sun (2019)](#_bookmark226) for a recent review. A number of open questions remain in the theory of first-order methods for variational in- equalities and saddle-point problems, mainly in the case of variational inequalities with non-monotone operators. In particular, recently the

authors of ([Cohen et al., 2021](#_bookmark154)) observed a connection between extra- gradient methods for monotone variational inequalities and accelerated first-order methods. Thus, as we emphasize in this survey, new con- nections, that are still continuously being discovered between different methods and different formulations, can lead to new understanding and developments in this lively field of first-order methods.

### Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

### Acknowledgment

The authors are grateful to Yurii Nesterov and Alexander Gas- nikov for fruitful discussions. Mathias Staudigl thanks the COST Action CA16228 (European Network for Game Theory), the FMJH Program PGMO and from the support of EDF (Project ”Privacy preserving algo- rithms for distributed control of energy markets”) for its support. The research of Pavel Dvurechensky is supported by the Ministry of Science and Higher Education of the Russian Federation (Goszadaniye) No.075- 00337-20-03, project No. 0714-2020-0005.

### References

[Alacaoglu, A., Tran Dinh, Q., Fercoq, O., Cevher, V., 2017. Smooth primal-dual co- ordinate descent algorithms for nonsmooth convex optimization. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (Eds.), Advances in Neural Information Processing Systems, Vol. 30. Curran Asso- ciates, Inc., pp. 5852–5861.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0001)

[Allen-Zhu, Z., 2017. Katyusha: The first direct acceleration of stochastic gradient methods. In: Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017. ACM, New York, NY, USA, pp. 1200–1205.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0002)

[Allen-Zhu, Z., Hazan, E., 2016. Optimal black-box reductions between optimization ob- jectives. In: Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., Garnett, R. (Eds.), Ad- vances in Neural Information Processing Systems, Vol. 29. Curran Associates, Inc.,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0003)

[pp. 1614–1622.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0003)

[Alvarez, F., Bolte, J., Brahic, O., 2004. Hessian Riemannian gradient flows in convex pro- gramming. SIAM Journal on Control and Optimization 43 (2), 477–501.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0004)

[Andersen, E.D., Andersen, K.D., 2000. The Mosek Interior Point Optimizer for Linear Pro- gramming: An Implementation of the Homogeneous Algorithm. Springer US, Boston, MA, pp. 197–232.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0005)

[Anikin, A.S., Gasnikov, A.V., Dvurechensky, P.E., Tyurin, A.I., Chernov, A.V., 2017. Dual approaches to the minimization of strongly convex functionals with a simple structure under aﬃne constraints. Computational Mathematics and Mathematical Physics 57 (8), 1262–1276.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0006)

[Arrow, K., Hurwicz, L., Uzawa, H., 1958. Studies in linear and non-linear programming. In: Chenery, H., Johnson, S., Karlin, S., Marschak, T., Solow, R. (Eds.), Stanford Math- ematical Studies in the Social Sciences, Vol. II. Stanford University Press, Stanford.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0007)

[Attouch, H., Bolte, J., Redont, P., Teboulle, M., 2004. Singular Riemannian barrier meth- ods and gradient-projection dynamical systems for constrained optimization. Opti- mization 53 (5-6), 435–454.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0008)

[Attouch, H., Cabot, A., Frankel, P., Peypouquet, J., 2011. Alternating proximal algorithms for linearly constrained variational inequalities: application to domain decomposition for pde’s. Nonlinear Analysis: Theory, Methods & Applications 74 (18), 7455–7473.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0009)

Attouch, H., Chbani, Z., Fadili, J., Riahi, H., 2020. First-order optimization algo- rithms via inertial systems with hessian driven damping. Mathematical Programming doi:[10.1007/s10107-020-01591-1](https://doi.org/10.1007/s10107-020-01591-1).

[Attouch, H., Chbani, Z., Peypouquet, J., Redont, P., 2018. Fast convergence of inertial dy- namics and algorithms with asymptotic vanishing viscosity. Mathematical Program- ming 168 (1-2), 123–175.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0011)

[Attouch, H., Redont, P., Soubeyran, A., 2007. A new class of alternating proximal min- imization algorithms with costs-to-move. SIAM Journal on Optimization 18 (3), 1061–1081.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0012)

[Attouch, H., Teboulle, M., 2004. Regularized lotka-volterra dynamical system as contin- uous proximal-like method in optimization. Journal of Optimization Theory and Ap- plications 121 (3), 541–570.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0013)

[Auslender, A., Teboulle, M., 2006. Asymptotic cones and functions in optimization and variational inequalities. Springer Science & Business Media.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0014)

[Auslender, A., Teboulle, M., 2006. Interior gradient and proximal methods for convex and conic optimization. SIAM Journal on Optimization 16 (3), 697–725.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0015)

[Auslender, A., Teboulle, M., 2009. Projected subgradient methods with non-euclidean dis- tances for non-differentiable convex minimization and variational inequalities. Math- ematical Programming 120 (1), 27–48.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0016)

[Bach, F., 2015. Duality between subgradient and conditional gradient methods. SIAM Journal on Optimization 25 (1), 115–129.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0017)

[Bach,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0018) [F., Levy, K.Y., 2019. A universal algorithm for variational inequalities adaptive to smoothness and noise. In: Beygelzimer, A., Hsu, D. (Eds.), Proceedings of the Thir- ty-Second Conference on Learning Theory, Vol. 99 of Proceedings of Machine Learn- ing Research, PMLR, Phoenix, USA, pp. 164–194.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0018)

[Baes,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0019) [M., 2009. Estimate sequence methods: extensions and approximations, Institute for Operations Research. ETH, Zürich, Switzerland.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0019)

[Bah,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0020) [B., Rauhut, H., Terstiege, U., Westdickenberg, M., 2019. Learning deep linear neural networks: Riemannian gradient flows and convergence to global minimizers. Infor- mation and Inference: A Journal of the IMAIaaa039.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0020)

[Baimurzina, D.R.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0021), [Gasnikov, A.V.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0021), [Gasnikova, E.V.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0021), [Dvurechensky, P.E.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0021), [Ershov, E.I.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0021), [Kubentaeva, M.B., Lagunovskaya, A.A., 2019. Universal method of searching for equi- libria and stochastic equilibria in transportation networks. Computational Mathemat- ics and Mathematical Physics 59 (1), 19–33.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0021)

[Banert,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0022) [S., BoŢ, R.I., Csetnek, E.R., 2021. Fixing and extending some recent results on the admm algorithm. Numerical Algorithms 86 (3), 1303–1325.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0022)

[Bauschke,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0023) [H., Borwein, J., Combettes, P., 2003. Bregman monotone optimization algo- rithms. SIAM Journal on Control and Optimization 42 (2), 596–636.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0023)

[Bauschke,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0024) [H.H., Bolte, J., Teboulle, M., 2016. A descent lemma beyond lipschitz gradient continuity: First-order methods revisited and applications. Mathematics of Operations Research 42 (2), 330–348.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0024)

[Bauschke,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0025) [H.H., Combettes, P.L., 2016. Convex Analysis and Monotone Operator Theory in Hilbert Spaces. Springer - CMS Books in Mathematics.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0025)

[Bayandina,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0026) [A., Dvurechensky, P., Gasnikov, A., Stonyakin, F., Titov, A., 2018. Mirror de- scent and convex optimization problems with non-smooth inequality constraints. In: Giselsson, P., Rantzer, A. (Eds.), Large-Scale and Distributed Optimization. Springer International Publishing, pp. 181–215.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0026) [Ch. 8](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0026)

[Beck,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0027) [A., 2017. First-Order Methods in Optimization. Society for Industrial and Applied Mathematics.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0027)

[Beck,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0028) [A., Guttmann-Beck, N., 2019. FOM –a matlab toolbox of first-order methods for solving convex optimization problems. Optimization Methods and Software 34 (1), 172–193.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0028)

[Beck,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0029) [A., Shtern, S., 2017. Linearly convergent away-step conditional gradient for non-strongly convex functions. Mathematical Programming 164 (1-2), 1–27.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0029)

[Beck,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0030) [A., Teboulle, M., 2003. Mirror descent and nonlinear projected subgradient methods for convex optimization. Operations Research Letters 31 (3), 167–175.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0030)

[Beck,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0031) [A., Teboulle, M., 2004. A conditional gradient method with linear rate of conver- gence for solving convex linear systems. Mathematical Methods of Operations Re- search 59 (2), 235–247.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0031)

[Beck,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0032) [A., Teboulle, M., 2009. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM Journal on Imaging Sciences 2 (1), 183–202.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0032)

[Beck,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0033) [A., Teboulle, M., 2009. Gradient-based algorithms with applications to signal re- covery. In: Palomar, D.P., Eldar, Y.C. (Eds.), Convex optimization in signal processing and communications. Cambridge University Press, pp. 42–88.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0033)

[Beck,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0034) [A., Teboulle, M., 2012. Smoothing and first order methods: A unified framework.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0034)

[SIAM Journal on Optimization 22 (2), 557–580.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0034)

[Ben-Tal,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0035) [A., Margalit, T., Nemirovski, A., 2001. The ordered subsets mirror descent opti- mization method with applications to tomography. SIAM Journal on Optimization 12 (1), 79–108.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0035)

Ben-Tal, A., Nemirovski, A., 2020. Lectures on Modern Convex Opti- mization (Lecture Notes). Personal web-page of A. Nemirovski. URL <https://www2.isye.gatech.edu/~nemirovs/LMCOLN2020WithSol.pdf>

[Benaïm,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0037) [M., 1998. Recursive algorithms, urn processes, and the chaining number of chain recurrent sets. Ergodic Theory and Dynamical Systems 18, 53–87.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0037)

[Benveniste,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0038) [A., Métivier, M., Priouret, P., 1990. Adaptive Algorithms and Stochastic Ap- proximations. Springer, Berlin.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0038)

[Bian,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0039) [W., Chen, X., 2015. Linearly constrained non-lipschitz optimization for image restoration. SIAM Journal on Imaging Sciences 8 (4), 2294–2322.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0039)

[Bian,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0040) [W., Chen, X., Ye, Y., 2015. Complexity analysis of interior point algorithms for non-lipschitz and nonconvex minimization. Mathematical Programming 149 (1), 301–327.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0040)

[Bolte,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0041) [J., Nguyen, T.P., Peypouquet, J., Suter, B.W., 2017. From error bounds to the com- plexity of first-order descent methods for convex functions. Mathematical Program- ming 165 (2), 471–507.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0041)

[Bolte,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0042) [J., Teboulle, M., 2003. Barrier operators and associated gradient-like dynamical systems for constrained minimization problems. SIAM Journal on Control and Opti- mization 42 (4), 1266–1292.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0042)

[Bomze,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0043) [I.M., Mertikopoulos, P., Schachinger, W., Staudigl, M., 2019. Hessian barrier algo- rithms for linearly constrained optimization problems. SIAM Journal on Optimization 29 (3), 2100–2127.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0043)

[Bottou,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0044) [L., Curtis, F., Nocedal, J., 2018. Optimization methods for large-scale machine learning. SIAM Review 60 (2), 223–311.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0044)

[Boyd,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0045) [S., Parikh, N., Chu, E., Peleato, B., Eckstein, J., 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. Foundations and Trends® in Machine learning 3 (1), 1–122.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0045)

[Boyd, S.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0046), [Vandenberghe, L., 2004. Convex optimization. Cambridge university press](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0046).

[Bruckstein,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0047) [A., Donoho, D., Elad, M., 2009. From sparse solutions of systems of equations to sparse modeling of signals and images. SIAM Review 51 (1), 34–81.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0047)

[Bubeck,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0048) [S., 2015. Convex optimization: Algorithms and complexity. Foundations and Trends in Machine Learning 8 (3-4), 231–357.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0048)

[Bühlmann,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0049) [P., van de Geer, S., 2011. Statistics for high-dimensional data. Springer Series in Statistics. Springer-Verlag Berlin Heidelberg.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0049)

[Bùi,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0050) [M.N., Combettes, P.L., 2021. Bregman forward-backward operator splitting. Set-Val- ued and Variational Analysis 29 (3), 583–603.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0050)

[Candes,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0051) [E., Tao, T., 2007. The Dantzig selector: Statistical estimation when *𝑝* is much](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0051)

[larger than *𝑛*. The Annals of Statistics 35 (6), 2313–2351.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0051)

[Candes, E.J., Romberg, J., Tao, T., 2006. Robust uncertainty principles: exact signal re- construction from highly incomplete frequency information. IEEE Transactions on Information Theory 52 (2), 489–509.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0052)

[Canon, M.D., Cullum, C.D., 1968. A tight upper bound on the rate of convergence of frank-wolfe algorithm. SIAM Journal on Control 6 (4), 509–516.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0053)

[Carderera, A., Besancon, M., Pokutta, S.. Simple steps are all you need: Frank-wolfe and generalized self-concordant functions.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0054)

[Carderera, A., Diakonikolas, J., Lin, C.Y., Pokutta, S., 2021. Parameter-free locally accel- erated conditional gradients. In: Meila, M., Zhang, T. (Eds.), Proceedings of the 38th International Conference on Machine Learning, Vol. 139 of Proceedings of Machine Learning Research, PMLR, pp. 1283–1293.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0055)

[Censor, Y., Zenios, S.A., 1992. Proximal minimization algorithm withd-functions. Journal of Optimization Theory and Applications 73 (3), 451–464.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0056)

[Cesa-Bianchi, N., Lugosi, G., 2006. Prediction, Learning, and Games. Cambridge University Press.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0057)

[Chambolle, A., Pock, T., 2011. A first-order primal-dual algorithm for convex problems with applications to imaging. Journal of Mathematical Imaging and Vision 40 (1), 120–145.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0058)

[Chen, C., He, B., Ye, Y., Yuan, X., 2016. The direct extension of admm for multi-block con- vex minimization problems is not necessarily convergent. Mathematical Programming 155 (1-2), 57–79.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0059)

[Chen, G., Teboulle, M., 1993. Convergence analysis of a proximal-like minimization algo- rithm using Bregman functions. SIAM Journal on Optimization 3 (3), 538–543.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0060)

[Chen, Y., Lan, G., Ouyang, Y., 2017. Accelerated schemes for a class of variational in- equalities. Mathematical Programming.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0061) [10.1007/s10107-017-1161-4](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0061)

[Chernov, A., Dvurechensky, P., Gasnikov, A., 2016. Fast primal-dual gradient method for strongly convex minimization problems with linear constraints. In: Kochetov, Y., Khachay, M., Beresnev, V., Nurminski, E., Pardalos, P. (Eds.), Discrete Optimiza- tion and Operations Research: 9th International Conference, DOOR 2016, Vladivos- tok, Russia, September 19-23, 2016, Proceedings. Springer International Publishing,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0062)

[pp. 391–403.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0062)

[Cohen, M., Diakonikolas, J., Orecchia, L., 2018. On acceleration with noise-corrupted gra- dients. In: Dy, J., Krause, A. (Eds.), Proceedings of the 35th International Conference on Machine Learning, Vol. 80 of Proceedings of Machine Learning Research, PMLR. Stockholmsmässan, Stockholm Sweden, pp. 1019–1028.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0063)

Cohen, M.B., Sidford, A., Tian, K., 2021. Relative lipschitzness in extragradient methods and a direct recipe for acceleration. In: Lee, J.R. (Ed.), 12th Innova- tions in Theoretical Computer Science Conference (ITCS 2021), pp. 62:1–62:18. doi:[10.4230/LIPIcs.ITCS.2021.62](https://doi.org/10.4230/LIPIcs.ITCS.2021.62). Vol. 185 of Leibniz International Proceedings in Informatics (LIPIcs), Schloss Dagstuhl–Leibniz-Zentrum für Informatik, Dagstuhl, Ger- many

[Combettes, P.L., Pesquet, J.C., 2011. Proximal splitting methods in signal processing.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0065)

[Springer, pp. 185–212.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0065)

[Combettes, P.L., Wajs, V.R., 2005. Signal recovery by proximal forward-backward split- ting. Multiscale Modeling & Simulation 4 (4), 1168–1200.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0066)

[Cox, B., Juditsky, A., Nemirovski, A., 2014. Dual subgradient algorithms for large-scale nonsmooth learning problems. Mathematical Programming 148 (1), 143–180.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0067)

[Damla Ahipasaoglu, S., Sun, P., Todd, M.J., 2008. Linear convergence of a modified frank–wolfe algorithm for computing minimum-volume enclosing ellipsoids. Optimi- sation Methods and Software 23 (1), 5–19.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0068)

[Danilova, M., Dvurechensky, P., Gasnikov, A., Gorbunov, E., Guminov, S., Kam-](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0069) [zolov, D., Shibaev, I., 2020. Recent theoretical advances in non-convex optimization. arXiv:2012.06188Accepted to be a part of Springer volume ”High Dimensional Opti- mization and Probability”.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0069)

[d’Aspremont, A., 2008. Smooth optimization with approximate gradient. SIAM J. on Op- timization 19 (3), 1171–1183.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0070)

[d’Aspremont, A.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0071), [Scieur, D.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0071), [Taylor, A.. Acceleration methods](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0071).

[Daubechies, I., Defrise, M., De Mol, C., 2004. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. Communications on pure and applied mathematics 57 (11), 1413–1457.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0072)

[Davis, D., Drusvyatskiy, D., Kakade, S., Lee, J.D., 2020. Stochastic subgradient method converges on tame functions. Foundations of Computational Mathematics 20 (1), 119–154.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0073)

[Devolder, O., 2013. Exactness, inexactness and stochasticity in first-order methods for large-scale convex optimization Ph.d. thesis.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0074)

[Devolder, O., Glineur, F., Nesterov, Y., 2014. First-order methods of smooth convex opti- mization with inexact oracle. Mathematical Programming 146 (1), 37–75.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0075)

[Diakonikolas, J., Orecchia, L., 2018. Alternating randomized block coordinate descent. In: Dy, J., Krause, A. (Eds.), Proceedings of the 35th International Conference on Ma- chine Learning, Vol. 80 of Proceedings of Machine Learning Research, PMLR, Stock- holmsmässan, Stockholm Sweden, pp. 1224–1232.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0076)

[Doljansky, M., Teboulle, M., 1998. An interior proximal algorithm and the exponential multiplier method for semidefinite programming. SIAM Journal on Optimization 9 (1), 1–13.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0077)

[Donoho, D.L., 2006. Compressed sensing. IEEE Transactions on Information Theory 52 (4), 1289–1306.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0078)

Dragomir, R.-A., Taylor, A.B., d’Aspremont, A., Bolte, J., 2021. Optimal complex- ity and certification of bregman first-order methods. Mathematical Programming doi:[10.1007/s10107-021-01618-1](https://doi.org/10.1007/s10107-021-01618-1).

[Duchi, J., Hazan, E., Singer, Y., 2011. Adaptive subgradient methods for online learning and stochastic optimization. Journal of Machine Learning Research 12, 2121–2159.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0080) [(Jul.)](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0080)

[Duchi, J., Shalev-Shwartz, S., Singer, Y., Tewari, A., 2010. Composite objective mirror descent. In: COLT 2010 - The 23rd Conference on Learning Theory, pp. 14–26.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0081)

[Duchi, J.C., Ruan, F., 2018. Stochastic methods for composite and weakly convex opti- mization problems. SIAM Journal on Optimization 28 (4), 3229–3259.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0082)

[Dunn,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0083) [J.C., 1979. Rates of convergence for conditional gradient algorithms near singu- lar and nonsingular extremals. SIAM Journal on Control and Optimization 17 (2), 187–211.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0083)

[Dvinskikh,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0084) [D., Gorbunov, E., Gasnikov, A., Dvurechensky, P., Uribe, C.A., 2019. On primal and dual approaches for distributed stochastic convex optimization over networks. In: 2019 IEEE 58th Conference on Decision and Control (CDC), pp. 7435–7440.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0084)

[Dvinskikh,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0085) [D., Ogaltsov, A., Gasnikov, A., Dvurechensky, P., Spokoiny, V., 2020. On the line-search gradient methods for stochastic optimization. IFAC-PapersOnLine 53 (2), 1715–1720.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0085)

[Dvurechensky, P., Dvinskikh, D., Gasnikov, A., Uribe, C.A., Nedić, A., 2018. Decentral- ize and randomize: Faster algorithm for Wasserstein barycenters. In: Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (Eds.), Ad- vances in Neural Information Processing Systems, Vol. 31. Curran Associates, Inc.,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0086)

[pp. 10783–10793.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0086) [NeurIPS 2018](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0086)

[Dvurechensky, P., Gasnikov, A., 2016. Stochastic intermediate gradient method for convex problems with stochastic inexact oracle. Journal of Optimization Theory and Appli- cations 171 (1), 121–145.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0087)

[Dvurechensky, P., Gasnikov, A., Gasnikova, E., Matsievsky, S., Rodomanov, A., Usik, I., 2016. Primal-dual method for searching equilibrium in hierarchical congestion popu- lation games. In: Supplementary Proceedings of the 9th International Conference on Discrete Optimization and Operations Research and Scientific School (DOOR 2016) Vladivostok, Russia, September 19 - 23, 2016, pp. 584–595.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0088)

[Dvurechensky, P., Gasnikov, A., Kroshnin, A., 2018. Computational optimal transport: Complexity by accelerated gradient descent is better than by Sinkhorn’s algorithm. In: Dy, J., Krause, A. (Eds.), Proceedings of the 35th International Conference on Machine Learning, Vol. 80 of Proceedings of Machine Learning Research, pp. 1367–1376.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0089)

[Dvurechensky, P., Gasnikov, A., Tiurin, A., Zholobov, V.. Unifying framework for acceler- ated randomized methods in convex optimization.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0090)

[Dvurechensky, P., Gorbunov, E., Gasnikov, A., 2021. An accelerated directional derivative method for smooth stochastic convex optimization. European Journal of Operational Research 290 (2), 601–621.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0091)

[Dvurechensky, P.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0092), [Kamzolov, D.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0092), [Lukashevich, A.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0092), [Lee, S.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0092), [Ordentlich, E.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0092), [Uribe, C.A.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0092), [Gasnikov, A.. Hyperfast second-order local solvers for eﬃcient statistically precon- ditioned distributed optimization.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0092)

[Dvurechensky, P., Ostroukhov, P., Safin, K., Shtern, S., Staudigl, M., 2020. Self-concordant analysis of frank-Wolfe algorithms. In: Daumé III, H., Singh, A. (Eds.), Proceedings of the 37th International Conference on Machine Learning, Vol. 119 of Proceedings of Machine Learning Research, PMLR, Virtual, pp. 2814–2824.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0093)

[Dvurechensky, P., Safin, K., Shtern, S., Staudigl, M.. Generalized self-concordant analysis of Frank-Wolfe algorithms.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0094)

[Dvurechensky, P.E., Gasnikov, A.V., Nurminski, E.A., Stonyakin, F.S., 2020. Advances in Low-Memory Subgradient Optimization. Springer International Publishing, Cham,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0095)

[pp. 19–59.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0095)

[Eckstein,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0096) [J., 1994. Some saddle-function splitting methods for convex programming. Op- timization Methods and Software 4 (1), 75–83.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0096)

[Eckstein,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0097) [J., Bertsekas, D.P., 1992. On the douglas—rachford splitting method and the proximal point algorithm for maximal monotone operators. Mathematical Program- ming 55 (1-3), 293–318.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0097)

[Epelman,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0098) [M., Freund, R.M., 2000. Condition number complexity of an elementary algo- rithm for computing a reliable solution of a conic linear system. Mathematical Pro- gramming 88 (3), 451–485.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0098)

[Facchinei,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0099) [F., Pang, J.S., 2003. Finite-dimensional variational inequalities and comple- mentarity problems - volume i and volume II. Springer Series in Operations Research.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0099)

[Feizollahi,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0100) [M.J., Ahmed, S., Sun, A., 2017. Exact augmented lagrangian duality for mixed integer linear programming. Mathematical Programming 161 (1), 365–387.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0100)

[Fercoq,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0101) [O., Qu, Z., 2020. Restarting the accelerated coordinate descent method with a rough strong convexity estimate. Computational Optimization and Applications 75 (1), 63–91.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0101)

[Fercoq,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0102) [O., Richtárik, P., 2015. Accelerated, parallel, and proximal coordinate descent.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0102)

[SIAM Journal on Optimization 25 (4), 1997–2023.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0102)

[Frank,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0103) [M., Wolfe, P., et al., 1956. An algorithm for quadratic programming. Naval research logistics quarterly 3 (1-2), 95–110.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0103)

[Freund,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0104) [R.M., Grigas, P., 2016. New analysis and results for the frank–wolfe method.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0104)

[Mathematical Programming 155 (1-2), 199–230.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0104)

[Frostig,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0105) [R., Ge, R., Kakade, S., Sidford, A., 2015. Un-regularizing: approximate proximal point and faster stochastic algorithms for empirical risk minimization. In: Bach, F., Blei, D. (Eds.), Proceedings of the 32nd International Conference on Machine Learn- ing, Vol. 37 of Proceedings of Machine Learning Research, PMLR, Lille, France,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0105)

[pp. 2540–2548.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0105)

[Gabay,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0106) [D., 1983. Applications of the method of multipliers to variational inequalities, 15.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0106)

[Elsevier, pp. 299–331.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0106) [Ch. ix](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0106)

[Garber,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0107) [D., Hazan, E., 2015. Faster rates for the frank-wolfe method over strongly-convex sets. In: 32nd International Conference on Machine Learning, ICML 2015.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0107)

[Garber,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0108) [D., Hazan, E., 2016. A linearly convergent variant of the conditional gradient algo- rithm under strong convexity, with applications to online and stochastic optimization. SIAM Journal on Optimization 26 (3), 1493–1528.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0108)

[Gasnikov, A.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0109), [Dvurechensky, P.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0109), [Gorbunov, E.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0109), [Vorontsova, E.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0109), [Selikhanovych, D.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0109), [Uribe, C.A., Jiang, B., Wang, H., Zhang, S., Bubeck, S., Jiang, Q., Lee, Y.T., Li, Y., Sid-](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0109) [ford, A., 2019. Near optimal methods for minimizing convex functions with lipschitz](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0109)

[*𝑝*-th derivatives. In: Beygelzimer, A., Hsu, D. (Eds.), Proceedings of the Thirty-Second](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0109)

[Conference on Learning Theory, Vol. 99 of Proceedings of Machine Learning Research, PMLR, Phoenix, USA, pp. 1392–1393.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0109)

[Gasnikov,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0110) [A.V., Dvurechensky, P.E., 2016. Stochastic intermediate gradient method for convex optimization problems. Doklady Mathematics 93 (2), 148–151.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0110)

[Gasnikov,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0111) [A.V., Nesterov, Y.E., 2018. Universal method for stochastic composite optimiza- tion problems. Computational Mathematics and Mathematical Physics 58 (1), 48–64.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0111)

[lems with an oracle producing a (*𝛿*, l)-model of function at the requested point. Com- Gasnikov, A.V., Tyurin, A.I., 2019. Fast gradient descent for convex minimization prob-](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0112)

[putational Mathematics and Mathematical Physics 59 (7), 1085–1097.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0112)

[Ghadimi, S., Lan, G., 2012. Optimal stochastic approximation algorithms for strongly con- vex stochastic composite optimization i: A generic algorithmic framework. SIAM Jour- nal on Optimization 22 (4), 1469–1492.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0113)

[Ghadimi, S., Lan, G., 2013. Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization, ii: Shrinking procedures and optimal al- gorithms. SIAM Journal on Optimization 23 (4), 2061–2089.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0114)

[Ghadimi, S., Lan, G., Zhang, H., 2019. Generalized uniformly optimal methods for non- linear programming. Journal of Scientific Computing 79 (3), 1854–1881.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0115)

[Glowinski, R., Tallec, P.L., 1989. Augmented Lagrangian and operator-splitting methods in nonlinear mechanics. SIAM.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0116)

[Goodfellow, I.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0117), [Pouget-Abadie, J.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0117), [Mirza, M.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0117), [Xu, B.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0117), [Warde-Farley, D.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0117), [Ozair, S.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0117), [Courville, A., Bengio, Y., 2014. Generative adversarial nets. In: Advances in neural information processing systems, pp. 2672–2680.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0117)

[Gorbunov, E., Danilova, M., Gasnikov, A., 2020. Stochastic optimization with heavy-tailed noise via accelerated gradient clipping. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H. (Eds.), Advances in Neural Information Processing Systems, Vol. 33. Curran Associates, Inc., pp. 15042–15053.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0118)

[Gorbunov, E., Danilova, M., Shibaev, I., Dvurechensky, P., Gasnikov, A.. Near-optimal high probability complexity bounds for non-smooth stochastic optimization with heavy– tailed noise.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0119)

[Gorbunov, E., Dvinskikh, D., Gasnikov, A.. Optimal decentralized distributed algorithms for stochastic convex optimization.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0120)

[Gorbunov, E., Dvurechensky, P., Gasnikov, A.. An accelerated method for derivative-free smooth stochastic convex optimization.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0121)

Gorbunov, E., Rogozin, A., Beznosikov, A., Dvinskikh, D., Gasnikov, A., d. Recent the- oretical advances in decentralized distributed convex optimization. arXiv preprint arXiv:2011.13259Accepted to be a part of Springer volume ”High Dimensional Opti- mization and Probability”.

[Gower, R.M., Schmidt, M., Bach, F., Richtárik, P., 2020. Variance-reduced methods for machine learning. In: Proceedings of the IEEE, Vol. 108, pp. 1968–1983.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0123)

[Guélat, J., Marcotte, P., 1986. Some comments on wolfe’s ‘away step’. Mathematical Pro- gramming 35 (1), 110–119.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0124)

[Guminov, S., Dvurechensky, P., Tupitsa, N., Gasnikov, A., 2021. On a combination of alternating minimization and Nesterov’s momentum. In: Meila, M., Zhang, T. (Eds.), Proceedings of the 38th International Conference on Machine Learning, Vol. 139 of Proceedings of Machine Learning Research, PMLR, Virtual, pp. 3886–3898.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0125)

[Guminov, S.V., Nesterov, Y.E., Dvurechensky, P.E., Gasnikov, A.V., 2019. Accelerated pri- mal-dual gradient descent with linesearch for convex, nonconvex, and nonsmooth optimization problems. Doklady Mathematics 99 (2), 125–128.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0126)

Haeser, G., Liu, H., Ye, Y., 2018. Optimality condition and complexity analysis for linearly- constrained optimization without differentiability on the boundary. Mathematical Programming doi:[10.1007/s10107-018-1290-4](https://doi.org/10.1007/s10107-018-1290-4).

[Hanzely, F., Richtárik, P., Xiao, L., 2021. Accelerated bregman proximal gradient methods for relatively smooth convex optimization. Computational Optimization and Applica- tions 79 (2), 405–440.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0128)

[Helmke, U., Moore, J.B., 1996. Optimization and dynamical systems. Communications & Control Engineering. Springer Berlin Heidelberg.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0129)

[Hendrikx, H., Xiao, L., Bubeck, S., Bach, F., Massoulie, L., 2020. Statistically precondi- tioned accelerated gradient method for distributed optimization. In: Daumé III, H., Singh, A. (Eds.), Proceedings of the 37th International Conference on Machine Learn- ing, Vol. 119 of Proceedings of Machine Learning Research, PMLR, pp. 4203–4227.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0130)

[Hiriart-Urrut, J.-B.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0131), [Lemaréchal, C., 2001. Fundamentals of Convex Analysis. Springer](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0131). [Holloway, C.A., 1974. An extension of the frank and wolfe method of feasible directions.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0132)

[Mathematical Programming 6 (1), 14–27.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0132)

Ivanova, A., Dvurechensky, P., Gasnikov, A., Kamzolov, D., 2020. Composite optimization for the resource allocation problem. Optimization Methods and Software 0 (0), 1–35. doi:[10.1080/10556788.2020.1712599](https://doi.org/10.1080/10556788.2020.1712599).

[Jaggi, M., 2013. Revisiting frank-wolfe: Projection-free sparse convex optimization. Inter- national Conference on Machine Learning 427–435.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0134)

[Jain, P., Kar, P., 2017. Non-convex optimization for machine learning. Found. Trends Mach. Learn. 10 (3-4), 142–336.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0135)

[with applications to 𝓁1 -minimization. Mathematical Programming 142 (1), 269–310. Juditsky, A., Kılınç Karzan, F., Nemirovski, A., 2013. Randomized first order algorithms](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0136)

[Juditsky, A., Nazin, A.V., Tsybakov, A.B., Vayatis, N., 2005. Recursive aggregation of estimators by the mirror descent algorithm with averaging. Problems of Information Transmission 41 (4), 368–384.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0137)

[Juditsky, A., Nemirovski, A., 2011. First order methods for nonsmooth convex large-scale optimization, i: general purpose methods. In: Optimization for Machine Learning. MIT Press, pp. 121–148.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0138) [Ch. 5](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0138)

[Juditsky, A., Nemirovski, A., 2011. First order methods for nonsmooth convex large-scale optimization, ii: utilizing problems structure. In: Optimization for Machine Learning. MIT Press, pp. 149–183.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0139) [Ch. 6](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0139)

[Juditsky, A., Nesterov, Y., 2014. Deterministic and stochastic primal-dual subgradient algorithms for uniformly convex minimization. Stochastic Systems 4 (1), 44–80.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0140)

Kairouz, P., McMahan, H.B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A.N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., D’Oliveira, R.G.L., Eichner, H., Rouay- heb, S.E., Evans, D., Gardner, J., Garrett, Z., Gascón, A., Ghazi, B., Gibbons, P.B.,

Gruteser, M., Harchaoui, Z., He, C., He, L., Huo, Z., Hutchinson, B., Hsu, J., Jaggi, M., Javidi, T., Joshi, G., Khodak, M., Konecný, J., Korolova, A., Koushanfar, F., Koyejo, S., Lepoint, T., Liu, Y., Mittal, P., Mohri, M., Nock, R., Özgür, A., Pagh, R., Qi, H., Ram-

age, D., Raskar, R., Raykova, M., Song, D., Song, W., Stich, S.U., Sun, Z., Suresh, A.T.,

Tramèr, F., Vepakomma, P., Wang, J., Xiong, L., Xu, Z., Yang, Q., Yu, F.X., Yu, H.,

Zhao, S., 2021. Advances and open problems in federated learning. Foundations and Trends®in Machine Learning 14 (1-2), 1–210. doi:[10.1561/2200000083](https://doi.org/10.1561/2200000083).

Kamzolov, D., Dvurechensky, P., Gasnikov, A.V., 2020. Universal intermediate gradient method for convex problems with inexact oracle. Optimization Methods and Software 0 (0), 1–28. doi:[10.1080/10556788.2019.1711079](https://doi.org/10.1080/10556788.2019.1711079).

[Kerdreux,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0143) [T., d’Aspremont, A., Pokutta, S., 2019. Restarting frank-wolfe. In: Chaudhuri, K., Sugiyama, M. (Eds.), Proceedings of Machine Learning Research, Vol. 89 of Proceed- ings of Machine Learning Research, PMLR, pp. 1275–1283.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0143)

[Kroshnin,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0144) [A., Tupitsa, N., Dvinskikh, D., Dvurechensky, P., Gasnikov, A., Uribe, C., 2019. On the complexity of approximating Wasserstein barycenters. In: Chaudhuri, K., Salakhutdinov, R. (Eds.), Proceedings of the 36th International Conference on Ma- chine Learning, Vol. 97 of Proceedings of Machine Learning Research, PMLR, Long Beach, California, USA, pp. 3530–3540.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0144)

[Kuczyński, J., Woźniakowski, H., 1992. Estimating the largest eigenvalue by the power and lanczos algorithms with a random start. SIAM Journal on Matrix Analysis and Applications 13 (4), 1094–1122.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0145)

[Kushner,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0146) [H.J., 1984. Approximation and Weak Convergence Methods for Random Pro- cesses. The MIT Press.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0146)

[Lacoste-Julien, S., Jaggi, M., 2015. On the global linear convergence of frank-wolfe opti- mization variants. Advances in neural information processing systems 28, 496–504.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0147)

[Lan,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0148) [G.. The complexity of large-scale convex programming under a linear optimization oracle.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0148)

[Lan,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0149) [G., 2012. An optimal method for stochastic composite optimization. Mathematical Programming 133 (1), 365–397.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0149)

[Lan,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0150) [G., 2020. First-order and Stochastic Optimization Methods for Machine Learning.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0150)

[Springer.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0150)

Lan, G., Zhou, Y., 2017. An optimal randomized incremental gradient method. Mathemat- ical Programming doi:[10.1007/s10107-017-1173-0](https://doi.org/10.1007/s10107-017-1173-0).

[Lan,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0152) [G., Zhou, Y., 2016. Conditional gradient sliding for convex optimization. SIAM Jour- nal on Optimization 26 (2), 1379–1409.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0152)

[Lee,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0153) [Y.T., Sidford, A., 2013. Eﬃcient accelerated coordinate descent methods and faster algorithms for solving linear systems. In: Proceedings of the 2013 IEEE 54th Annual Symposium on Foundations of Computer Science, FOCS ’13. IEEE Computer Society, Washington, DC, USA, pp. 147–156.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0153)

[Levitin,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0154) [E.S., Polyak, B.T., 1966. Constrained minimization methods. USSR Computational mathematics and mathematical physics 6 (5), 1–50.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0154)

[Levy,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0155) [K.Y., Yurtsever, A., Cevher, V., 2018. Online adaptive methods, universality and ac- celeration. In: Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (Eds.), Advances in Neural Information Processing Systems, Vol. 31. Cur- ran Associates, Inc., pp. 6500–6509.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0155)

[Lin,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0156) [F., Fardad, M., Jovanović, M.R., 2012. Sparse feedback synthesis via the alternating direction method of multipliers. In: 2012 American Control Conference (ACC). IEEE,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0156)

[pp. 4765–4770.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0156)

[Lin,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0157) [H., Mairal, J., Harchaoui, Z., 2015. A universal catalyst for first-order optimization. In: Proceedings of the 28th International Conference on Neural Information Processing Systems, NIPS’15. MIT Press, Cambridge, MA, USA, pp. 3384–3392.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0157)

[Lin,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0158) [Q., Lu, Z., Xiao, L., 2014. An accelerated proximal coordinate gradient method. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q. (Eds.), Advances in Neural Information Processing Systems, Vol. 27. Curran Associates, Inc.,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0158)

[pp. 3059–3067.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0158)

[Lin,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0159) [T., Ho, N., Chen, X., Cuturi, M., Jordan, M., 2020. Fixed-support wasserstein barycen- ters: Computational hardness and fast algorithm. In: Larochelle, H., Ranzato, M., Had- sell, R., Balcan, M.F., Lin, H. (Eds.), Advances in Neural Information Processing Sys- tems, Vol. 33. Curran Associates, Inc., pp. 5368–5380.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0159)

[Lin,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0160) [T., Ho, N., Cuturi, M., Jordan, M.I.. On the complexity of approximating multimarginal optimal transport.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0160)

[Lin,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0161) [T., Ho, N., Jordan, M., 2019. On eﬃcient optimal transport: An analysis of greedy and accelerated mirror descent algorithms. In: Chaudhuri, K., Salakhutdinov, R. (Eds.), Proceedings of the 36th International Conference on Machine Learning, Vol. 97 of Proceedings of Machine Learning Research, PMLR, Long Beach, California, USA,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0161)

[pp. 3982–3991.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0161)

[Ljung,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0162) [L., Pflug, G., Walk, H., 2012. Stochastic approximation and optimization of random systems, 17. Birkhäuser.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0162)

[Lu,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0163) [H., Freund, R., Nesterov, Y., 2018. Relatively smooth convex optimization by first-order methods, and applications. SIAM Journal on Optimization 28 (1), 333–354.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0163) [Malitsky,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0164) [Y., Pock, T., 2018. A first-order primal-dual algorithm with linesearch. SIAM](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0164)

[Journal on Optimization 28 (1), 411–432.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0164)

[Martinet,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0165) [B., 1970. Régularisation d’inéquations variationnelles par approximations suc- cessives. Revue française d’informatique et de recherche opérationnelle. Série rouge 4 (R3), 154–158.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0165)

[Mertikopoulos, P., Staudigl, M., 2018. On the convergence of gradient-like flows with noisy gradient input. SIAM Journal on Optimization 28 (1), 163–197.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0166)

[Mertikopoulos, P., Staudigl, M., 2018. Stochastic mirror descent dynamics and their con- vergence in monotone variational inequalities. Journal of Optimization Theory and Applications 179 (3), 838–867.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0167)

[Monteiro,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0168) [R., Svaiter, B., 2013. An accelerated hybrid proximal extragradient method for convex optimization and its implications to second-order methods. SIAM Journal on Optimization 23 (2), 1092–1125.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0168)

[Moreau,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0169) [J.J., 1965. Proximité et dualité dans un espace hilbertien. Bulletin de la Société mathématique de France 93, 273–299.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0169)

[Necoara,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0170) [I., Nesterov, Y., Glineur, F., 2019. Linear convergence of first order methods for non-strongly convex optimization. Mathematical Programming 175 (1), 69–107.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0170)

[Nemirovski,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0171) [A., 1982. Orth-method for smooth convex optimization. Izvestia AN SSSR, Transl.: Eng. Cybern. Soviet J. Comput. Syst. Sci 2, 937–947.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0171)

[Nemirovski, A., 2004. Prox-method with rate of convergence *𝑜*(1∕*𝑡*) for variational in-](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0172)

[equalities with lipschitz continuous monotone operators and smooth convex-concave saddle point problems. SIAM Journal on Optimization 15 (1), 229–251.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0172)

[Nemirovski, A.S., Yudin, D.B., 1983. Problem Complexity and Method Eﬃciency in Opti- mization. Wiley, New York, NY.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0173)

[Nemirovskii, A., Nesterov, Y., 1985. Optimal methods of smooth convex minimization.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0174)

[USSR Computational Mathematics and Mathematical Physics 25 (2), 21–30.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0174)

Nesterov, Y., 2019. Implementable tensor methods in unconstrained convex optimization.

Mathematical Programming doi:[10.1007/s10107-019-01449-1](https://doi.org/10.1007/s10107-019-01449-1).

[rate *𝑜*(1∕*𝑘*2 ). Soviet Mathematics Doklady 27 (2), 372–376. Nesterov, Y., 1983. A method of solving a convex programming problem with convergence](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0176)

[Nesterov, Y., 2005. Excessive gap technique in nonsmooth convex minimization. SIAM Journal on Optimization 16 (1), 235–249.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0177)

[Nesterov, Y., 2005. Smooth minimization of non-smooth functions. Mathematical Pro- gramming 103 (1), 127–152.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0178)

[Nesterov, Y., 2007. Dual extrapolation and its applications to solving variational inequal- ities and related problems. Mathematical Programming 109 (2), 319–344.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0179)

[Nesterov, Y., 2008. Accelerating the cubic regularization of newton’s method on convex problems. Mathematical Programming 112 (1), 159–181.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0180)

[Nesterov, Y., 2009. Primal-dual subgradient methods for convex problems. Mathematical Programming 120 (1), 221–259.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0181)

[Nesterov, Y., 2012. Eﬃciency of coordinate descent methods on huge-scale optimization problems. SIAM Journal on Optimization 22 (2), 341–362.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0182)

[Nesterov, Y., 2013. Gradient methods for minimizing composite functions. Mathematical Programming 140 (1), 125–161.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0183)

[Nesterov, Y., 2015. Universal gradient methods for convex optimization problems. Math- ematical Programming 152 (1), 381–404.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0184)

[Nesterov, Y., 2018. Complexity bounds for primal-dual methods minimizing the model of objective function. Mathematical Programming 171 (1-2), 311–330.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0185)

[Nesterov, Y., 2018. Lectures on convex optimization. Vol. 137 of Springer Optimization and Its Applications. Springer International Publishing.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0186)

Nesterov, Y., Gasnikov, A., Guminov, S., Dvurechensky, P., 2020. Primal-dual accelerated gradient methods with small-dimensional relaxation oracle. Optimization Methods and Software 1–28. doi:[10.1080/10556788.2020.1731747](https://doi.org/10.1080/10556788.2020.1731747).

[Nesterov, Y., Nemirovski, A., 1994. Interior Point Polynomial methods in Convex pro- gramming. SIAM Publications.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0188)

[Nesterov, Y., Spokoiny, V., 2017. Random gradient-free minimization of convex functions.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0189)

[Found. Comput. Math. 17 (2), 527–566.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0189)

Nesterov, Y., Stich, S.U., 2017. Eﬃciency of the accelerated coordinate descent method on structured optimization problems. SIAM Jour- nal on Optimization 27 (1), 110–123. First presented in May 2015 http://www.mathnet.ru:8080/PresentFiles/11909/7\_nesterov.pdf

[Odor, G., Li, Y.-H., Yurtsever, A., Hsieh, Y.-P., Tran-Dinh, Q., Halabi, M.E., Cevher, V., 2016. Frank-wolfe works for non-lipschitz continuous gradient objectives: scalable poisson phase retrieval. In: 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). Ieee, pp. 6230–6234.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0191)

[Opial, Z., 1967. Weak convergence of the sequence of successive approximations for non- expansive mappings. Bulletin of the American Mathematical Society 73 (4), 591–597.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0192) [Parikh, N., Boyd, S., 2014. Proximal algorithms. Foundations and Trends® in Optimization](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0193)

[1 (3), 127–239.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0193)

[Pedregosa, F., Negiar, G., Askari, A., Jaggi, M., 2020. Linearly convergent frank-wolfe with backtracking line-search. In: International Conference on Artificial Intelligence and Statistics, PMLR, pp. 1–10.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0194)

[Pokutta, S., 2020. Restarting algorithms: Sometimes there is free lunch. In: Hebrard, E., Musliu, N. (Eds.), Integration of Constraint Programming, Artificial Intelligence, and Operations Research. Springer International Publishing, Cham, pp. 22–38.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0195)

[Polyak, B.T., 1964. Some methods of speeding up the convergence of iteration methods.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0196)

[USSR Computational Mathematics and Mathematical Physics 4 (5), 1–17.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0196)

[Polyak, B.T., 1987. Introduction to Optimization. Optimization Software](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0197).

[Robbins, H., Monro, S., 1951. A stochastic approximation method. The Annals of Mathe- matical Statistics 22 (3), 400–407.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0198)

[Robinson, S.M., 1982. Generalized equations and their solutions, part ii: applications to nonlinear programming. In: Optimality and Stability in Mathematical Programming. Springer, pp. 200–221.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0199)

[Rockafellar, R.T., 1970. Convex Analysis. Princeton University Press, Princeton](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0200).

[Rockafellar, R.T., 1976. Augmented lagrangians and applications of the proximal point algorithm in convex programming. Mathematics of operations research 1 (2), 97–116.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0201) [Rockafellar, R.T., 1976. Monotone operators and the proximal point algorithm. SIAM jour-](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0202)

[nal on control and optimization 14 (5), 877–898.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0202)

[Rockafellar, R.T., Wets, R.J.B., 1998. Variational analysis. Vol. 317 of A Series of Com- prehensive Studies in Mathematics. Springer-Verlag, Berlin.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0203)

[Rogozin, A., Bochko, M., Dvurechensky, P., Gasnikov, A., Lukoshkin, V., 2021. An acceler- ated method for decentralized distributed stochastic optimization over time-varying graphs. In: 2021 60th IEEE Conference on Decision and Control (CDC).](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0204)

[Roulet, V., d’Aspremont, A., 2017. Sharpness, restart and acceleration. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (Eds.), Advances in Neural Information Processing Systems, Vol. 30. Curran Asso- ciates, Inc., pp. 1119–1129.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0205)

[Shalev-Shwartz, S., Zhang, T., 2014. Accelerated proximal stochastic dual coordinate as- cent for regularized loss minimization. In: Xing, E.P., Jebara, T. (Eds.), Proceedings of the 31st International Conference on Machine Learning, Vol. 32 of Proceedings of Machine Learning Research, PMLR, Bejing, China, pp. 64–72.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0206)

[Shapiro,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0207) [A., Dentcheva, D., Ruszczyński, A., 2009. Lectures on stochastic programming.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0207)

[Society for Industrial and Applied Mathematics.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0207)

[Shefi,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0208) [R., Teboulle, M., 2014. Rate of convergence analysis of decomposition methods based on the proximal method of multipliers for convex minimization. SIAM Journal on Optimization 24 (1), 269–297.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0208)

[Shi,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0209) [B., Du, S.S., Su, W., Jordan, M.I., 2019. Acceleration via symplectic discretization of high-resolution differential equations. Advances in Neural Information Processing Systems 5744–5752.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0209)

[Sorin, S., 2000. A First-Course on Zero-Sum Repeated Games. Springer](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0210).

[Stonyakin,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0211) [F., Gasnikov, A., Dvurechensky, P., Alkousa, M., Titov, A.. Generalized Mirror Prox for monotone variational inequalities: Universality and inexact oracle.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0211)

Stonyakin, F., Tyurin, A., Gasnikov, A., Dvurechensky, P., Agafonov, A., Dvinskikh, D., Alkousa, M., Pasechnyuk, D., Artamonov, S., Piskunova, V., 2020. Inexact model: A framework for optimization and variational inequalities. Optimization Methods and Software doi:[10.1080/10556788.2021.1924714](https://doi.org/10.1080/10556788.2021.1924714).

[Stonyakin,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0213) [F.S., Dvinskikh, D., Dvurechensky, P., Kroshnin, A., Kuznetsova, O., Aga-](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0213) [fonov, A., Gasnikov, A., Tyurin, A., Uribe, C.A., Pasechnyuk, D., Artamonov, S., 2019. Gradient methods for problems with inexact model of the objective. In: Khachay, M., Kochetov, Y., Pardalos, P. (Eds.), Mathematical Optimization Theory and Operations Research. Springer International Publishing, Cham, pp. 97–114.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0213)

[Sturm,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0214) [J.F., 1999. Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones. Optimization methods and software 11 (1-4), 625–653.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0214)

[Su,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0215) [W., Boyd, S., Candes, E.J., 2016. A differential equation for modeling nesterov’s accel- erated gradient method: Theory and insights. Journal of Machine Learning Research.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0215) [Sun,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0216) [A.X., Phan, D.T., Ghosh, S., 2013. Fully decentralized ac optimal power flow algo-](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0216)

[rithms. In: 2013 IEEE Power & Energy Society General Meeting. IEEE, pp. 1–5.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0216)

[Sun, R.. Optimization for deep learning: theory and algorithms](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0217).

[Teboulle,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0218) [M., 1992. Entropic proximal mappings with applications to nonlinear program- ming. Mathematics of Operations Research 17, 670–690.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0218)

[Teboulle,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0219) [M., 2018. A simplified view of first order methods for optimization. Mathemat- ical Programming 170 (1), 67–96.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0219)

[Todd,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0220) [M.J., 2016. Minimum-volume ellipsoids. Society for Industrial and Applied Mathe- matics.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0220)

[Tran-Dinh,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0221) [Q., Alacaoglu, A., Fercoq, O., Cevher, V., 2020. An adaptive primal-dual frame- work for nonsmooth convex minimization. Mathematical Programming Computation 12 (3), 451–491.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0221)

[Tran-Dinh,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0222) [Q., Cevher, V., 2014. Constrained convex minimization via model-based exces- sive gap. In: Proceedings of the 27th International Conference on Neural Information Processing Systems, NIPS’14. MIT Press, Cambridge, MA, USA, pp. 721–729.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0222)

[Tran-Dinh,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0223) [Q., Fercoq, O., Cevher, V., 2018. A smooth primal-dual optimization frame- work for nonsmooth composite convex minimization. SIAM Journal on Optimization 28 (1), 96–134.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0223)

[Tseng,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0224) [P., 1991. Applications of a splitting algorithm to decomposition in convex program- ming and variational inequalities. SIAM Journal on Control and Optimization 29 (1), 119–138.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0224)

[Tseng,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0225) [P., 2008. On accelerated proximal gradient methods for convex-concave optimiza- tion. Tech. rep.. MIT.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0225)

[Tupitsa,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0226) [N., Dvurechensky, P., Gasnikov, A., Uribe, C.A., 2020. Multimarginal optimal transport by accelerated alternating minimization. In: 2020 59th IEEE Conference on Decision and Control (CDC), pp. 6132–6137.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0226)

[Uribe,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0227) [C.A., Dvinskikh, D., Dvurechensky, P., Gasnikov, A., Nedić, A., 2018. Distributed computation of Wasserstein barycenters over networks. In: 2018 IEEE Conference on Decision and Control (CDC), pp. 6544–6549.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0227)

[Van](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0228) [Nguyen, Q., 2017. Forward-backward splitting with bregman distances. Vietnam Journal of Mathematics 45 (3), 519–539.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0228)

[Von](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0229) [Hohenbalken, B., 1977. Simplicial decomposition in nonlinear programming algo- rithms. Mathematical Programming 13 (1), 49–68.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0229)

[Vorontsova,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0230) [E.A., Gasnikov, A.V., Gorbunov, E.A., 2019. Accelerated directional search with non-euclidean prox-structure. Automation and Remote Control 80 (4), 693–707.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0230) [Vorontsova,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0231) [E.A., Gasnikov, A.V., Gorbunov, E.A., Dvurechenskii, P.E., 2019. Accelerated gradient-free optimization methods with a non-euclidean proximal operator. Automa-](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0231)

[tion and Remote Control 80 (8), 1487–1501.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0231)

[Wibisono,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0232) [A., Wilson, A.C., Jordan, M.I., 2016. A variational perspective on accelerated methods in optimization. Proceedings of the National Academy of Sciences 113 (47), E7351.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0232)

[Wolfe,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0233) [P., 1970. Convergence theory in nonlinear programming. In: Abadie, J. (Ed.), In- teger and nonlinear programming. North-Holland, Amsterdam.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0233)

[Wright,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0234) [S.J., 2018. Optimization algorithms for data analysis. The Mathematics of Data 25, 49.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0234)

[Yang,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0235) [J., Zhang, Y., 2011. Alternating direction algorithms for ∖ell\_1-problems in com-](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0235)

[pressive sensing. SIAM journal on scientific computing 33 (1), 250–278.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0235)

[Yuan,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0236) [X., 2012. Alternating direction method for covariance selection models. Journal of Scientific Computing 51 (2), 261–273.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0236)

[Zhang,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0237) [Y., Xiao, L., 2015. Stochastic primal-dual coordinate method for regularized em- pirical risk minimization. In: Bach, F., Blei, D. (Eds.), Proceedings of the 32nd Interna- tional Conference on Machine Learning, Vol. 37 of Proceedings of Machine Learning Research, PMLR, Lille, France, pp. 353–361.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0237)

[Zhao,](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0238) [R., Freund, R.M.. Analysis of the Frank-Wolfe method for logarithmically-homoge- neous barriers, with an extension.](http://refhub.elsevier.com/S2192-4406(21)00142-8/sbref0238)