Journal of Optimization Theory and Applications
https://doi.org/10.1007/s10957-024-02412-7

Polyak Minorant Method for Convex Optimization

Nikhil Devanathan[1] · **Stephen Boyd**[1]

Received: 10 October 2023 / Accepted: 18 February 2024

Abstract
In 1963 Boris Polyak suggested a particular step size for gradient descent methods, now known as the Polyak step size, that he later adapted to subgradient methods. The Polyak step size requires knowledge of the optimal value of the minimization problem, which is a strong assumption but one that holds for several important problems. In this paper we extend Polyak's method to handle constraints and, as a generalization of subgradients, general minorants, which are convex functions that tightly lower bound the objective and constraint functions. We refer to this algorithm as the Polyak Minorant Method (PMM). It is closely related to cutting-plane and bundle methods.

Keywords Convex optimization · First-order methods · Cutting-plane methods · Bundle methods

Mathematics Subject Classification 49M20 · 49M37 · 90C25 · 90C55

1 Introduction

1.1 The Problem

We consider the convex optimization problem

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leq 0, \quad i = 1, \ldots, m \\ & Ax = b, \end{array} \tag{1}$$

Communicated by Arkadi Nemirovski.

✉ Nikhil Devanathan
ndev@stanford.edu

Stephen Boyd
boyd@stanford.edu

1 Stanford University, Stanford, CA 94305, USA

Published online: 30 March 2024

with variable $x \in \mathbf{R}^n$, where $f_0 : \Omega \to \mathbf{R}$ and $f_i : \mathbf{R}^n \to \mathbf{R}$, $i = 1, \ldots, m$ are closed proper convex functions, $A \in \mathbf{R}^{p \times n}$, and $b \in \mathbf{R}^p$. We can have $m = 0$ (no inequality constraints) or $p = 0$ (no equality constraints).

We let $\mathcal{F}$ denote the set of feasible points for (1). For $\rho > 0$, $\mathcal{F}_\rho$ will denote the ρ-violated constraint set,

$$\mathcal{F}_\rho = \{x \mid f_i(x) \le \rho,\ i = 1, \ldots, m,\ Ax = b\}.$$

We will assume that $\mathcal{F}_\rho \subseteq \Omega$ for some $\rho > 0$, which means that when x is ρ-close to feasible, the objective $f_0(x)$ is defined. We assume that the optimal value

$$f^\star = \inf\{f_0(x) \mid f_i(x) \le 0,\ i = 1, \ldots, m,\ Ax = b\}$$

is finite and achieved, i.e., there exists at least one optimal point $x^\star$, with $f_i(x^\star) \le 0$ for $i = 1, \ldots, m$ and $f_0(x^\star) = f^\star$. We define $\mathcal{X}^\star$ to be the set of optimal points. For our convergence proofs, we will also assume that f_i for $i = 0, \ldots, m$ are Lipschitz continuous with constant G.

We define the (maximum) violation at a point $x \in \Omega$ that satisfies $Ax = b$ as

$$v(x) = \max\{f_0(x) - f^\star, f_1(x), \ldots, f_m(x)\}, \tag{2}$$

and take $v(x) = \infty$ when $x \notin \Omega$ or $Ax \ne b$. The violation is zero if and only if x is a solution of (1). An algorithm solves the problem (1) if it produces a sequence x^k with $v(x^k) \to 0$.

1.2 Known Optimal Value

Like Polyak's original method, the algorithm we present in this paper assumes knowledge of $f^\star$. Although requiring that $f^\star$ is known before solving the problem is restrictive, there are common generic cases where it holds.

Feasibility problems. A feasibility problem is the special case of (1) with $f_0 = 0$,

$$\begin{array}{ll} \text{minimize} & 0 \\ \text{subject to} & f_i(x) \le 0, \quad i = 1, \ldots, m \\ & Ax = b, \end{array} \tag{3}$$

with variable $x \in \mathbf{R}^n$, and f_i, A, and b as in (1).

Primal–dual problems. A primal–dual problem includes both primal and dual variables and constraints, and includes the duality gap (the difference of the primal objective and the dual objective) as the objective or as a constraint (that it is zero). Such a problem has known objective value 0 when strong duality holds and the primal problem has a solution. As a specific example consider the primal and dual cone programs

$$\begin{array}{llll} \text{minimize} & c^T u & \text{maximize} & b^T v \\ \text{subject to} & Au = b & \text{subject to} & c - A^T v = s \\ & u \in \mathcal{K}, & & s \in \mathcal{K}^* \end{array} \tag{4}$$

with variables $u \in \mathbf{R}^n$, $s \in \mathbf{R}^n$, $v \in \mathbf{R}^p$, where $\mathcal{K} \subseteq \mathbf{R}^n$ is a closed convex cone and $\mathcal{K}^*$ is its dual cone, and $c \in \mathbf{R}^n$, $A \in \mathbf{R}^{p \times n}$, and $b \in \mathbf{R}^p$ are parameters. Expressing the condition that the duality gap $c^T u - b^T v$ is zero as as a linear constraint, we arrive at the primal–dual feasibility problem

$$\begin{array}{ll} \text{minimize} & 0 \\ \text{subject to} & d_{\mathcal{K}}(u) \le 0, \quad d_{\mathcal{K}^*}(s) \le 0 \\ & \begin{bmatrix} s \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 & -A^T & c \\ A & 0 & -b \\ -c^T & b^T & 0 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}, \end{array} \tag{5}$$

where $d_{\mathcal{K}}$ is the ℓ_2 distance to $\mathcal{K}$ and $d_{\mathcal{K}^*}$ is the distance to $\mathcal{K}^*$. This has the form (1) with variable $x = (u, v, s)$ and known optimal value $f^\star = 0$.

1.3 Pointwise Lower Bounds and Minorants

Pointwise lower bound. Suppose $f : \Omega \to \mathbf{R}$, with $\Omega \subseteq \mathbf{R}^n$. We say that $\hat{f} : \mathbf{R}^n \to \mathbf{R} \cup \{\infty\}$ is a pointwise lower bound (PLB) on f if $\hat{f}(z) \le f(z)$ for all $z \in \Omega$. We write this as $\hat{f} \le f$.

Minorant. We will use the basic idea of a minorant of a convex function at a point. Suppose $f : \Omega \to \mathbf{R}$ is a closed convex function. We say that $\hat{f} : \mathbf{R}^n \to \mathbf{R} \cup \{\infty\}$ is a minorant of f at $z \in \Omega$ if the following hold:

- $\hat{f}$ is closed convex;
- $\hat{f} \le f$, i.e., $\hat{f}$ is a PLB on f;
- $\hat{f}(z) = f(z)$, i.e., the lower bound is tight at the point z.

We write $\hat{f}(x)$ as $\hat{f}(x; z)$ to indicate that $\hat{f}$ is a minorant at the point z.

The simplest minorant of f at z is affine,

$$\hat{f}(x; z) = f(z) + g^T (x - z), \qquad g \in \partial f(z), \tag{6}$$

with $\partial f(z)$ denoting the subdifferential of f at z. At the other extreme, we can take f as its own minorant, $\hat{f}(x; z) = f(x)$. All other minorants are in between these two, in the sense that they are pointwise between f and at least one affine minorant defined by a subgradient. We will say more about minorants and how to construct them in Sect. 3.

If f has Lipschitz constant G, we will also assume that any minorant inherits the same Lipschitz constant. (Some minorants do not inherit the Lipschitz constant, but all the typical methods for constructing minorants do have this property.)

Other notation. Many authors have used different names for what we call a PLB and a minorant. Some papers define (what we call) a minorant to be (what we call) a

PLB [12, 42]. Other authors use minorant to be more restrictive, for example affine [25, 30]. In [21], a minorant is defined as the largest convex function whose epigraph contains a set of points. Throughout this paper, we use the definitions of PLB and minorant above.

1.4 This Paper

In this paper, we develop a method that solves (1) where the only access to the objective and constraint functions is via minorants. That is, we can find a minorant of f_i, $i = 1, \ldots, m$ at any z, and of f_0 for any $z \in \Omega$. Our method is inspired by and is an extension of the subgradient method with Polyak step size. As we will discuss below, it is closely connected to many other methods including subgradient methods, cutting-plane methods, and bundle methods. Much like bundle and cutting-plane methods, each iteration of our method requires solving a relatively simple convex optimization problem. One benefit of the method is that it has no parameters that need to be tuned. To honor Boris Polyak, we name the method the Polyak minorant method (PMM).

1.5 Prior Work

Subgradient methods. Boris Polyak introduced the Polyak gradient step for minimizing continuously differentiable functionals in [38]. Following the development of subgradient methods for non-differentiable optimization [43, 46], Polyak adapted his gradient step to the subgradient case [41]. Since then various extensions of the Polyak subgradient method have been developed, including Polyak step variations for stochastic gradient descent [1, 7, 14, 27, 28, 35, 37], Polyak-like steps that do not require knowledge of $f^\star$ [19, 51], a Polyak-like step for momentum-accelerated gradient descent [4, 18, 49], a Polyak step method for mirror descent [50], a Polyak-like step for convex problems with box constraints [9], and a Polyak step for weakly convex functions [11]. We will see that PMM reduces to the subgradient method with Polyak step sizes when there are no constraints and a subgradient-based affine minorant is used.

Cutting-plane methods. Cutting-plane methods originate from the works [8] and [22]. These methods solve convex problems by iteratively shrinking a polygonal superset of the optimal set. This shrinking is done using a cutting plane in each iteration, i.e., a halfspace known to contain the optimal set [6]. Cutting-plane methods differ in how they choose the next iterate; see, e.g., the survey [47]. Cutting-plane methods can be viewed as iteratively refining a piecewise-affine minorant on the objective and constraints. Conversely, PMM can be thought of as a cutting-plane method, when the minorants are piecewise affine.

Bundle methods. Bundle methods extend cutting-plane methods in two ways. First, the next iterate is found by minimizing a minorant plus an additional (typically quadratic) stabilization term. Second, bundle methods include logic that only updates the current iterate if a sufficient descent condition holds. The original and most common bundle method is the proximal bundle method [13, 23]. Alternatively, the level bundle method [13, 26] projects the current point onto the sublevel set of a minorant, exactly as PMM does. Yet another variation is the trust-region bundle method [13, 31]. A history of bundle methods can be found in [20, Ch. XIV, XV]. We refer to [42] for a more thorough review of modern bundle literature. PMM is structurally similar to a level-set bundle method, but lacks a sufficient descent condition, and admits any minorant instead of only cutting-plane minorants.

2 Polyak Minorant Method

We let $x^k \in \mathbf{R}^n$ denote the kth iterate of PMM for $k = 1, 2, \ldots$. PMM maintains a PLB on the objective and constraint functions, which vary with iterations, denoted $\hat{f}_i^k, i = 0, \ldots, m$. The constraint function lower bounds $\hat{f}_i^k, i = 1, \ldots, m$, are always minorants of f_i at x^k, while the PLB $\hat{f}_0^k$ is a minorant only for some iterations. We define

$$\mathcal{X}^k = \{x \mid \hat{f}_0^k(x) \le f^\star,\ \hat{f}_i^k(x) \le 0,\ i = 1, \ldots, m,\ Ax = b\}. \tag{7}$$

Since $\hat{f}_i^k \le f_i$ for $i = 1, \ldots, m$ and $\hat{f}_0^k \le f_0$, we have that $\hat{f}_i^k(x^\star) \le f_i(x^\star) \le 0$ for $i = 1, \ldots, m$ and $\hat{f}_0^k(x^\star) \le f_0(x^\star) = f^\star$, so $x^\star \in \mathcal{X}^k$.

The next iterate x^{k+1} is the projection of x^k onto $\mathcal{X}_k$, i.e.,

$$x^{k+1} = \Pi_{\mathcal{X}^k}(x^k) = \underset{x \in \mathcal{X}^k}{\operatorname{argmin}} \|x - x^k\|_2.$$

PMM is summarized in algorithm 2.1.

Algorithm 2.1 Polyak minorant method

given $x^1 \in \mathbf{R}^n$, optimal value $f^\star$, tolerance $\epsilon > 0$.

for $k = 1, 2, \ldots$

1. *Constraint minorants.* Find minorants $\hat{f}_i^k$ of f_i at x^k for $i = 1, \ldots, m$.
2. *Objective minorant.*
   If $x^k \in \Omega$, find a minorant $\hat{f}_0^k$ of f_0 at x^k.
   Else, set $\hat{f}_0^k$ to be any PLB on f_0.
3. *Update.* $x^{k+1} = \Pi_{\mathcal{X}^k}(x^k)$.
4. *Check stopping criterion.* Stop if $v(x^{k+1}) \le \epsilon$.

Comments. Recall that $\mathbf{dom}\, f_i = \mathbf{R}^n$ for $i = 1, \ldots, m$, so we can always find constraint minorants in step 1. In step 2, if $x^k \in \Omega = \mathbf{dom}\, f_0$, $\hat{f}_0^k$ is a minorant of f_0 at x^k. If $x^k \notin \Omega$, $\hat{f}_0^k$ is any PLB, such as the constant function $\hat{f}_0^k = f^\star$, or a minorant of f_0 found in any previous iteration.

The PMM method is generic since we have not specified what minorants to use. We will discuss many different types of minorants in Sect. 3. Depending on the minorants used, PMM can be considered a subgradient-type method, a cutting-plane method, or a level-set bundle method [48].

Connection to proximal operator. Define

$$\hat{F}^k(x) = \hat{f}_0(x) + \mathcal{I}(\hat{f}_i(x) \leq 0,\ i = 1, \ldots, m,\ Ax = b),$$

where $\mathcal{I}$ is the $\{0, \infty\}$-indicator function. The projection of x^k onto $\mathcal{X}^k$ also minimizes $t^k \hat{F}^k(x) + (1/2)\|x - x^k\|_2^2$ for some $t^k > 0$. In other words, we have $x^{k+1} = \mathbf{prox}_{t^k \hat{F}^k}(x^k)$, where $\mathbf{prox}$ is the proximal operator [36]. In standard proximal operator methods, t^k is specified. In our case, however, x^{k+1} is given as a projection, and we determine t^k only after the update is computed.

Polyak subgradient method. Consider the special case $\Omega = \mathbf{R}^n$ and $m = p = 0$. We use the subgradient-based affine minorant (6) for f_0. The set $\mathcal{X}^k$ is the halfspace $\{x \mid f_0(x^k) + (g^k)^T(x - x^k) \leq f^\star\}$, where $g^k \in \partial f_0(x^k)$. The projection in step 3 is then

$$x^{k+1} = x^k - \frac{f_0(x^k) - f^\star}{\|g^k\|_2^2} g^k,$$

which coincides with the subgradient method with Polyak's step size. (This assumes $g^k \neq 0$; if $g^k = 0$, we can terminate since x^k is optimal.) So PMM generalizes the subgradient method with Polyak step size.

Alternating-update PMM. We mention one simple variation of PMM in which the projection is replaced with a projection onto the objective sublevel set or a projection onto the constraint minorant sublevel set. We define

$$\begin{aligned}
\mathcal{X}_0^k &= \{x \mid \hat{f}_0^k(x) \leq f^\star\}, \\
\mathcal{X}_1^k &= \{x \mid \hat{f}_i^k(x) \leq 0,\ i = 1, \ldots, m,\ \ Ax = b\}.
\end{aligned}$$

We replace the projection in step 3 of PMM 2.1 with projection onto $\mathcal{X}_0^k$ for even k and $\mathcal{X}_1^k$ for odd k. This alternating-update variation is very similar to existing alternating-update subgradient methods for constrained convex optimization [2, 29, 39]. This modified PMM converges under the same assumptions as the original PMM; we provide a brief proof of convergence in Sect. 2.2.

2.1 Convergence Proof

Here we give a short convergence proof for PMM, i.e., we show that $v(x^k) \to 0$ as $k \to \infty$, which implies that the stopping criterion is eventually satisfied. (A very similar proof can be constructed to show that the alternating-update version also converges.) We give the proof not because it is novel, but because it is short and simple. It uses basic convex analysis and standard ideas that trace back to the subgradient methods of the 1960 s.

Since x^{k+1} is the projection of x^k onto $\mathcal{X}^k$ (which contains $x^\star$), we have

$$(x^k - x^{k+1})^T(x^\star - x^{k+1}) \le 0.$$

It follows that

$$\begin{aligned}\|x^{k+1} - x^\star\|_2^2 &= \|x^k - x^\star\|_2^2 - \|x^k - x^{k+1}\|_2^2 + 2(x^k - x^{k+1})^T(x^\star - x^{k+1}) \\ &\le \|x^k - x^\star\|_2^2 - \|x^k - x^{k+1}\|_2^2.\end{aligned}$$

This shows that the algorithm is Fejér monotone, i.e., each iteration does not increase the distance to any optimal point. Iterating the inequality above yields

$$\sum_{k=1}^{\infty} \|x^k - x^{k+1}\|_2^2 \le \|x^1 - x^\star\|_2^2,$$

which shows that

$$\|x^k - x^{k+1}\|_2 \to 0. \tag{8}$$

We use the Lipschitz continuity of $\hat{f}_i^k$, the fact that $x^{k+1} \in \mathcal{X}^k$ (which implies $\hat{f}_i^k(x^{k+1}) \le 0$), and that $\hat{f}_i^k$ is a minorant of f_i at x^k to conclude that

$$G\|x^{k+1} - x^k\|_2 \ge \hat{f}_i^k(x^k) - \hat{f}_i^k(x^{k+1}) \ge f_i(x^k) \tag{9}$$

for $i = 1, \ldots, m$.

Combining (8) and (9), we see that $\max\{f_i(x^k), 0\} \to 0$ as $k \to \infty$, for $i = 1, \ldots, m$. In other words, the iterates are eventually almost feasible. It follows that there exists a K for which $f_i(x^k) \le \rho$, $i = 1, \ldots, m$, for all $k \ge K$, which implies $x^k \in \mathcal{F}_\rho$, and thus $x^k \in \Omega$. This in turn implies that $\hat{f}_0^k$ is a minorant of f_0 at x^k for $k \ge K$.

We now show that $f_0(x^k) \to f^\star$. For $k \ge K$, a similar argument as above for f_i, $i = 1, \ldots, m$, gives

$$G\|x^{k+1} - x^k\|_2 \ge \hat{f}_0^k(x^k) - \hat{f}_0^k(x^{k+1}) \ge \hat{f}_0(x^k) - f^\star = f_0(x^k) - f^\star. \tag{10}$$

(Here we use $\hat{f}_0(x^{k+1}) \le f^\star$ and $\hat{f}_0(x^k) = f_0(x^k)$.) Combined with (8), we deduce that $f_0(x^k) \to f^\star$. It follows that $v(x^k) \to 0$, so the stopping criterion is eventually satisfied.

Comments. The convergence proof shows that we can relax several assumptions. For example, the assumption of Lipschitz continuity can be relaxed by requiring it to hold for the minorants, and only on the bounded set defined by $\|x - x^1\|_2 \le \|x^\star - x^1\|_2$. (We do not know the righthand side here, but we never use the Lipschitz constant in the algorithm.)

While we have assumed above that all constraint functions have domain $\mathbf{R}^n$, our proof shows that it suffices for just one to be defined on all $\mathbf{R}^n$, with the other constraint functions playing a similar role to the objective, i.e., they are defined only when the one constraint is nearly satisfied. Instead of a minorant, we take $\hat{f}_i$ to be any PLB for f_i for the constraints with $x^k \notin \textbf{dom}\, f_i$, as we do above for the objective.

2.2 Convergence with Alternating Update

We follow the steps of the proof in Sect. 2.1 to show that $f_0(x^k) \to f^\star$ as $k \to \infty$ for the alternating-update variant of the PMM. By construction, $Ax^k = b$. Since $x^\star \in X_0^k$ and $x^\star \in X_1^k$, we deduce that for both odd and even k,

$$(x^k - x^{k+1})^T(x^\star - x^{k+1}) \le 0.$$

It follows that

$$\|x^{k+1} - x^\star\|_2^2 \le \|x^k - x^\star\|_2^2 - \|x^k - x^{k+1}\|_2^2.$$

Iterating these inequalities yields that

$$\sum_{k=1}^{\infty} \|x^k - x^{k+1}\|_2^2 \le \|x^1 - x^\star\|_2^2.$$

It now follows that $\|x^k - x^{k+1/2}\|_2 \to 0$. We conclude using (9), (10), and the Lipschitz continuity of $\hat{f}_i$ for $i = 0, \dots m$ that $f_0(x^k) \to f^\star$ as $k \to \infty$.

2.3 Convergence with Sharpness

A convex function ϕ on Ω, with $\phi^\star = \inf_{x \in \Omega} \phi(x)$ finite, is μ-sharp if for some $\mu > 0$ and all $x \in \Omega$

$$\phi(x) - \phi^\star \ge \mu \inf_{x^\star \in \mathcal{X}^\star} \|x - x^\star\|_2.$$

It is a well-known result (see [11, 17, 40, 45]) that for minimizing a G-Lipschitz μ-sharp closed proper convex function, the original Polyak subgradient step produces iterates that converge linearly:

$$\inf_{x^\star \in \{x | \phi(x) \le \phi^\star\}} \|x^{k+1} - x^\star\|_2 \le \sqrt{1 - \frac{\mu^2}{G^2}} \inf_{x^\star \in \{x | f(x) \le f^\star\}} \|x^k - x^\star\|_2.$$

We can derive a similar convergence result for PMM. Define

$$h(x) = \max(f(x) - f^\star, f_1(x), \dots, f_m(x))$$

on Ω. The function h is G-Lipschitz, closed, proper, and convex, and its set of minimizers is $\mathcal{X}^\star$. Now suppose that h is μ-sharp. From (9), (10), and sharpness, we have

$$G\|x^{k+1} - x^k\|_2 \geq h(x^k) \geq \mu \inf_{x^\star \in \mathcal{X}^\star} \|x^k - x^\star\|_2.$$

Letting $\Pi_{\mathcal{X}^\star} x^k$ denote the projection of x^k onto $\mathcal{X}^\star$, we have

$$\begin{aligned}\inf_{x^\star \in \mathcal{X}^\star} \|x^{k+1} - x^\star\|_2^2 &\leq \|x^{k+1} - \Pi_{\mathcal{X}^\star} x^k\|_2^2 \\ &\leq \|x^k - \Pi_{\mathcal{X}^\star} x^k\|_2^2 - \frac{\mu^2}{G^2} \inf_{x^\star \in \mathcal{X}^\star} \|x^k - x^\star\|_2^2 \\ &= \left(1 - \frac{\mu^2}{G^2}\right) \inf_{x^\star \in \mathcal{X}^\star} \|x^k - x^\star\|_2^2.\end{aligned}$$

Thus when the function h is μ-sharp, PMM attains the same linear rate of convergence as the original Polyak subgradient method.

2.4 Cost of an Iteration

We address here a question that could just as well be asked about cutting-plane or bundle methods: What is the computational cost of an iteration of PMM, specifically the projection step? Of course, this depends very much on the minorants used. For example, if we take the functions themselves as minorants, PMM converges in one step, which consists of solving the problem; PMM is correct in this case, but silly. To be useful, carrying out the projection step should be, at a very minimum, cheaper than solving the original problem.

Typical minorants are piecewise affine, defined as the maximum of a set of affine functions. For such problems, the projection can be solved in time that is linear in n, and quadratic in the number of terms in the minorants plus equality constraints. In typical cases the latter number is kept substantially smaller than n as the algorithm proceeds, using limited memory minorants described in Sect. 3.4.

While the details depend on the specific form of the minorants, we give them here for a specific generic case, where the projection can be expressed as

$$\begin{array}{ll}\text{minimize} & \|x - x^k\|_2^2 \\ \text{subject to} & Fx \leq g, \quad Ax = b,\end{array} \tag{11}$$

where $F \in \mathbf{R}^{q \times n}$ and $g \in \mathbf{R}^q$, where q is the number of terms in the piecewise affine minorants. We assume here that $q \ll n$, and show how to solve this problem efficiently.

From the optimality condition for this quadratic program (QP), we find that the solution to (11) has the form

$$x^{k+1} = x^k - F^T \lambda - A^T \nu,$$

for some dual variables $\lambda \in \mathbf{R}^q$ and $\nu \in \mathbf{R}^p$, with $\lambda \geq 0$ [5, §5.5.3]. So we can reformulate (11) using variables λ and ν as

$$\begin{array}{ll} \text{minimize} & \|F^T\lambda + A^T\nu\|_2^2 \\ \text{subject to} & Fx^k - FF^T\lambda - FA^T\nu \leq g \\ & Ax^k - AF^T\lambda - AA^T\nu = b. \end{array} \tag{12}$$

This a QP with variables $(\lambda, \nu) \in \mathbf{R}^{q+p}$. We solve this (smaller) QP and then set $x^{k+1} = x^k - F^T\lambda^\star - A^T\nu^\star$. When $q + p \ll n$, this has far fewer variables than the original projection QP 11.

Without exploiting any structure, the small QP (12) can be solved in $O((p+q)^3)$ flops [5, §11]. In many cases the computational cost of the projection is dominated by forming the matrices

$$FF^T, \quad FA^T, \quad AA^T, \tag{13}$$

which has cost $O(n(p+q)^2)$ flops.

To illustrate this, we consider a specific example with $n = 10^6$ and $p = q = 50$. Computing the matrices (13) costs $O(10^{10})$ flops, whereas solving the small QP (12) costs $O(10^6)$ flops, which is negligible in comparison. Carrying out this computation using CVXPY [10] and OSQP [44] on an instance of this problem yields results that are compatible with these rough flop counts. Computing the matrices requires around 0.3 s, and solving the small QP requires 0.007 s. In comparison, solving the original QP directly takes around 700 s. (These numbers are for a laptop with a Ryzen 9 5900HX processor.)

Similar methods to efficiently compute the projection can be used for minorants of a more general form, i.e., not the maximum of q affine functions. As specific examples, the minorants could be the sum of terms, each a maximum of affine functions, second-order cone representable, with q being something like the total number of terms involved. The main point here is while each iteration of PMM requires solving a (possibly large) convex optimization problem that does not have an analytical solution, this can be done efficiently provided q is not too big.

3 Minorants

In this section, we look at methods for constructing minorants. We have already mentioned some simple minorants, such as the affine subgradient-based minorant (6) and the function itself. We start by mentioning simple minorants for functions that satisfy additional conditions, such as strongly convex or self-concordant [34]. We then give some rules for constructing minorants, which can be extended to an automated method that relies on disciplined convex programming [15].

3.1 Strongly Convex and Self-Concordant Functions

Strongly convex functions. The subgradient-based minorant (6) can be replaced with a quadratic minorant when f is strongly convex with parameter $\delta > 0$, i.e., $f(x) - (\delta/2)\|x\|_2^2$ is convex. Here the minorant is

$$\hat{f}(x; z) = f(z) + g^T(x - z) + (\delta/2)\|x - z\|_2^2,$$

where $g \in \partial f(z)$.

Self-concordant functions. As another example, suppose f is self-concordant [32, §2.2.VI]. In this case, we have the minorant

$$\hat{f}(x; z) = f(z) + \nabla f(z)^T(x - z) + u - \log(1 + u),$$

where $u = \|\nabla^2 f(z)^{1/2}(x - z)\|_2$. (This is convex since $u - \log(1 + u)$ is increasing on $u \geq 0$.)

3.2 Rules for Constructing Minorants

Scaling and sum. If $\hat{f}_i(x; z)$ is a minorant for f_i at z and $\alpha_i \geq 0$, then $\sum_{i=1}^M \hat{f}_i(x; z)$ is a minorant for $\sum_{i=1}^M f_i$ at z. As an example, if each minorant $\hat{f}_i(x; z)$ is the maximum of affine functions, then the minorant for the sum is also piecewise affine, with the specific form of a sum of functions, each the maximum of affine functions.

Selective minorization. As a specific example, suppose

$$f(x) = l(x) + \lambda r(x),$$

where l is a convex (loss) function, r is convex (regularizer) function, and $\lambda > 0$ is a parameter [5, §6.3.2], [33, §6.4.1]. (This function arises in regularized empirical risk minimization problems.) We can use the minorant

$$\hat{f}(x; z) = \hat{l}(x; z) + \lambda r(x),$$

where $\hat{l}(x; z)$ is a minorant of l at z. Here we form a minorant of the loss function, but keep the regularizer (which is its own minorant).

Supremum. Suppose f is the pointwise supremum of a set of convex functions,

$$f(x) = \sup_{\alpha \in \mathcal{A}} f_\alpha(x),$$

where $f_\alpha : \Omega \to \mathbf{R}$ are convex. We will assume that the supremum is achieved for each x. (When f is the objective, the problem (1) is then a minimax problem [5].) We can construct a minorant as follows. Find $\alpha' \in \mathcal{A}$ with $f(z) = f_{\alpha'}(z)$. Then

$$\hat{f}(x; z) = f_{\alpha'}(x)$$

is a minorant. We can generalize this in several ways. We can replace the righthand side with a minorant of $f_{\alpha'}$ at z. We can form a (pointwise) larger minorant as the maximum over multiple values of α, as long as one of them maximizes $f_\alpha(z)$.

Maximum eigenvalue. As a specific example, we consider $f : \mathbf{S}^m \to \mathbf{R}$ defined as

$$f(X) = \lambda_{\max}(X) = \sup_{\|u\|_2=1} u^T X u, \tag{14}$$

where $X \in \mathbf{S}^m$, the set of symmetric $m \times m$ matrices. Using the method above, we find a minorant at $Z \in \mathbf{S}^m$ by finding an eigenvector v of Z, with $\|v\|_2 = 1$, associated with its maximum eigenvalue [24]. This gives the minorant

$$\hat{f}(X; Z) = v^T X v,$$

which is linear. (This coincides with the minorant constructed from the subgradient vv^T of f at Z.) For more sophisticated (and larger) minorants, we can take

$$\hat{f}(X; Z) = \lambda_{\max}(V^T X V), \tag{15}$$

or

$$\hat{f}(X; Z) = \max \mathbf{diag}(V^T X V), \tag{16}$$

where V is an $m \times r$ matrix whose columns are orthonormal eigenvectors associated with the top r eigenvalues of Z and **diag** gives the diagonal entries of a matrix. These minorants reduce to (14) when $r = 1$. For $r \geq 2$, (15) and (16) are not equivalent, and when $r = 2$, (15) is second-order cone representable. The minorant (16) is always piecewise linear.

3.3 DCP Expressions

Disciplined convex programming (DCP) is a system for constructing expressions with known curvature, convex, concave, or affine [15]. Expressions are built from a library of atomic or basic functions with known sign, monotonicity, and curvature. These are combined in such a way that a composition rule, sufficient to establish convexity or concavity, holds for each subexpression. The leaves of the expression tree are constants or variables. In addition to curvature of subexpressions, we can also track sign and monotonicity. Roughly speaking, we determine the signs of the zeroth, first, and second derivatives of all subexpressions using the composition rule.

DCP composition rule. Consider the expression given by

$$\psi_0 = \phi(\psi_1, \ldots, \psi_k),$$

where ϕ is an atom and $\psi_1, \ldots, \psi_k$ are expressions. The composition rule for convexity is: ψ_0 is convex provided ϕ is convex and for each $i = 1, \ldots, k$, one of the following holds:

1. ψ_i is affine,
2. ψ_i is convex and ϕ is non-decreasing in argument i,
3. ψ_i is concave and ϕ is non-increasing in argument i.

One subtlety is that the monotonicity in conditions 2 and 3 must be of the extended-valued function; see [5, §3.2.4]. There is a similar rule for concavity.

An expression is called DCP (or DCP-compliant) if every subexpression satisfies the composition rule. DCP is a sufficient condition for convexity or concavity. We can think of a DCP-convex expression as a function that is syntactically convex, since its convexity can be established by recursively applying the composition rule. Note that it only requires knowing the sign, monotonicity, and curvature of the atoms used, and not their specific values.

Minorant for DCP expression. Suppose ψ is DCP-convex. We can construct a minorant for it by replacing every atom in it with a minorant at its argument, provided the minorants satisfy two additional conditions: The minorant preserves the sign and the monotonicity of the atom. These properties do not hold for general minorants, but they do when the minorants are constructed from subgradients or the other methods described above. Handling the sign requirement is easy; if ϕ is an atom that is known to be nonnegative, we replace any minorant $\hat{\phi}(x; z)$ with $\max\{\hat{\phi}(x; z), 0\}$, a minorant that is also nonnegative.

We omit the proof that this simple method produces a minorant of ϕ, since it uses standard arguments used in DCP. We note that the sum, scaling, and maximum rule above (i.e., the supremum rule when $\mathcal{A}$ is finite) are special cases of DCP-constructed minorants.

3.4 Memory-Based Minorants

Suppose we have found minorants $\hat{f}^i$ of f for iterations $i = 1, \ldots, k$. Then their pointwise maximum

$$\max\{\hat{f}^1, \ldots, \hat{f}^k\} \tag{17}$$

is also a minorant, because $\hat{f}^1, \ldots, \hat{f}^{k-1}$ are PLBs, and $\hat{f}^k$ is a minorant. The minorant (17) is pointwise larger than $\hat{f}^k$. We say that the minorant (17) has *memory* (of the previously found minorants). We can also limit the memory to, say, the last M minorants, as

$$\max\{\hat{f}^{k-M}, \ldots, \hat{f}^k\}. \tag{18}$$

We refer to this as a limited-memory or finite-memory minorant.

When the minorants in each iteration are affine, we can express the minorant (17) as

$$\hat{f}^k(x) = \max_{i=1,\ldots,k} \left(f(x^i) + (g^i)^T (x - x^i) \right), \tag{19}$$

where $g^i \in \partial f(x^i)$. These minorants are piecewise affine and require only the evaluation of a subgradient of f, and its value, at each point. When these minorants are used, PMM looks very much like a cutting-plane or bundle method.

4 Numerical Experiments

We present two numerical experiments to illustrate the PMM. Python code for these experiments is available as a Jupyter notebook at

https://github.com/cvxgrp/polyak_minorant

The data for both problems was generated with a seeded pseudorandom number generator, so our numerical experiments can be reproduced exactly.

We used CVXPY [10] with the SOCP solver Clarabel [16] to compute the PMM updates. This incurs some inefficiency, but our goal is only to illustrate how PMM works with various minorants, with a trade-off of per-iteration cost and overall iteration cost. The numerical examples were run on a laptop with Ryzen 9 5900HX processor and 32 GB of DDDR4 memory.

4.1 Second-Order Cone Program

We consider an instance of the primal–dual cone program, given in (4) and (5), with

$$\mathcal{K} = \mathcal{K}_1 \times \cdots \times \mathcal{K}_l,$$

where $\mathcal{K}_i = \{(s_i, t_i) \mid \|s\|_2 \leq t\}$ are second-order cones. These cones are self-dual, i.e., $\mathcal{K}_i^* = \mathcal{K}_i$. In (5) we list the cone distances separately as

$$d_{\mathcal{K}_i}(x_i) \leq 0, \quad i = 1, \ldots, l,$$

and similarly for the dual cone constraints. In the form (1), we have $m = 2l$ inequality constraint functions. There is an analytical expression for the projection onto a second-order cone, and from this, we can derive an analytical expression for a subgradient of $d_{\mathcal{K}_i}$.

Data generation. We generate an instance of (4) with $n = 500$ variables, $p = 200$ equality constraints, and $l = 10$ cones each of dimension 50.

We generate the data A, b, and c as follows. We first generate a vector $z \in \mathbf{R}^n$ with standard normal entries. We project z onto $\mathcal{K}$ to obtain u, and we set $s = u - z$, which guarantees that $s \in \mathcal{K}^* = \mathcal{K}$ [43][§14]. These two vectors satisfy $s^T u = 0$, i.e., they are complementary with respect to $\mathcal{K}$. Then we generate $A \in \mathbf{R}^{p \times n}$ and $v \in \mathbf{R}^p$ with standard normal entries. We set $b = Au$ and $c = s + A^T v$. The zero-gap equality constraint $c^T u = b^T v$ holds for this data. For our experiment we only use the data A, b, and c, and not the primal–dual solution (u, v, s).

Minorant construction and initial point. We use a basic subgradient minorant for $d_{\mathcal{K}}$ and $d_{\mathcal{K}^*}$, and explore different memories M. Our initial point is $x^1 = 0$.

Results Figure 1 shows the maximum violation $v(x^k)$ versus k, the number of iterations, for memory values $M = 0, 5, 20, 100$. Not surprisingly we get a strong speedup with $M = 20$ and $M = 100$ compared to smaller memory.

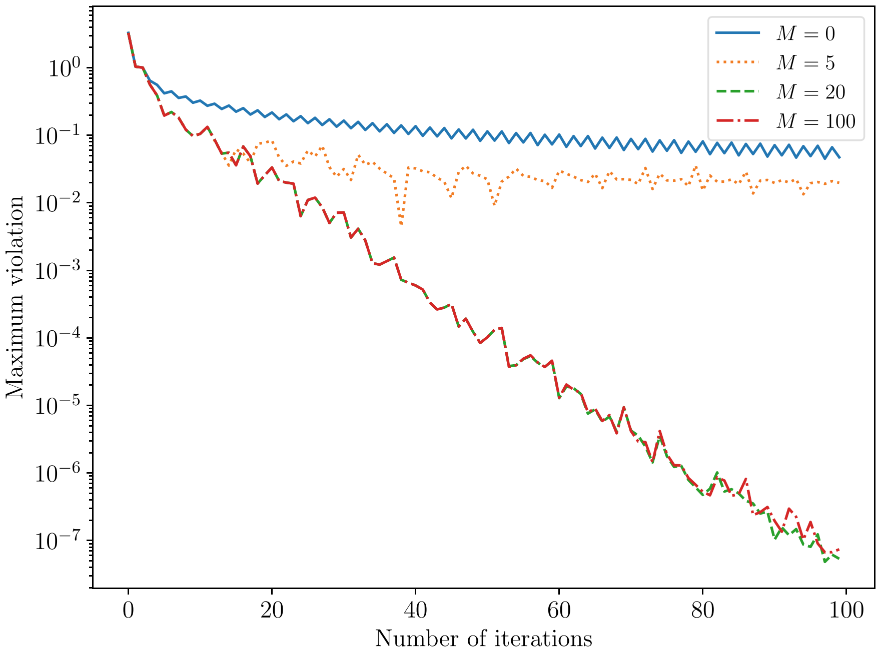

Fig. 1 Maximum violation versus iterations for primal–dual cone problem

Figure 2 shows the maximum violation $v(x^k)$ versus the total elapsed time, which takes into account the varying complexity of the subproblems solved in each iteration. Here we see a clear best value of memory $M = 20$.

4.2 Linear Matrix Inequality

Our second example is a linear matrix inequality (LMI). The goal is to find a matrix $X \in \mathbf{S}^q$ that satisfies

$$X \geq I, \quad A_i^T X + X A_i \leq 0, \quad i = 1, \ldots, k,$$

where the inequalities are with respect to the positive semidefinite cone, and $A_i \in \mathbf{R}^{q \times q}$, $i = 1, \ldots, m$, are given data. Problems of this form arise in the analysis of control systems; see, e.g., [3].

We express this as the feasibility problem

$$\begin{array}{ll} \text{minimize} & 0 \\ \text{subject to} & \lambda_{\max}(I - X) \leq 0 \\ & \lambda_{\max}(A_i^T X + X A_i) \leq 0, \quad i = 1, \ldots, k. \end{array} \tag{20}$$

The variable is $X \in \mathbf{S}^q$, which has dimension $n = q(q+1)/2$. There are no equality constraints and $m = k + 1$ inequality constraints.

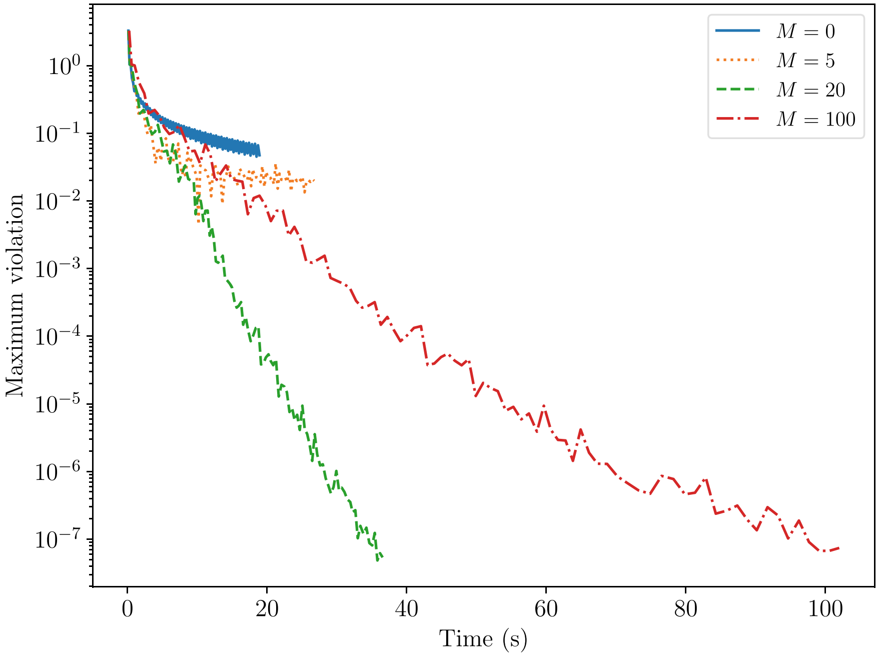

Fig. 2 Maximum violation versus time for primal–dual cone problem

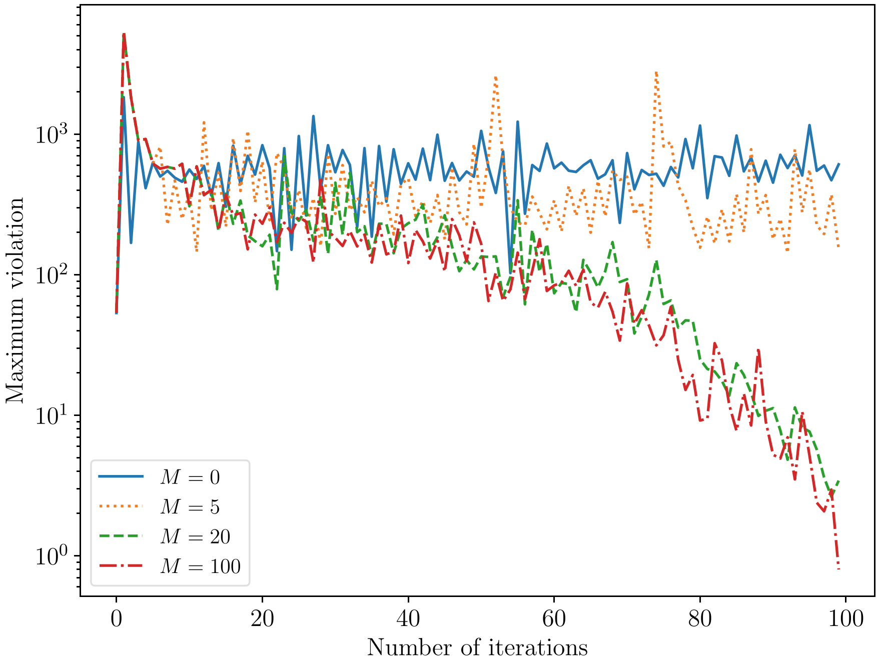

Fig. 3 Maximum violation versus iterations for LMI problem

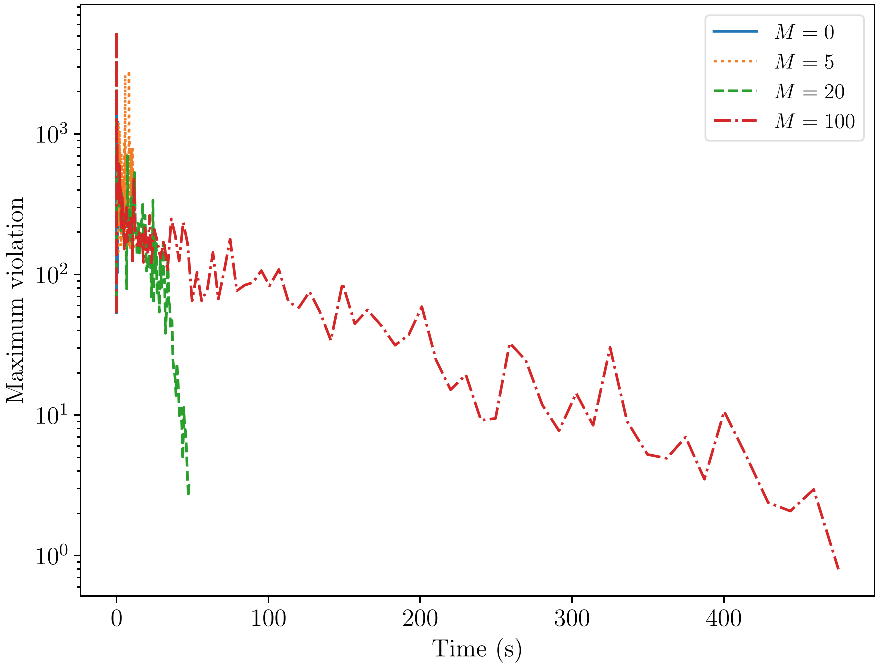

Fig. 4 Maximum violation versus time for primal–dual cone problem

Data generation. We set $q = 20$ and $k = 10$, so $n = 210$ and $m = 11$. To generate A_i we proceed as follows. First, we generate

$$\tilde{A}_i = -B_i B_i^T + C_i - C_i^T,$$

where the entries of the $q \times q$ matrices B_i and C_i are standard normals. This means that, with probability one,

$$\tilde{A}_i^T + \tilde{A}_i \preceq 0,$$

i.e., they satisfy the constraints $\tilde{A}_i^T X + X\tilde{A}_i \preceq 0$ with $X = I$. Now we generate a $q \times q$ matrix F with standard normal entries, so F is invertible with probability one, and form

$$A_i = F^{-1}\tilde{A}_i F.$$

Then $X = F^T F$ satisfies $A_i^T X + X A_i \preceq 0$, $i = 1, \ldots, k$. Since $X \succ 0$ (with probability one) we can scale it to obtain a solution of the LMI (20). For our experiment, we only use the data A_i, and not the solution X.

Minorant construction and initial point. Each constraint is the composition of a linear function with $\lambda_{\max}$. For $\lambda_{\max}$ we use the minorant (15), with dimension 2, so the minorants are second-order cone representable, and the projection can be computed using a SOCP solver. We run PMM with memories $M = 0, 5, 20, 100$.

Results Figure 3 shows the maximum violation $v(x^k)$ versus k, the number of iterations, for memory values $M = 0, 5, 20, 100$, and Fig. 4 shows the maximum violation versus elapsed time. The results are very similar to the previous example, with memory $M = 20$ giving the fastest (in time) convergence.

Acknowledgements This paper builds on notes written around 2010 for the Stanford course EE364B, *Convex Optimization II*, to which Lieven Vandenberghe, Almir Mutapcic, Jaehyun Park, Lin Xiao, and Jacob Mattingley contributed. We thank Tetiana Parshakova, Fangzhao Zhang, Parth Nobel, Logan Bell, and Thomas Schmeltzer for useful discussions. The authors thank an anonymous reviewer for suggesting the sharpness convergence result, as well as pointing us to some very relevant literature that we had missed in an earlier version of this paper. Stephen Boyd would like to dedicate this paper to Boris Polyak, his hero and friend.

Funding Funding was provided by ACCESS (AI Chip Center for Emerging Smart Systems) and Office of Naval Research (Grant No. N00014-22-1-2121).

References

1. Abdukhakimov, F., Xiang, C., Kamzolov, D., Takáč, M.: Stochastic gradient descent with preconditioned Polyak step-size. https://arxiv.org/abs/2310.02093 (2023)
2. Beck, A., Ben-Tal, A., Guttmann-Beck, N., Tetruashvili, L.: The comirror algorithm for solving nonsmooth constrained convex problems. Oper. Res. Lett. **38**(6), 493–498 (2010)
3. Boyd, S., El Ghaoui, L., Feron, E., Balakrishnan, V.: Linear Matrix Inequalities in System and Control Theory. Society for Industrial and Applied Mathematics, Philadelphia (1994)
4. Barré, M., Taylor, A., d'Aspremont, A.: Complexity guarantees for Polyak steps with momentum. In: Proceedings of Thirty Third Conference on Learning Theory, vol. 125, pp. 452–478. PMLR, 09-12 Jul 2020
5. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press, Cambridge (2004)
6. Boyd, S., Vandenberghe, L.: Localization and cutting-plane methods. Lecture notes for EE364b. Stanford University (2007)
7. Berrada, L., Zisserman, A., Kumar, M.: Comment on stochastic Polyak step-size: Performance of ALI-G. https://arxiv.org/abs/2105.10011 (2021)
8. Cheney, E., Goldstein, A.: Newton's method for convex programming and Tchebycheff approximation. Numer. Math. **1**, 253–268 (1959)
9. Cheng, W., Li, D.: An active set modified Polak–Ribiére–Polyak method for large-scale nonlinear bound constrained optimization. J. Optim. Theory Appl. **155**(3), 1084–1094 (2012)
10. Diamond, S., Boyd, S.: CVXPY: a Python-embedded modeling language for convex optimization. J. Mach. Learn. Res. **17**(83), 1–5 (2016)
11. Davis, D., Drusvyatskiy, D., MacPhee, K., Paquette, C.: Subgradient methods for sharp weakly convex functions. J. Optim. Theory Appl. **179**(3), 962–982 (2018)
12. Drusvyatskiy, D.: Convex analysis and nonsmooth optimization. https://sites.math.washington.edu/~ddrusv/crs/Math_516_2020/bookwithindex.pdf (2020)
13. Frangioni, A.: Standard bundle methods: Untrusted models and duality. In: Numerical Nonsmooth Optimization, pp. 61–116. Springer (2020)
14. Gower, R., Blondel, M., Gazagnadou, N., Pedregosa, F.: Cutting some slack for SGD with adaptive Polyak stepsizes. https://arxiv.org/abs/2202.12328 (2022)
15. Grant, M., Boyd, S., Ye, Y.: Disciplined convex programming. In: Global Optimization, pp. 155–210. Springer (2006)
16. Goulart, P., Chen, Y.: Clarabel: A library for optimization and control (2021). https://oxfordcontrol.github.io/ClarabelDocs/stable/

17. Goffin, J.: On convergence rates of subgradient optimization methods. Math. Program. **13**(1), 329–347 (1977)
18. Goujaud, B., Taylor, A., Dieuleveut, A.: Quadratic minimization: from conjugate gradients to an adaptive heavy-ball method with Polyak step-sizes. In: OPT 2022: Optimization for Machine Learning (NeurIPS 2022 Workshop) (2022)
19. Hazan, E., Kakade, S.: Revisiting the Polyak step size. https://arxiv.org/abs/1905.00313 (2022)
20. Hiriart-Urruty, J., Lemaréchal, C.: Convex Analysis and Minimization Algorithms II: Advanced Theory and Bundle Methods. Grundlehren der mathematischen Wissenschaften. Springer, Berlin (1996)
21. Jongbloed, G.: The iterative convex minorant algorithm for nonparametric estimation. J. Comput. Graph. Stat. **7**(3), 310 (1998)
22. Kelley, J.: The cutting-plane method for solving convex programs. J. Soc. Ind. Appl. Math. **8**(4), 703–712 (1960)
23. Kiwiel, K.: Proximity control in bundle methods for convex nondifferentiable minimization. Math. Program. **46**(1–3), 105–122 (1990)
24. Kowalewski, G.: Einführung in die determinantentheorie einschliesslich der unendlichen und der Fredholmschen determinanten. Veit & comp. (1909)
25. Lin, Q., Ma, R., Yang, T.: Level-set methods for finite-sum constrained convex optimization. In: Proceedings of the 35th International Conference on Machine Learning, vol.80, pp. 3112–3121. PMLR, 10–15 Jul (2018)
26. Lemaréchal, C., Nemirovskii, A., Nesterov, Y.: New variants of bundle methods. Math. Program. **69**(1), 111–147 (1995)
27. Li, S., Swartworth, W., Takáč, M., Needell, D., Gower, R.: SP2: A second order stochastic Polyak method. In: The Eleventh International Conference on Learning Representations (2023)
28. Loizou, N., Vaswani, S., Hadj Laradji, I., Lacoste-Julien, S.: Stochastic Polyak step-size for SGD: an adaptive learning rate for fast convergence, 13–15 Apr 2021
29. Lan, G., Zhou, Z.: Algorithms for stochastic optimization with function or expectation constraints. Comput. Optim. Appl. **76**(2), 461–498 (2020)
30. McLinden, L.: Affine minorants minimizing the sum of convex functions. J. Optim. Theory Appl. **24**(4), 569–583 (1978)
31. Marsten, R., Hogan, W., Blankenship, J.: The boxstep method for large-scale optimization. Oper. Res. **23**(3), 389–405 (1975)
32. Nemirovski, A.: Interior point polynomial time methods in convex programming. https://www2.isye.gatech.edu/~nemirovs/Lect_IPM.pdf (1996)
33. Nesterov, Y.: Lectures on Convex Optimization. Springer, Berlin (2018)
34. Nesterov, Y., Nemirovskii, A.: Interior-Point Polynomial Algorithms in Convex Programming. Society for Industrial and Applied Mathematics, Philadelphia (1994)
35. Orvieto, A., Lacoste-Julien, S., Loizou, N.: Dynamics of SGD with stochastic polyak stepsizes: truly adaptive variants and convergence to exact solution. In: Advances in Neural Information Processing Systems (2022)
36. Parikh, N., Boyd, S.: Proximal algorithms. Found. Trends Optim. **1**(3), 127–239 (2014)
37. Prazeres, M., Oberman, A.: Stochastic gradient descent with Polyak's learning rate. J. Sci. Comput. **89**(1), 1–16 (2021)
38. Polyak, B.: Gradient methods for the minimisation of functionals. USSR Comput. Math. Math. Phys. **3**(4), 864–878 (1963)
39. Polyak, B.: A general method of solving extremum problems. Sov. Math. Dokl. **8**, 593–597 (1967)
40. Polyak, B.: Minimization of unsmooth functionals. USSR Comput. Math. Math. Phys. **9**(3), 14–29 (1969)
41. Polyak, B.: Introduction to Optimization. Optimization Software Inc., New York (1987)
42. Parshakova, T., Zhang, F., Boyd, S.: Implementation of an oracle-structured bundle method for distributed optimization. Optim. Eng. (2023). https://doi.org/10.1007/s11081-023-09859-z
43. Rockafellar, R.: The Theory of Subgradients and its Applications to Problems of Optimization. Heldermann Verlag, Berlin (1981)
44. Stellato, B., Banjac, G., Goulart, P., Bemporad, A., Boyd, S.: OSQP: an operator splitting solver for quadratic programs. Math. Program. Comput. **12**(4), 637–672 (2020)
45. Shor, N.: Convergence rate of the gradient descent method with dilatation of the space. Cybernetics **6**(2), 102–108 (1973)
46. Shor, N.: Minimization Methods for Non-differentiable Functions, vol. 3. Springer, Berlin (2012)

47. Sra, S., Nowozin, S., Wright, S.: Optimization for Machine Learning. MIT Press, Cambridge (2012)
48. van Ackooij, W., Frangioni, A., de Oliveira, W.: Inexact stabilized Benders' decomposition approaches with application to chance-constrained problems with finite support. Comput. Optim. Appl. **65**, 637–669 (2016)
49. Wang, X., Johansson, M., Zhang, T.: Generalized Polyak step size for first order optimization with momentum. https://arxiv.org/abs/2305.12939 (2023)
50. You, J., Cheng, H., Li, Y.: Minimizing quantum Rényi divergences via mirror descent with Polyak step size. In: 2022 IEEE International Symposium on Information Theory (ISIT), pp. 252–257 (2022)
51. You, J., Li, Y.: Two Polyak-type step sizes for mirror descent. https://arxiv.org/abs/2210.01532 (2022)