OPTIMAL STEEPEST DESCENT ALGORITHMS FOR UNCONSTRAINED CONVEX PROBLEMS: FINE TUNING NESTEROV'S METHOD

CLÓVIS C. GONZAGA[†] AND ELIZABETH W. KARAS[‡]

Abstract. We modify the first order algorithm for convex programming proposed by Nesterov [5]. The resulting algorithm keeps the optimal complexity obtained by Nesterov with no need of a known Lipschitz constant for the gradient, and performs better in practically all examples in a set of test problems.

1. Introduction.

We study the nonlinear programming problem

$$(P) \qquad \begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in \mathbb{R}^n, \end{array}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is convex and continuously differentiable, with a Lipschitz constant $L > 0$ for the gradient and a convexity parameter $\mu \geq 0$. It means that for all $x, y \in \mathbb{R}^n$,

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\| \tag{1.1}$$

and

$$f(x) \geq f(y) + \nabla f(y)^T (x - y) + \frac{1}{2}\mu\|x - y\|^2. \tag{1.2}$$

If $\mu > 0$, the function is said to be *strongly convex*. Note that $\mu \leq L$. The algorithms described in this paper use the convexity parameter, but do not assume that it is positive. But we do assume that the problem has an optimal solution. We do not assume that the Lipschitz constant is known, although its knowledge may be very helpful. In fact, our main algorithms do not need the assumption that $L < +\infty$, unless for the complexity analysis. Hence it can be used for instance in problems involving logarithmic barriers.

The most widely known method for solving this problem is the steepest descent algorithm, devised by Cauchy in the nineteenth century. It constructs a sequence (x^k) in which $x^{k+1} = x^k - \nu_k \nabla f(x^k)$. The step length must produce a large decrease of $f(\cdot)$. There are several methods for this line search, described in nonlinear programming textbooks. If L is known, then the constant step length $\nu_k = 1/L$ ensures the global convergence of the algorithm.

In the second half of last century there was much activity in the development of quasi-Newton algorithms, which iteratively construct matrices which in a certain sense approach the Hessian matrices $\nabla^2 f(x^k)$, to endow the methods with superlinear convergence. All these methods compute only first derivatives, so that each iteration is based on the accumulated first order information from the previous iterates. The main objective of these methods is to obtain high asymptotic convergence speeds.

[†] Department of Mathematics, Federal University of Santa Catarina, Florianópolis, SC, Brazil; e-mail: `clovis@mtm.ufsc.br`. Research done while visiting LIMOS – Université Blaise Pascal, Clermont- Ferrand, France. This author is supported by CNPq and PRONEX - Optimization.

[‡] Department of Mathematics, Federal University of Paraná, Curitiba, PR, Brazil; e-mail :`karas@mat.ufpr.br`. Research done while visiting the Tokyo Institute of Technology, Global Edge Institute. Supported by PRONEX - Optimization and MEXT's program.

The last quarter of the twentieth century saw the coming of computational complexity theory into the realm of continuous convex optimization. Linear and quadratic programming were revolutionized by the introduction of interior point methods, and for the first time methods for differentiable optimization had their practical efficiency motivated by complexity theory.

Complexity results for nonlinear programming are limited to convex problems, but they are impressive (see Nemirovsky and Yudin [3] and Shor [7]): no method for solving (P) based on accumulated first order information can achieve an iteration-complexity bound of less than $O(1/\sqrt{\varepsilon})$, where $\varepsilon > 0$ is the absolute precision of the final objective function value. The steepest descent method cannot achieve a complexity better than $O(1/\varepsilon)$. These results and much more are explained in Yurii Nesterov's book [5]. This reference will be continuously cited in this text.

This paper is about Nesterov's ingenious treatment of steepest descent, first published in 1983 [4]. The theory laid dormant for many years, and is now calling the attention of the continuous optimization community. The method relies heavily on complexity results, and is not easy (although very worthwhile) to understand: he proves that with insignificant increase in computational effort per iteration, the computational complexity of the steepest descent method is lowered to the optimal value $O(1/\sqrt{\varepsilon})$. Besides the impact on actual computation that may be anticipated, the beauty of this work is in showing how highly non-trivial theoretical complexity results can beat the existing intuition and improve a century old method.

Our paper does a fine tuning of Nesterov's algorithm. His method relies heavily on the knowledge of the Lipschitz constant L, and achieves optimal complexity with the smallest possible computational time per iteration: only one gradient computation and no function evaluations (besides the ones used in the line search, if it is done at all). We trade this economy in function evaluations for an extra inexact line search, eliminating the need to know L or μ and improving the efficiency of each iteration, while keeping the optimal complexity in number of iterations.

We believe that our best algorithm is the one presented in Section 4, based on a decreasing sequence of estimates for the parameter μ.

Structure of the method.

A detailed explanation of the method is in Nesterov's book. Here we give a rough description of its main features. It uses local properties of $f(\cdot)$ (the steepest descent directions) and global properties of convex functions, by generating a sequence functional upper bounds imposed on the epigraph of $f(\cdot)$. The functional upper bounds are simple distance functions with the shape

$$\phi_k(x) = \phi_k^* + \frac{\gamma_k}{2}\|x - v^k\|^2,$$

where $\phi_k^* \in \mathbb{R}$, $\gamma_k > 0$ and $v^k \in \mathbb{R}^n$. Note that these functions are strictly convex with minimum ϕ_k^* at v^k. At start, $v^0 = x^0$ is a given point and $\gamma_0 > 0$ is given. If L is available, then one should take $\gamma_0 = L$.

The method will compute a sequence of points x^k, a sequence of positive parameters $\lambda_k \to 0$ and a sequence of functions $\phi_k(\cdot)$. The construction is such that at each iteration $\phi_k^* \geq f(x^k)$ (in the present paper, we require $\phi_k^* = f(x^k)$). Hence any optimal solution x^* satisfies $f(x^*) \leq \phi_k^* \leq \phi_k(x^k)$, and so we can think of $\phi_k(\cdot)$ as an upper bound imposed to the epigraph of $f(\cdot)$. This is shown in Fig. 1.1 .

The second order constants are defined as $\gamma_k - \mu = \lambda_k(\gamma_0 - \mu)$. So, as k increases, the functions $\phi_k(\cdot)$ become flatter, as shown in Fig. 1.1.

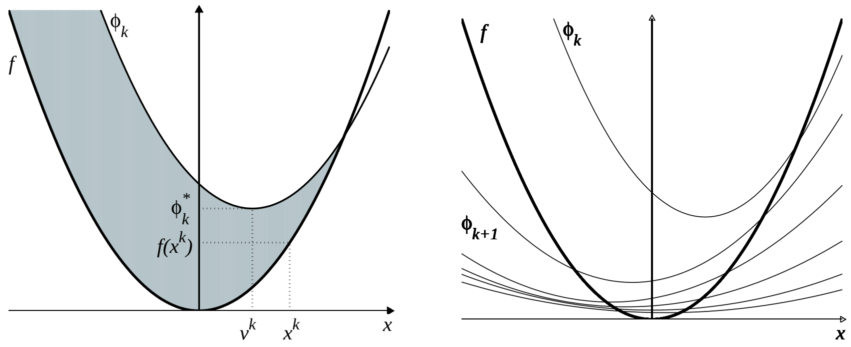

Fig. 1.1. *The mechanics of the algorithm.*

Now, the very non-trivial construction: Nesterov shows how to define these functions so that

$$f(x^k) \le \phi_k(\cdot) \le \lambda_k \phi_0(\cdot) + (1-\lambda_k) f(\cdot).$$

With this property (which will be developed in detail in Sec. 2), an immediate examination of an optimal solution results in

$$f(x^k) \le \phi_k(x^*) \le \lambda_k \phi_0(x^*) + (1-\lambda_k) f(x^*) = f(x^*) + \lambda_k(\phi_0(x^*) - f(x^*)),$$

and we see that $f(x^k) \to f(x^*)$ with the speed with which $\lambda_k \to 0$. Nesterov's method ensures $\lambda_k = O(1/k^2)$, and this means optimal complexity: an error of $\varepsilon > 0$ is achieved in $O(1/\sqrt{\varepsilon})$ iterations.

The practical efficiency of the method depends on how much each iteration explores the local properties around each iterate x^k to push λ_k down. Of course, the worst case complexity cannot be improved, but the speed can be improved in practical problems. This is the goal of this paper.

Structure of the paper. Section 2 begins by presenting in detail a prototype algorithm and its main variants, as they should be implemented. Then, after some technical results on quadratic functions, we describe all variables, functions and parameters needed in the analysis. Section 3 discusses some details on how the methods can be implemented and simplified in several different environments, keeping the the optimal complexity. Section 4 describes a modified algorithm based on adaptive estimates of the strong convexity parameter and does a complete complexity analysis, proving that Nesterov's complexity bound is preserved. Finally, Section 5 compares numerically the performance of the algorithms. The numerical results show that the new algorithm performs better in practically all examples in a set of randomly generated toy test problems.

2. Description of the main algorithms.

In this section we present a prototype algorithm which includes the choice of two parameters (α_k and θ_k) in each iteration: different choices of these parameters give different algorithms. There are several ways of choosing these parameters. Nesterov has a fixed rule for choosing them, based on the Lipschitz constant L. In our methods only the choice of θ_k is needed, while α_k is determined by solving a second degree equation.

Section 2.1 states the algorithms as they should be implemented, with no explanation of their theoretical bases. Understanding the motivation of the algorithm depends on the study of the convex combination of two simple quadratic functions, which is done separately in Section 2.2. Then we can describe all variables and functions used in the theoretical development in Section 2.3.

2.1. The algorithm. Here we state the prototype algorithm, followed by three different ways of implementing the steps. At this moment it is not possible to understand the meaning of each procedure: this will be the subject of the rest of this section.

Consider the problem (P). Here we assume that the strong convexity parameter μ is given (possibly null). In Section 4 we modify the algorithm to use an adaptive decreasing sequence of estimates for μ.

Algorithm 2.1. *Prototype algorithm.*

```
Data: $x^0 \in \mathbb{R}^n$, $v^0 = x^0$, $\gamma_0 > \mu$ ($\gamma_0 = L$ if $L$ is known).
$k = 0$
REPEAT
    $d^k = v^k - x^k$.
    Choose $\theta_k \in [0,1]$.
    $y^k = x^k + \theta_k d^k$.
    IF $\nabla f(y^k) = 0$, then STOP with $y^k$ as an optimal solution.
    Steepest descent step: $x^{k+1} = y^k - \nu \nabla f(y^k)$.
    Choose $\alpha_k \in (0,1]$.
    $\gamma_{k+1} = (1-\alpha_k)\gamma_k + \alpha_k \mu$.
    $v^{k+1} = \dfrac{1}{\gamma_{k+1}} \big((1-\alpha_k)\gamma_k v^k + \alpha_k(\mu y^k - \nabla f(y^k))\big)$.
    $k = k+1$.
```

Steepest descent step. Any efficient line search procedure may be used in the steepest descent step. We know that if L is known, then the step length $\nu = 1/L$ ensures that $f(y^k) - f(x^{k+1}) \geq \|\nabla f(y^k)\|^2/(2L)$. We require that the line search used must be at least as good as this, whenever L is known. Of course, a perfect line search would do, but it is not practically feasible. An Armijo search with well chosen parameters is usually the best choice, and even if L is not known it ensures a decrease of at least $\|\nabla f(y^k)\|^2/(4L)$, which is also acceptable for the complexity analysis. Hence we shall assume that at all iterations

$$f(y^k) - f(x^{k+1}) \geq \frac{1}{4L}\|\nabla f(y^k)\|^2. \tag{2.1}$$

Choice of the parameters.

- Our algorithm

  Choice of θ_k: compute $\theta_k \in [0,1]$ such that

  $f(x^k + \theta_k d^k) \leq f(x^k)$ and

  either $\theta_k = 1$ or $f'(x^k + \theta_k d^k, d^k) \geq 0$. (These are an Armijo condition with parameter 0 and a Wolfe condition with parameter 0.)

  Choice of α_k: compute $\alpha_k \in [0,1]$ as the largest root of the equation

$$A\alpha^2 + B\alpha + C = 0 \tag{2.2}$$

with

$$\begin{aligned}
Q &= \gamma_k \left(\frac{\mu}{2}\|v^k - y^k\| + \nabla f(y^k)^T (v^k - y^k)\right),\\
A &= Q + \frac{1}{2}\|\nabla f(y^k)\|^2 + (\mu - \gamma_k)\left(f(x^k) - f(y^k)\right),\\
B &= (\mu - \gamma_k)\left(f(x^{k+1}) + f(x^k)\right) - \gamma_k\left(f(y^k) - f(x^k)\right) - Q,\\
C &= \gamma_k\left(f(x^{k+1}) - f(x^k)\right).
\end{aligned}$$

We shall prove that this equation always has a real root and its largest root is in [0,1].

- Nesterov's algorithm
  Choice of θ_k: compute α_N as the positive solution of the equation

$$2L\alpha^2 - (1-\alpha)\gamma_k - \alpha\mu = 0.$$

Compute

$$\theta_k = \frac{\gamma_k}{\gamma_k + \alpha_N \mu}\alpha_N. \tag{2.3}$$

Original choice of α_k: set $\alpha_k = \alpha_N$ or
Modified choice of α_k: compute α_k as the largest root of (2.2), as in our algorithm.

Nesterov's choice ($\alpha_k = \alpha_N$ and θ_k computed in (2.3)) is the best possible in the following sense: it uses only one gradient computation and no function computations per iteration, and has optimal complexity. It has the following disadvantages:

- It does not explore the local efficiency of the Cauchy step, relying on the knowledge of the Lipschitz constant to compute α_k. As we shall see, the speed of the algorithm depends on how fast γ_k is reduced, which implies that in each iteration the value of α_k should be as large as the theory allows.
- The sequence of function values $(f(x^k))$ is not monotonically decreasing, although it converges to $f(x^*)$ with optimal complexity.

Our method does not depend on the knowledge of the Lipschitz constant, and computes α_k which in a sense is the largest possible. But this computation requires an inexact line search along the direction d, which may need several function evaluations. It has the following positive points:

- It does not depend on the knowledge of L, or even on the existence of L, but make use of it whenever available. Of course, the complexity results only makes sense when L exists (although unknown).
- The sequence $(f(x^k))$ decreases toward $f(x^*)$ with optimal complexity.
- When L is known, the extra line search can be abbreviated to limit the number of function calculations to a predetermined number: this will be described in Section 3.

2.2. Combining simple quadratic functions. The algorithm is based on the construction of a sequence of simple quadratic functions $\phi_k(\cdot)$. In this section we concentrate on the study of combinations of these quadratics. An example of this construction is in Fig. 2.1.

Here we are calling "simple" a quadratic function whose Hessian has the shape tI, with $t \geq 0$. We shall consider two such functions:

$$\begin{aligned} \phi(x) &= \phi^* + \frac{\gamma}{2}\|x - v\|^2, \\ \ell(x) &= \ell_0 + g^T(x - y) + \frac{\mu}{2}\|x - y\|^2, \end{aligned}$$

where $\phi^*, \ell_0, \mu, \gamma \in \mathbb{R}$, $g, y, v \in \mathbb{R}^n$. For simplicity, we assume that

$$\gamma > \mu \geq 0,$$

because this is the case in this paper. Both functions have scalar Hessians, but $\ell(\cdot)$ may be linear. If $\mu > 0$, then $\ell(\cdot)$ is minimized at $\hat{x} = y - \dfrac{g}{\mu}$, and

$$\min_{x \in \mathbb{R}^n} \ell(x) = \ell(\hat{x}) = \ell_0 - \frac{\|g\|^2}{2\mu}, \tag{2.4}$$

and $\ell(\cdot)$ may be expressed as $\ell(x) = \ell(\hat{x}) + \dfrac{\mu}{2}\|x - \hat{x}\|^2$.

We shall study the convex combinations of these two functions, defining for $\alpha \in \mathbb{R}$ and $x \in \mathbb{R}^n$,

$$\phi_+(\alpha, x) = (1 - \alpha)\phi(x) + \alpha\ell(x), \tag{2.5}$$

$$\phi_+^*(\alpha) = \inf_{x \in \mathbb{R}^n} \phi_+(\alpha, x) \quad \in \quad \mathbb{R} \cup \{-\infty\}, \tag{2.6}$$

$$\gamma_+(\alpha) = (1 - \alpha)\gamma + \alpha\mu. \tag{2.7}$$

The function $\phi_+(\alpha, \cdot)$ is also a simple quadratic function, with Hessian $\gamma_+(\alpha)I$. If $\gamma_+(\alpha) > 0$, then $\phi_+^*(\alpha)$ is finite. If $\gamma_+(\alpha) < 0$, then $\phi_+^*(\alpha) = -\infty$. For the case in this paper we have the following fact:

FACT 2.2. *Assuming that* $\gamma > \mu \geq 0$,

$$\{\alpha \in \mathbb{R} \mid \gamma_+(\alpha) > 0\} = (-\infty, \alpha_{\max}),$$

with $\alpha_{\max} = \dfrac{\gamma}{\gamma - \mu}$.

Proof. Trivial from (2.7). □

So $\phi_+^*(\alpha)$ is well defined for any $\alpha \neq \alpha_{\max}$. Here we shall look at what happens for $\alpha = \alpha_{\max}$, and eliminate an annoying "pathological" situation. When $\alpha = \alpha_{\max}$, $\gamma_+(\alpha) = 0$ and $\phi_+(\alpha_{\max}, \cdot)$ is linear. In general, $\phi_+^*(\alpha_{\max}) = -\infty$, but it may happen that $\phi_+(\alpha_{\max}, \cdot)$ is constant and $\phi_+^*(\alpha_{\max}) > -\infty$. We shall see that in this unlikely situation everything is very simple. We state this as a lemma.

LEMMA 2.3. *Consider two simple quadratic functions $\phi(\cdot)$ and $\ell(\cdot)$, as above, with $\gamma > \mu$ and assume that $\ell(\cdot)$ is not constant. Define $\alpha_{max} = \gamma/(\gamma - \mu)$. Then $\phi_+^*(\alpha_{max})$ is finite only if $\mu > 0$, and the minimizers of $\phi(\cdot)$ and $\ell(\cdot)$ coincide. In this case, $\phi_+^*(\cdot)$ is linear in $(-\infty, \alpha_{max}]$.*

Proof. By construction, $\phi_+(\alpha_{max}, \cdot)$ is linear, because $\gamma_+(\alpha_{max}) = 0$. Assuming that $\inf_{x \in \mathbb{R}^n} \phi_+(\alpha_{max}, x)$ is finite, we conclude that

$$\phi_+(\alpha_{max}, \cdot) = (1 - \alpha_{max})\phi(\cdot) + \alpha_{max}\ell(\cdot)$$

is constant.

If $\mu = 0$, then $\alpha_{max} = 1$ and $\ell(\cdot)$ is constant, contradicting the hypothesis. Hence $\mu > 0$, and we may write $\ell(x) = \ell^* + \mu\|x - \hat{x}\|$, where $\hat{x}$ is the minimizer of $\ell(\cdot)$. Differentiating the expression for ϕ_+ in relation to x, we have for any $x \in \mathbb{R}^n$,

$$(1 - \alpha_{max})\gamma(x - v) + \alpha_{max}\mu(x - \hat{x}) = 0.$$

If, by contradiction, $v \neq \hat{x}$, then substituting $x = v$, we get $\alpha_{max}\mu = 0$, which cannot be true, proving that $v = \hat{x}$. Now, since v is the minimizer of both $\phi(\cdot)$ and $\ell(\cdot)$, we have for any $\alpha \in (-\infty, \alpha_{max}]$, $\phi_+^*(\alpha) = (1 - \alpha)\phi^* + \alpha\ell(v)$, which is linear, completing the proof. □

The following lemma was essentially proved by Nesterov.

Lemma 2.4. *For any $\alpha \in (-\infty, \alpha_{\max})$, the function $x \mapsto \phi_+(\alpha, x)$ is a simple quadratic function given by*

$$\phi_+(\alpha, x) = \phi_+^*(\alpha) + \frac{\gamma_+(\alpha)}{2}\|x - v_+(\alpha)\|^2, \tag{2.8}$$

with

$$\begin{aligned} \phi_+^*(\alpha) &= \ell_0 - \frac{\alpha^2}{2\gamma_+(\alpha)}\|g\|^2 + \\ & +(1-\alpha)\left(\phi^* - \ell_0 + \frac{\alpha\gamma}{\gamma_+(\alpha)}\left(\frac{\mu\|y - v\|^2}{2} + g^T(v - y)\right)\right) \end{aligned} \tag{2.9}$$

and

$$v_+(\alpha) = \frac{1}{\gamma_+(\alpha)}\left[(1 - \alpha)\gamma v + \alpha(\mu y - g)\right]. \tag{2.10}$$

In particular, if $\mu > 0$ then

$$\phi_+^*(1) = \ell_0 - \frac{\|g\|^2}{2\mu}. \tag{2.11}$$

Proof. The proof is straightforward and done in detail in Nesterov's book [5]. The expression (2.10) is obtained by differentiation of (2.5). Now, $x = v_+(\alpha)$ is the unique minimizer of (2.5). Substituting the expression (2.10) into (2.5), we obtain

$$\phi_+^*(\alpha) = (1 - \alpha)\phi^* + \alpha\ell_0 - \frac{\alpha^2}{2\gamma_+(\alpha)}\|g\|^2 + \frac{\alpha(1 - \alpha)\gamma}{\gamma_+(\alpha)}\left(\frac{\mu}{2}\|y - v\|^2 + g^T(v - y)\right).$$

Adding and subtracting ℓ_0, we have (2.9). Finally (2.11) follows from (2.9) with $\alpha = 1$ and $\gamma_+(\alpha) = \mu > 0$, completing the proof. □

We now use Danskin's theorem in the format presented in Bertsekas [1] to prove that $\phi_+^*(\alpha)$ is concave.

Lemma 2.5. *The function $\alpha \in \mathbb{R} \mapsto \phi_+^*(\alpha)$ constructed above is concave in $\mathbb{R}$ and differentiable in $(-\infty, \alpha_{\max})$.*

Proof. We know that $\phi_+^*(\alpha) = -\infty$ for $\alpha > \alpha_{max}$.

If $\phi_+^*(\alpha_{max})$ is finite, then Lemma 2.3 ensures that $\phi_+^*(\cdot)$ is linear in $(-\infty, \alpha_{max}]$ and hence this case is proved. Assume then that $\phi_+^*(\alpha_{max}) = -\infty$. We must prove that $\phi_+^*(\cdot)$ is concave and differentiable in $(-\infty, \alpha_{max})$. We will prove the concavity in an arbitrary interval $[\alpha_1, \alpha_2] \subset (-\infty, \alpha_{\max})$. For each $\alpha \in [\alpha_1, \alpha_2]$, $\phi_+(\alpha, \cdot)$ has a unique minimizer $v_+(\alpha)$. By continuity of $v_+(\cdot)$ in (2.10), $v_+([0, \bar{\alpha}])$ is bounded, i.e., there exists a compact set $V \subset \mathbb{R}^n$ such that

$$\phi_+^*(\alpha) = \min_{x \in V} \phi_+(\alpha, x).$$

So, the hypotheses for Danskin's theorem are satisfied:
• $\phi_+(\cdot, x)$ is concave (in fact linear) for each $x \in \mathbb{R}^n$,
• $\phi_+(\alpha, \cdot)$ has a unique minimizer in the compact set V for each $\alpha \in [\alpha_1, \alpha_2]$.
We conclude that $\phi_+^*(\cdot)$ is concave and differentiable in $[\alpha_1, \alpha_2]$. Since $[\alpha_1, \alpha_2]$ was arbitrary, this proves the concavity and differentiability in $(-\infty, \alpha_{\max})$, completing the proof. □

We can now study the equation

$$\phi_+^*(\alpha) = P,$$

for a given $P \in \mathbb{R}$. We shall be interested in the case in which $\phi^* > \inf_{x \in \mathbb{R}^n} \ell(x)$ and $\ell(\cdot)$ is not constant.

LEMMA 2.6. *Assume that* $\phi^* \geq \inf_{x \in \mathbb{R}^n} \ell(x)$. *Given a constant* $P \geq \inf_{x \in \mathbb{R}^n} \ell(x)$, *then*

- *either* $\phi_+^*(\alpha) < P$ *for all* $\alpha \in [0, 1]$ *(this situation will always be avoided),*
- *or* $\phi_+^*(\alpha) = P$ *has one or two real roots and the largest root belongs to* $[0, 1]$. *The roots are computed by solving the second degree equation*

$$A\alpha^2 + B\alpha + C = 0 \tag{2.12}$$

with

$$\begin{aligned} Q &= \gamma\left(\frac{\mu}{2}\|v - y\| + g^T(v - y)\right), \\ A &= Q + \frac{1}{2}\|g\|^2 + (\mu - \gamma)(\phi^* - \ell_0), \\ B &= (\mu - \gamma)(P + \phi^*) - \gamma(\ell_0 - \phi^*) - Q, \\ C &= \gamma(P - \phi^*). \end{aligned}$$

Proof. If $\phi_+^*(\alpha) < P$ for all $\alpha \in [0, 1]$, then trivially there are no solutions for $\phi_+^*(\alpha) = P$. Otherwise, let $\bar{\alpha}$ be the largest $\alpha \in [0, 1)$ such that $\phi_+^*(\bar{\alpha}) = \max_{\alpha \in [0,1]} \phi_+^*(\alpha)$ (which exists because $\phi_+^*(\cdot)$ is concave and continuous in $[0, 1)$, with $\phi_+^*(1) < \phi_+^*(\bar{\alpha})$). Since $\phi_+^*(1) < \phi_+^*(\bar{\alpha})$ and $P \geq \phi_+^*(1)$, the concave function decreases from $\bar{\alpha}$ to 1, and crosses the value P exactly once in $[\bar{\alpha}, +\infty)$, at some point $\alpha' \in [0, 1]$. Setting $\phi_+^*(\alpha) = P$ in (2.9) and multiplying both sides of the equation by $\gamma_+(\alpha) = (1 - \alpha)\gamma + \alpha\mu$, we obtain a second degree equation in α. The largest root of this equation must be α', completing the proof. □

2.3. Analysis of the algorithm. The algorithm constructs a sequence of points (x^k) and a sequence of functions $(\phi_k(\cdot))$. Each iteration constructs $\phi_{k+1}(\cdot)$ so that $\phi_{k+1}(\cdot) \leq (1 - \alpha_k)\phi_k(\cdot) + \alpha_k f(\cdot)$. $\phi_k(\cdot)$ is a simple quadratic function with Hessian $\gamma_k I$, and $\gamma_{k+1} = (1 - \alpha_k)\gamma_k + \alpha_k\mu$. So, γ_k converges to μ at the same rate as $\phi_k(x^*)$ converges to $f(x^*)$, where x^* is an optimal solution.

This section describes in detail the construction of these functions, and we should keep in mind that we have two main objectives:

- Prove that the algorithm is well defined, i.e., that we can compute α_k such that $f(x^{k+1}) = \min_{x\in\mathbb{R}^n} \phi_{k+1}(x)$;
- Prove that α_k is large, i.e., $\alpha_k = \Omega(\sqrt{\gamma_{k+1}})$, which ensures the optimal complexity.

The geometry of one iteration is shown in Fig. 2.1: the figure at the left shows the points present in an iteration for a 2-dimensional problem. The figure at the right shows the functions involved in an iteration of a 1-dimensional problem.

A complete proof of the optimal complexity will be done in Section 4, for an algorithm based on adaptive estimates of μ. That proof will be trivially true for the case of a fixed μ.

Now we describe the variables and functions associated with each iteration $k = 0, 1, \ldots$ of the algorithm.

- $x^k, v^k \in \mathbb{R}^n$, with x^0 given and $v^0 = x^0$.
- $\alpha_k \in [0, 1)$.
- $\gamma_k \in \mathbb{R}_{++}$: second order constants. $\gamma_0 > \mu$ is given, and should be taken as $\gamma_0 = L$ whenever L is available.
- $y^k \in \mathbb{R}^n$: a point in the segment joining x^k and v^k
$$y^k = x^k + \theta_k(v^k - x^k).$$
It is from y^k that a steepest descent step is computed in each iteration to obtain x^{k+1}. This paper will discuss in detail the computation of θ_k.
- $\phi_k(\cdot)$: function defined for $x \in \mathbb{R}^n$ by
$$\phi_k(x) = \phi_k^* + \frac{\gamma_k}{2}\|x - v^k\|^2, \tag{2.13}$$
with $\phi_0^* = f(x^0)$. By construction, $\phi_k(\cdot)$ is a simple quadratic function which assumes the minimum ϕ_k^* at v^k. These are the functions discussed in the introduction, whose construction and properties are described from now on. Nesterov constructs these functions so that $\phi_k^* \geq f(x^k)$. We shall require that $\phi_k^* = f(x^k)$ at all iterations.
- $x^{k+1} = y^k - \nu\nabla f(y^k)$: next iterate, computed by a line search. As we commented above, a good line search ensures
$$f(x^{k+1}) \leq f(y^k) - \frac{1}{4L}\nabla f(y^k). \tag{2.14}$$

Construction of the functions $\phi_k(\cdot)$.. Now we quote Nesterov [5] to describe the construction of the functions $\phi_k(\cdot)$. We start from given $x^k, v^k, \phi_k(\cdot), \gamma_k$. At this point we assume that y^k is also given (its calculation will be the task of this paper). Assume also that x^{k+1} has been computed and that (2.14) holds.

Once y^k and x^{k+1} is given, we must compute α_k. Here we use an abuse of notation to include α as a variable and define the function
$$\alpha \in \mathbb{R}, x \in \mathbb{R}^n \mapsto \phi_k(\alpha, x).$$

The function $\phi_{k+1}(\cdot,\cdot)$ is obtained by a convex combination of $\phi_k(\cdot,\cdot)$ and a lower quadratic approximation of $f(\cdot)$ around y^k:
$$\phi_{k+1}(\alpha, x) = (1 - \alpha)\phi_k(x) + \alpha\ell_k(x), \tag{2.15}$$

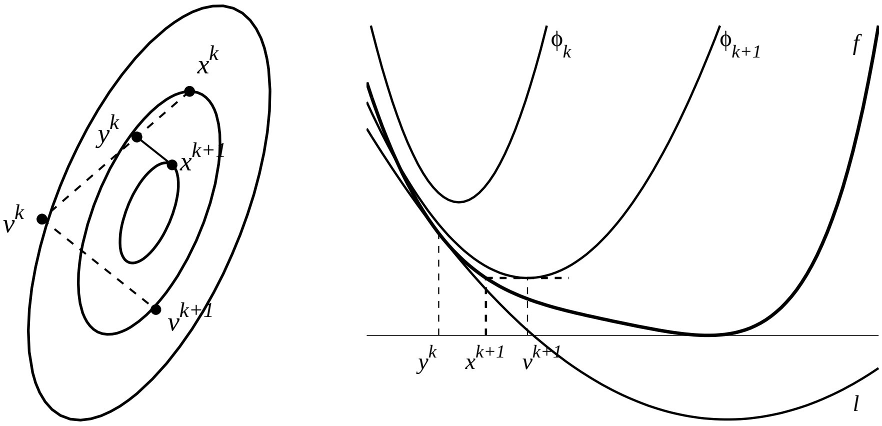

FIG. 2.1. *One iteration of the algorithm*

where

$$\ell_k(x) = f(y^k) + \nabla f(y^k)^T(x - y^k) + \frac{\mu}{2}\|x - y^k\|^2.$$

We shall assume that at all iterations $\ell(\cdot)$ is not constant, otherwise y^k would be an optimal solution for the problem (P). Let $\phi^*_{k+1}(\alpha) = \min\{\phi_{k+1}(\alpha, x) \,|\, x \in \mathbb{R}^n\}$. The choice of α must be such that $\phi^*_{k+1}(\alpha) = f(x^{k+1})$: we must show under what conditions this is feasible.

We see that $\phi_{k+1}(\cdot)$ is the combination of two simple quadratic functions. Using the results of Section 2.2, we obtain directly from Lemma 2.4 with $y = y^k$, $v = v^k$, $\gamma = \gamma_k$, $\gamma_+(\alpha) = \gamma_{k+1}$, $\ell_0 = f(y^k)$, $g = \nabla f(y^k)$ the following:

$$v^{k+1}(\alpha) = \frac{(1-\alpha)\gamma_k v^k + \alpha(\mu y^k - \nabla f(y^k))}{\gamma_{k+1}}. \tag{2.16}$$

$$\begin{aligned} \phi^*_{k+1}(\alpha) = \;& f(y^k) - \frac{\alpha^2}{2\gamma_{k+1}}\|\nabla f(y^k)\|^2 + (1-\alpha)[\phi^*_k - f(y^k) + \\ & + \frac{\alpha\gamma_k}{\gamma_{k+1}}\left(\frac{\mu}{2}\|y^k - v^k\|^2 + \nabla f(y^k)^T(v^k - y^k)\right)]. \end{aligned} \tag{2.17}$$

Setting $y^k = x^k + \theta_k d^k$, with $d^k = v^k - x^k$ and $\theta_k \in [0,1]$ this may be written as

$$\phi^*_{k+1}(\alpha) = f(y^k) - \frac{\alpha^2}{2\gamma_{k+1}}\|\nabla f(y^k)\|^2 + (1-\alpha)\zeta(\alpha, \theta_k) \tag{2.18}$$

with

$$\begin{aligned} \zeta(\alpha,\theta) \;\; = \;\; & \phi^*_k - f(x^k + \theta d^k) + \\ & + \frac{\alpha\gamma_k}{(1-\alpha)\gamma_k + \alpha\mu}\left(\frac{\mu}{2}(1-\theta)^2\|d^k\|^2 + (1-\theta)f'(x^k + \theta d^k, d^k)\right). \end{aligned} \tag{2.19}$$

LEMMA 2.7. *Assume that $f(x^k) \le \phi^*_k$, $y^k = x^k + \theta_k d^k$ with $d^k = v^k - x^k$ and $\theta_k \in [0,1]$. Assume also that x^{k+1} is obtained by a Cauchy step from y^k, satisfying (2.14), where the Lipschitz constant L is possibly infinite. Then*

$$\phi^*_{k+1}(\alpha) \ge f(x^{k+1}) + \left(\frac{1}{4L} - \frac{\alpha^2}{2\gamma_{k+1}}\right)\|\nabla f(y^k)\|^2 + (1-\alpha)\zeta(\alpha, \theta_k). \tag{2.20}$$

Furthermore

$$\zeta(\alpha,\theta) \geq \left(-\theta + \frac{\alpha\gamma_k}{(1-\alpha)\gamma_k + \alpha\mu}(1-\theta)\right) f'(x^k + \theta d^k, d^k). \tag{2.21}$$

Proof. The expression (2.20) follows directly from (2.18) and the fact that the steepest descent step satisfies (2.14).

Using the facts that $f(x^k) \leq \phi_k^*$ and $\mu \geq 0$ in the definition of $\zeta(\cdot,\cdot)$, we can write

$$\zeta(\alpha,\theta) \geq f(x^k) - f(x^k + \theta d^k) + \frac{\alpha\gamma_k}{(1-\alpha)\gamma_k + \alpha\mu}(1-\theta) f'(x^k + \theta d^k, d^k).$$

By the convexity of $f(\cdot)$, we know that $f(x^k) - f(x^k + \theta d^k) \geq -\theta f'(x^k + \theta d^k, d^k)$, and so

$$\zeta(\alpha,\theta) \geq \left(-\theta + \frac{\alpha\gamma_k}{(1-\alpha)\gamma_k + \alpha\mu}(1-\theta)\right) f'(x^k + \theta d^k, d^k),$$

completing the proof. □

Now we are ready to show the two main results cited above:

- For the choices of θ_k taken by Algorithm 2.1, it is always possible to compute α_k such that $\phi_{k+1}^*(\alpha_k) = f(x^{k+1})$.
- For these choices, the value of α_k is large, i.e., $\alpha_k = \Omega(\sqrt{\gamma_{k+1}})$.

We begin by describing Nesterov's choice.

Nesterov's choice.. When a Lipschitz constant L is known, Nesterov's choices are:
• α_N is computed so that the central term in (2.20) is null. So, α_N is the positive solution of the second degree equation

$$2L\alpha^2 - (1-\alpha)\gamma_k - \alpha\mu = 0. \tag{2.22}$$

• θ_N is so that the right hand side of (2.21) is null:

$$\theta_N = \frac{\gamma_k}{\gamma_k + \alpha_N\mu}\alpha_N. \tag{2.23}$$

Note: If L is unknown but exists, α_N and θ_N are well defined (but unknown). Otherwise we set $\alpha_N = 0$, $\theta_N = 0$.

Nesterov's original method sets $\alpha_k = \alpha_N$ and $\theta_k = \theta_N$, obtaining directly from (2.20) and $\zeta(\alpha_N, \theta_N) = 0$ that

$$\phi_{k+1}^*(\alpha_N) \geq f(x^{k+1}) \qquad \text{and} \qquad \alpha_N = \sqrt{\frac{\gamma_{k+1}}{2L}}. \tag{2.24}$$

The next lemma uses the definition of α_N (possibly null if L is unknown) to show under what conditions we can compute $\alpha_k \geq \alpha_N$ such that $\phi_{k+1}^*(\alpha_k) = f(x^{k+1})$. We use directly Lemma 2.6.

By construction, we have

$$\phi_{k+1}^*(0) = \phi_k^* \geq f(x^k) \tag{2.25}$$

$$\phi_{k+1}^*(1) = \inf_{x\in\mathbb{R}^n} \ell_k(x) \leq f(x^*) \leq f(x^{k+1}). \tag{2.26}$$

Let us first eliminate a trivial case: due to (2.26), if $\phi^*_{k+1}(1) = f(x^{k+1})$ then x^{k+1} is an optimal solution, and we may terminate the algorithm. Let us assume that $\phi^*_{k+1}(1) < f(x^{k+1})$.

Lemma 2.8. *Consider an iteration k of Algorithm 2.1. Assume that $\phi^*_{k+1}(1) < f(x^{k+1})$. Assume that $y^k = x^k + \theta_k d^k$ is chosen so that $\zeta(\alpha_N, \theta_k) \geq 0$. Then there exists a value $\alpha_k \in [\alpha_N, 1]$ that solves the equation $\phi^*_{k+1}(\alpha_k) = f(x^{k+1})$ in (2.17).*

Proof. Assume that α_N and $y^k = x^k + \theta_k d^k$ have been computed as in the Lemma. Now use the result in Lemma 2.6, with the same attribution of variables as we used in (2.16). Then

$$\phi^*_k = f(x^k) > \ell_k(x^k) \geq \inf_{x \in \mathbb{R}^n} \ell_k(x).$$

Taking $P = f(x^{k+1})$ and $\alpha = \alpha_N$ with $\zeta(\alpha_N, \theta_k) \geq 0$ we get

$$\phi^*_{k+1}(\alpha_N) \geq f(x^{k+1}),$$

and by Lemma 2.6 the equation $\phi^*_{k+1}(\alpha) = f(x^{k+1})$ has a real root, and the largest root α' is in $[0, 1]$. Note that $\alpha' \geq \alpha_N$, because α' is the largest root of the concave function $\phi^*_{k+1}(\cdot) - P$. □

We conclude from this lemma that the equality $\phi^*_{k+1}(\alpha_k) = f(x^{k+1})$ can always be achieved whenever $\zeta(\alpha_N, \theta_k) \geq 0$.

Theorem 2.9. *Algorithm 2.1 with both choices (modified Nesterov and ours) of θ_k is well defined and generates sequences (x^k) and (ϕ^*_k) such that for all $k = 0, 1, \ldots$, $\phi^*_k = f(x^k)$. Besides this, the parameters α_k and γ_k satisfy*

$$\gamma_k - \mu = \alpha_k(\gamma_0 - \mu) \qquad \text{and} \qquad \alpha_k \geq \sqrt{\frac{\gamma_{k+1}}{2L}}. \tag{2.27}$$

Proof. (i) Let us first prove that both versions of the algorithm are well defined: Nesterov: immediate. Taking $\theta_k = \theta_N$ we obtain by definition of α_N and θ_N, $\zeta(\alpha_N, \theta_N) = 0$ and Lemma 2.8 ensures that $\alpha_k \in [\alpha_N, 1)$ can be computed as desired.
Our method: since $f(x^k) - f(x^k + \theta_k d^k) \geq 0$ and $f'(x^k + \theta_k d^k, d^k) \geq 0$ by construction, then $\zeta(\alpha, \theta_k) \geq 0$ for any $\alpha \in [0, 1]$, and Lemma 2.8 can be used similarly.
(ii) In both cases we obtain $\zeta(\alpha_k, \theta_k) \geq 0$. Hence from (2.20),

$$f(x^{k+1}) = \phi^*_{k+1}(\alpha_k) \geq f(x^{k+1}) + \left(\frac{1}{4L} - \frac{\alpha^2}{2\gamma_{k+1}}\right) \|\nabla f(y^k)\|^2,$$

which immediately gives

$$\alpha_k \geq \sqrt{\frac{\gamma_{k+1}}{2L}},$$

completing the proof. □

We conclude from this that Nesterov's algorithm, which uses the value $\theta_k = \theta_N$ with $\alpha_k = \alpha_N$ can be improved by re-calculating α_k using (2.17) (and keeping $\theta_k = \theta_N$).

Remark.. Even computing the best value for α_k, the sequence $(f(x^k))$ is not necessarily decreasing.

Remark.. In the case $\mu = 0$, the expressions get simplified. Then α_N is the positive solution of $L\alpha_k^2 = (1-\alpha_k)\gamma_k$, and expression (2.21) is reduced to

$$\zeta(\alpha,\theta) \geq \frac{\alpha-\theta}{1-\alpha} f'(x^k+\theta d, d)$$

In this case, $\theta_N = \alpha_N$.

3. More on the choice of θ_k.

The choice of θ_k proposed by us depends on a line search along d^k. The effort needed by this search depends very much on the comparative difficulties of computing function values and gradients. If a gradient computation demands much more time than a function evaluation, then the line search can be very good. Otherwise, the effort in the line search must be reduced. In this section we show some results on how the search can be simplified.

The scope of this section is to discuss the amount of computation needed in this line search, and to show that when L and μ are known the line search may be done in polynomial time. The results in this section may have limited interest in a first reading.

We conclude from the preceding analysis that what must be shown for a given choice of $y^k = x^k + \theta_k d^k$ in an iteration k is that $\zeta(\alpha_N, \theta_k) \geq 0$. The following lemma summarizes the conditions that ensure this property. These conditions will lead to new algorithms.

Lemma 3.1. *Let $y^k = x^k + \theta_k d^k$ be the choice made by Algorithm 2.1 in iteration k, and let θ_N be the Nesterov choice* (2.23)*. Assume that one of the following conditions is satisfied:*
(i) $f(y^k) \leq f(x^k)$ and $f'(y^k, d^k) \geq 0$;
(ii) $f(y^k) \leq f(x^k)$ and $\theta_k \geq \theta_N$;
(iii) $f'(y^k, d^k) \geq 0$ and $\theta_k \leq \theta_N$;
(iv) $f'(x^k, d^k) \geq -\dfrac{\mu}{2}\|d^k\|^2$ and $\theta_k = 0$.
Then $\zeta(\alpha_N, \theta_k) \geq 0$ (and $\alpha_k \geq \alpha_N$ may be computed using (2.17)*).*

Proof. (i) It has been proved in Theorem 2.9.

Let us substitute $\theta_k = \theta_N + \Delta\theta$ in (2.21), with $\gamma_N = (1-\alpha_N)\gamma_k + \alpha_N\mu$:

$$\zeta(\alpha_N, \theta_k) \geq \left(-\theta_N + \frac{\alpha_N\gamma_k}{\gamma_N}(1-\theta_N) - \Delta\theta - \frac{\alpha_N\gamma_k}{\gamma_N}\Delta\theta \right) f'(x^k+\theta_k d^k, d^k).$$

By definition of θ_N, $-\theta_N + \alpha_N\gamma_k(1-\theta_N)/\gamma_N = 0$. Hence

$$\zeta(\alpha_N, \theta_k) \geq -\left(1 + \frac{\alpha_N\gamma_k}{\gamma_N}\right)\Delta\theta\, f'(x^k+\theta_k d^k, d^k). \tag{3.1}$$

(ii) Assume that $f(y^k) \leq f(x^k)$ and $\theta_k \geq \theta_N$.
If $f'(y^k, d^k) \leq 0$, then the result follows from (3.1) because $\Delta\theta \geq 0$. Otherwise, it follows from (i).
(iii) Assume now that $f'(y^k, d^k) \geq 0$ and $\theta_k \leq \theta_N$.
Then the result follows from (3.1) because $\Delta\theta \leq 0$.
(iv) This case is also trivial by (2.19), completing the proof. □

All that we must do is to specify in Algorithm 2.1 the choice of θ_k and α_k as follows:
Choose $\theta_k \in [0,1]$ satisfying one of the conditions in Lemma 3.1.
Compute α_k such that $\phi_{k+1}^* = f(x^{k+1})$ using (2.17).

We must discuss the computation of θ_k, which requires a line search. Of course, if L is given, we can use $\theta_k = \theta_N$ and ensure the optimal complexity, but larger values of α_k may be obtained.

We would like to obtain θ_k (and then α_k) such that $\zeta(\alpha_k, \theta_k)$ is as large as possible. Since the determination of α_k depends on the steepest descent step from $y^k = x^k + \theta_k d^k$, we cannot maximize $\zeta(\cdot, \cdot)$. But positive values of $\zeta(\alpha, \theta_k)$ will be obtained if θ_k is such that $f(x^k) - f(x^k + \theta_k d^k)$ and $f'(x^k + \theta_k d^k, d^k)$ are made as large as possible (of course, these are conflicting objectives).

- We define (but do not compute) two points:

$$\theta' \in \operatorname{argmin}\{f(x^k + \theta d^k) \mid \theta \geq 0\}, \tag{3.2}$$

$$\theta'' = \max\{\theta \in [0, 1] \mid f(x^k + \theta d^k) \leq f(x^k)\}. \tag{3.3}$$

Let us examine two important cases:

- If $f'(x^k, d^k) \geq 0$, then we should choose $\theta_k = 0$: a Cauchy step from x^k follows.
- If $f(x^k + d^k) \leq f(x^k)$, then we may choose $\theta_k = 1$: a Cauchy step from v^k follows.

If none of these two special situations occur, then $0 < \theta' < \theta'' < 1$. Any point $\theta_k \in [\theta', \theta'']$ gives $\zeta(\alpha, \theta_k) \geq 0$ for any $\alpha \in [0, 1]$, and $y^k = x^k + \theta_k d^k$ satisfies condition (i) of Lemma 3.1. A point in this interval can be computed by the interval reduction algorithm described in the Appendix.

We have no complexity estimates for a line search procedure such as this if L and μ are not available. When these constants are known, it is possible to limit the number of iterations in the line search to keep Nesterov's complexity. We shall discuss briefly the choice of θ_k in three possible situations, according to our knowledge of the constants.

Case 1. No knowledge of L or μ.. We start the iteration checking the special cases above: If $f(x^k + d^k) \leq f(x^k)$, then we choose $\theta_k = 1$: a Cauchy step from v^k will follow. Otherwise, we must decide whether $f'(x^k, d^k) \geq 0$. This may be done as follows:

Compute $f(x^k + \tilde{\theta} d^k)$ for a *small* value of $\tilde{\theta} > 0$
If $f(x^k + \tilde{\theta} d^k) \geq f(x^k)$, compute $\nabla f(x^k)$ and then $f'(x^k, d^k) = \nabla f(x^k)^T d^k$.
Else $f'(x^k, d^k) < 0$.

If $f'(x^k, d^k) \geq 0$, set $\theta_k = 0$. Else, do a line search as in the Appendix, starting with the points 0, $\tilde{\theta}$ and 1.

Remarks.. Of course we cannot specify the meaning of "small $\tilde{\theta}$", but this is usually easy in practical applications. An intellectually satisfying guess is the following: Let L be a very large upper bound for the (unknown) Lipschitz constant. Compute α_N and $\tilde{\theta} = \theta_N$ by (2.22) and (2.23).

Case 2: L is given.. In this case we start the search with θ_N computed as in (2.23). The line search can be abbreviated to keep the number of function calculations within a given bound, resulting in a step satisfying one of the three first conditions in Lemma 3.1. There are two cases to consider, depending on the sign of $f(x^k) - f(x^k + \theta_N d^k)$.

- $f(x^k + \theta_N d^k) < f(x^k)$: Condition (ii) of Lemma 3.1 is satisfied for any $\theta_k \geq \theta_N$ such that $f(x^k + \theta_k d^k) \leq f(x^k)$. We may use an interval reduction method as in the Appendix, or simply take steps to increase θ. A trivial method is the following, using a constant $\beta > 1$:

```
$\theta_k = \theta_N$
WHILE $\beta\theta_k \leq 1$ and $f(x^k + \beta\theta_k) \leq f(x^k)$, set $\theta_k = \beta\theta_k$.
```

In any method, the number of function calculations may be limited by a given constant, adopting as θ_k the largest computed step length which satisfies the descent condition.

- $f(x^k + \theta_N d^k) > f(x^k)$: Condition (iii) of Lemma 3.1 holds for θ_N, and we intend to reduce the value of θ. For an interval reduction search between 0 and θ_N we need an intermediate point ν such that $f(x^k + \nu d^k) \leq f(x^k)$, if it exists. We propose the following procedure:

Compute $f(x^k + \nu d^k)$ for $\nu << \theta_N$
If $f(x^k + \nu d^k) \geq f(x^k)$, set $\theta_k = \nu$
Else compute θ_k by an interval reduction method as in the Appendix, starting with the points 0, ν and θ_N. The algorithm can be interrupted at any time, setting $\theta_k = B$.

Case 3: L and $\mu > 0$ are given.. This is a special case of case 2, with an interesting property: a point satisfying condition (i) in Lemma 3.1 can be computed in polynomial time without computing $\nabla f(x^k)$, using the next lemma.

LEMMA 3.2. *Consider* $\theta' = \operatorname{argmin}\{f(x^k + \theta d^k) \,|\, \theta \geq 0\}$ *and* $\theta'' > \theta'$ *such that* $f(x^k + \theta'' d^k) = f(x^k)$, *if it exists. Define* $\bar\theta = \mu/(2L)$.
If $f(x^k + \bar\theta d^k) \leq f(x^k) - \dfrac{\mu^2}{8L}\|d\|^2$, *then* $\theta' \geq \bar\theta$ *and* $\theta'' - \theta' \geq \bar\theta$,
otherwise $f'(x^k, d^k) \geq -\dfrac{\mu}{2}\|d\|^2$.

Proof. Assume that $f(x^k + \theta' d^k) \leq f(x^k + \bar\theta d^k) \leq f(x^k) - \dfrac{\mu^2}{8L}\|d\|^2$.
Using the Lipschitz condition at θ' with $f'(x^k + \theta' d^k, d^k) = 0$, we get for $\theta \in \mathbb{R}$,

$$f(x^k + \theta d^k) \leq f(x^k + \theta' d^k) + \frac{L}{2}(\theta - \theta')^2\|d\|^2.$$

Using the assumption,

$$f(x^k + \theta d^k) \leq f(x^k) + \left(L(\theta - \theta')^2 - \frac{\mu^2}{4L}\right)\frac{\|d\|^2}{2}.$$

For $\theta = 0$, this gives $L\theta'^2 - \mu^2/(4L) \geq 0$, or equivalently $\theta'^2 \geq \bar\theta^2$, proving the first inequality.
For $\theta = \theta''$, $f(x^k + \theta'') = f(x^k)$, and we obtain immediately $(\theta'' - \theta')^2 \geq \bar\theta^2$.
Assume now that $f(x^k + \bar\theta d^k) > f(x^k) - \dfrac{\mu^2}{8L}\|d\|^2$.
By the Lipschitz condition for $\nabla f(\cdot)$, we know that

$$f(x^k + \bar\theta d^k) \leq f(x^k) + f'(x^k, d^k)\bar\theta + \frac{L}{2}\bar\theta^2\|d\|^2.$$

Assuming by contradiction that $f'(x^k, d^k) < -\mu\|d\|^2/2$, we obtain

$$\begin{aligned} f(x^k + \bar\theta d^k) &< f(x^k) - \frac{\mu}{2}\bar\theta\|d\|^2 + \frac{L}{2}\bar\theta^2\|d\|^2 \\ &= f(x^k) - \frac{\mu^2}{4L}\|d\|^2 + \frac{\mu^2}{8L}\|d\|^2 \\ &= f(x^k) - \frac{\mu^2}{8L}\|d\|^2, \end{aligned}$$

contradicting the hypothesis and completing the proof. □

In this case we may use a golden search, as follows:
If $f(x^k+\bar\theta d^k) \geq f(x^k) - \dfrac{\mu^2}{8L}\|d\|^2$, set $\theta_k = 0$ and stop (condition (iv) in Lemma 3.1 holds).
If $f(x^k+d^k) \leq f(x^k)$, set $\theta_k = 1$ and stop (condition (ii) in Lemma 3.1 holds).
(Now we know that $\theta' \leq \theta'' - \mu/(2L) < 1$).
Use an interval reduction search as in the Appendix in the interval $[\bar\theta, 1]$

If we use a golden section search, then at each iteration j before the stopping condition is met, we have $A \leq \theta' \leq \theta'' \leq B$, and the interval length satisfies $B - A \leq \beta^j$, with $\beta = (\sqrt{5}-1)/2 \approx 0.62$. Using the lemma above, a point $B \in [\theta', \theta'']$ will have been found when $\beta^j \leq \mu/(2L)$. Hence the number of iterations of the search will be bounded by

$$j_{max} = \frac{\log(\mu/(2L))}{\log\beta}.$$

This provides a bound on the computational effort per iteration of $O(\log(2L/\mu))$ function evaluations.

4. Adaptive convexity parameter μ. In computational tests (see next section) we see that the use of a strong convexity constant is very effective in reducing the number of iterations. But very seldom such constant is available. In this section we state an algorithm which uses a decreasing sequence of estimates for the value of the constant μ. We assume that a strong convexity parameter μ^* (possibly null) is given, and the method will generate a sequence $\mu_k \to \mu^*$, starting with $\mu_0 \in [\mu^*, \gamma_0)$. We still need an extra hypothesis: that the level set associated with x^0 is bounded. Note that since $f(\cdot)$ is convex, this is equivalent to the hypothesis that the optimal set is bounded. We begin by the algorithm as it may be implemented, and comment afterwards.

The algorithm iterations are the same as in Algorithm 2.1, using a parameter $\mu_k \geq \mu^*$. The parameter is reduced (we do this by setting $\mu_k = \max\{\mu^*, \mu_k/10\}$) in two situations:
• When $\gamma_k - \mu^* < \beta(\mu_k - \mu^*)$, where $\beta > 1$ is fixed (we used $\beta = 1.02$), meaning that γ_k is too close to μ_k.
• When it is impossible to satisfy the equation $\phi_+^*(\alpha) = f(x^{k+1})$ for $\alpha \in [0,1]$. By Lemma 2.6, this can only happen when $P = f(x^{k+1}) < \min_{x\in\mathbb{R}^n} \ell(x)$. This minimum is given by $\phi_{k+1}^*(1) = f(y^k) - \|\nabla f(y^k)\|^2/(2\mu_k)$, using expression (2.11). So, μ_k cannot be larger than

$$\tilde\mu = \frac{\|\nabla f(y^k)\|^2}{2(f(y^k) - f(x^{k+1}))}.$$

If $\mu_k > \tilde\mu$, we reduce μ_k.

Notation: Denote $\phi_{k+1}(x) = \phi_{k+1}(\alpha_k, x)$ and $\phi_{k+1}^* = \phi_{k+1}^*(\alpha_k)$.

Algorithm 4.1. *Algorithm with adaptive convexity parameter.*

```
Data: $x^0 \in \mathbb{R}^n$, $v^0 = x^0$, $\gamma_0 > 0$, $\beta > 1$, $\mu^* = \mu$, $\mu_0 \in [\mu^*, \gamma_0)$.
(we suggest $\mu_0 = \max\{\mu^*, \gamma_0/100\}$, $\beta = 1.02$, $\gamma_0 = L$ if $L$ is known.)
$k = 0$
REPEAT
    $d^k = v^k - x^k$.
    Choose $\theta_k \in [0,1]$ as in Algorithm 2.1.
```

```
$y^k = x^k + \theta_k d^k$.
IF $\nabla f(y^k) = 0$, then STOP with $y^k$ as an optimal solution.
Steepest descent step: $x^{k+1} = y^k - \nu \nabla f(y^k)$. If $L$ is known, $\nu \geq 1/L$.
IF $\gamma_k - \mu^* < \beta(\mu_k - \mu^*)$, THEN $\mu_k = \max\{\mu^*, \mu_k/10\}$.
Compute $\tilde{\mu} = \dfrac{\|\nabla f(y)\|^2}{2(f(y^k) - f(x^{k+1}))}$.
IF $\mu_k > \tilde{\mu}$, then $\mu_k = \max\{\mu^*, \tilde{\mu}/10\}$.
Compute $\alpha_k$ as the largest root of (2.2) with $\mu = \mu_k$.
Set $\mu_{k+1} = \mu_k$.
$\gamma_{k+1} = (1 - \alpha_k)\gamma_k + \alpha_k \mu_k$.
$v^{k+1} = \dfrac{1}{\gamma_{k+1}} \big((1 - \alpha_k)\gamma_k v^k + \alpha_k(\mu_k y^k - \nabla f(y^k))\big)$.
$k = k + 1$.
```

We now show the optimality of the algorithm, using an additional hypothesis.

Hypothesis.. Given $x^0 \in \mathbb{R}^n$, assume that the level set associated with x^0 is bounded, i.e.,

$$D = \sup\{\|x - y\| \mid f(x) \leq f(x^0), f(y) \leq f(x^0)\} < \infty.$$

Define $Q = \dfrac{D^2}{2}$.

LEMMA 4.2. *Consider $\gamma_0 > \mu^*$. Let x^* be an optimal solution of (P). Then, at all iteration of Algorithm 4.1,*

$$\phi_k(x^*) - f(x^*) \leq \frac{\gamma_0 + L}{\gamma_0 - \mu^*} Q(\gamma_k - \mu^*). \tag{4.1}$$

Proof. First let us prove (4.1) for $k = 0$. By definition of ϕ_0 and convexity of f with Lipschitz constant L for the gradient, we have

$$\begin{aligned} \phi_0(x^*) - f(x^*) &= f(x^0) - f(x^*) + \frac{\gamma_0}{2}\|x^* - x^0\|^2 \\ &\leq \frac{L + \gamma_0}{2}\|x^* - x^0\|^2 \\ &\leq (L + \gamma_0)Q. \end{aligned}$$

Thus, we can use induction. By (2.15) and (1.2), we have

$$\begin{aligned} \phi_{k+1}(x) &= (1 - \alpha_k)\phi_k(x) + \alpha_k \left(f(y^k) + \nabla f(y^k)^T(x - y^k) + \frac{\mu_k}{2}\|x - y^k\|^2\right) \\ &\leq (1 - \alpha_k)\phi_k(x) + \alpha_k \left(f(x) + \frac{\mu_k - \mu^*}{2}\|x - y^k\|^2\right). \end{aligned}$$

Thus,

$$\phi_{k+1}(x^*) - f(x^*) \leq (1 - \alpha_k)\left(\phi_k(x^*) - f(x^*)\right) + \alpha_k Q(\mu_k - \mu^*).$$

Using the induction hypothesis and the definition of γ_{k+1}, we have

$$\begin{aligned} \phi_{k+1}(x^*) - f(x^*) &\leq (1 - \alpha_k)\frac{\gamma_0 + L}{\gamma_0 - \mu^*}Q(\gamma_k - \mu^*) + \alpha_k Q(\mu_k - \mu^*) \\ &\leq \frac{\gamma_0 + L}{\gamma_0 - \mu^*}Q\left((1 - \alpha_k)\gamma_k + \alpha_k\mu_k - \mu^*\right) \\ &\leq \frac{\gamma_0 + L}{\gamma_0 - \mu^*}Q(\gamma_{k+1} - \mu^*), \end{aligned}$$

completing the proof. □

Now we prove a technical lemma, following a very similar result in Nesterov [5].

LEMMA 4.3. *Consider a positive sequence (λ_k). Assume that there exists $M > 0$ such that $\lambda_{k+1} \leq (1 - M\sqrt{\lambda_{k+1}})\lambda_k$. Then, for all $k > 0$,*

$$\lambda_k \leq \frac{4\lambda_0}{M^2}\frac{1}{k^2}.$$

Proof. Denote $a_k = \dfrac{1}{\sqrt{\lambda_k}}$. Since $\{\lambda_k\}$ is a decreasing sequence, we have

$$a_{k+1} - a_k = \frac{\sqrt{\lambda_k} - \sqrt{\lambda_{k+1}}}{\sqrt{\lambda_k \lambda_{k+1}}} = \frac{\lambda_k - \lambda_{k+1}}{\sqrt{\lambda_k \lambda_{k+1}}\left(\sqrt{\lambda_k} + \sqrt{\lambda_{k+1}}\right)} \geq \frac{\lambda_k - \lambda_{k+1}}{2\lambda_k\sqrt{\lambda_{k+1}}}.$$

Using the hypothesis,

$$a_{k+1} - a_k \geq \frac{\lambda_k - (1 - M\sqrt{\lambda_{k+1}})\lambda_k}{2\lambda_k\sqrt{\lambda_{k+1}}} = \frac{M}{2}.$$

Thus, $a_k \geq a_0 + \dfrac{Mk}{2} \geq \dfrac{Mk}{2}$ and consequently $\lambda_k \leq \dfrac{4}{M^2k^2}$, completing the proof. □

Now we can discuss how fast $(\gamma_k - \mu^*)$ goes to zero for estimating the rate of convergence of the algorithm.

LEMMA 4.4. *Consider $\gamma_0 > \mu$. Then, for all $k \geq 0$,*

$$\gamma_k - \mu^* \leq \frac{8\beta^2 L}{(\beta-1)^2}\frac{1}{k^2}(\gamma_0 - \mu^*).$$

Proof. By the definition of γ_{k+1},

$$\gamma_{k+1} - \mu^* = (1 - \alpha_k)(\gamma_k - \mu^*) + \alpha_k(\mu_k - \mu^*).$$

By the algorithm, $(\gamma_k - \mu^*) \geq \beta(\mu_k - \mu^*)$. Thus,

$$\begin{aligned} \gamma_{k+1} - \mu^* &\leq (1 - \alpha_k)(\gamma_k - \mu^*) + \frac{\alpha_k}{\beta}(\gamma_k - \mu^*) \\ &= \left(1 - \frac{\beta - 1}{\beta}\alpha_k\right)(\gamma_k - \mu^*). \end{aligned}$$

By Theorem 2.9

$$\alpha_k \geq \sqrt{\frac{\gamma_{k+1}}{2L}} \geq \sqrt{\frac{\gamma_{k+1} - \mu^*}{2L}},$$

thus

$$\gamma_{k+1} - \mu^* \leq \left(1 - \frac{\beta - 1}{\beta\sqrt{2L}}\sqrt{\gamma_{k+1} - \mu^*}\right)(\gamma_k - \mu^*).$$

Applying Lemma 4.3 with $\lambda_k = \gamma_k - \mu^*$, for $k = 1, 2, \ldots$, and $M = \dfrac{\beta - 1}{\beta\sqrt{2L}}$, we complete the proof. □

The next theorem discusses the complexity bound of the algorithm.

THEOREM 4.5. *Consider $\gamma_0 > \mu^* \geq 0$. Then Algorithm 4.1 generates a sequence (x^k) such that, for all $k > 0$,*

$$f(x^k) - f(x^*) \leq \frac{8\beta^2 QL(L+\gamma_0)}{(\beta-1)^2 k^2}.$$

Proof. By construction, $f(x^k) \leq \phi_k(x)$ for all $x \in \mathbb{R}^n$, in particular for x^*. Using this, Lemma 4.2 and then Lemma 4.4, we have:

$$f(x^k) - f(x^*) \leq \frac{(L+\gamma_0)}{\gamma_0 - \mu^*} Q(\gamma_k - \mu^*) \leq \frac{8\beta^2 QL(L+\gamma_0)}{(\beta-1)^2 k^2},$$

completing the proof. □

This lemma ensures that an error of $\varepsilon > 0$ for the final objective function value is achieved in $O(1/\sqrt{\varepsilon})$ iterations.

5. Numerical results.

In this section, we report the results of our computational experiments, comparing the variants of Algorithm 2.1 and Algorithm 4.1. The codes are written in MATLAB.

We solved 60 quadratic problems with Hessian matrix and initial point generated randomly, with space dimension from 50 to 10000, L from 100 to 10000 and $\mu = 1$.

We considered as stopping criterion the value of the objective function with $\varepsilon = 10^{-6}$.

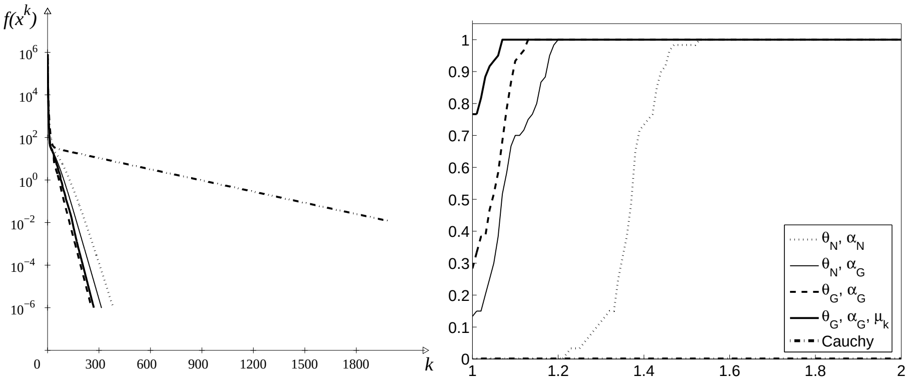

FIG. 5.1. *Case in which μ and L are given.*

We show in Figs. 5.1 to 5.3 three different situations. In each of them the figure at the left is the sequence of function values for a typical case generated at random with $\mu = 1$, $L = 1000$, $n = 100$. The figure at the right shows performance profiles for the 60 problems, using the number of iterations as criterion. In all cases we test four algorithms, indicated by:

- θ_N, α_N: Nesterov's original algorithm.
- θ_N, α_G: Nesterov's choice of θ and α computed by (2.2).
- θ_G, α_G: parameters chosen in our version of Algorithm 2.1.
- θ_G, α_G, μ_k: Algorithm 4.1

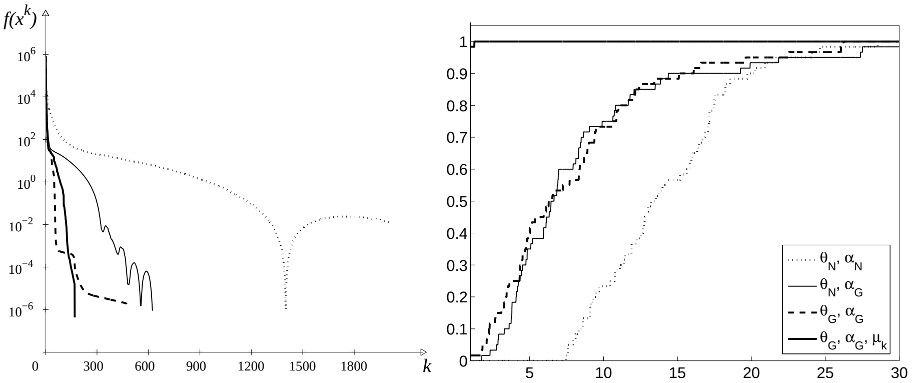

FIG. 5.2. *Case in which μ and L are unknown.*

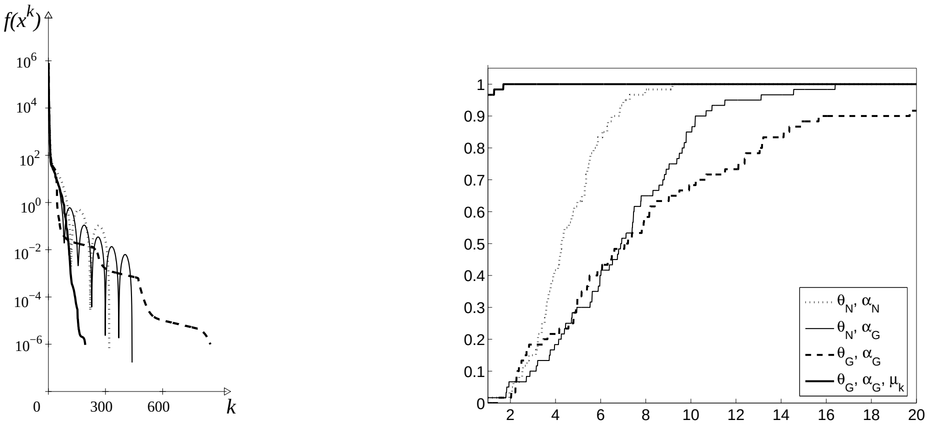

FIG. 5.3. *Case in which L is given.*

First test - Fig. 5.1.. : assuming that L and μ are known. We see that all algorithms behave similarly. If we assume that our methods use twice the time per iteration as Nesterov's (by performing two line searches), Nesterov has the best performance because it solves all problems with less than double of the number of iterations of the best method for each problem.

Second test - Fig. 5.2.. : L and μ are unknown. We used $\mu = 0$ and $\gamma_0 = 100L$, a large hint for the Lipschitz constant. We see from the figure that Algorithm 4.1 is the best. It is the fastest in 98% of the cases and Nesterov's method uses more than 14 times the best number of iteration in 50% of the cases (more than 7 times in all cases). The usage of the adaptive parameter μ_k improves very much the behavior of our method as well.

Third test - Fig. 5.3.. : L known and μ unknown. This case is interesting. Algorithm 4.1 is again the best, but our method θ_G, α_G performs very poorly. The reason is that γ_k decreased too fast, causing the functions $\phi_k(\cdot)$ to be too flat.

We conclude from these preliminary tests that Algorithm 4.1 has a good chance of being a good method. More tests, using non-quadratic functions are needed.

Appendix: interval reduction search. Let $g : [0,1] \to \mathbb{R}$ be a convex differentiable function. Assume that $g(0) = 0$, $g'(0) < 0$ and $g'(1) > 0$. We shall study the problem of finding $\theta \in [0,1]$ such that $g(\theta) \leq 0$ and $g'(\theta) \geq 0$.

This can be seen as a line search problem with an Armijo constant 0 and a Wolfe condition with constant 0, for which there are algorithms in the literature (see for instance [6, 2]).

In the present case, in which the function is convex, we can use a simple interval reduction algorithm, based on the following step:
Assume that three points $0 \leq A < \nu < B \leq 1$ are given, satisfying $g(A) \geq g(\nu) \leq g(B)$. Note that as consequences of the convexity of g, the interval $[A, B]$ contains a minimizer of g and $g'(B) \geq 0$. The problem can be solved by the following interval reduction scheme:

ALGORITHM 5.1. *Interval reduction algorithm*

```
WHILE $g(B) > 0$,
    Choose $\xi \in [0,1]$, $\xi \neq \nu$.
    Set $u = \min\{\nu, \xi\}$, $v = \max\{\nu, \xi\}$.
    If $g(u) \leq g(v)$, set $B = v$, $\nu = u$, else set $A = u$, $\nu = v$.
```

The initial value of ν and the values of ξ in each iteration may be such that u and v define the golden section of the interval $[A, B]$, and then the interval length will be reduced by a factor of $(\sqrt{5} - 1)/2 \approx 0.62$ in each iteration.

We can also choose ξ as the minimizer of the quadratic function through $g(A)$, $g(\nu)$, $g(B)$. In this case one must avoid the case $\xi = \mu$, in which ξ must be perturbed.

Acknowledgements. The authors would like to thank Masakazu Kojima and Mituhiro Fukuda for valuable comments.

REFERENCES

[1] D. P. Bertsekas, A. Nedić, and A. E. Ozdaglar. *Convex Analysis and Optimization.* Athena Scientific, Belmont, USA, 2003.

[2] J. J. Moré and D. J. Thuente. Line search algorithms with guaranteed sufficient decrease. Technical Report MCS-P330-1092, Mathematics and Computer Science Division, Argonne National Laboratory, 1992.

[3] A. S. Nemirovski and D. B. Yudin. *Problem Complexity and Method Efficiency in Optimization.* John Wiley, New York, 1983.

[4] Y. Nesterov. A method for unconstrained convex minimization problem with the rate of convergence $O(1/k^2)$. *Doklady AN SSSR (translated as Soviet Math Docl)*, 269:543 – 547, 1983.

[5] Y. Nesterov. *Introductory Lectures on Convex Optimization. A basic course.* Kluwer Academic Publishers, Boston, 2004.

[6] J. Nocedal and S. J. Wright. *Numerical Optimization.* Springer Series in Operations Research. Springer-Verlag, 1999.

[7] N. Shor. *Minimization Methods for Non-Differentiable Functions.* Springer Verlag, Berlin, 1985.