Handling Nonpositive Curvature in a Limited Memory Steepest Descent Method

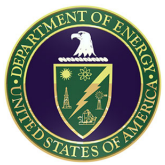
Frank E. Curtis, Lehigh University

joint work with

Wei Guo, Lehigh University

MOPTA Conference — Bethlehem, PA

14 August 2014



Outline

Outline

Context

Given $f : \mathbb{R}^n \to \mathbb{R}$ with large n, consider the unconstrained optimization problem

$$\min_{x \in \mathbb{R}^n} f(x).$$

We are interested in steepest descent methods:

Algorithm 1 Steepest Descent Framework

1: Input $x_0 \in \mathbb{R}^n$.
2: **for** $k \in \mathbb{N} := \{0, 1, 2, \dots\}$ **do**
3: Compute $g_k \leftarrow \nabla f(x_k)$.
4: Choose $\alpha_k \in (0, \infty)$.
5: Set $x_{k+1} \leftarrow x_k - \alpha_k g_k$.
6: **end for**

All that remains to be determined are the stepsizes $\{\alpha_k\}$.

Why steepest descent?

Is it because of widespread interest in "optimal" steepest descent methods?

Why steepest descent?

Is it because of widespread interest in "optimal" steepest descent methods?

- No. We are not interested in algorithm complexity analyses (yet).

Why steepest descent?

Is it because of widespread interest in "optimal" steepest descent methods?

- No. We are not interested in algorithm complexity analyses (yet).

Is it because we believe they can outperform quasi-Newton methods?

Why steepest descent?

Is it because of widespread interest in "optimal" steepest descent methods?

- No. We are not interested in algorithm complexity analyses (yet).

Is it because we believe they can outperform quasi-Newton methods?

- Not for convex problems. For those, we only hope to be competitive.

Why steepest descent?

Is it because of widespread interest in "optimal" steepest descent methods?

- No. We are not interested in algorithm complexity analyses (yet).

Is it because we believe they can outperform quasi-Newton methods?

- Not for convex problems. For those, we only hope to be competitive.

Then why?

- When function/gradient evaluations are relatively cheap, then it may be beneficial to "move quickly" as opposed to "sitting" and computing a step.
- Handling nonpositive curvature continues to be a pervasive difficulty. There may be more efficient ways of handling it in a steepest descent context.

Exploiting previously computed information

For a given $k \in \mathbb{N}_+ := \{1, 2, 3, \dots\}$, our strategy for computing α_k may involve

$$\{x_k, x_{k-1}, \dots, x_{k-m}\} \quad \text{and} \quad \{g_k, g_{k-1}, \dots, g_{k-m}\}.$$

Barzilai and Borwein (1988):

- $m = 1$
- "two-point step size gradient method"

Fletcher (2012):

- $m \geq 1$
- "limited memory steepest descent method"

In both cases:

- Ideas based on minimizing strictly convex quadratics.
- Ideas generalize when minimizing other convex functions.
- However, unclear how to handle nonpositive curvature.

Preview of contributions

- New strategies for handling nonpositive curvature in steepest descent.
- Consider both BB-type and LMSD methods. (Former is special case of latter.)
- Ideas based on employing cubic models when nonpositive curvature is present.
- Globalization is straightforward with nonmonotone line search.
- Maintain local convergence properties near strict local minimizers.
- Numerical experiments are promising so far (though work is ongoing).

Outline

Main idea

For a given $k \in \mathbb{N}$, how should we choose α_k?

- Define the displacements

$$s_k := x_k - x_{k-1} \quad \text{and} \quad y_k := g_k - g_{k-1}.$$

(Recall the classical secant equation $H_k s_k = y_k$.)

- Consider the one-dimensional least-squares problems

$$\min_{\bar{q} \in \mathbb{R}} \tfrac{1}{2} \|(\bar{q} I) s_k - y_k\|_2^2 \quad \text{and} \quad \min_{\hat{q} \in \mathbb{R}} \tfrac{1}{2} \|s_k - (\hat{q}^{-1} I) y_k\|_2^2,$$

which have the unique solutions[1]

$$\bar{q}_k := \frac{s_k^T y_k}{s_k^T s_k} \quad \text{and} \quad \hat{q}_k := \frac{y_k^T y_k}{s_k^T y_k}.$$

Both $\bar{q} I$ and $\hat{q} I$ represent simple approximations of the Hessian $\nabla^2 f(x_k)$.

[1] For simplicity, assume here that $s_k \neq 0$, $y_k \neq 0$, and $s_k^T y_k \neq 0$.

Main idea (continued)

- Consider minimizing, along $s = -\alpha g_k$, the quadratic model given by

$$f(x_k) + g_k^T s + \tfrac{1}{2} s^T (q_k I) s \approx f(x_k + s).$$

- For $q_k = \bar{q}_k$ and $q_k = \hat{q}_k$, respectively, we obtain the stepsizes

$$\bar{\alpha}_k := \frac{1}{\bar{q}_k} = \frac{s_k^T s_k}{s_k^T y_k} \quad \text{and} \quad \hat{\alpha}_k := \frac{1}{\hat{q}_k} = \frac{s_k^T y_k}{y_k^T y_k}.$$

These represent the two BB stepsize alternatives.

Theoretical analyses and extensions

For strictly convex quadratics, R-linearly convergent / R-superlinear local rate:

- Barzilai and Borwein (1988)
- Raydon (1993)
- Dai and Liao (2002)

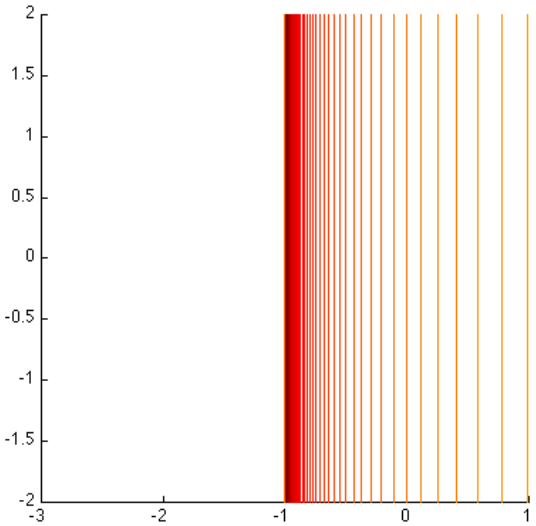
Algorithm extensions:

- Raydon (1997)
- Dai, Yuan, and Yuan (2002)
- Yuan (2006)
- De Asmundis, Serafino, Riccio, and Toraldo (2013)
- Xiao, Wang, and Wang (2010)
- Biglari and Solimanpur (2013)
- Kafaki and Fatemi (2013)

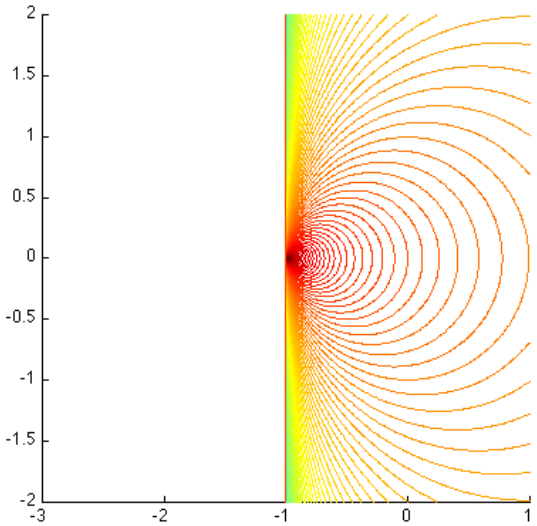
All except the last essentially ignore issues related to nonpositive curvature. The typical strategy, when $s_k^T y_k < 0$, is to set α_k to a predetermined constant.

Visualizing the BB stepsizes

Figure: Suppose $g_{k-1} = (-1, 0)$ and $\alpha_{k-1} = 1$ so that $s_k = -\alpha_{k-1} g_{k-1} = (1, 0)$. The contours illustrate the stepsize α_k as a function of the gradient g_k.



(a) $q_k = \bar{q}_k$



(b) $q_k = \hat{q}_k$

Key observations:

- Extremely different stepsizes when $s_k^T y_k > 0$ and vectors are ∼orthogonal.
- No contours in left-hand sides since $s_k^T y_k < 0$ leads to constant stepsize!

A closer look

Letting θ_k be the angle between s_k and y_k, we have

$$\bar{q}_k := \frac{s_k^T y_k}{s_k^T s_k} = \cos\theta_k \frac{\|y_k\|_2}{\|s_k\|_2}$$
$$\text{and} \quad \hat{q}_k := \frac{y_k^T y_k}{s_k^T y_k} = \frac{1}{\cos\theta_k} \frac{\|y_k\|_2}{\|s_k\|_2}.$$

Letting $y_k = u_k + v_k$ where u_k is the projection of y_k onto $\text{span}(s_k)$, we have

$$\bar{q}_k := \frac{s_k^T y_k}{s_k^T s_k} = \frac{s_k^T u_k}{s_k^T s_k}$$
$$\text{and} \quad \hat{q}_k := \frac{y_k^T y_k}{s_k^T y_k} = \frac{u_k^T u_k + v_k^T v_k}{s_k^T u_k} = \bar{q}_k + \frac{v_k^T v_k}{s_k^T u_k}.$$

Key observations:

- $|\bar{q}_k| \leq |\hat{q}_k|$, which implies $|\bar{\alpha}_k| \geq |\hat{\alpha}_k|$.
- The "bar" quantities only observe displacement of gradient along s_k, whereas the "hat" quantities observe the entire gradient displacement.

Basics of our strategy

Compute $\hat{q}_k$ (which seems better intuitively).

- If $\hat{q}_k > 0$, then set $\alpha_k \leftarrow 1/\hat{q}_k$.
- If $\hat{q}_k < 0$, then consider the cubic model

$$m_k(s) = f(x_k) + g_k^T s + \tfrac{1}{2}\hat{q}_k \|s\|_2^2 + \tfrac{1}{6} c_k \|s\|_2^3 \approx f(x_k + s).$$

  Choose $c_k > 0$ so that minimizing m_k along $s = -\alpha g_k$ yields a good stepsize.

- If $\hat{q}_k = 0$, then handle as a special case (see later).

Choosing the cubic term coefficient

Idea #1:

- Choosing c_k to minimize the least-squares error

$$\|\nabla m_k(-s_k) - g_{k-1}\|_2^2$$

- . . . leads to the choice

$$c_k \leftarrow \frac{1}{2\|s_k\|_2}(\bar{q}_k - \hat{q}_k).$$

Idea #2:

- Choosing c_k so the curvature of m_k at $-s_k$ along s_k is equal to $\bar{q}_k$, i.e.,

$$s_k^T \nabla^2 m_k(-s_k) s_k = \bar{q}_k \|s_k\|_2^2,$$

- . . . leads to the choice

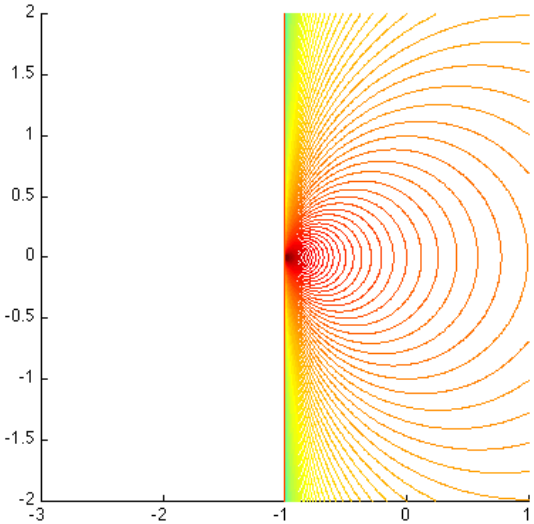
$$c_k \leftarrow \frac{1}{\|s_k\|_2}(\bar{q}_k - \hat{q}_k).$$

Key observations:

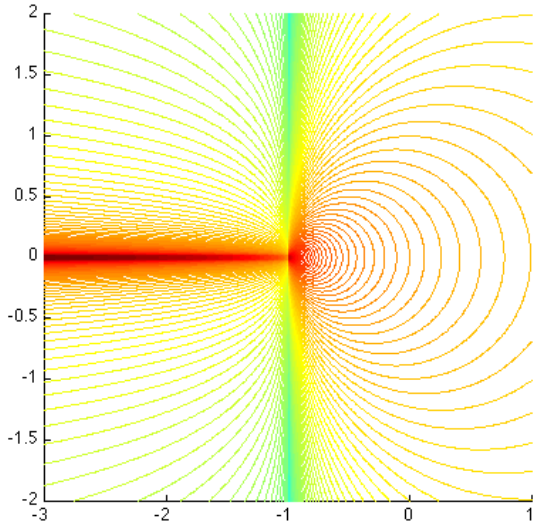
- Both suggest a similar strategy! (Coefficients only differ by a constant.)
- If $s_k^T y_k < 0$ and $s_k \nparallel y_k$, then $\bar{q}_k > \hat{q}_k$, so $c_k > 0$.

Visualizing our stepsizes

Figure: Suppose $g_{k-1} = (-1, 0)$ and $\alpha_{k-1} = 1$ so that $s_k = -\alpha_{k-1} g_{k-1} = (1, 0)$. The contours illustrate the stepsize α_k as a function of the gradient g_k.



(a) $q_k = \hat{q}_k$, constant stepsize for $q_k < 0$



(b) $q_k = \hat{q}_k$, $c_k \geq 0$

Special cases

Terminating if $g_k = 0$ and always choosing $\alpha_k > 0$ ensures $s_k \neq 0$ for all $k \in \mathbb{N}$.

- If $y_k = 0$, then function "appears affine", so set $\alpha_k \leftarrow \Omega$ (large constant).
- If $y_k \neq 0$, but $s_k^T y_k = 0$, then we have no useful information along the new direction $-g_k$, so set $\alpha_k \leftarrow \omega$ (small constant).
- If $y_k \neq 0$, $s_k^T y_k < 0$, and $s_k \parallel y_k$, then function "appears affine" along $-g_k$, so set $\alpha_k \leftarrow \Omega$ (large constant).

Observe consistency between these and the extremes in plot on previous slide.

Complete algorithm

Algorithm 2 BB-type Method with Cubic Regularization

Choose $(\omega, \Omega) \in \mathbb{R} \times \mathbb{R}$ satisfying $0 < \omega \leq \Omega$ and $c \in \mathbb{R}_+ := \{c \in \mathbb{R} : c > 0\}$.
Choose $x_0 \in \mathbb{R}^n$ and set $f_0 \leftarrow f(x_0)$.
Choose $\alpha_0 \in [\omega, \Omega]$.
Set $g_0 \leftarrow \nabla f(x_0)$.
if $g_0 = 0$ **then return** the stationary point x_0. **end if**
Set $x_1 \leftarrow x_0 - \alpha_0 g_0$ and $f_1 \leftarrow f(x_1)$.
Set $k \leftarrow 1$.
loop
    Set $g_k \leftarrow \nabla f(x_k)$, $s_k \leftarrow x_k - x_{k-1}$, and $y_k \leftarrow g_k - g_{k-1}$.
    if $g_k = 0$ **then return** the stationary point x_k. **end if**
    if $y_k = 0$ or $s_k^T y_k = -\|s_k\|_2 \|y_k\|_2 < 0$ **then**
        Set $\alpha_k \leftarrow \Omega$.
    else if $s_k^T y_k = 0$ **then**
        Set $\alpha_k \leftarrow \omega$.
    else
        Set $q_k \leftarrow y_k^T y_k / s_k^T y_k$.
        if $q_k > 0$ **then** Set $c_k \leftarrow 0$. **else** Set $c_k \leftarrow \frac{c}{\|s_k\|_2} \left(\frac{s_k^T y_k}{s_k^T s_k} - q_k \right)$. **end if**
        if $q_k > 0$ **then** Set $\alpha_k \leftarrow 1/q_k$. **else** Set $\alpha_k \leftarrow \frac{-q_k + \sqrt{q_k^2 + 2c_k \|g_k\|_2}}{c_k \|g_k\|_2}$. **end if**
        Replace α_k by its projection onto the interval $[\omega, \Omega]$.
    end if
    Set $x_{k+1} \leftarrow x_k - \alpha_k g_k$ and $f_{k+1} \leftarrow f(x_{k+1})$.
    Set $k \leftarrow k + 1$.
end loop

Outline

Key result

Consider the minimization of $\frac{1}{2}x^T Ax$ for $A \succ 0$.

Theorem (Finite termination of steepest descent)

Suppose that A has n distinct eigenvalues

$$0 < \lambda_1 < \lambda_2 < \cdots < \lambda_n.$$

If $\alpha_{k+i-1} \leftarrow \lambda_i^{-1}$ for all $i \in \{1, \ldots, n\}$, then $g_{k+n} = 0$.

Fletcher's main idea

Obtain stepsizes by approximating reciprocals of the eigenvalues of A.

- At x_k and with $m \geq 1$, consider the matrix of previous gradients

$$G_k := \begin{bmatrix} g_{k-m} & \cdots & g_{k-1} \end{bmatrix}.$$

- For all $j \in \{1, \ldots, m\}$, we have the following property:

$$x_k - x_{k-j} \in \text{span}\{g_{k-j}, Ag_{k-j}, A^2 g_{k-j}, \ldots, A^{j-1} g_{k-j}\}.$$

- This Krylov sequence provides m distinct eigenvalue estimates (Ritz values).
- In particular, with the QR-decomposition $G_k = Q_k R_k$, the Ritz values are

$$\text{eigenvalues of } T_k, \text{ where } T_k = Q_k^T A Q_k \text{ is tridiagonal.}$$

Computational efficiency

In fact, T_k can be obtained without access to A.

- Computing the partially extended Cholesky factorization

$$G_k^T \begin{bmatrix} G_k & g_k \end{bmatrix} = R_k^T \begin{bmatrix} R_k & r_k \end{bmatrix},$$

  we have

$$T_k = \begin{bmatrix} R_k & r_k \end{bmatrix} J_k R_k^{-1},$$

  where J_k is a matrix with only $2m$ nonzeros depending on previous stepsizes.
- With $m = 1$, we obtain the first BB alternative! That is, $T_k = \bar{q}_k$.

The second BB alternative can be obtained by computing harmonic Ritz values.

Fletcher's LMSD method

Main components:

- Construct G_k and compute factorization of $G_k^T \begin{bmatrix} G_k & g_k \end{bmatrix} \in \mathbb{R}^{m \times (m+1)}$.
- Construct $T_k \in \mathbb{R}^{m \times m}$ and compute its eigenvalues.
- Employ reciprocals of eigenvalues as stepsizes in next m iterations.

Issues for nonquadratics:

- T_k is not tridiagonal (but is upper Hessenberg).
- Eigenvalues are not necessarily real.
- Real eigenvalues are not necessarily positive.
- Globalization?

Our approach

We essentially employ the following procedure:

- Follow Fletcher's strategy of computing $\tilde{T}_k$ (by symmetrizing T_k).
- Compute Ritz and harmonic Ritz values (ordered largest to smallest):

$$\{\bar{q}_k, \bar{q}_{k+1}, \ldots, \bar{q}_{k+m-1}\}$$
$$\text{and} \quad \{\hat{q}_k, \hat{q}_{k+1}, \ldots, \hat{q}_{k+m-1}\}$$

- Under favorable conditions, these eigenvalues are interlaced! That is,

$$\hat{q}_{k+m-1} \leq \bar{q}_{k+m-1} \leq \cdots \leq 0 \leq \cdots \leq \bar{q}_k \leq \hat{q}_k$$

- For iterations $k, k+1, \ldots, k+m-1$, take the corresponding pair and apply a similar approach as in the $m = 1$ case.

Since this is ongoing work, the details are secret ;-)

Outline

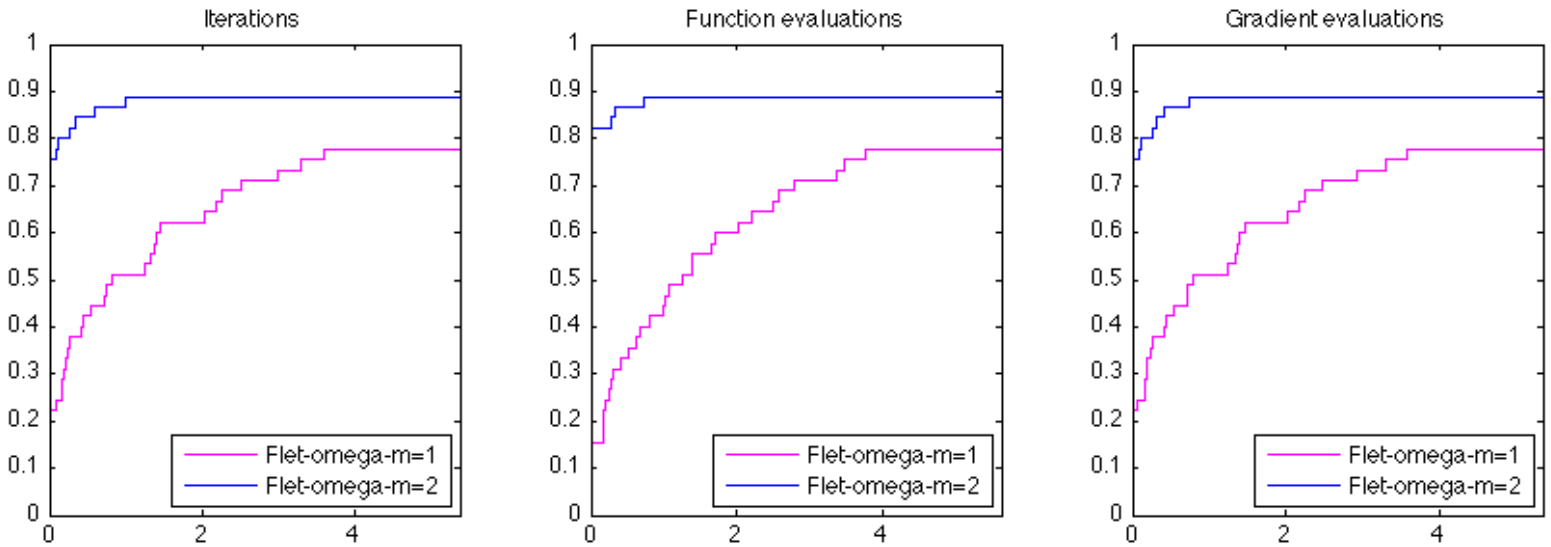
Implementation

- Matlab implementation of our approach versus...
- Matlab implementation of Fletcher's method.
  - Minor modifications to ensure consistency with our method.
  - For $m = 1$, reduces to a BB-type method.
  - For handling nonpositive curvature, perform line search and "clear the stack".
  - If nonpositive curvature, line search initialized with ω or Ω (two variants).
- Test only $m \in \{1, 2\}$ for now. (Working on larger m.)
- Ran all unconstrained CUTEst problems with $n \geq 3$, successful if/when

$$\|g_k\|_\infty \leq 10^{-4} \max\{\|g_0\|_\infty, 1\}.$$

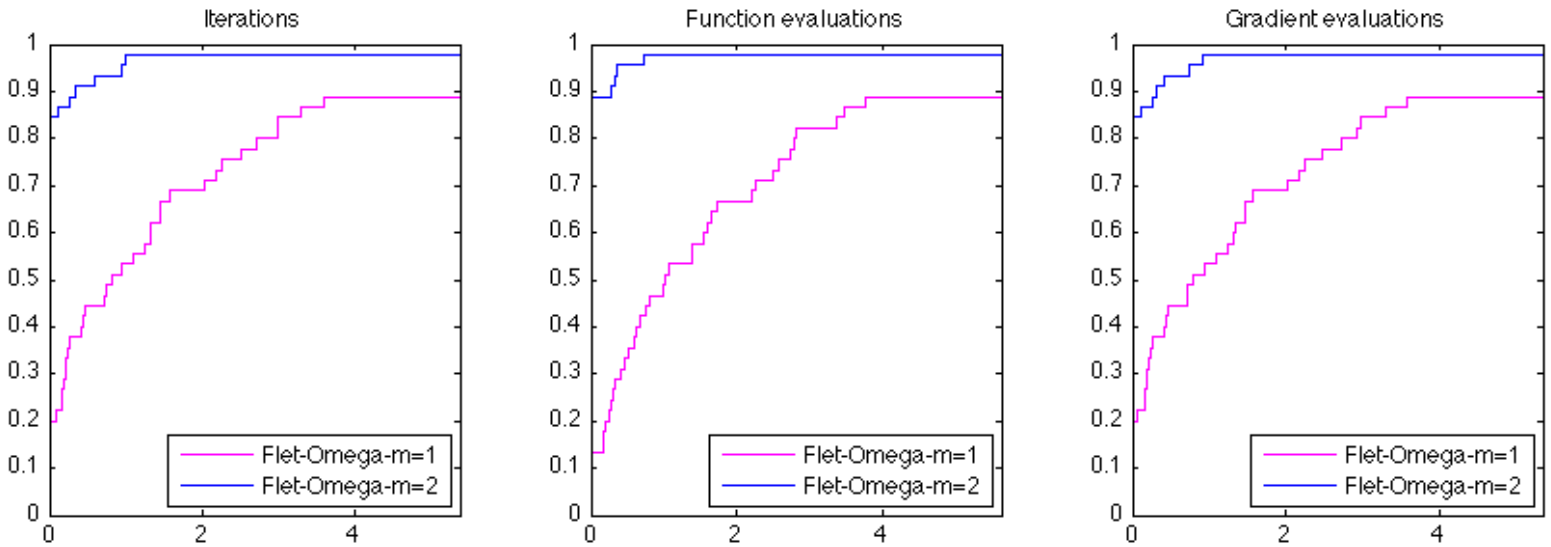
- Results only for problems on which at least one algorithm was successful...

Performance profiles: Fletcher with $\alpha_k \leftarrow \omega$ initially for line search



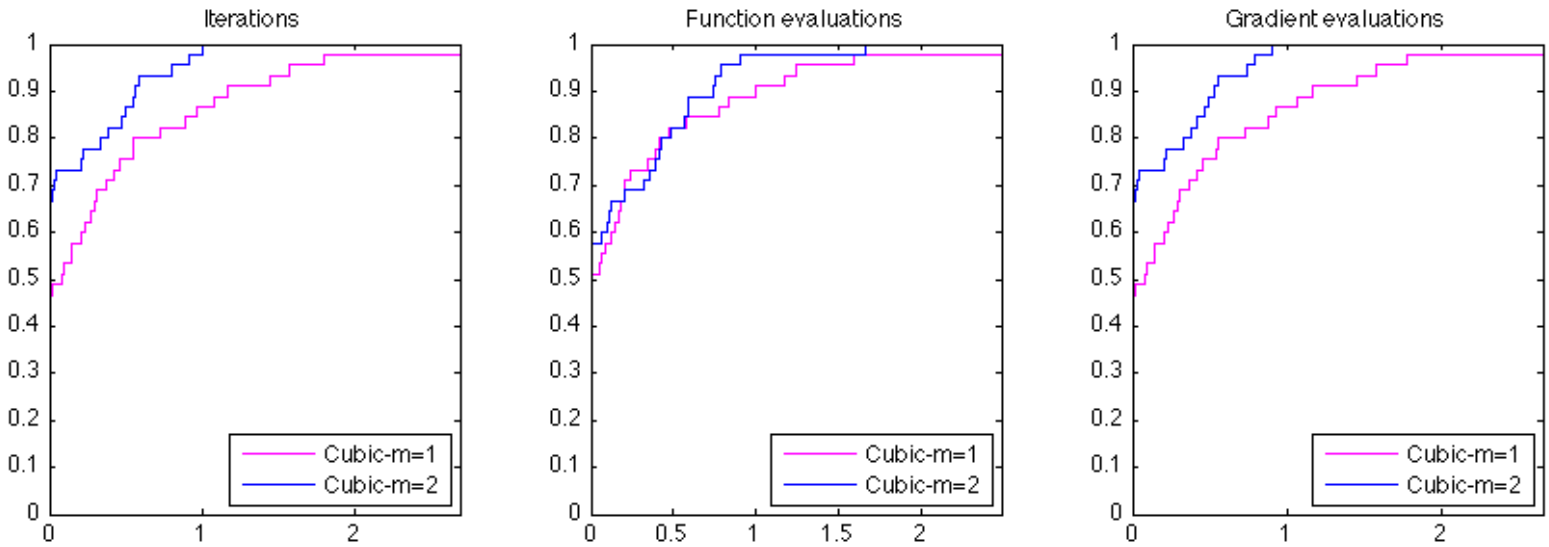
Larger m is beneficial in Fletcher's method.

Performance profiles: Fletcher with $\alpha_k \leftarrow \Omega$ initially for line search



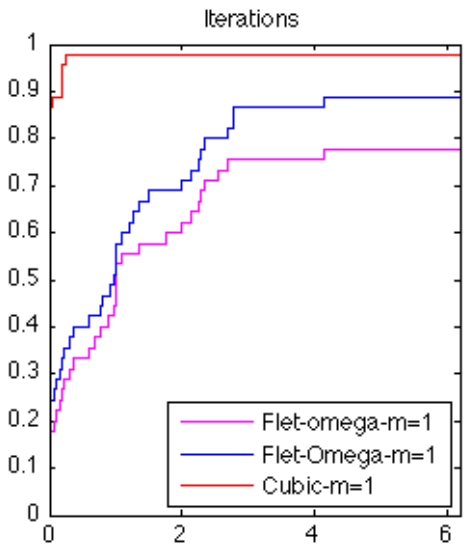
Larger m is still beneficial in Fletcher's method.

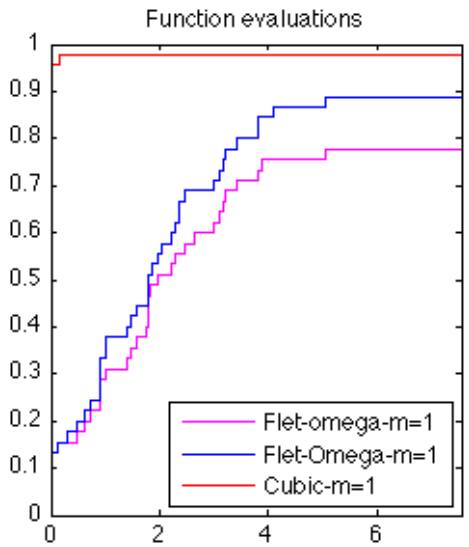
Performance profiles: Our method

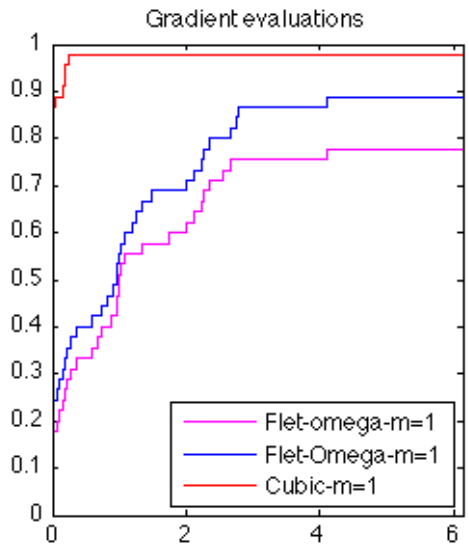


Larger m is beneficial in our method (though we believe we can improve further).

Performance profiles: $m = 1$

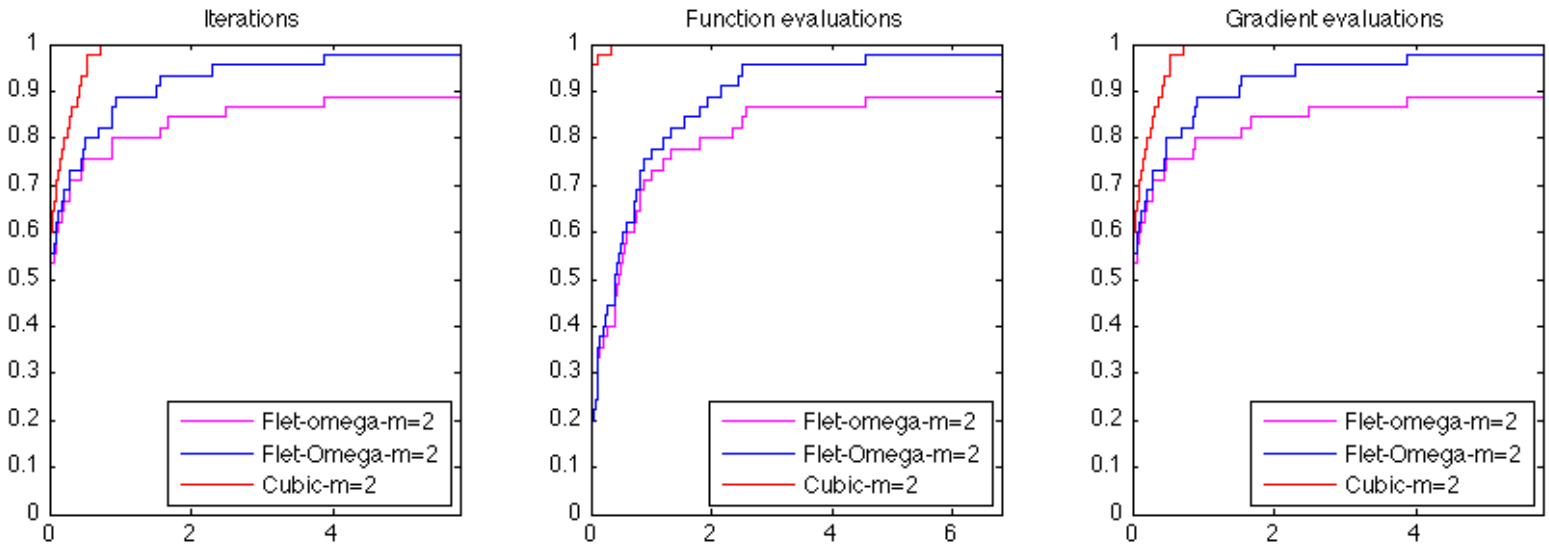






Cubic strategy is beneficial.

Performance profiles: $m = 2$



Cubic strategy is still beneficial (though still working on larger m).

Outline

Contributions

- New strategies for handling nonpositive curvature in steepest descent.
- Consider both BB-type and LMSD methods. (Former is special case of latter.)
- Ideas based on employing cubic models when nonpositive curvature is present.
- Globalization is straightforward with nonmonotone line search.
- Maintain local convergence properties near strict local minimizers.
- Numerical experiments are promising so far (though work is ongoing).