NonOpt: **Non**(-linear/-smooth/-convex) **Opt**imizer

Frank E. Curtis, Lehigh University

presented at

INFORMS Annual Meeting

Anaheim, California, USA

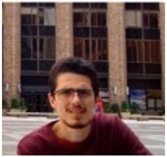
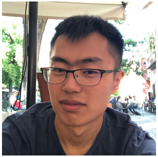
October 24, 2021

References







Outline

Outline

Why?!

Do we need *complicated* “non” optimization software?

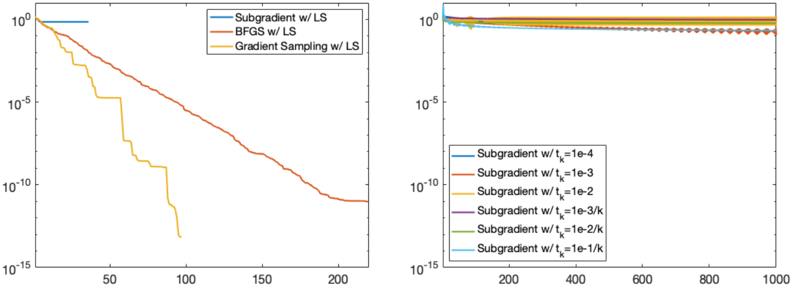
- Many problems are nonlinear, nonsmooth, and nonconvex,
- . . . but people say these can be solved with *simple* algorithms.

For example, trend is to analyze/use:

- subgradient method
- BFGS with line search

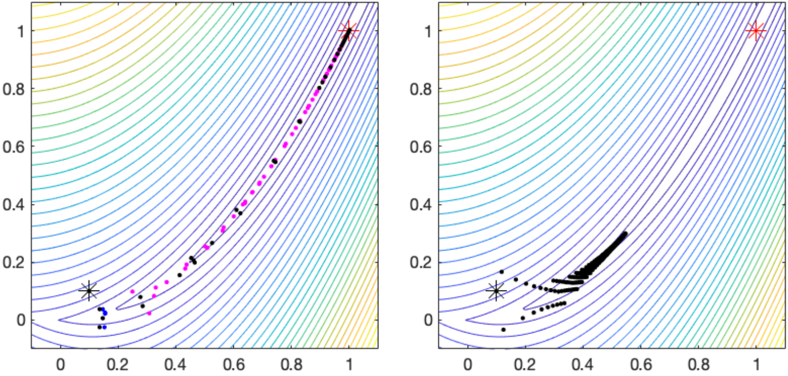
Nonsmooth Rosenbrock

$$f(x) = 8|x_1^2 - x_2| + (1 - x_1)^2$$



Nonsmooth Rosenbrock

$$f(x) = 8|x_1^2 - x_2| + (1 - x_1)^2$$



NonOpt (https://github.com/frankecurtis/NonOpt)

NonOpt is an open-source C++ software package to solve

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

where $f : \mathbb{R}^n \to \mathbb{R}$ is

- locally Lipschitz,
- continuously differentiable over $\mathcal{D}_f$ with full measure in $\mathbb{R}^n$,
- (potentially) nonsmooth, and
- (potentially) nonconvex.[†]

The code is designed to be extensible.

- Ideas borrowed from Ipopt.
- Additional features (e.g., constraints) forthcoming.

[†] Theory requires f to be weakly semismooth.

Central tenets of NonOpt

Quasi-Newton methods are surprisingly effective for “non” optimization.

- Self-correcting updates

However, they must be guided with cutting planes and/or gradient sampling.

- Point sets are critical for nonsmooth optimization.
- QP subproblems need to be solved.
- IMPORTANT: Specialized QP solvers, gradient aggregation, and inexact subproblem solutions mean that added per-iteration cost can be negligible compared to “simple” algorithms.

Outline

Search direction computation

At $x_k \in \mathbb{R}^n$, a smooth optimization algorithm computes $d_k \leftarrow x_k^* - x_k$, where

$$x_k^* \in \arg\min_{x \in \mathbb{R}^n} \; f(x_k) + \nabla f(x_k)^T (x - x_k) + \tfrac{1}{2}(x - x_k)^T H_k (x - x_k)$$
$$\text{s.t. } \|x - x_k\| \le \delta_k.$$

For nonsmooth f, with sets of points, scalars, and (sub)gradients

$$\{x_{k,j}\}_{j=1}^m, \quad \{f_{k,j}\}_{j=1}^m, \quad \text{and} \quad \{g_{k,j}\}_{j=1}^m,$$

NonOpt solves the primal subproblem

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \; & \left(\max_{j \in \{1,\dots,m\}} \{f_{k,j} + g_{k,j}^T (x - x_{k,j})\} + \tfrac{1}{2}(x - x_k)^T H_k (x - x_k) \right) \\ \text{s.t. } & \|x - x_k\| \le \delta_k. \end{aligned} \tag{P}$$

Examples

Multiple types of algorithms involve subproblems of the form

$$\min_{x\in\mathbb{R}^n} \left(\max_{j\in\{1,\dots,m\}} \{f_{k,j} + g_{k,j}^T(x - x_{k,j})\} + \tfrac{1}{2}(x - x_k)^T H_k (x - x_k) \right) \quad \text{s.t. } \|x - x_k\| \leq \delta_k. \tag{P}$$

Bundle methods: (Schramm & Zowe, 1992)

- $\{x_{k,j}\}$ from previous / current iterates and trial points
- $\{f_{k,j}\}$ with $f_{k,j} = \min\{f(x_{k,j}), f(x_k) + g_{k,j}^T(x_{k,j} - x_k) - c\|x_{k,j} - x_k\|_2^2\}$
- $\{g_{k,j}\}$ with $g_{k,j} \in \partial f(x_{k,j})$

Gradient sampling methods: (Burke, Lewis, & Overton, 2005)

- $\{x_{k,j}\}$ from current iterate and randomly sampled points
- $\{f_{k,j}\}$ with $f_{k,j} = f(x_k)$
- $\{g_{k,j}\}$ with $g_{k,j} = \nabla f(x_{k,j})$

Dual subproblem

The primal subproblem is nonsmooth:

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \ & \left(\max_{j \in \{1,\dots,m\}} \{f_{k,j} + g_{k,j}^T (x - x_{k,j})\} + \tfrac{1}{2}(x - x_k)^T H_k (x - x_k) \right) \\ \text{s.t. } & \|x - x_k\| \leq \delta_k. \end{aligned} \tag{P}$$

With $G_k \leftarrow [g_{k,1} \ \cdots \ g_{k,m}]$, it is typically more efficient to solve the dual

$$\begin{aligned} \sup_{(\omega,\gamma) \in \mathbb{R}_+^m \times \mathbb{R}^n} \ & -\tfrac{1}{2}(G_k\omega + \gamma)^T W_k (G_k\omega + \gamma) + b_k^T \omega - \delta_k \|\gamma\|_* \\ \text{s.t. } & \mathbb{1}_m^T \omega = 1. \end{aligned} \tag{D}$$

The primal solution can then be recovered by

$$x_k^* \leftarrow x_k - W_k \underbrace{(G_k \omega_k + \gamma_k)}_{\tilde{g}_k}.$$

NonOpt has specialized active-set QP solver for (D) based on Kiwiel, 1985.

Self-correcting properties of BFGS updates with $\{(s_k, v_k)\}$

Theorem 1 (Byrd & Nocedal, 1989)

Suppose that, for all k, there exists $\{\eta, \theta\} \subset \mathbb{R}_{++}$ *such that*

$$\eta \leq \frac{s_k^T v_k}{\|s_k\|_2^2} \quad \text{and} \quad \frac{\|v_k\|_2^2}{s_k^T v_k} \leq \theta. \tag{KEY}$$

Then, for any $p \in (0,1)$, *there exist constants* $\{\iota, \kappa, \lambda\} \subset \mathbb{R}_{++}$ *such that, for any* $K \geq 2$, *the following relations hold for at least* $\lceil pK \rceil$ *values of* $k \in \{1, \dots, K\}$:

$$\iota \leq \frac{s_k^T H_k s_k}{\|s_k\|_2 \|H_k s_k\|_2} \quad \text{and} \quad \kappa \leq \frac{\|H_k s_k\|_2}{\|s_k\|_2} \leq \lambda.$$

Corollary 2

Suppose the conditions of Theorem 1 hold. Then, for any $p \in (0,1)$, *there exist constants* $\{\mu, \nu\} \subset \mathbb{R}_{++}$ *such that, for any* $K \geq 2$, *the following relations hold for at least* $\lceil pK \rceil$ *values of* $k \in \{1, \dots, K\}$:

$$\mu \|g_k\|_2^2 \leq g_k^T W_k g_k \quad \text{and} \quad \|W_k g_k\|_2^2 \leq \nu \|g_k\|_2^2$$

Algorithm NonOpt Algorithm Framework

Choose $x_1 \in \mathbb{R}^n$.
Choose a symmetric positive definite $W_1 \in \mathbb{R}^{n \times n}$.
Choose $\alpha \in (0,1)$
for all $k \in \mathbb{N} := \{1, 2, \dots\}$ **do**
    Solve (P)–(D) such that setting

$$G_k \leftarrow \begin{bmatrix} g_{k,1} & \cdots & g_{k,m} \end{bmatrix},$$
$$s_k \leftarrow -W_k(G_k\omega_k + \gamma_k),$$
$$\text{and} \;\; x_{k+1} \leftarrow x_k + s_k$$

    yields (potentially after a line search)

$$f(x_{k+1}) \leq f(x_k) - \tfrac{1}{2}\alpha(G_k\omega_k + \gamma_k)^T W_k (G_k\omega_k + \gamma_k).$$

    Choose $y_k \in \mathbb{R}^n$.
    Set $\beta_k \leftarrow \min\{\beta \in [0,1] : v(\beta) := \beta s_k + (1-\beta) y_k$ satisfies (KEY)$\}$.
    Set $v_k \leftarrow v(\beta_k)$.
    Set

$$W_{k+1} \leftarrow \left(I - \frac{v_k s_k^T}{s_k^T v_k}\right)^T W_k \left(I - \frac{v_k s_k^T}{s_k^T v_k}\right) + \frac{s_k s_k^T}{s_k^T v_k}.$$

end for

Search direction computation

Algorithm 2 NonOpt Subproblem Solver

Require: $\{x_{k,j}\}_{j=1}^m$, $\{f_{k,j}\}_{j=1}^m$, $\{g_{k,j}\}_{j=1}^m$, and W_k

for $j = m, m+1, \ldots$ **do**

  Solve (D) for $(\omega_{k,j}, \gamma_{k,j})$.

  Set $s_{k,j} \leftarrow -W_k(G_{k,j}\omega_{k,j} + \gamma_{k,j})$.

  Set $x_{k,j+1} \leftarrow x_k + s_{k,j}$.

  if either

$$f(x_{k,j+1}) \leq f(x_k) - \tfrac{1}{2}\alpha(G_{k,j}\omega_{k,j} + \gamma_{k,j})^T W_k (G_{k,j}\omega_{k,j} + \gamma_{k,j})$$

  or

$$\begin{aligned} \|W_k(G_{k,j}\omega_{k,j} + \gamma_{k,j})\| &\leq \xi\delta_k, \\ \|G_{k,j}\omega_{k,j} + \gamma_{k,j}\| &\leq \xi\delta_k, \\ \text{and} \quad \|G_{k,j}\omega_{k,j}\| &\leq \xi\delta_k \end{aligned}$$

  then return

  else add $x_{k,j+1}$ and appropriate $(f_{k,j+1}, g_{k,j+1})$ to point set

  end if

end for

Additional features:

- Inexact subproblem solves
- (Sub)gradient aggregation

Outline

Basics

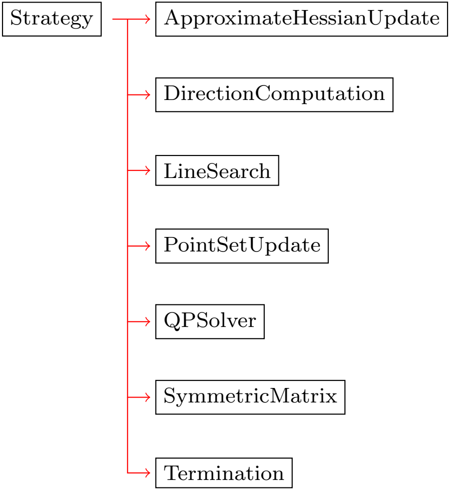
Low-level classes:

- Vector
- SymmetricMatrix
- Point
- PointSet

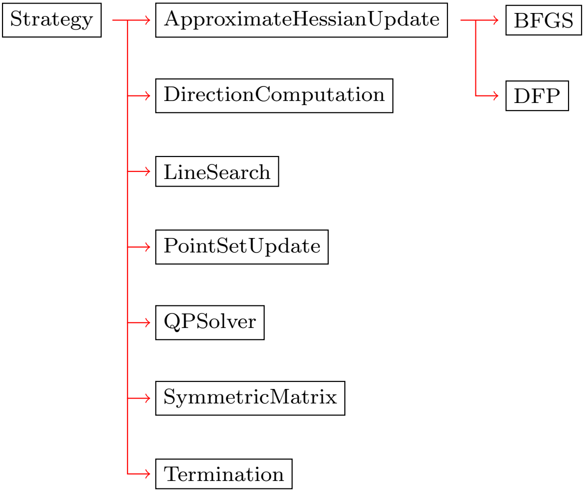
Solver classes:

- Reporter
- Options
- Quantities
- Strategies

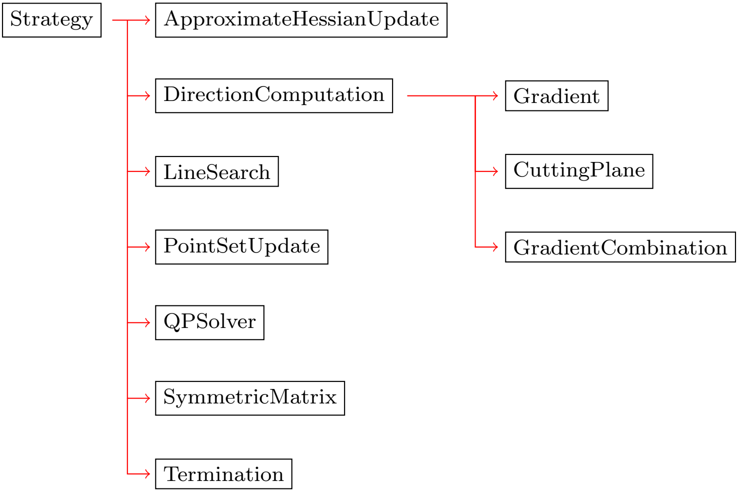
Strategies


Strategy
ApproximateHessianUpdate
DirectionComputation
LineSearch
PointSetUpdate
QPSolver
SymmetricMatrix
Termination

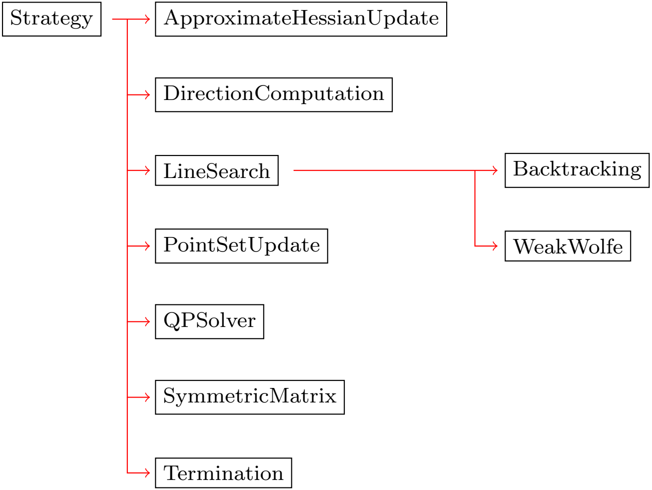
Strategies


Strategy
ApproximateHessianUpdate
BFGS
DFP
DirectionComputation
LineSearch
PointSetUpdate
QPSolver
SymmetricMatrix
Termination

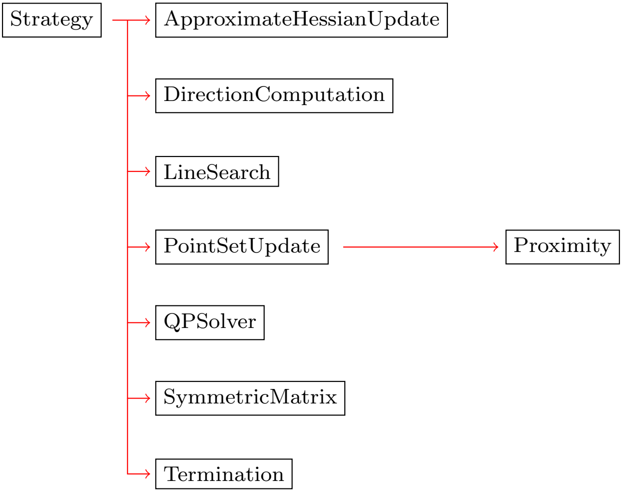
Strategies


Strategy
ApproximateHessianUpdate
DirectionComputation
LineSearch
PointSetUpdate
QPSolver
SymmetricMatrix
Termination
Gradient
CuttingPlane
GradientCombination

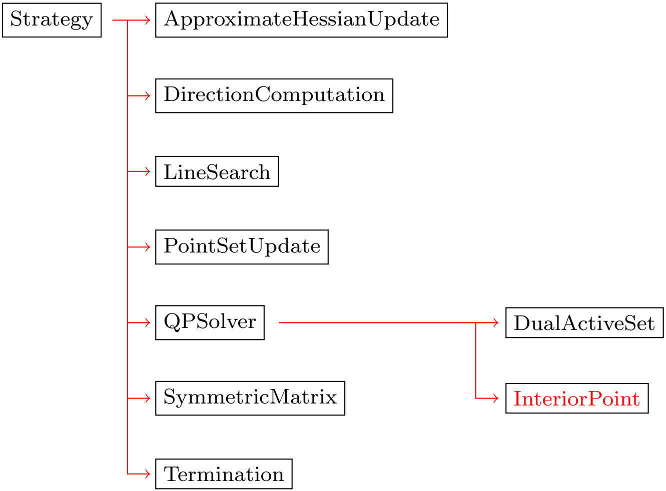
Strategies


Strategy
ApproximateHessianUpdate
DirectionComputation
LineSearch
Backtracking
WeakWolfe
PointSetUpdate
QPSolver
SymmetricMatrix
Termination

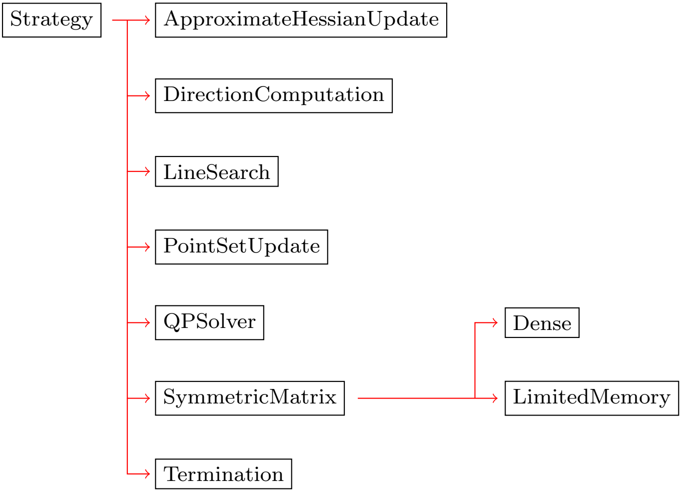
Strategies


Strategy
ApproximateHessianUpdate
DirectionComputation
LineSearch
PointSetUpdate
Proximity
QPSolver
SymmetricMatrix
Termination

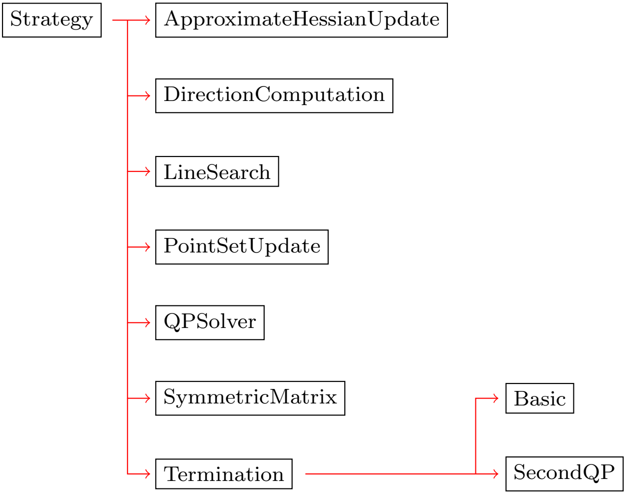
Strategies


Strategy
ApproximateHessianUpdate
DirectionComputation
LineSearch
PointSetUpdate
QPSolver
SymmetricMatrix
Termination
DualActiveSet
InteriorPoint

Strategies


Strategy
ApproximateHessianUpdate
DirectionComputation
LineSearch
PointSetUpdate
QPSolver
SymmetricMatrix
Termination
Dense
LimitedMemory

Strategies


Strategy
ApproximateHessianUpdate
DirectionComputation
LineSearch
PointSetUpdate
QPSolver
SymmetricMatrix
Termination
Basic
SecondQP

Brown Function 2

```
+----------------------------------------------------------+
|       NonOpt = Nonlinear/Nonconvex/Nonsmooth Optimizer      |
| NonOpt is released as open source code under the MIT License |
| Please visit http://coral.ise.lehigh.edu/frankecurtis/nonopt |
+----------------------------------------------------------+

This is NonOpt version 1.0rc2

Number of variables................. : 100
Approximate Hessian update strategy.. : BFGS
Derivative checker strategy.......... : FiniteDifference
Direction computation strategy....... : CuttingPlane
Line search strategy................. : WeakWolfe
Point set update strategy............ : Proximity
QP solver strategy................... : DualActiveSet
Symmetric matrix strategy............ : Dense
Termination strategy................. : Basic

EXIT: Stationary point found.

Objective............................ : 5.092179e-10
Objective (unscaled)................. : 5.092179e-10

Number of iterations................. : 73
Number of inner iterations........... : 97
Number of QP iterations.............. : 207
Number of function evaluations....... : 420
Number of gradient evaluations....... : 96

CPU seconds.......................... : 0.066282
CPU seconds in evaluations........... : 0.008750
CPU seconds in direction computations : 0.043033
CPU seconds in line searches......... : 0.010376
```

Numerical experiments with NonOpt

Experiments with randomized problems of the form ($n = 500$, $m = 400$):

$$\min_{x \in \mathbb{R}^n} \; g^T x + \tfrac{1}{2} x^T H x + \max_{i \in [m]} (a_i^T x + b_i)$$

Table: Results for GS-exact versus GS-inexact.

problem #	iters	qp-iters	f-evals	g-evals	objective	% ↓ qp-iters
0	551	2732	4176	29695	+1.239436e-03	
1	562	2661	4262	30286	+1.234747e-03	
2	567	3110	4278	30449	+1.064397e-03	
3	562	3051	4257	30449	+1.084893e-03	
4	616	3109	4669	32627	+1.098173e-03	
5	533	2798	4042	29155	+1.152184e-03	
6	462	2309	3514	25950	+1.201356e-03	
7	498	2213	3753	27147	+1.326716e-03	
8	538	2699	4067	28979	+1.090619e-03	
9	439	2313	3360	25298	+1.241042e-03	
0	548	2614	4136	29387	+1.237740e-03	-4.310055
1	543	2591	4093	29080	+1.239542e-03	-2.616570
2	559	2799	4228	30206	+1.062914e-03	-10.003279
3	554	2812	4195	30081	+1.085707e-03	-7.803190
4	611	3043	4596	32275	+1.099544e-03	-2.149265
5	523	2626	3972	28794	+1.148177e-03	-6.123893
6	460	2243	3483	25312	+1.206694e-03	-2.863402
7	498	2223	3733	26680	+1.328899e-03	+0.445497
8	539	2624	4069	29208	+1.093068e-03	-2.778601
9	441	2277	3381	25389	+1.245562e-03	-1.528737

Numerical experiments with NonOpt

Table: GS-inexact versus GS-inexact-agg.

problem #	iters	qp iters	f-evals	g-evals	objective	% improvement
0	548	2614	4136	29387	+1.237740e-03	
1	543	2591	4093	29080	+1.239542e-03	
2	559	2799	4228	30206	+1.062914e-03	
3	554	2812	4195	30081	+1.085707e-03	
4	611	3043	4596	32275	+1.099544e-03	
5	523	2626	3972	28794	+1.148177e-03	
6	460	2243	3483	25312	+1.206694e-03	
7	498	2223	3733	26680	+1.328899e-03	
8	539	2624	4069	29208	+1.093068e-03	
9	441	2277	3381	25389	+1.245562e-03	
0	553	2230	4160	29184	+1.238634e-03	-14.701136
1	552	2191	4134	28841	+1.237933e-03	-15.463027
2	571	2455	4259	29752	+1.063100e-03	-12.295404
3	568	2527	4249	29819	+1.082229e-03	-10.142980
4	602	2550	4487	31146	+1.101189e-03	-16.194883
5	523	2175	3906	27390	+1.148150e-03	-17.168026
6	459	1816	3436	24424	+1.201954e-03	-19.057806
7	492	1769	3671	25964	+1.328518e-03	-20.427147
8	537	2128	4004	27967	+1.092298e-03	-18.899768
9	434	1831	3251	23276	+1.245693e-03	-19.598184

Numerical experiments with NonOpt

Common test problems ($n = 1000$); NonOpt limited by time required by LMBM

Table: LMBM versus NonOpt

	LMBM			NonOpt			
name	iters	f-evals	$f(x)$	iters	qp-iters	f-evals	$f(x)$
MaxQ	21940	22808	+4.987830e-06	5443	5447	14283	+9.920069e-05
MxHilb	441	861	+6.166410e-03	142	142	674	+4.648503e-05
Chained_LQ	300	1824	-1.412780e+03	55	283	444	-1.412570e+03
Chained_CB3_1	291	1690	+1.998000e+03	119	272	818	+2.003039e+03
Chained_CB3_2	66	150	+1.998000e+03	91	252	606	+1.998000e+03
ActiveFaces	523	569	+1.376680e-14	15	788	394	+3.961526e-05
Brown_Function_2	493	4217	+2.136910e-09	51	288	405	+9.411906e-02
Chained_Mifflin_2	546	3892	-7.064510e+02	53	305	462	-7.061542e+02
Chained_Crescent_1	177	817	+3.681010e-08	39	60	232	+5.335065e-10
Chained_Crescent_2	903	9626	+1.369240e-04	52	276	456	+2.556368e-01
Test29_2	62	63	+9.815390e-01	1366	1487	8175	+7.677696e-02
Test29_5	1230	4563	+6.434430e-06	125	1563	777	+3.224645e-05
Test29_6	44	48	+2.000000e+00	60	292	478	+2.000982e+00
Test29_11	283	1336	+1.203580e+04	20	293	252	+1.207612e+04
Test29_13	3747	7092	+5.665460e+02	116	1629	1420	+5.700497e+02
Test29_17	962	2247	+3.574260e-03	25	269	271	+1.091778e-03
Test29_19	143	1012	+1.000000e+00	54	310	503	+1.002374e+00
Test29_20	277	3087	+5.000010e-01	83	280	762	+5.000672e-01
Test29_22	21	172	+1.966970e-06	30	279	362	+1.042170e-04
Test29_24	315	1945	+4.232150e-02	44	295	507	+1.099555e-01

Outline

NonOpt (https://github.com/frankecurtis/NonOpt)

NonOpt is an open-source C++ software package to solve

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

where $f : \mathbb{R}^n \to \mathbb{R}$ is

- locally Lipschitz,
- continuously differentiable over $\mathcal{D}_f$ with full measure in $\mathbb{R}^n$,
- (potentially) nonsmooth, and
- (potentially) nonconvex.

References

⋆ James V. Burke, Frank E. Curtis, Adrian S. Lewis, Michael L. Overton, and Lucas E. A. Sim oes.
Gradient Sampling Methods for Nonsmooth Optimization.
In *Numerical Nonsmooth Optimization*, chapter 6, pages 201–225. Springer, 2020.

⋆ Frank E. Curtis and Minhan Li.
Gradient Sampling Methods with Inexact Subproblem Solutions and Gradient Aggregation.
arXiv 2005.07822, 2020.

⋆ Frank E. Curtis, Tim Mitchell, and Michael L. Overton.
A BFGS-SQP Method for Nonsmooth, Nonconvex, Constrained Optimization and its Evaluation using Relative Minimization Profiles.
Optimization Methods and Software, 32(1):148–181, 2017.

⋆ Frank E. Curtis and Michael L. Overton.
A Sequential Quadratic Programming Algorithm for Nonconvex, Nonsmooth Constrained Optimization.
SIAM Journal on Optimization, 22(2):474–500, 2012.

⋆ Frank E. Curtis and Xiaocun Que.
An Adaptive Gradient Sampling Algorithm for Nonsmooth Optimization.
Optimization Methods and Software, 28(6):1302–1324, 2013.

⋆ Frank E. Curtis and Xiaocun Que.
A Quasi-Newton Algorithm for Nonconvex, Nonsmooth Optimization with Global Convergence Guarantees.
Mathematical Programming Computation, 7(4):399–428, 2015.

⋆ Frank E. Curtis, Daniel P. Robinson, and Baoyu Zhou.
A Self-Correcting Variable-Metric Algorithm Framework for Nonsmooth Optimization.
IMA Journal of Numerical Analysis, 40(2):1154–1187, 2020.