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

**Frank E. Curtis**, Lehigh University

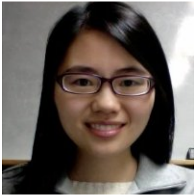
presented at

International Conference on Continuous Optimization

Berlin, Germany

August 5, 2019

# References


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

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

- ⋆ F. E. Curtis, D. P. Robinson, and B. Zhou.
  A Self-Correcting Variable-Metric Algorithm Framework for Nonsmooth Optimization.
  *IMA Journal of Numerical Analysis*, 10.1093/imanum/drz008, 2019.

# Outline

Motivation

Algorithm

NonOpt

Summary

# Outline

Motivation

Algorithm

NonOpt

Summary

# Why?!

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

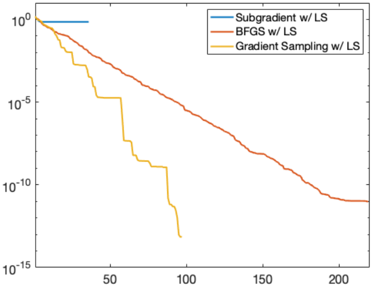
- Many problems are nonlinear, nonsmooth, and nonconvex,
- . . . but people say these can be solved with *simple* algorithms.

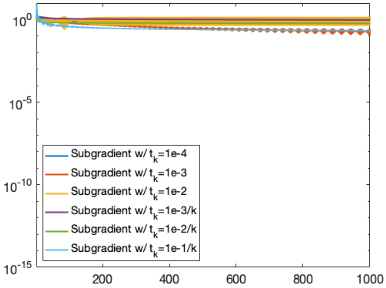
For example, trend is to analyze/use:

- subgradient method
- BFGS with line search

# Nonsmooth Rosenbrock

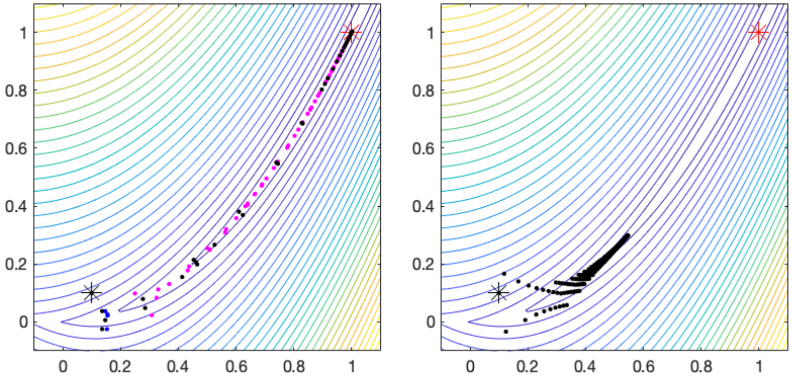
$$f(x) = 8|x_1^2 - x_2| + (1 - x_1)^2$$



# Nonsmooth Rosenbrock

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



# NonOpt

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 a full-measure subset of $\mathbb{R}^n$,
- (potentially) nonsmooth, and
- (potentially) nonconvex.†

The code is designed to be extensible.

- Ideas borrowed from Ipopt. (Thanks, Andreas!)
- 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 BFGS update
- Offers exactly the inequalities needed for convergence
- Careful radii updates; Curtis, Robinson, & Zhou, 2019

Quasi-Newton 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

Motivation

Algorithm

NonOpt

Summary

# Search direction computation

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

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

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

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

# Examples

Multiple types of algorithms involve subproblems of the form

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

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

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

  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** $j = m, m+1, \dots$ **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

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

  **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

Features being finalized before release:

- Inexact subproblem solves: Primal value $p_k^j$ and dual value $q_k^j$ yielding

$$p_k^j < 0 \quad \text{and} \quad \frac{q_k^j - q_k^0}{p_k^j - q_k^0} \geq \zeta \in (0,1)$$

- Gradient aggregation (new for gradient sampling!); Kiwiel, 1985

# Outline


Motivation

Algorithm

NonOpt

Summary

# Basics

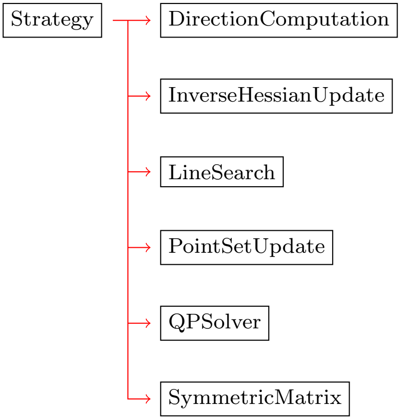
Low-level classes:

- Vector
- SymmetricMatrix
- Point
- PointSet

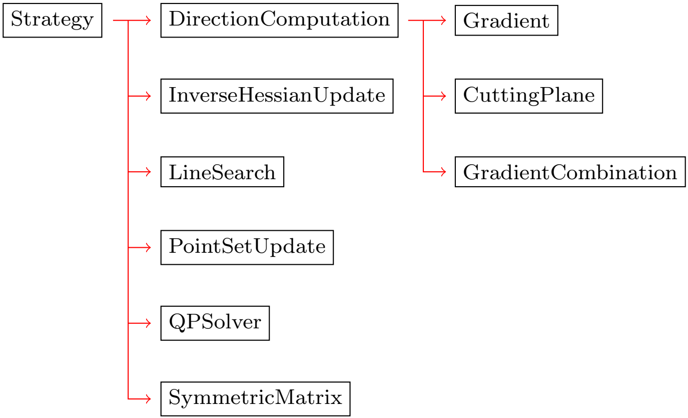
Solver classes:

- Reporter (similar to Journalist in Ipopt)
- Options
- Quantities
- Strategies

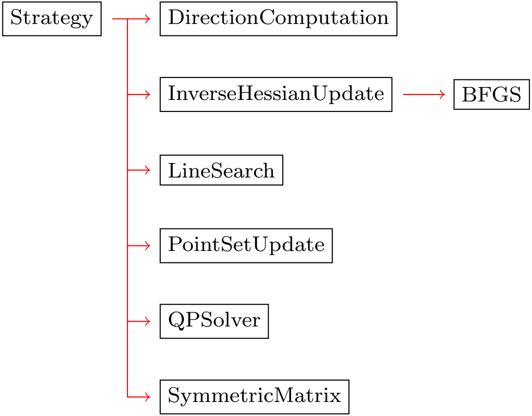
# Strategies


Strategy
DirectionComputation
InverseHessianUpdate
LineSearch
PointSetUpdate
QPSolver
SymmetricMatrix

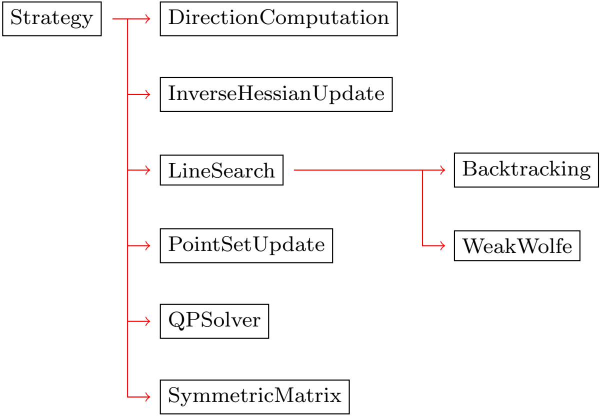
# Strategies


Strategy
DirectionComputation
Gradient
InverseHessianUpdate
CuttingPlane
LineSearch
GradientCombination
PointSetUpdate
QPSolver
SymmetricMatrix

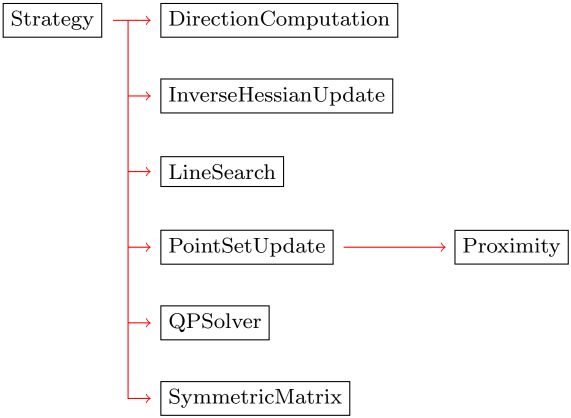
# Strategies


Strategy
DirectionComputation
InverseHessianUpdate
BFGS
LineSearch
PointSetUpdate
QPSolver
SymmetricMatrix

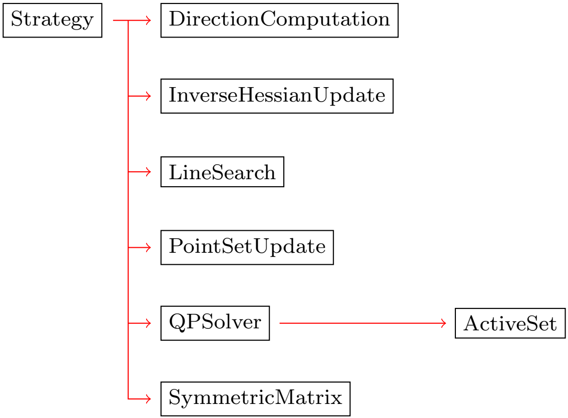
# Strategies


Strategy
DirectionComputation
InverseHessianUpdate
LineSearch
Backtracking
WeakWolfe
PointSetUpdate
QPSolver
SymmetricMatrix

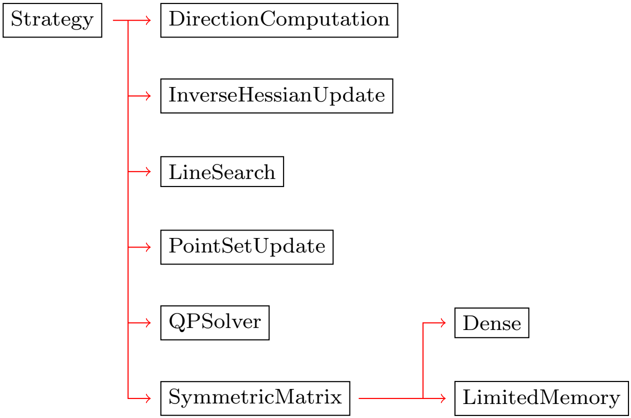
# Strategies


Strategy
DirectionComputation
InverseHessianUpdate
LineSearch
PointSetUpdate
Proximity
QPSolver
SymmetricMatrix

# Strategies


Strategy
DirectionComputation
InverseHessianUpdate
LineSearch
PointSetUpdate
QPSolver
ActiveSet
SymmetricMatrix

# Strategies


Strategy
DirectionComputation
InverseHessianUpdate
LineSearch
PointSetUpdate
QPSolver
SymmetricMatrix
Dense
LimitedMemory

# Nonsmooth test problems

| Name | Convex? | $f(x_0)$ | $f(x_*)$ |
|---|---|---|---|
| maxq | Yes | 2500.0 | 0.0 |
| mxhilb | Yes | 4.5 | 0.0 |
| chained lq | Yes | 49.0 | -69.3 |
| chained cb3 1 | Yes | 980.0 | 98.0 |
| chained cb3 2 | Yes | 980.0 | 98.0 |
| active faces | No | 3.9 | 0.0 |
| brown function 2 | No | 98.0 | 0.0 |
| chained mifflin 2 | No | 232.8 | -34.8 |
| chained crescent 1 | No | 292.3 | 0.0 |
| chained crescent 2 | No | 292.3 | 0.0 |

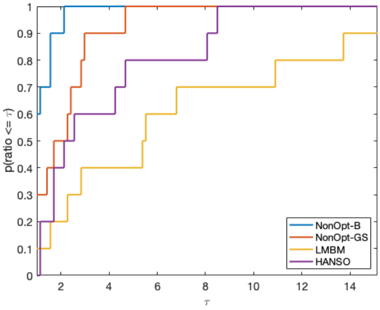
Codes:

- NonOpt
- LMBM: Karmitsa/Haarala, Miettinen, & Mäkelä, 2004 & 2007
- HANSO: Lewis & Overton, 2012; Burke, Lewis, & Overton, 2005

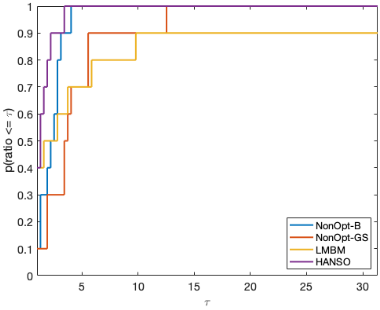
Termination:

- NonOpt terminates for all problems with $\|G_k \omega_k\| \leq 10^{-4}$.
- LMBM terminates 9/10 times due to small objective improvement.
- HANSO terminates 5/10 times due to small objective improvement.

# Numerical results, nonsmooth test problems




Iterations



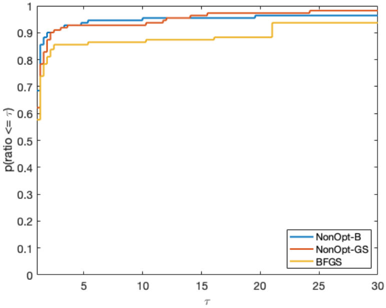

Function evaluations

# Chained Crescent 1

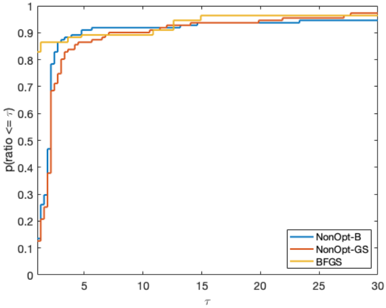
```
+-------------------------------------------------------------+
|            NonOpt = Nonsmooth Optimization Solver           |
| Please visit http://coral.ise.lehigh.edu/frankecurtis/nonopt |
+-------------------------------------------------------------+

Number of variables............. : 50
Direction computation strategy... : CuttingPlane
Inverse Hessian update strategy.. : BFGS
Line search strategy............ : WeakWolfe
Point set update strategy........ : Proximity
QP solver strategy.............. : ActiveSet
Symmetric matrix strategy........ : Dense

------------------------------------------------------------------------------------------------------------------------
  Iter.   Objective    Stat. Rad.    Trust Rad.   |Points|    QP Error    |G. Combo.|      |Step|      Stepsize    Correction
------------------------------------------------------------------------------------------------------------------------
      0  +2.9225e+02  +7.0000e-01  +7.0000e+04          0  +0.0000e+00  +7.0000e+00  +7.0000e+00  +5.0000e-01  +0.0000e+00
      1  +2.8775e+02  +7.0000e-01  +7.0000e+04          1  +0.0000e+00  +7.0000e+00  +1.7567e+00  +1.0000e+00  +0.0000e+00
...
     19  +5.2006e-08  +1.0000e-04  +7.0000e+00         12  +7.7753e-16  +1.7149e-04  +8.5751e-05  +1.0000e+00  +0.0000e+00
     20  +2.2457e-08  +1.0000e-04  +7.0000e+00         14  +2.2204e-16  +8.3220e-05  +4.1619e-05

EXIT: Stationary point found.

Objective....................... : 2.245677e-08

Number of iterations............ : 20
Number of inner iterations....... : 35
Number of QP iterations.......... : 16
Number of function evaluations... : 82
Number of gradient evaluations... : 35

CPU seconds..................... : 0.006477
CPU seconds in NonOpt........... : 0.005924
CPU seconds in evaluations....... : 0.000553
```

# Numerical results, CUTE/r/st (smooth) problems

Nonsmooth algorithms for solving difficult smooth problems?




Iterations




Function evaluations

# Outline

Motivation

Algorithm

NonOpt

Summary

# Summary

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 a full-measure subset of $\mathbb{R}^n$,
- (potentially) nonsmooth, and
- (potentially) nonconvex.


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

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

⋆ F. E. Curtis, D. P. Robinson, and B. Zhou.
A Self-Correcting Variable-Metric Algorithm Framework for Nonsmooth Optimization.
*IMA Journal of Numerical Analysis*, 10.1093/imanum/drz008, 2019.