AN INTRODUCTION TO DIRECT OPTIMIZATION METHODS

By Arnaud Bistoquet
03/30/04

INTRODUCTION

- **Goal**: minimization of a real-valued function \( f(x), x \in \mathbb{R}^n \)
- \( f(x) \) is differentiable
- \( \forall f(x) \) can be computed
- **Question**: Suppose explicit info about \( \forall f(x) \) is unavailable or untrustworthy; can this minimization problem be solved?
- Yes! **Direct Search Methods**: optimization of a function without using its derivatives

SIMPLE EXAMPLE: COMPASS SEARCH

- \( f(x) = 2D \) modified Broyden tridiagonal function:
  \[ f(x,y) = |(3-2x)x-2y+1|^{7/3} + |(3-2y)y-x+1|^{7/3} \]
- Minimization problem with only 2 variables
- Level curves of \( f(x) \): solution of the problem

SIMPLE EXAMPLE: COMPASS SEARCH (Con't)

**Algo**:
- Choose an initial point \((x_0, y_0)\)
- Choose an initial step size
- Try steps to the East, West, North, South:
  - If one of these steps yields to a smaller \( f(x,y) \) \( \Rightarrow \) new iterate \((x_k, y_k)\)
  - Otherwise: try again with a step half as long
- As \((x_k, y_k)\) approaches the solution, the algo reduces the length of the steps
- Stopping criteria: step length falls below a certain tolerance

SIMPLE EXAMPLE: COMPASS SEARCH (Con't)

**Move North**

Initial pattern
Initial point
New iterate
SIMPLE EXAMPLE: COMPASS SEARCH (Con’t)

Move East
New iterate

Move North
New iterate

Move North
New iterate
SIMPLE EXAMPLE: COMPASS SEARCH (Con’t)

Move East

New iterate

SIMPLE EXAMPLE: COMPASS SEARCH (Con’t)

Contract

New iterate

First observations on this algo:
- Easy to describe
- Easy to implement
- May quickly approach a minimizer
- But may be slow to detect it (it has to reduce the step size)

GENERAL REMARKS

- Problem: Optimization of a function
  - Data:
    - 1st order derivatives
    - 1st + 2nd order derivatives
    - No derivative info
  - Methods:
    - Gradient-based
    - Newton-based
    - Direct search

CONVERGENCE DEFINITIONS

- In optimization, the order refers to the order of associated derivatives:
  - Order:
    - 0th order method: Direct search
    - 1st order method: Gradient descent
  - Necessary 1st order optimality condition:
    \[ \nabla f(x) = 0 \iff \text{stationary point of } f(x) \]
CONVERGENCE DEFINITIONS (Cont)

- 1st order convergence of an optimization method:
  - one (or some) of the limit points of the iterates is a stationary point of f(x)
- Global convergence: convergence from an arbitrary starting point
- Local convergence: convergence when the initial point is close enough to a minimizer
- 1st order convergence convergence to a minimizer

LINE SEARCH METHODS

- Descent direction: d
  - If f(x) is differentiable: - ∇f(x)^T.d > 0
  - f(x+εd) = f(x) + ε ∇f(x)^T.d + o(ε) (ε small, >0)
    - x_k = x + ed reduces the value of f(x)
- Line search methods:
  - At iterate x_k, choose descent direction d_k and search along this direction for a point x_{k+1} = x_k + ε_k d_k that has a smaller objective value
    - d_k = - ∇f(x_k) Gradient descent method

CONVERGENCE ISSUES

- f(x_{k+1}) < f(x_k) convergence to a stationary point of f(x)
- Ex: f(x) = x^2, minimum: x_min = 0
  - x_0, x_1, ..., x_k = (-1)^k (1 + 2^{-k})
  - limits points: ±1
  - Steps are "too long" relative to the amount of decrease seen from one iterate to the next

CONVERGENCE ISSUES (Cont)

- Problems:
  - Poor choices of step lengths (too long or too short)
  - Poor choices of descent directions (too orthogonal to the direction of steepest descent)
  - f(x_{k+1}) < f(x_k) convergence to a stationary point of f(x)
- Constraints on the direction searches

GENERATING SET SEARCH (GSS)

- Not all direct search methods reliably find solutions
- Generating Set Search: identify a class of direct search methods which can be shown to have a mathematical foundation
  - Reliable in the sense of convergence result
- The algo must:
  - Have a search direction that is a descent direction
  - Avoid poor search directions to converge
  - Avoid poor choices of step lengths to a stationary point

GENERATING SETS

- The search directions of the algo belong to generating sets G
- Generating sets in R^n:
  - G = {d_1, ..., d_p}, p vectors (p>n+1) in R^n such that:
    - for any v in R^n, v = Σ λ_i d_i, i=1...p
  - Lemma: G generates R^n if and only if:
    - for any v in R^n (v<>0), there is d in G such that v^T.d > 0
    - So at every iteration k, there must be d_k in G_k such that:
      - -∇f(x_k)^T.d > 0, which means:
        - G_k is guaranteed to contain a descent direction whenever -∇f(x_k)<>0
ALGORITHM

- **Step 1**
  \( G_k \) = generating set for \( R^n \) satisfying:
  \[ \beta_{\min} \leq ||d|| \leq \beta_{\max} \]
  for all \( d \in G_k \) and \( \kappa(G_k) \geq \kappa_{\min} \)
  where \( \kappa(G_k) = \text{cosine measure of } G \)

- **Step 2**
  If there exists \( d_k \in G_k \) such that \( f(x_k + \Delta_k d_k) < f(x_k) - \rho(\Delta_k) \), then:
  \[ x_{k+1} = x_k + \Delta_k d_k \]
  (change the iterate)
  \[ \Delta_{k+1} = \phi_k \Delta_k \]
  (expand the step-length)

- **Step 3**
  Otherwise:
  \[ x_{k+1} = x_k \]
  (no change to the iterate)
  \[ \Delta_{k+1} = \theta_k \Delta_k \]
  (contract the step-length)
  if \( \Delta_{k+1} < \Delta_{\text{tol}} \), then terminate

STUDY OF THE CONSTRAINTS

- For all \( d \in G \), \( \beta_{\min} \leq ||d|| \leq \beta_{\max} \)
- \( \kappa(G_k) = \text{cosine measure of } G \)
- \( \kappa(G_k) = \min \{ \max \{ v^T d / (||v||.||d||) \} \} \)
- \( \kappa(G_k) = \kappa_{\text{min}} \)

GLOBAL CONVERGENCE

- **Theorem 1:**
  The GSS produces iterations satisfying for \( k \) in any subsequence of unsuccessful iteration:
  \[ \lim_{k \to \infty} \Delta_k = 0 \]
  Hypothesis used: properties of the function \( \rho(t) \)

- **Theorem 2:**
  For \( k \) in any subsequence of unsuccessful iteration:
  \[ \lim_{k \to \infty} \Delta_k = 0 \]
  \[ \lim_{k \to \infty} ||\nabla f(x_k)|| = 0 \]
  Hypothesis used: For all \( d \in G \), \( \beta_{\min} \leq ||d|| \leq \beta_{\max} \)
  \( \Rightarrow \) Global convergence of the GSS

EXTENSIONS

- There exist other techniques (than using the function \( \rho(t) \)) to ensure convergence of the GSS
- The method presented here is for the optimization without constraints but there are also direct search methods that take into account constraints

CONCLUSION

- Direct search methods are useful to optimize a function when there is no info on its derivatives
- Mathematical analysis of Direct Search Methods \( \Rightarrow \) reliable in the sense of convergence