Fundamentals of Unconstrained Optimization

WQD7011 Numerical Optimization

Introduction

- Remember the general form of optimization?
  - It can be classified according to:
    - Nature of objective function and constraints (linear, nonlinear, convex)
    - Number of variables (large or small)
    - Smoothness of the functions (differentiable or non-differentiable)
    - IMPORTANT: Problems that have constraints on variables and those that do not.

Constrained and Unconstrained Optimization

- Unconstrained optimization
  - no restrictions on the values of the variables
- Constrained optimization:
  - Constraints play an essential role
  - Simple bound such as 0 <= x_1 <= 100

Unconstrained Optimization

- Minimize an objective function that depends on real variables, with no restrictions at all on the values of these variables.

$$\min_x f(x)$$

- where $x \in \mathbb{R}^n$ is a real vector with $n \geq 1$ components and $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is a smooth function.

How to identify Solution? Introduction

- Local solution
  - A point at which the objective function is smaller than at all other feasible nearby points
- Global solution
  - The point with lowest function value among ALL feasible points.
  - Difficult to recognize and locate

How to identify Solution? Introduction

- Global minimizer of f, a point where the function attains its least value.
- Formal definition:
  - A point x^* is a *global minimizer* if $f(x^*) \leq f(x)$ for all x
  - where x ranges over all of $\mathbb{R}^n$ (or at least over the domain of interest to the modeler)
- Global minimizer can be difficult to find:
  - One possible reason: our knowledge of f is usually only local

How to identify Solution? Introduction

- Most algorithms are able to find only *local minimizer*, which is a point that achieves the smallest values of f in its neighborhood.
- Formal definition:
  - A point x^* is a *local minimizer* if there is a neighborhood N of x^* such that $f(x^*) \leq f(x)$ for all $x \in \mathrm{N}$.
  - A neighborhood of x^* is simply an open set that contains x^*.
  - If a point satisfies this definition => *weak local minimizer.*

How to identify Solution? Introduction

- Strict local minimizer, which is the outright winner in its neighborhood, is defined as:
  - A point x^* is a *strict local minimizer* (also called as *strong local minimizer*) if there is a neighborhood N of x^* such that $f(x^*) < f(x)$ for all $x \in \mathrm{N}$ with $x \neq x^*$
- Example:
  - Constant function $f(x) = 2$, every point x is a weak local minimizer, while the function $f(x) = (x-2)^4$ has a strict local minimizer at $x = 2$.

How to identify Solution? Introduction

- Isolated local minimizer – exotic type of local minimizer.
- Formal definition:
  - A point x^* is an *isolated local minimizer* if there is a neighborhood N of x^* such that x^* is the only local minimizer in N.
- Some strict local minimizers are not isolated but all isolated local minimizers are strict.

How to identify Solution? Introduction

- Fig. 2.2: a function with many local minimizer. In some optimization problems, can goes up to millions of local minima.

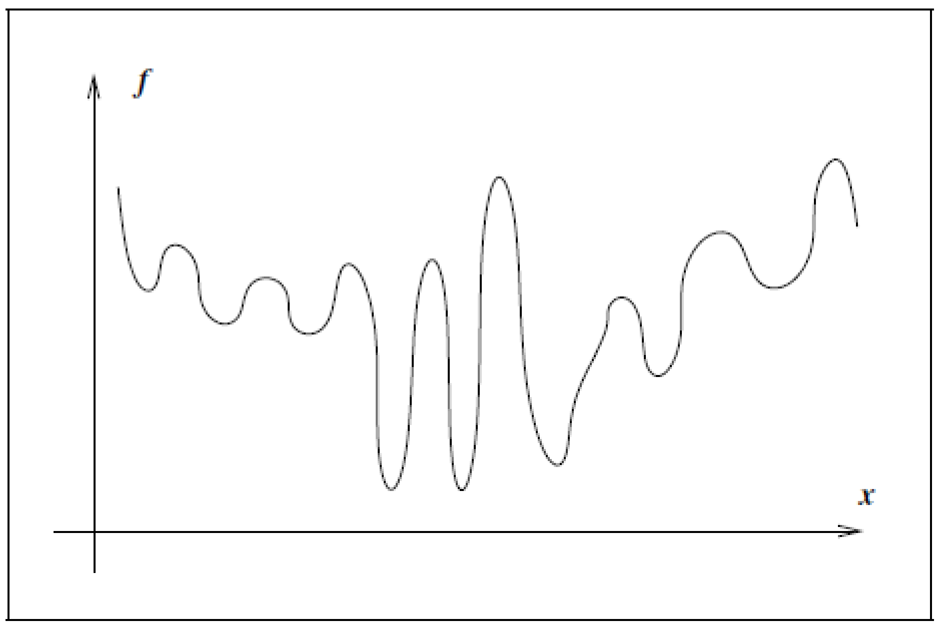

Figure 2.2 A difficult case for global minimization.

Recognizing a local minimum

- Only way to find out whether a point x^* is a local minimum is to examine all the points in its immediate vicinity, to make sure that non of them has a smaller function value.
- When function f is **smooth**, however, there are more efficient and practical ways to identify local minima.
- If f is twice continuously differentiable, we may be able to tell that x^* is a local minimizer (and possible strict one) by examining just the gradient $\nabla f(x^*)$ and the Hessian $\nabla^2 f(x^*)$.
- Mathematical tool used to study minimizer of smooth functions is **Taylor's theorem**.

Recognizing a local minimum

- Assumption on the function considered:
  - $x \in \mathbb{R}$
  - $f(x)$ is smooth
  - $f(x)$ is convex

Convex

- Convex applied to both sets and function
- A set $S \in R^n$ is a *convex* set if the straight line segment connecting any two points in S lies entirely inside S.
- For any two points $x \in S$ and $y \in S$, we have $\alpha x + (1-\alpha)y \in S$ for all $\alpha \in [0,1]$.
- Function f is a convex function if its domain S is convex set and if for any two points x and y in S, the following property is satisfied:

  $f(\alpha x + (1-\alpha)y) \leq \alpha f(x) + (1-\alpha)f(y)$ for all $\alpha \in [0,1]$

Taylor's Approximation around Local Minimum

- Refers to lecture demonstration.

Strategies for Optimization

- Generally, all algorithms start at an initial point x_0.
  - Either selected by experienced user or by algorithm (either systematically or arbitrary manner).
- Optimization algorithms generate a sequence of iterates $\{x_k\}_{k=0}^{\infty}$ that terminate when either no more progress can be made or when it seems that a solution point has been approximated with sufficient accuracy.
- How to move from one iterate to the next?
  - Algorithm use information about function f at x_k and possibly also information from earlier iterates.
  - Use this information to find a new iterate x_{k+1} with a lower function value than x_k.

Strategies for Optimization

- Two fundamental strategies :
  - Line search
  - Trust region

General Idea on Line Search Strategy

- Algorithm chooses a direction p_k and searches along this direction from the current iterate x_k for a new iterate with a lower function value/
- Distance to move along p_k can be found by approximately solving the following one dimensional minimization problem to find a step length α:

$$\min_{\propto > 0} f\ (x_k + \propto p_k)$$

- By solving the aforementioned problem, can derive the max benefit from direction p_k.
- Line search algorithm generates a limited number of trial step lengths until it finds one that loosely approximately the minimum of the problem.
- At new point, new search direction and step length are computed, and process repeated.

General Idea on Trust Region

- Information gathered about f is used to construct a *model function* m_k whose behavior near the current point x_k is similar to that of the actual objective function f.
- Because the model m_k may not posses a good approximation when it is far away from x_k, TRM restricts the search for a minimizer for m_k to some region around x_k.
- In other words, we find the candidate step p by approximately solving the following sub-problem:
- $\min_p m_k(x_k + p)$, where $x_k + p$ lies inside the trust region.

ONLINE EXERCISES