凸优化

11. Search Based Algorithms

李 力
清华大学
Email: li-li@tsinghua.edu.cn

2009-2021

11.0. 提纲

11.1. Overview

11.2. Zero-Order Optimization

11.3. First-Order Optimization

11.4. Second-Order Optimization

11.5. DIRECT Algorithm

11.1 Overview

Optimization algorithms tend to be iterative procedures. Starting from a given point $\mathbf{x}^0$, they generate a sequence $\{\mathbf{x}^k\}$ of iterates (or trial solutions) that converge to a "solution" – or at least they are designed to be so.

Recall that scalars $\{x^k\}$ converges to 0 if and only if for all real numbers $\varepsilon > 0$ there exists a positive integer K such that

$$|x^k| < \varepsilon \quad \text{for all } k \geq K.$$

Then $\{\mathbf{x}^k\}$ converges to solution $\mathbf{x}^*$ if and only if $\{\|\mathbf{x}^k - \mathbf{x}^*\|\}$ converges to 0.

We study algorithms that produce iterates according to

- well determined rules–Deterministic Algorithm
- random selection process–Randomized Algorithm.

The rules to be followed and the procedures that can be applied depend to a large extent on the characteristics of the problem to be solved.

11.1 Overview

选择求解方法的指标

- time/space complexity，因此运筹和计算复杂性理论相关
- effectiveness

搜索策略包括

i）确定策略deterministic（e.g., gradient-based methods, simplex method, enumeration）

ii）随机策略stochastic（e.g., sampling）

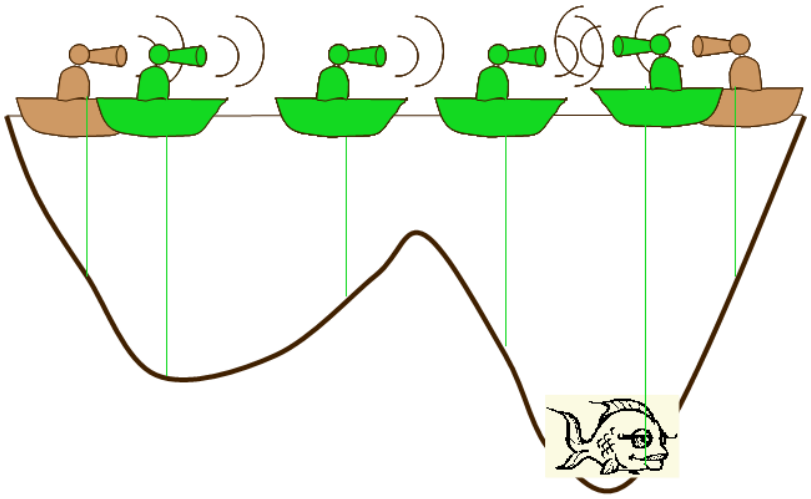
iii）启发式混合策略heuristically mixed（e.g., genetic algorithms, particle swarm optimization, simulated annealing）

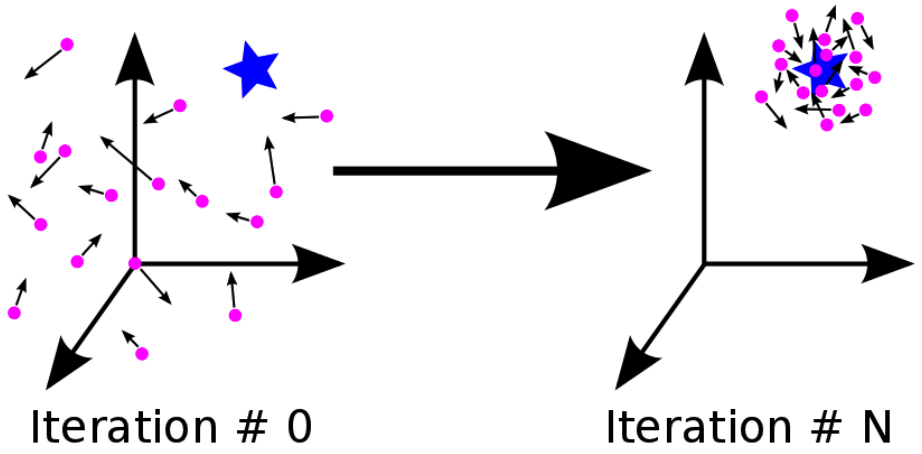
11.1 Overview

Particle Swarm Optimization

PSO算法

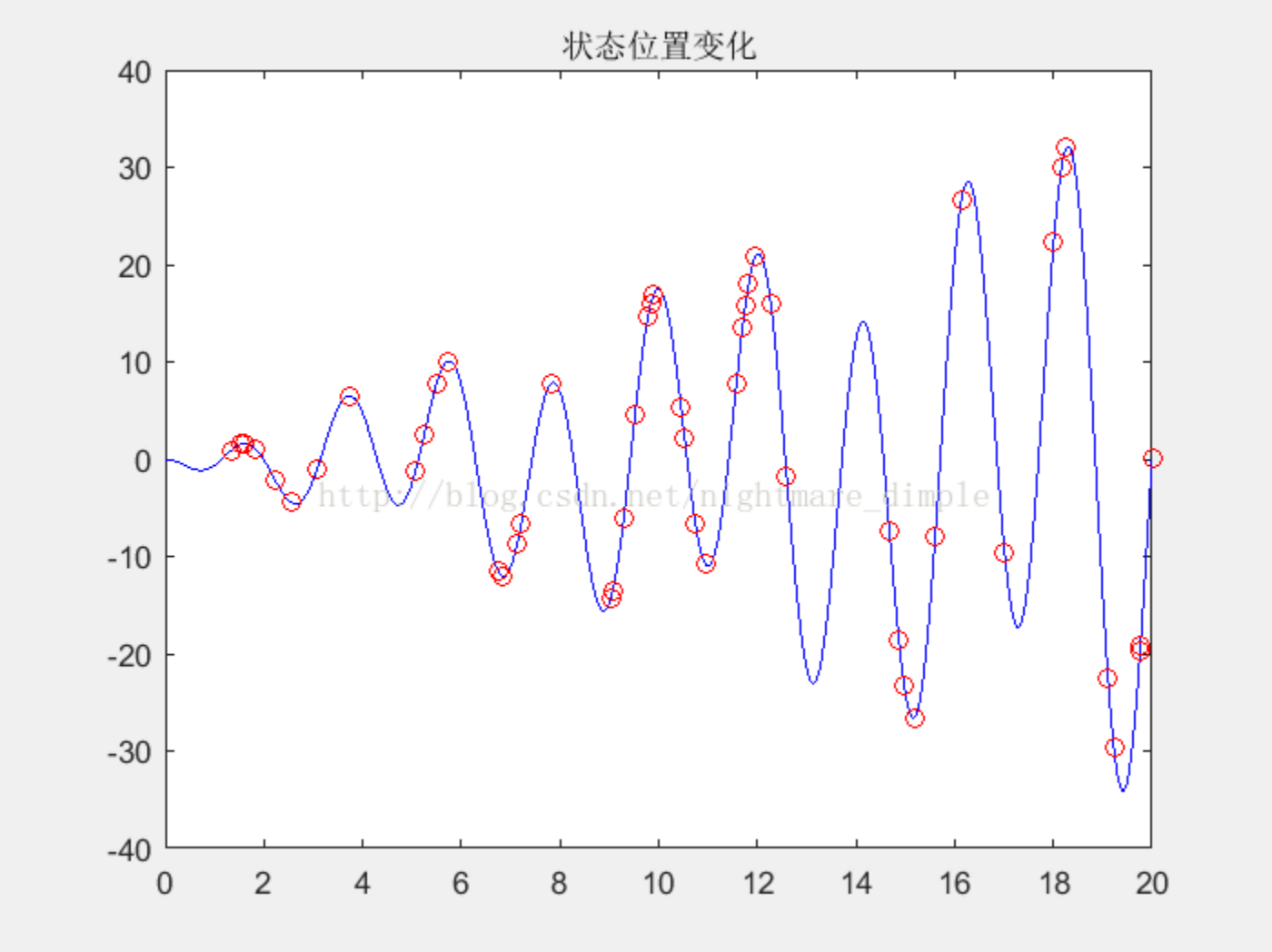
a swarm consists of n individuals that communicate either directly or indirectly with one another to adapt the search directions (gradients). Indeed, the particles are taken as some simple agents that fly through the search space and contiuously record/communicate the best solution that they have discovered.





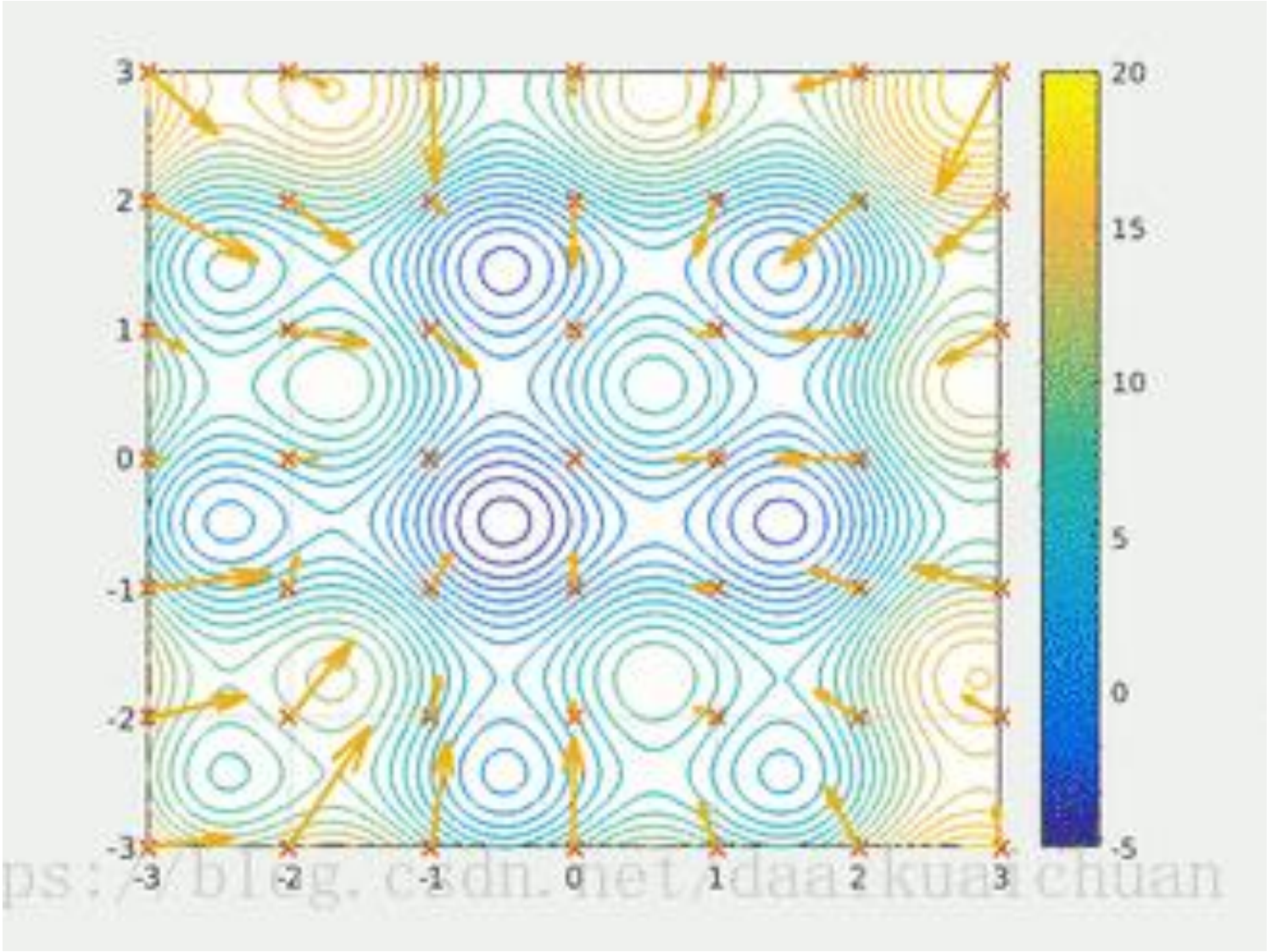
11.1 Overview

Particle Swarm Optimization PSO算法



11.1 Overview

Particle Swarm Optimization PSO算法



Particle Swarm Optimization PSO算法

假设在一个 D 维的目标搜索空间中，有 N 个粒子组成一个群落，其中第 i 个粒子表示为一个 D 维的向量

$$X_i=(x_{i1},x_{i2},\cdots,x_{iD}),\ i=1,2,\cdots,N。$$

第 i 个粒子的“飞行”速度也是一个 D 维的向量，记为

$$V_i=(v_{i1},v_{i2},\cdots,v_{iD}),\ i=1,2,\cdots 3。$$

第 i 个粒子迄今为止搜索到的最优位置称为个体极值，记为

$$p_{best}=(p_{i1},p_{i2},\cdots,p_{iD}),\ i=1,2,\cdots,N。$$

整个粒子群迄今为止搜索到的最优位置为全局极值，记为

$$g_{best}=(p_{g1},p_{g2},\cdots,p_{gD})$$

在找到这两个最优值时，粒子根据如下的公式(2-1)和(2-2)来更新自己的速度和位置：

公式（1）：

$$v_i=v_i+c_1\times rand()\times(pbest_i-x_i)+c_2\times rand()\times(gbest_i-x_i)$$

公式（2）：

$$x_i=x_i+v_i$$

11.1 Overview

Two Cultures!

11.1 Overview

A Generic Algorithm: A point to set mapping in a subspace of R^n.

Theorem 1 *(Page 201, L&Y) Let A be an "algorithmic mapping" defined over set X, and let sequence $\{\mathbf{x}^k\}$, starting from a given point $\mathbf{x}^0$, be generated from*

$$\mathbf{x}^{k+1} \in A(\mathbf{x}^k).$$

Let a solution set $S \subset X$ be given, and suppose

i) all points $\{\mathbf{x}^k\}$ are in a compact set;

ii) there is a continuous (merit) function $z(\mathbf{x})$ such that if $\mathbf{x} \notin S$, then $z(\mathbf{y}) < z(\mathbf{x})$ for all $\mathbf{y} \in A(\mathbf{x})$; otherwise, $z(\mathbf{y}) \leq z(\mathbf{x})$ for all $\mathbf{y} \in A(\mathbf{x})$;

iii) the mapping A is closed at points outside S ($\mathbf{x}^k \to \bar{\mathbf{x}} \in X$ and $A(\mathbf{x}^k) = \mathbf{y}^k \to \bar{\mathbf{y}}$ imply $\bar{\mathbf{y}} \in A(\bar{\mathbf{x}})$).

Then, the limit of any convergent subsequences of $\{\mathbf{x}^k\}$ is a solution in S.

从压缩映射的角度来看搜索算法

11.1 Overview

- Finite versus infinite convergence. For some classes of optimization problems there are algorithms that obtain an exact solution—or detect the unboundedness–in a finite number of iterations.
- Polynomial-time versus exponential-time. The solution time grows, in the worst-case, as a function of problem sizes (number of variables, constraints, accuracy, etc.).
- Convergence order and rate. If there is a positive number γ such that

$$\|\mathbf{x}^k - \mathbf{x}^*\| \leq \frac{O(1)}{k^\gamma}\|\mathbf{x}^0 - \mathbf{x}^*\|,$$

then $\{\mathbf{x}^k\}$ converges arithmetically to $\mathbf{x}^*$ with power γ. If there exists a number $\gamma \in [0, 1)$ such that

$$\|\mathbf{x}^{k+1} - \mathbf{x}^*\| \leq \gamma\|\mathbf{x}^k - \mathbf{x}^*\| \quad (\Rightarrow \|\mathbf{x}^k - \mathbf{x}^*\| \leq \gamma^k\|\mathbf{x}^0 - \mathbf{x}^*\|),$$

then $\{\mathbf{x}^k\}$ converges geometrically or linearly to $\mathbf{x}^*$ with rate γ. If there exists a number $\gamma \in [0, 1)$

$$\|\mathbf{x}^{k+1} - \mathbf{x}^*\| \leq \gamma\|\mathbf{x}^k - \mathbf{x}^*\|^2 \text{ after } \gamma\|\mathbf{x}^k - \mathbf{x}^*\| < 1$$

then $\{\mathbf{x}^k\}$ converges quadratically to $\mathbf{x}^*$ (such as $\left\{(\frac{1}{2})^{2^k}\right\}$).

11.1 Overview

Depending on information of the problem being used to create a new iterate, we have

(a) Zero-order algorithms. Popular when the gradient and Hessian information are difficult to obtain, e.g., no explicit function forms are given, functions are not differentiable, etc.

(b) First-order algorithms. Most popular now-days, suitable for large scale data optimization with low accuracy requirement, e.g., Machine Learning, Statistical Predictions.

(c) Second-order algorithms. Popular for optimization problems with high accuracy need, e.g., some scientific computing, etc.

以下以一维搜索问题（一维函数的优化问题）为例来讲解

11.2. Zero-Order Optimization

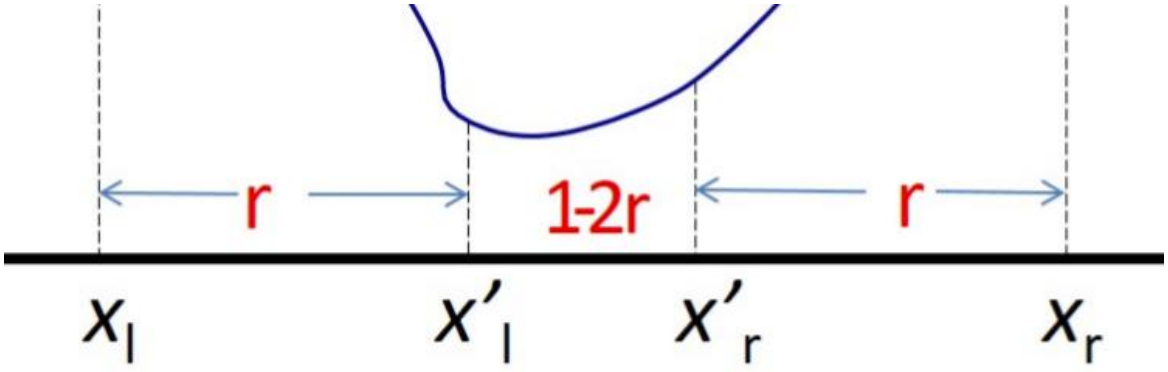
Assume that the one variable function $f(x)$ is Unimodel in interval $[a\ b]$, that is, for any point $x \in [a_r\ b_l]$ such that $a \le a_r < b_l \le b$, we have that $f(x) \le \max\{f(a_r),\ f(b_l)\}$. How do we find x^* within an error tolerance ϵ?

0) Initialization: let $x_l = a,\ x_r = b$, and choose a constant $0 < r < 0.5$;

1) Let two other points $\hat{x}_l = x_l + r(x_r - x_l)$ and $\hat{x}_r = x_l + (1-r)(x_r - x_l)$, and evaluate their function values.

2) Update the triple points $x_r = \hat{x}_r, \hat{x}_r = \hat{x}_l, x_l = x_l$ if $f(\hat{x}_l) < f(\hat{x}_r)$; otherwise update the triple points $x_l = \hat{x}_l, \hat{x}_l = \hat{x}_r, x_r = x_r$; and return to Step 1.

In either cases, the length of the new interval after one golden section step is $(1-r)$. If we set $(1-2r)/(1-r) = r$, then only one point is new in each step and needs to be evaluated. This give $r = 0.382$ and the linear convergence rate is 0.618.



Golden Section is Zero-Order

11.3. First-Order Optimization

For a one variable problem, an KKT point is the root of $g(x) := f'(x) = 0$.

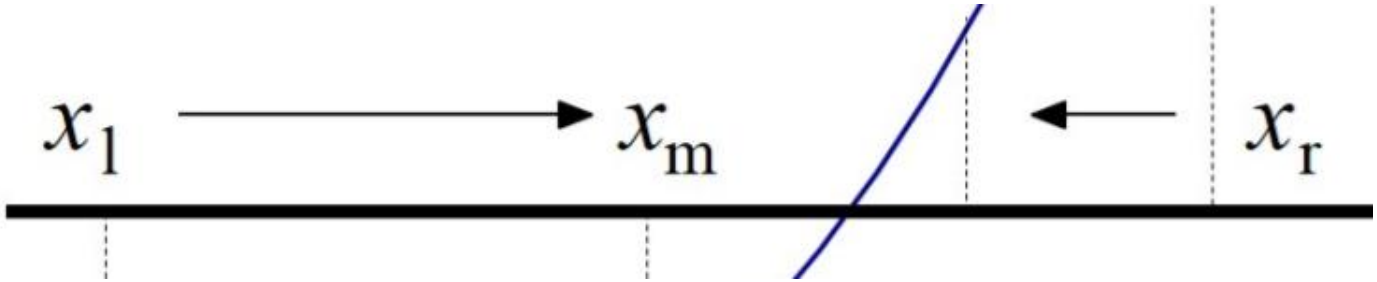
Assume we know an interval $[a\ b]$ such that $a < b$, and $g(a)g(b) < 0$. Then we know there exists an x^*, $a < x^* < b$, such that $g(x^*) = 0$; that is, interval $[a\ b]$ contains a root of g. How do we find x within an error tolerance ϵ, that is, $|x - x^*| \leq \epsilon$?

0) Initialization: let $x_l = a,\ x_r = b$.

1) Let $x_m = (x_l + x_r)/2$, and evaluate $g(x_m)$.

2) If $g(x_m) = 0$ or $x_r - x_l < \epsilon$ stop and output $x^* = x_m$. Otherwise, if $g(x_l)g(x_m) > 0$ set $x_l = x_m$; else set $x_r = x_m$; and return to Step 1.

The length of the new interval containing a root after one bisection step is $1/2$ which gives the linear convergence rate is $1/2$, and this establishes a linear convergence rate 0.5.



Bisection is First-Order

11.4. Second-Order Optimization

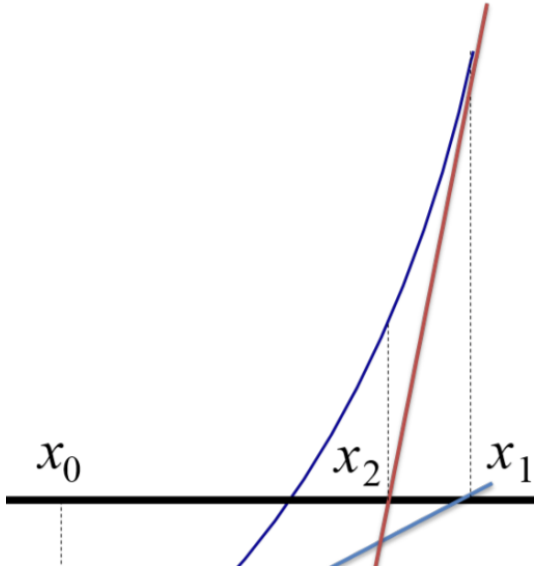
For functions of a single real variable x, the KKT condition is $g(x) := f'(x) = 0$. When f is twice continuously differentiable then g is once continuously differentiable, Newton's method can be a very effective way to solve such equations and hence to locate a root of g. Given a starting point x^0, Newton's method for solving the equation $g(x) = 0$ is to generate the sequence of iterates

$$x^{k+1} = x^k - \frac{g(x^k)}{g'(x^k)}.$$

The iteration is well defined provided that $g'(x^k) \neq 0$ at each step.

For strictly convex function, Newton's method has a linear convergence rate and, when the point is "close" to the root, the convergence becomes quadratic.

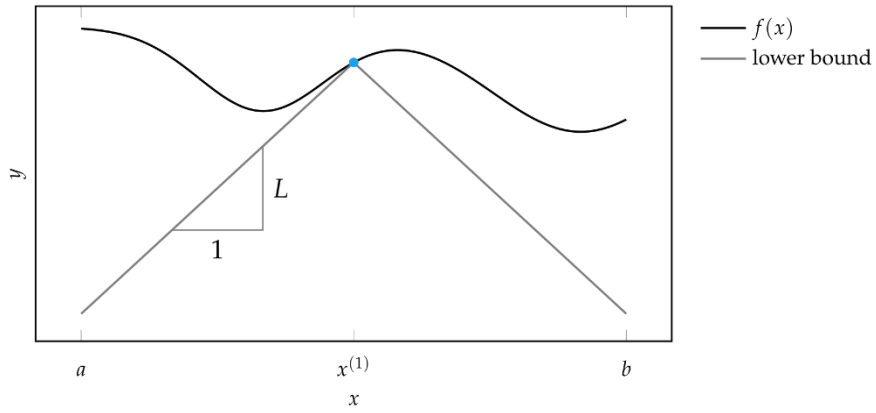
Newton's Method is Second-Order

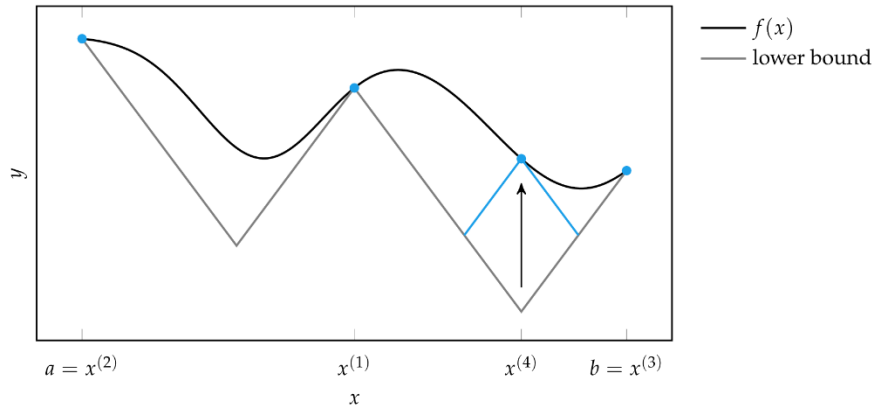


11.5. DIRECT Algorithm

Also called DIRECT for Divided RECTangles

A Lipshitz constant is used to provide a piece-wise affine approximation (lower bound) for the function, and a function evaluation is made where this bound is lowest

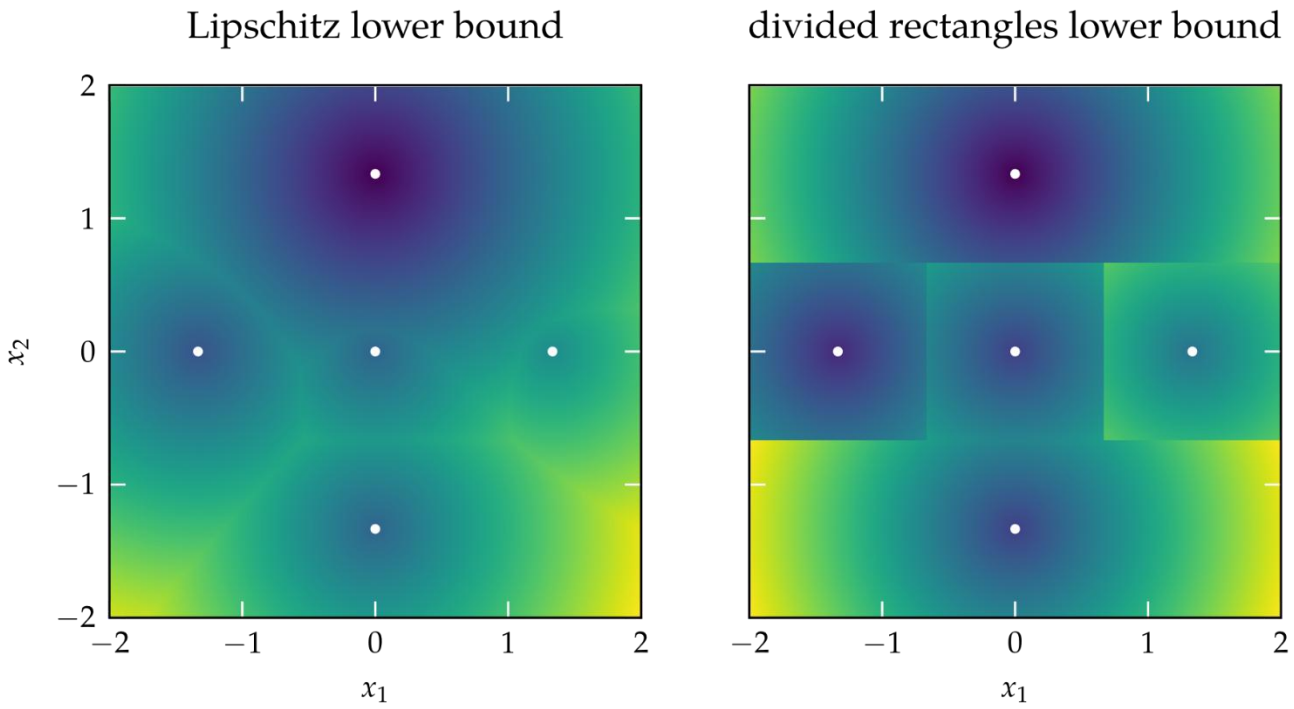




11.5. DIRECT Algorithm

In multiple dimensions, the geometry of these cone intersections can become very complicated

DIRECT simplifies geometry using subdivided rectangles

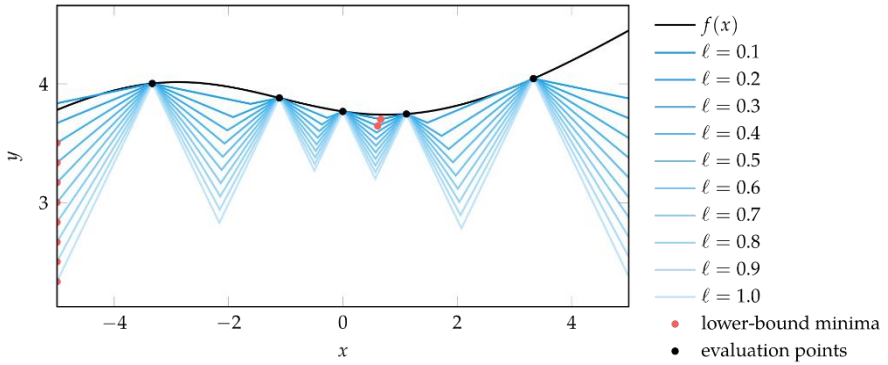


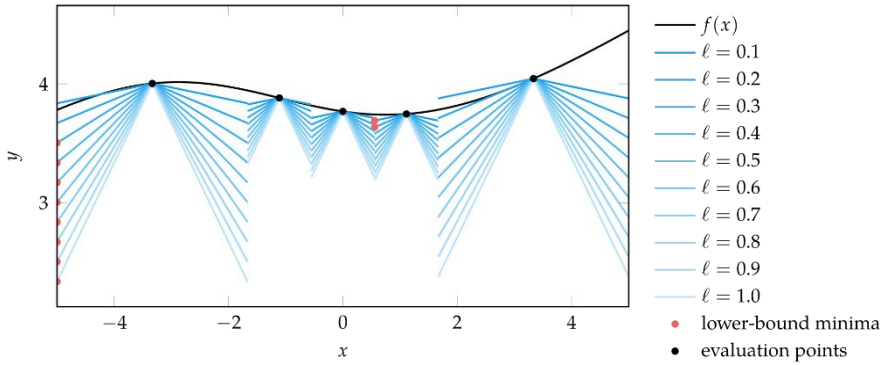
11.5. DIRECT Algorithm

DIRECT does not assume a Lipschitz constant is known.

Optimizer divides space into boxes and samples the vertices of each

A box is further divided based on estimated maximum rate of change of the function, K (Lipschitz constant)



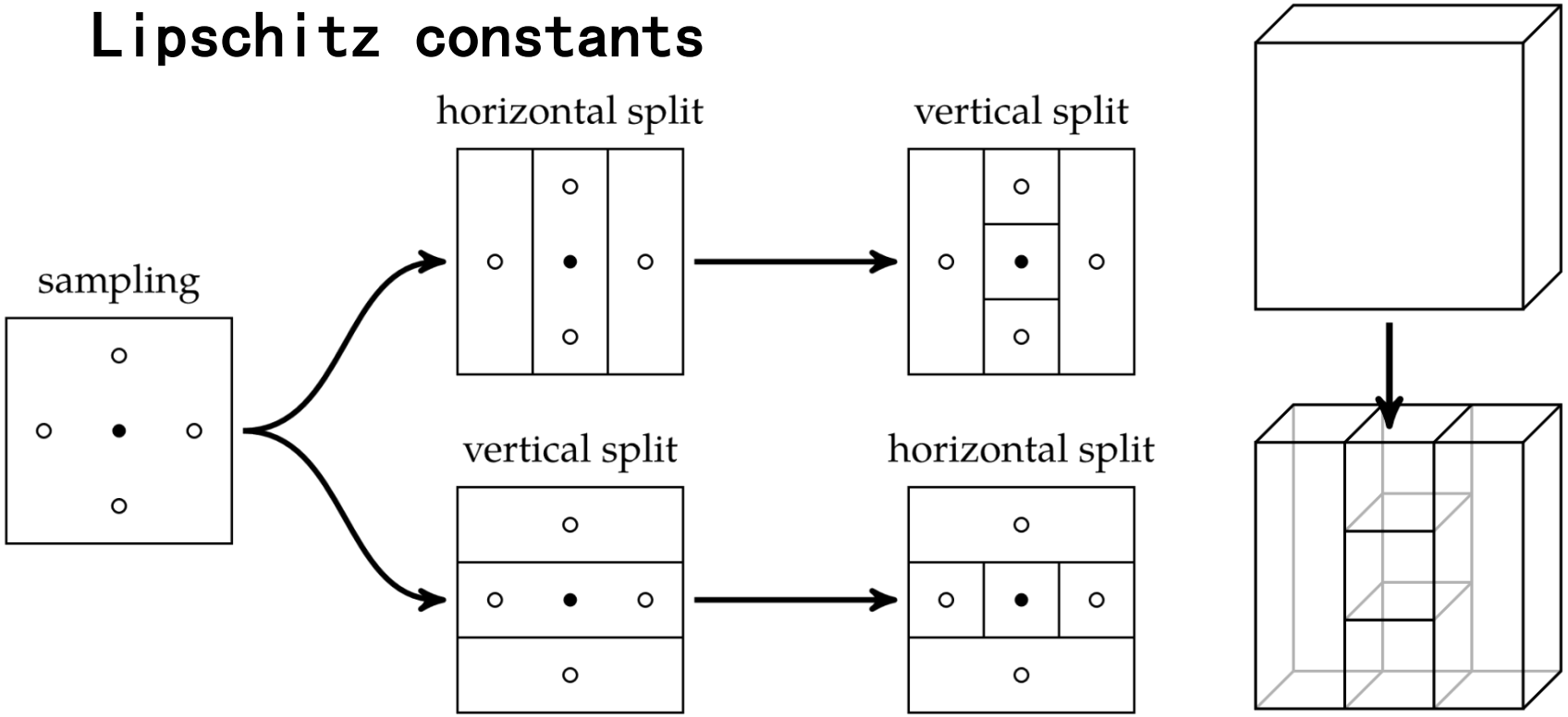


11.5. DIRECT Algorithm

The function value at the middle of each box and it's largest dimension are used to determine potentially optimal boxes

Each potentially optimal box is divided

Lipschitzian optimization for all possible Lipschitz constants

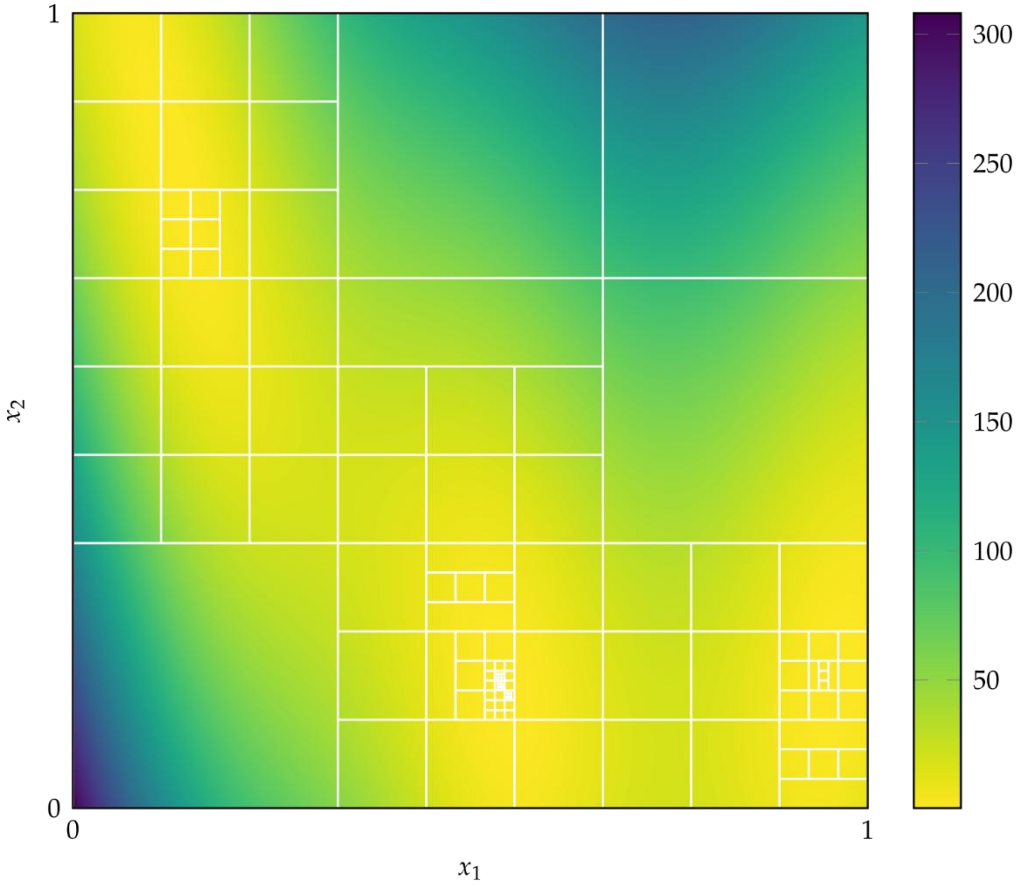


11.5. DIRECT Algorithm

DIRECT uses convex hull of box sizes to balance exploitation vs. exploration

With enough function evaluations every region in design space will be explored

This is clearly not feasible for high dimensional spaces (Adaptive block coordinate)



11.6 参考文献

- **CME307/MS&E311 Optimization Winter 2019-2020**
  https://web.stanford.edu/class/msande311/

- **Local Descent**
  https://web.stanford.edu/~mykel/algforopt/slides/7.pptx

- **Q. Tao, X. Huang, S. Wang, L. Li, "Adaptive block coordinate DIRECT algorithm," *Journal of Global Optimization*, vol. 69, no. 4, pp. 797-822, 2017.**

- **D. R. Jones, J. R. R. A. Martins, "The DIRECT algorithm: 25 years Later," Journal of Global Optimization,**
  https://link.springer.com/article/10.1007/s10898-020-00952-6

- **https://ctk.math.ncsu.edu/Finkel_Direct/**