Decentralized learning in control and optimization for networks and dynamic games

Part I: centralized optimization

Alexandre Proutiere
KTH

Outline of part I

- Gradient-free (or 0^{th} order) methods
- Gradient-descent (or 1^{st} order) methods
- Fixed point iterations

Gradient-free methods

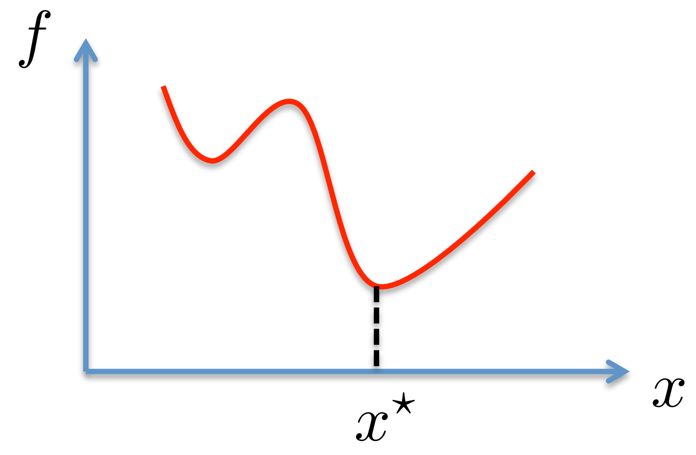
Gradient-free methods

Two surveys:

1. Optimization by direct search: new perspectives on some classical and modern methods, **Kolda-Lewis-Torczon**, SIAM rev. 2003
2. Derivative-free optimization: a review of algorithm, **Rios-Sahinidis**, submitted

Objective

$$\text{minimize } f(x)$$
$$\text{over } x \in \Omega$$
$$\Omega \subset \mathbb{R}^n$$

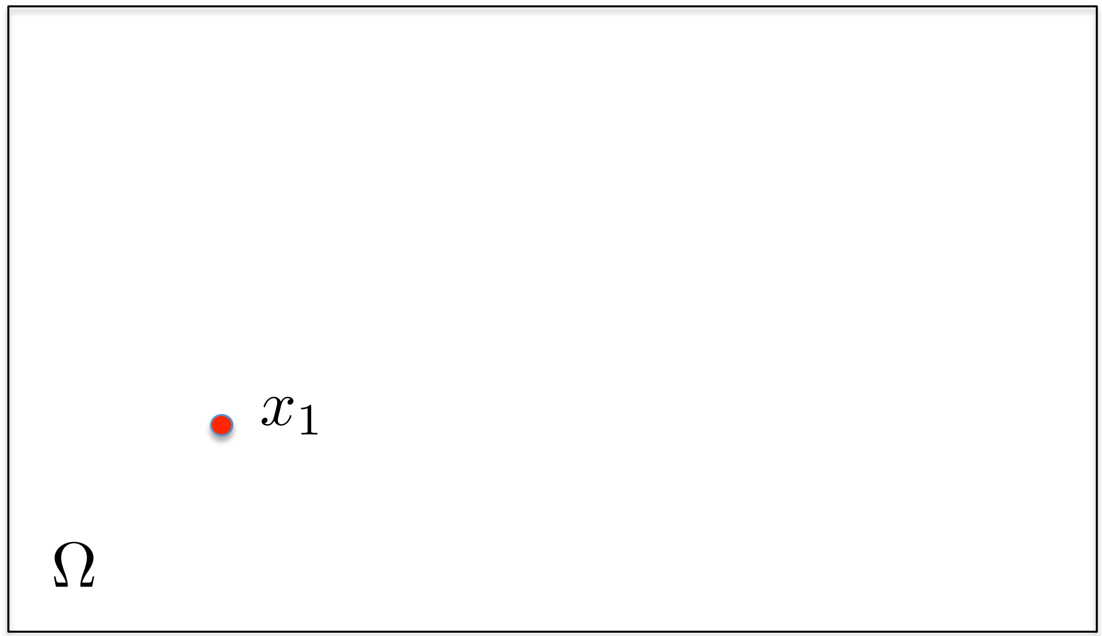


- The gradient $\nabla f(x)$ is not available
- Smooth function, and convex compact search space

A few algorithms

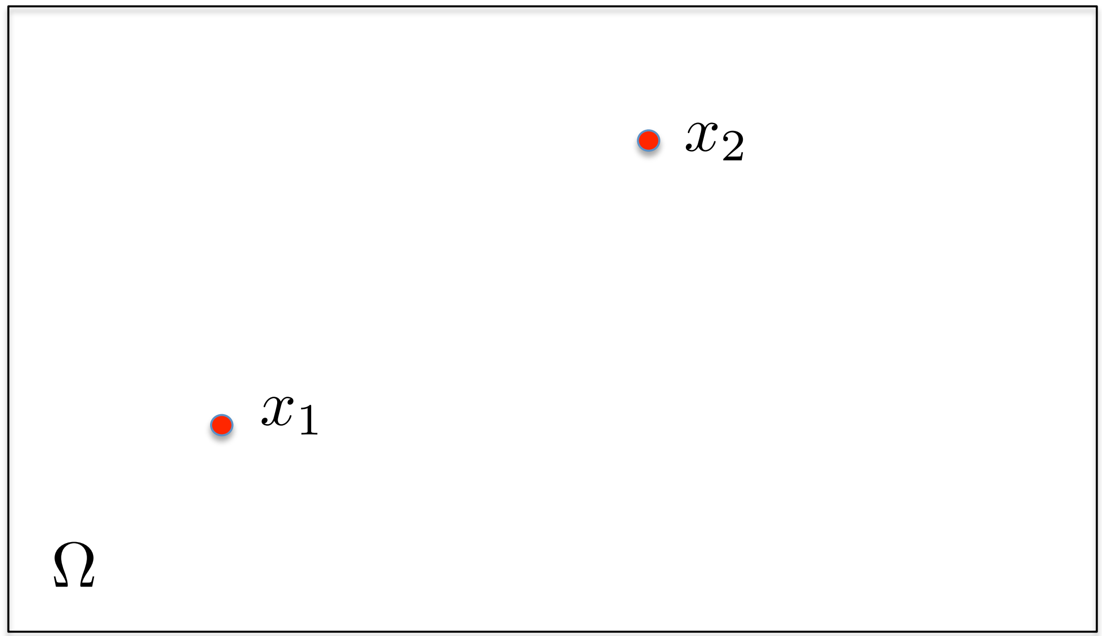
- Random global search: Hit and Run algorithm
- Random local search: Generating Set Search algorithm
- Simulated annealing
- Gradient-estimator methods

Random global search



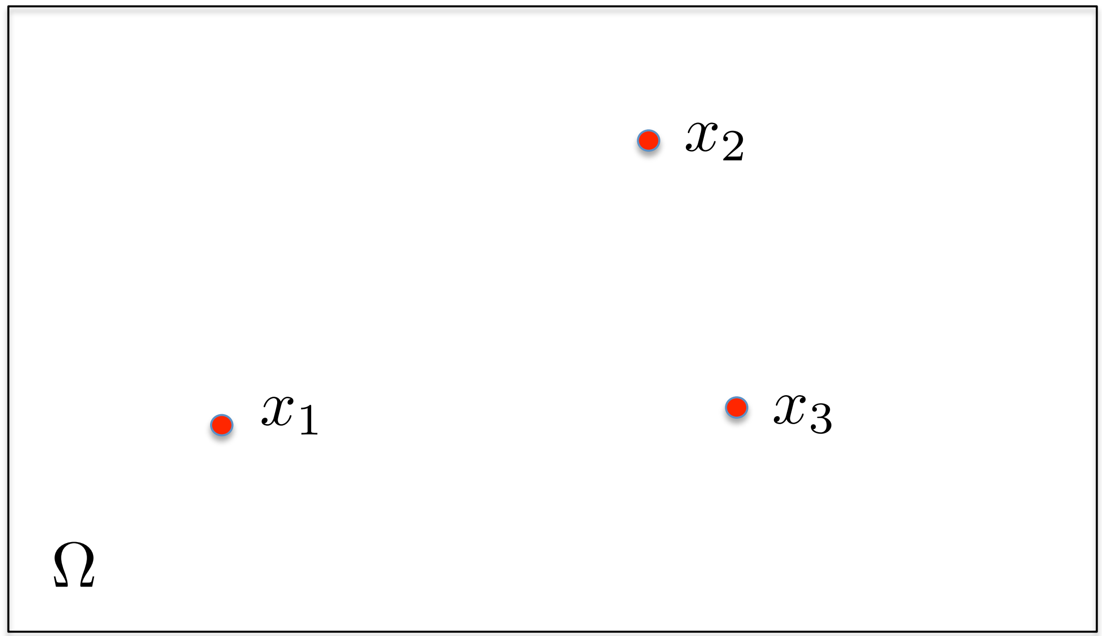
- Sequentially generate random points $(x_k, k = 1, 2, \ldots)$
- The candidate at a given time is the point with smallest value function observed so far

Random global search



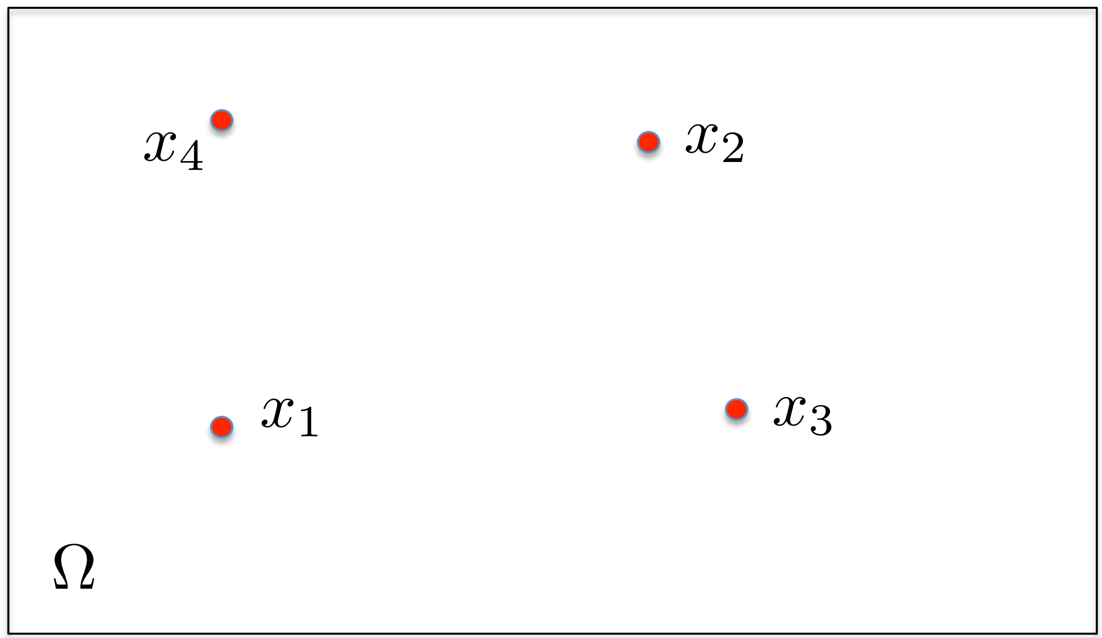
- Sequentially generate random points $(x_k, k = 1, 2, ...)$
- The candidate at a given time is the point with smallest value function observed so far

Random global search



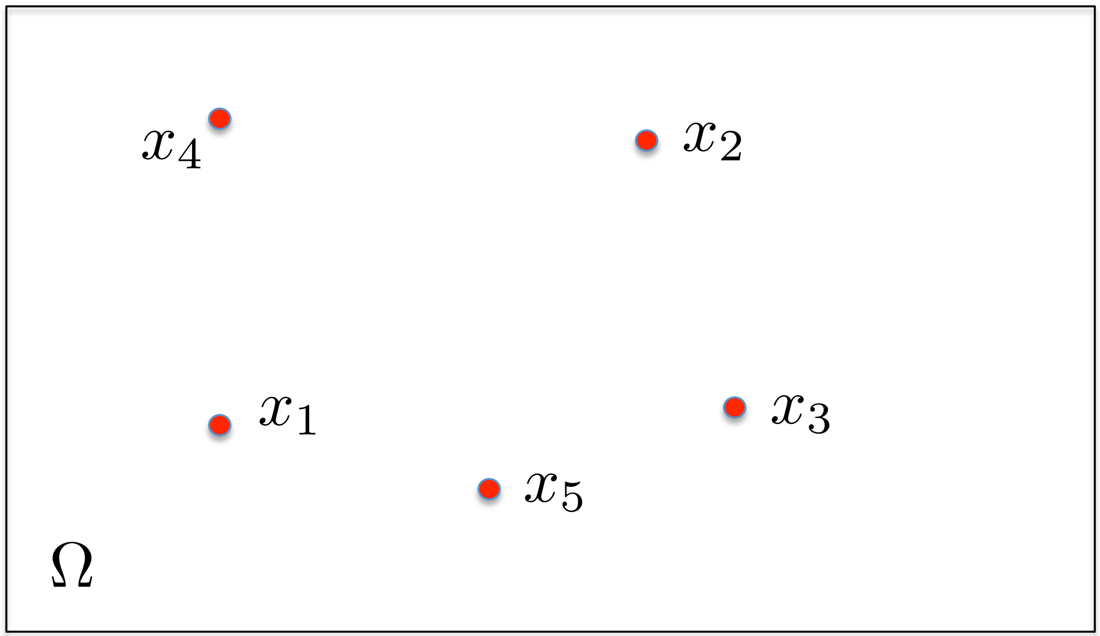
- Sequentially generate random points $(x_k, k = 1, 2, ...)$
- The candidate at a given time is the point with smallest value function observed so far

Random global search



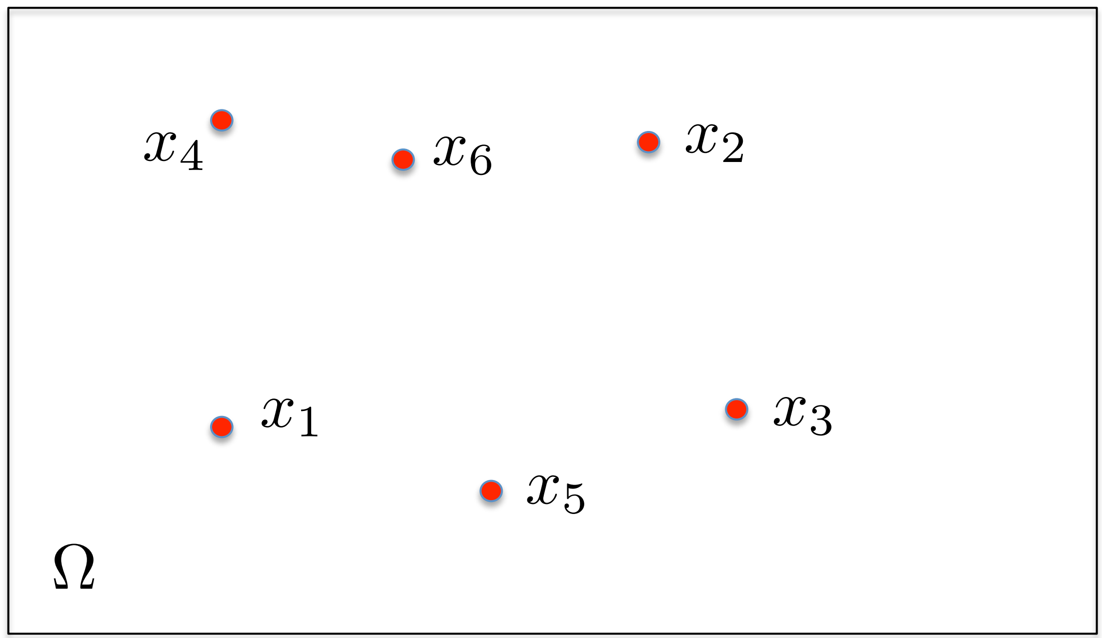
- Sequentially generate random points $(x_k, k = 1, 2, ...)$
- The candidate at a given time is the point with smallest value function observed so far

Random global search



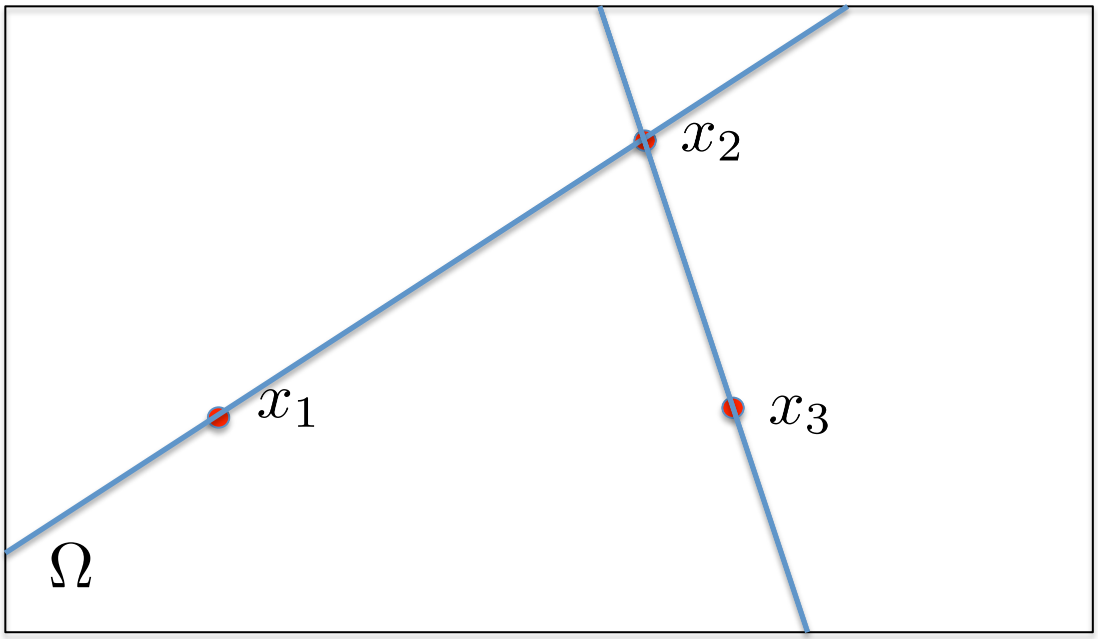
- Sequentially generate random points $(x_k, k = 1, 2, ...)$
- The candidate at a given time is the point with smallest value function observed so far

Random global search



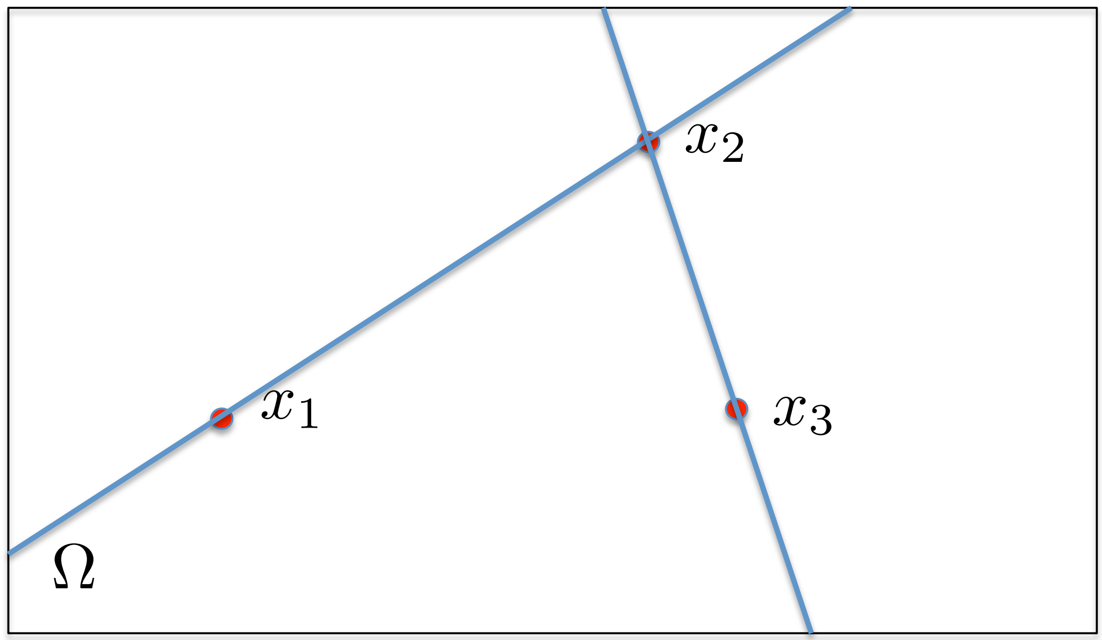
- Sequentially generate random points $(x_k, k = 1, 2, ...)$
- The candidate at a given time is the point with smallest value function observed so far

Hit and run algorithm



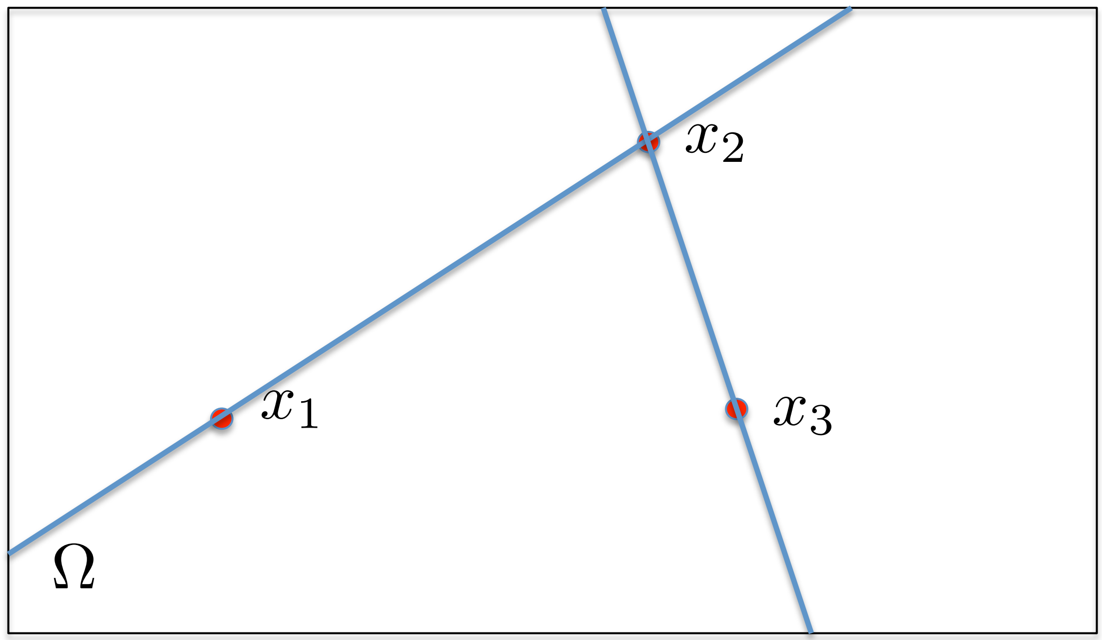
- Principle: generate a sequence of samples whose limited distribution is uniform over the search space

Hit and run algorithm



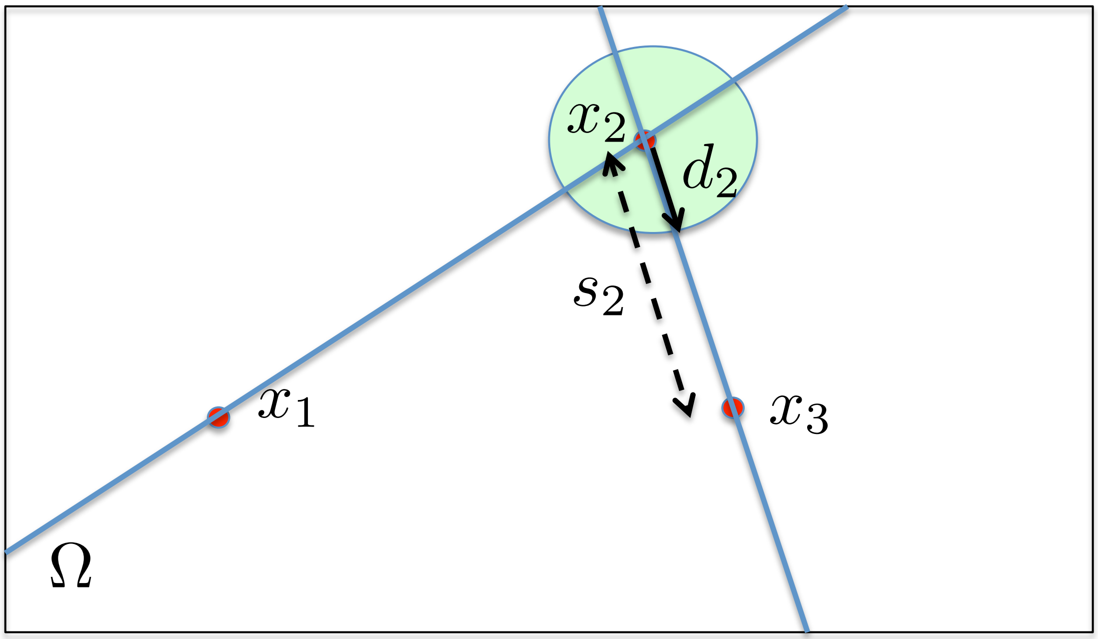
- Algorithm: select a direction uniformly at random, select a new point uniformly at random along this direction

Hit and run algorithm



- The fastest known procedure to generate samples with uniform distribution over Ω

Hit and run optimization



$$x_{k+1} = \begin{cases} x_k + s_k d_k, & \text{if } f(x_k + s_k d_k) < f(x_k) \\ x_k, & \text{otherwise} \end{cases}$$

Performance of HR

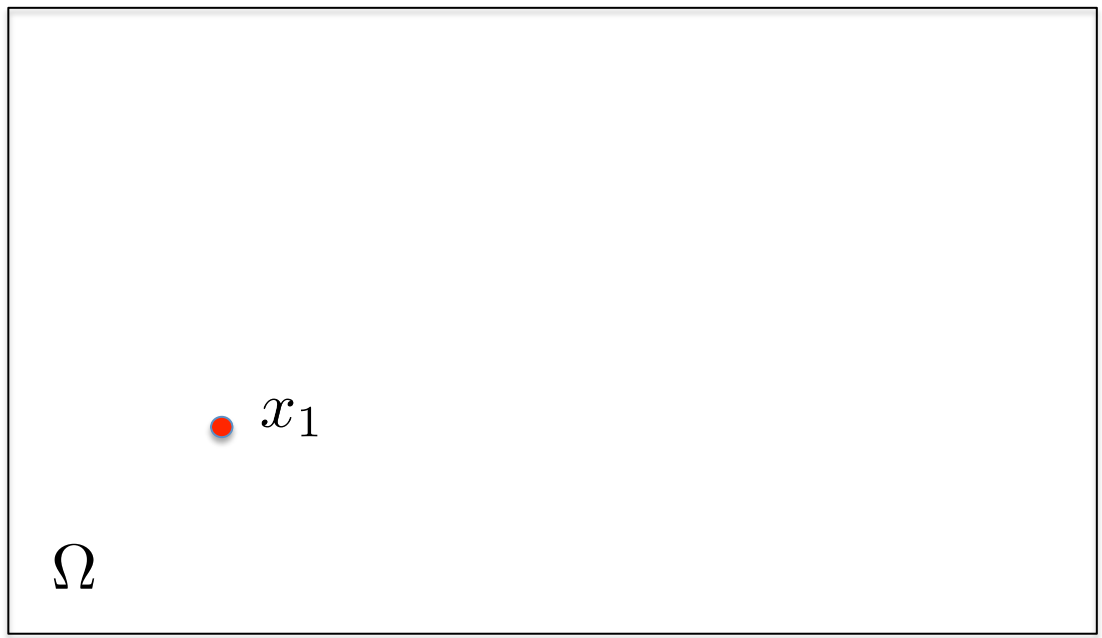
Theorem* Let *N(r)* be the number of samples required to be at a distance at least *r* of the minimum of a positive quadratic function. Then:

$$\mathbb{E}[N(r)] \leq \frac{\psi(n)}{r} n\mathbb{E}[K(r)^{PAS}] = O(n^{5.2})$$

$K(r)^{PAS}$ is the number of required improvements in the Pure Adaptive Search algorithm.

* Improving Hit-and-Run for Global Optimization, **Zabinsky et al.**, Journal of Global Optimization, 1993.

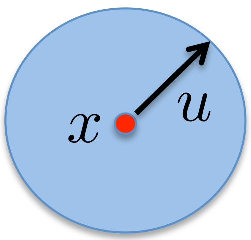
Oblivious Randomized Direct search*



- Isotropic random generation of improving points, ensuring fixed average improvement

- Oblivious randomized direct search for real parameter optimization, **Jagerskupper**, ESA, 2008.

Oblivious Randomized Direct Search

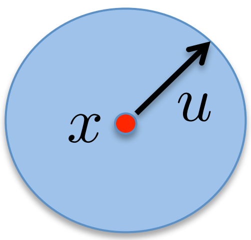


Candidate point: $y = x_k + L_k u_k$

Accepted if: $f(y) \leq f(X_k)$

- $u_1, u_2, \ldots$ i.i.d. sequence of unit vectors with uniformly random direction
- $L_1, L_2, \ldots$ i.i.d. sequence of step sizes, density μ
- How to choose μ such that the average improvement remains constant?

Oblivious Randomized Direct Search

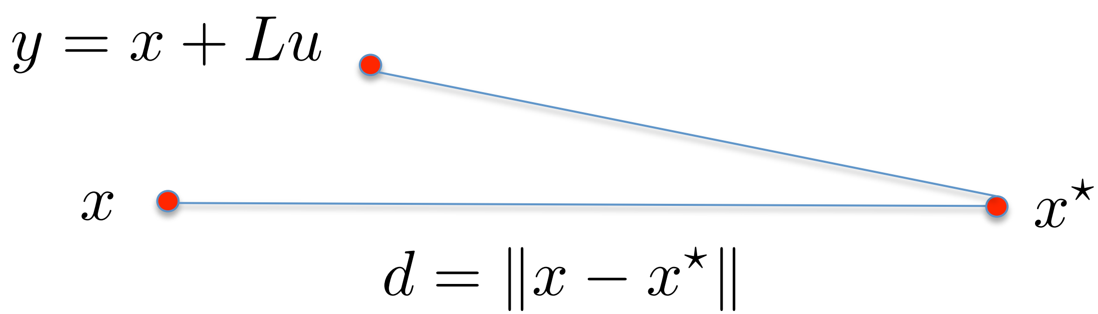


Candidate point: $y = x_k + L_k u_k$

Accepted if: $f(y) \leq f(X_k)$

- $u_1, u_2, \ldots$ i.i.d. sequence of unit vectors with uniformly random direction
- $L_1, L_2, \ldots$ i.i.d. sequence of step sizes, density μ
- How to choose μ such that the average improvement remains constant?

Oblivious Randomized Direct Search



$$p_{d,l,\alpha} = P[\|x + lu - x^\star\| \leq \alpha d]$$

Probability to reduce the distance *d* to the optimal point by a factor α when the step-size is *l*.

$$p_{d,\mu,\alpha} = P[\|y - x^\star\| \leq \alpha d] = \int_{(1-\alpha)d}^{(1+\alpha)d} p_{d,l,\alpha}\mu(l)dl$$

$$(p_{d,l,\alpha} = p_{1,l/d.\alpha})$$

Oblivious Randomized Direct Search

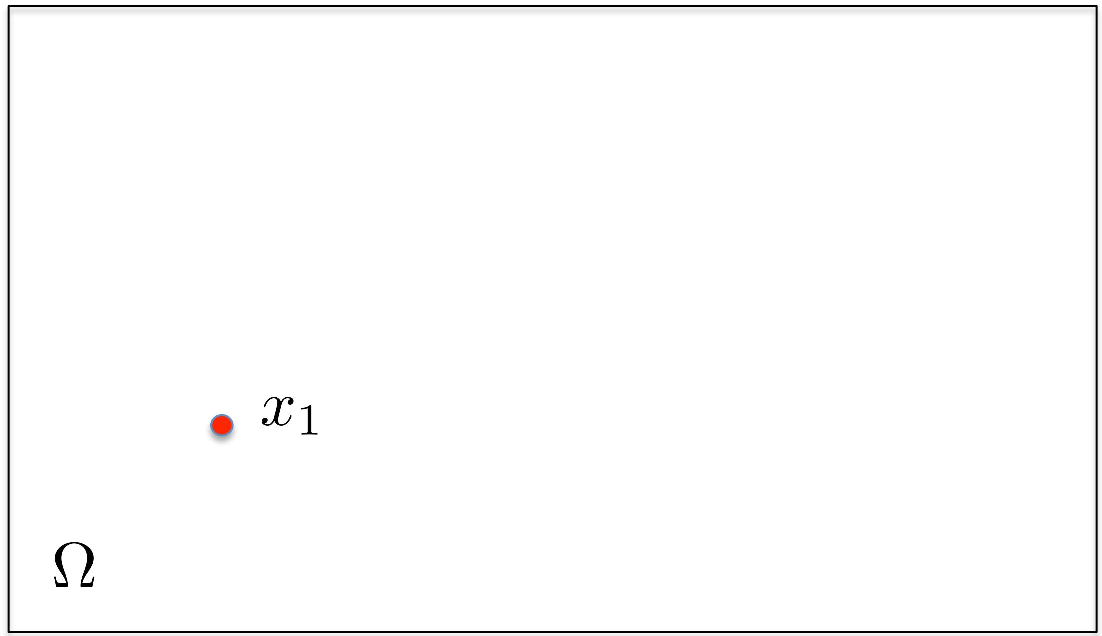
$$p_{d,\mu,\alpha} = \int_{(1-\alpha)}^{(1+\alpha)} d \times p_{1,v,\alpha}\mu(dv)dv$$

- Improvement independent of *d* if: $\mu(v) \sim \beta/v$
- Under support restriction:

$$\mu(v) = \frac{1_{v\in[a,b]}}{v\log(b/a)}$$

Theorem For $a = 0.1\epsilon/\sqrt{n+1}, \quad b = 2\sqrt{n+1}, \quad \Omega = [0,1]^n$
ORDS algorithm solves the sphere problem ($f(x) = \|x - x^\star\|_2^2$)
with precision $\epsilon + a$ in an expected number of steps scaling as
$O(n.\log^2(n/\epsilon))$

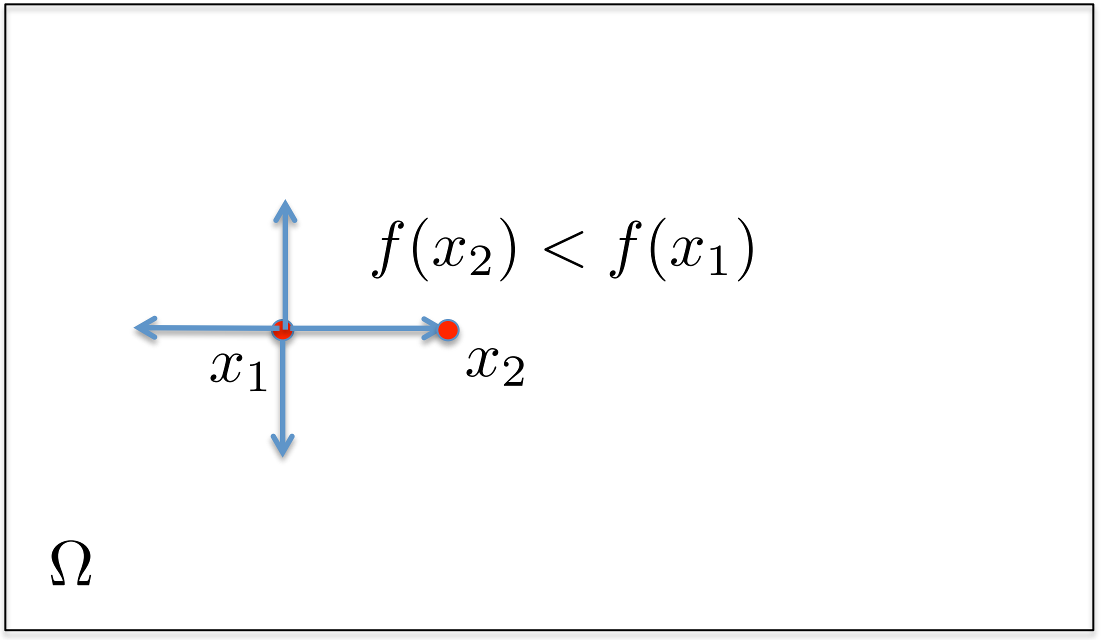
Oblivious Randomized Direct search*



- Isotropic random generation of improving points, ensuring fixed average improvement

- Oblivious randomized direct search for real parameter optimization, **Jagerskupper**, ESA, 2008.

Random local search



- Principle: Search locally for improving points, i.e., with smaller objective function

Generating Set Search

- Generic algorithm
  1. From the current point, generate neighboring trial points
  2. Evaluate the function at trial points
  3. If there is an improving point, move there
     Otherwise, modify the procedure to generate trial points

Compass search

Algorithm.

Initialization. Choose x_0, and Δ_0.

For each iteration $k \geq 1$:

Step 1. Generate trial $2n$ points: $x_{k-1} \pm \Delta_{k-1} e_i$, $\forall i = 1, \ldots, n$

Step 2. If there exists a trial point x' such that $f(x') < f(x_{k-1})$,

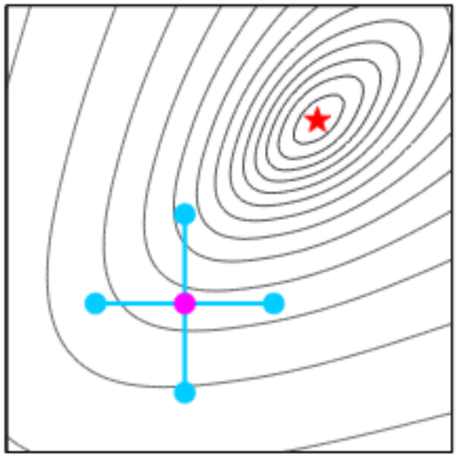
$x_k = x'$

$\Delta_k = \Delta_{k-1}$

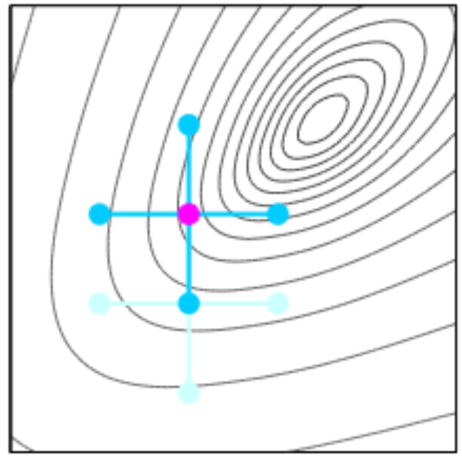
Step 3. Otherwise $x_k = x_{k-1}$

$\Delta_k = \alpha \Delta_{k-1} \quad (\alpha < 1)$

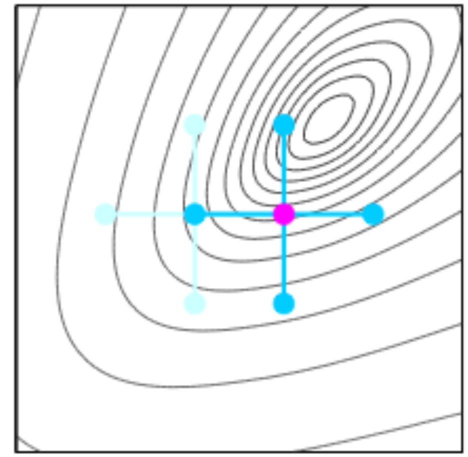
Compass search



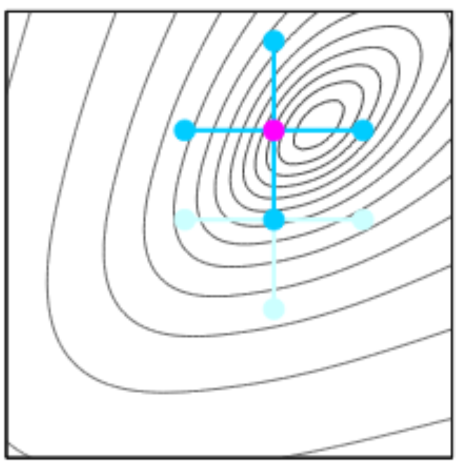
(a) Initial pattern



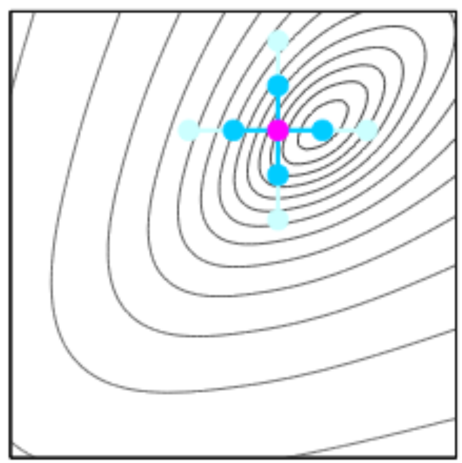
(b) Move North



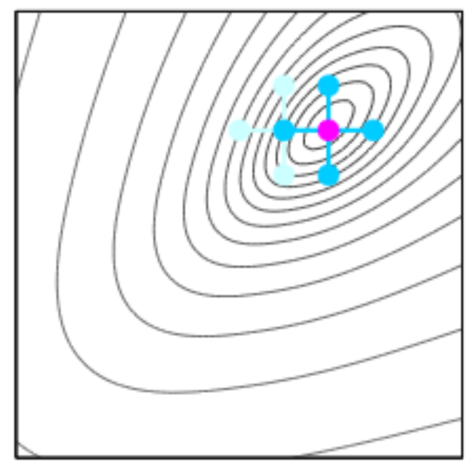
(c) Move West



(d) Move North



(e) Contract

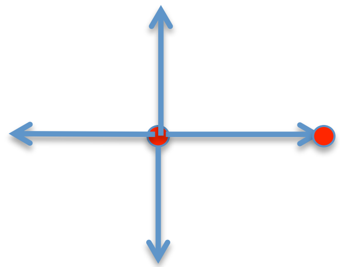


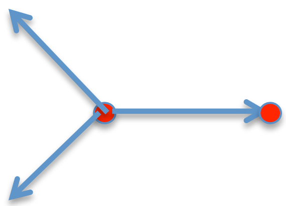
(f) Move West

Convergence of GSS

Theorem The compass search algorithm converges to a local minimizer of the objective function, if the latter is continuously differentiable and has Lipschitz gradient.

- The convergence result remains valid for generic GSS algorithms provided
  - the trial point generation algorithm is appropriate
  - the step-size sequence is appropriate





Simulated Annealing

- Proposed by **Kirkpatrick-Gelatt-Vecchi**, Science, 1983 (>24000 citations)
- Paradigm from statistical physics: at high temperature, molecules move freely forming a liquid; if the tmeperature is slowly decreased, thermal mobility disappears, and a crystal with minimum energy is created
- Principle: construct a Markov chain whose stationary distribution with fixed temperature is proportional to:

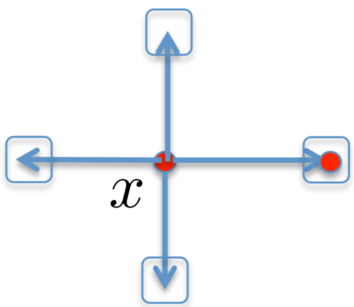
$$\pi_t(x) \propto \exp(-f(x)/T)$$

Discrete search space

- Components of the algorithm
  - A "cooling" schedule $T_1 \geq T_2 \geq \ldots$

$$\lim_{k \to \infty} T_k = 0$$

  - A distribution over possible moves $R(x, y)$



$$R(x, y) = 1/|N(x)|, \quad \forall y \in N(x)$$

  - Acceptance probability function:

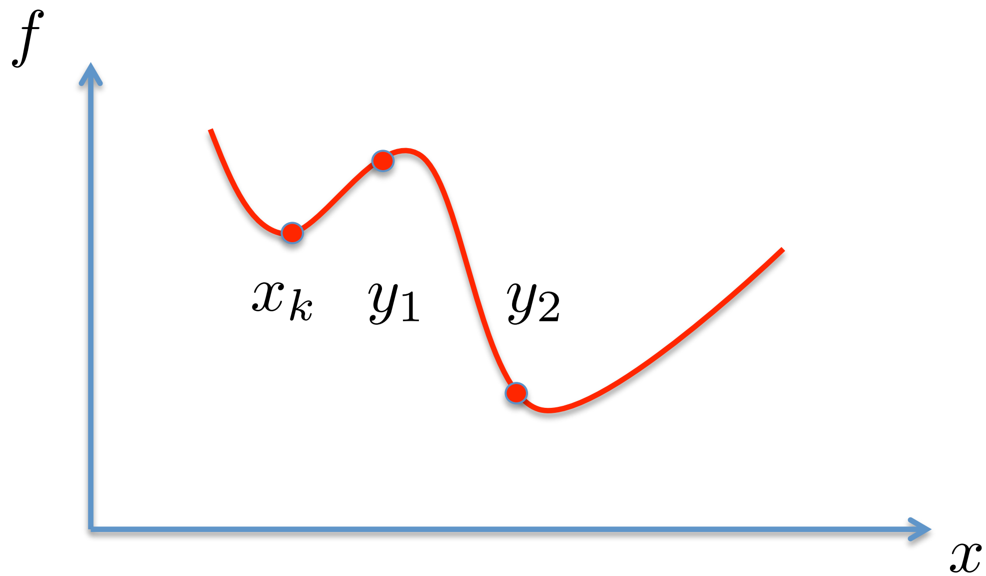
$$p_k(y) = \exp[\frac{-(f(y) - f(x))^+}{T_k}]$$

SA algorithm

Algorithm.
Initialization. Choose $x_0 \in \Omega$.
For each iteration $k \geq 1$:
Step 1. Generate y randomly according to $R(x_{k-1}, y)$,
Step 2. Accept the move, i.e., $x_k = y$, with probability

$$p_k = \exp[\frac{-[f(y) - f(x_{k-1})]^+}{T_k}]$$

SA algorithm: avoiding local minima



Move to y_2 accepted w.p. 1

Move to y_1 accepted with > 0 probability

Convergence

Theorem* Under SA algorithm, if the constructed Markov chain is irreducible and weakly reversible, then

$$\lim_{k\to\infty} P[x_k \in \arg\min_x f(x)] = 1 \Longleftrightarrow \sum_{k=1}^{\infty} \exp(-d/T_k) = +\infty$$

<u>Example:</u> $T_k = c/\log(k+1)$

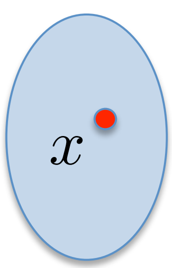
* Cooling schedules for optimal annealing, **Hajek**, Mathematics of Operations Research, 1988.

Continuous search space

- Similar components
  - A "cooling" schedule $T_1 \geq T_2 \geq \dots$

$$\lim_{k \to \infty} T_k = 0$$

  - A distribution over possible moves $R(x, y)$



  - Acceptance probability function:

$$p_k(y) = \exp[\frac{-(f(y) - f(x))^+}{T_k}]$$

Continuous search space

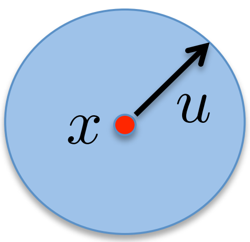
- Common justification of SA: avoids local minima
- Yet another justification of the acceptance probability of the form: $\exp(-f(y)/T)$

… it maximizes the convergence rate for convex optimization problems among all possible logconcave probabilties

* Simulated Annealing for Convex Optimization, **Kalai-Vempala**, Mathematics of Operations Research, 2006.

Gradient estimation

- Idea proposed by **Granichin**, 1989
- One-sample estimator of the gradient



$$S = \{y : \|y\| = 1\}$$

$$B = \{y : \|y\| \leq 1\}$$

$$\hat{f}(x) = \mathbb{E}_{v \in B}[f(x + \delta v)]$$

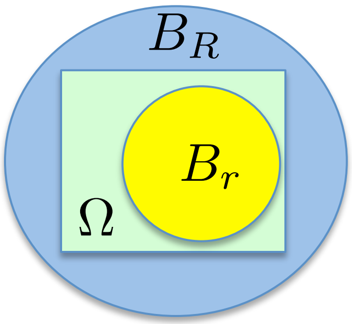
Gradient estimator: $f(x + \delta u)u$

Lemma $\forall \delta > 0, \quad \mathbb{E}_{u \in S}[f(x + \delta u)u] = \frac{\delta}{n} \nabla \hat{f}(x)$

Expected Gradient Descent

Algorithm.
Initialization. Choose $x_0 \in \Omega$.
For each iteration $k \geq 1$:
$x_k = x_{k-1} - \nu f(x_{k-1} + \delta u_k) u_k$

Expected Gradient Descent



$$\sup_{x \in \Omega} \|f(x)\| \leq F$$

$$\nu = \frac{R}{F\sqrt{K}}$$

$$\delta = \frac{1}{K^{1/4}} \sqrt{\frac{RrnF}{3(Lr + C)}}$$

Theorem* If *f* is convex and *L*-lipschitz:

$$\frac{1}{K}\mathbb{E}[\sum_{k=1}^{K} f(x_k)] \leq f(x^\star) + O(K^{-1/4})$$

* Online convex optimization in the bandit setting: gradient descent without the gradient, **Flaxman-Kalai-McMahan**, SODA, 2005.

Gradient-descent methods

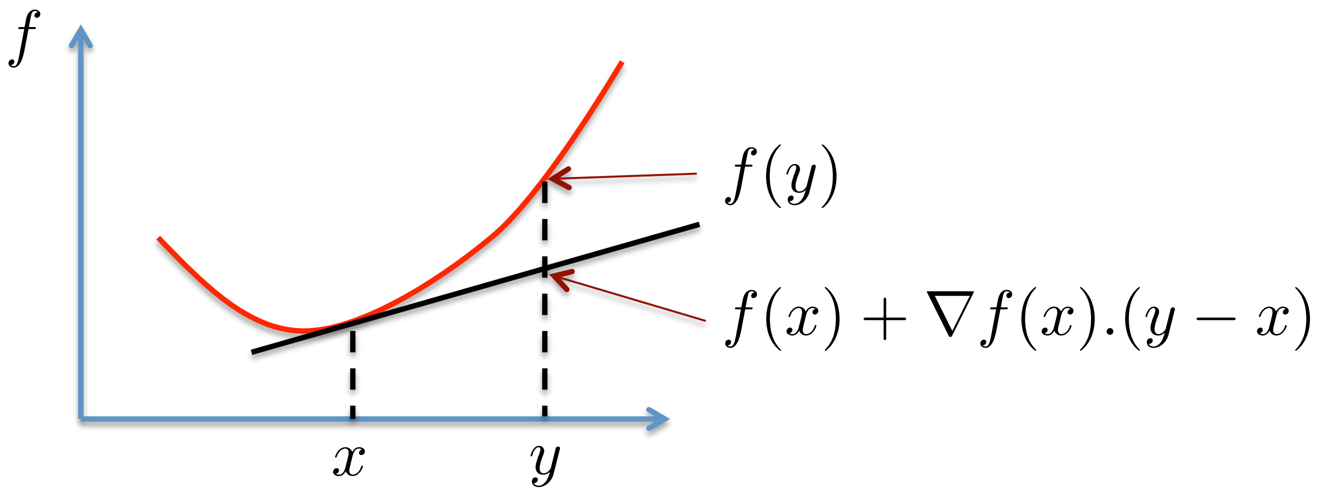
Gradient-descent methods

- A few words on convex analysis
  - Convexity, strong convexity
- Unconstrained smooth optimization
  - Gradient descent algorithms
  - Lower bounds on convergence rates
  - Heavy ball method
- Constrained smooth optimization
- Lagrange Duality
- Fixed point iteration

Convex analysis

- A continuously differentiable function f is convex if

$$\forall x, y \in \mathbb{R}^n, \quad f(y) \geq f(x) + \nabla f(x).(y - x)$$



- Convexity is equivalent to:

$$(\nabla f(x) - \nabla f(y)).(x - y) \geq 0$$

Convex analysis

- Continuously differentiable function with L-lipschitz gradient:

$$0 \leq f(y) - f(x) - \nabla f(x).(y-x) \leq \frac{L}{2}\|x-y\|^2$$

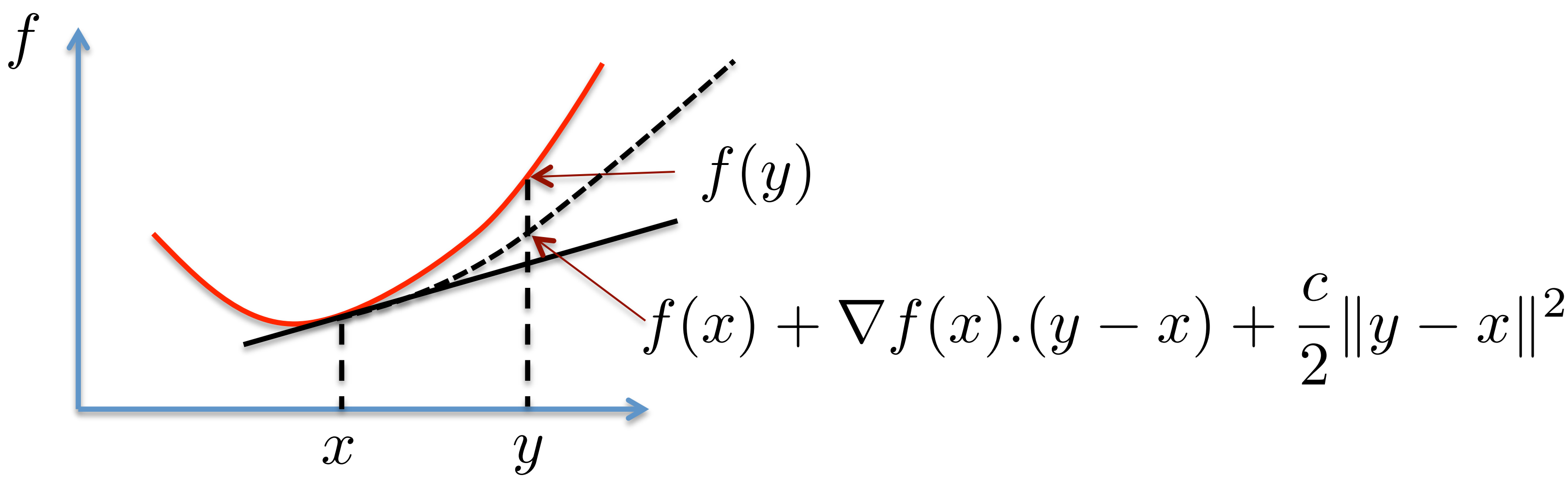
$$f(x) + \nabla f(x).(y-x) + \frac{1}{2L}\|\nabla f(x) - \nabla f(y)\|^2 \leq f(y)$$

$$(\nabla f(x) - \nabla f(y)).(x-y) \leq L\|x-y\|^2$$

Convex analysis

- Strong convex function:

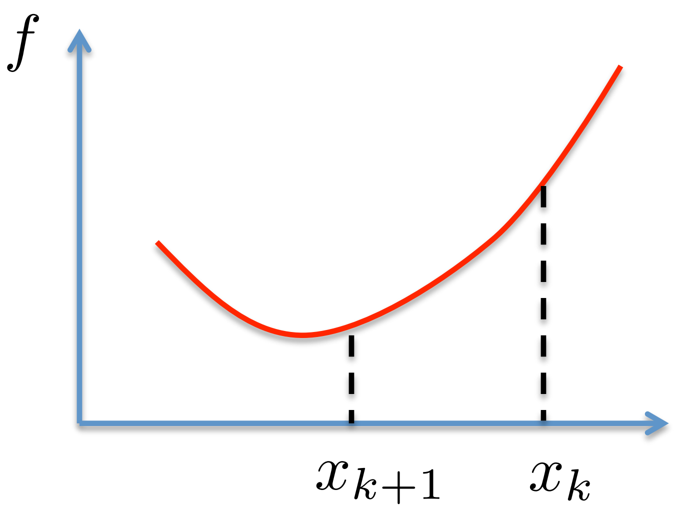
$$\forall x, y \in \mathbb{R}^n, \quad f(y) \geq f(x) + \nabla f(x).(y-x) + \frac{c}{2}\|y-x\|^2$$



GD for unconstrained opt.

minimize $f(x)$
over $x \in \mathbb{R}^n$

- Principle: move in the direction towards the minimizer



Algorithm.
Initialization. Choose $x_0 \in \mathbb{R}^n$.
For each iteration $k \geq 0$:
$$x_{k+1} = x_k - \alpha_k \nabla f(x_k)$$

Convergence

Theorem Let f be a convex and continuously differentiable function. Under GD algorithm:

$$f(x_T) - f^\star \le \frac{\|x_0 - x^\star\|^2 + \sum_{k=0}^{T} \alpha_k^2 \|\nabla f(x_k)\|^2}{2\sum_{k=0}^{T} \alpha_k}$$

Convex and L-lipschitz functions

Theorem Let f be a convex and continuously differentiable function. Assume: $\|x_0 - x^\star\| \leq R$.

(i) ε-optimality can be obtained in $(RL)^2/\epsilon^2$ steps (by choosing $\alpha_k = R/(L\sqrt{T})$)

(ii) For constant step size α, $\lim_{T\to\infty} f(x_T) \leq f^\star + \frac{\alpha L^2}{2}$

(iii) Assuming $\sum_k \alpha_k = \infty, \quad \sum_k \alpha_k^2 < \infty,$

$$\lim_{T\to\infty} f(x_T) = f^\star$$

Convex functions with L-lipschitz gradient

Theorem Let *f* be a convex and continuously differentiable function with *L*-lipschitz gradient. Fixed step size: $\alpha_k = 1/L$

$$f(x_T) - f^\star \leq \frac{2LR^2(f(x_0) - f^\star)}{2LR^2 + T(f(x_0) - f^\star)}$$

c-strongly convex functions with L-lipschitz gradient

Theorem Let *f* be a c-strongly convex and continuously differentiable function with *L*-lipschitz gradient. Fixed step size: $\alpha_k = 2/(c+L)$

$$f(x_T) - f^\star \leq \frac{L}{2}(\frac{\kappa - 1}{\kappa + 1})^{2T} \|x_0 - x^\star\|^2$$

Condition number: $\kappa = L/c$

Lower bounds

Theorem* Let *f* be a convex and continuously differentiable function with *L*-lipschitz gradient. There is no first-order method that guarantees a convergence rate faster than $1/T^2$ at least for $T < n/2$.

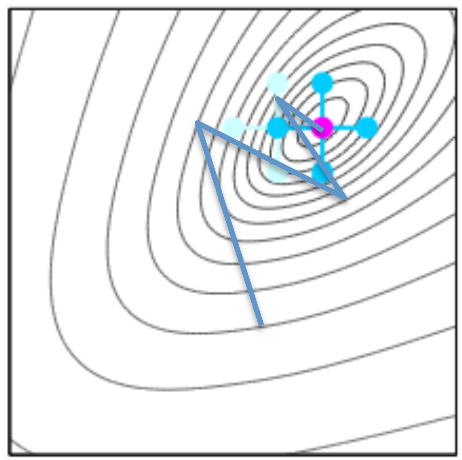
Theorem* Let *f* be a *c*-strongly convex and continuously differentiable function with *L*-lipschitz gradient. For all first order method, we have

$$f(x_T) - f^\star \geq \frac{c}{2}\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^{2T}\|x_0 - x^\star\|^2$$

* Introduction to convex optimization, **Nesterov**, 2004.

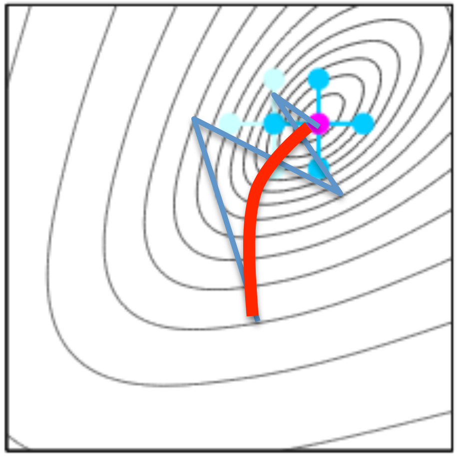
Heavy ball method

- Problem with classical GD: cannot avoid zig-zags



Heavy ball method

- Problem with classical GD: cannot avoid zig-zags



- Heavy ball method adds robustness by accounting for successive moves:

$$x_{k+1} = x_k - \alpha \nabla f(x_k) + \beta(x_k - x_{k-1})$$

- By an optimal choice of parameters, the convergence rate matches Nesterov's lower bound

Smooth convex unconstrained optimization

Class of functions	Algorithm	Complexity	1%
Lipschitz	GD	$1/\varepsilon^2$	10000
Lipschitz gradient	GD	$1/\varepsilon$	100
	Optimal	$1/\sqrt{\varepsilon}$	10
Strongly convex	GD	$\log(1/\varepsilon)$	2.7
	Optimal	$\log(1/\varepsilon)$	2.7

GD for constrained optimization

minimize $f(x)$
over $x \in \Omega \subset \mathbb{R}^n$

- Assumption: the search space is convex and closed
- Gradient projection:

Algorithm.
Initialization. Choose $x_0 \in \Omega$.
For each iteration $k \geq 0$:
$y = x_k - \alpha_k \nabla f(x_k)$
$x_{k+1} = \arg\min_{x \in \Omega} \|y - x\|$

- Similar convergence results as for unconstrained scenarios

Lagrange duality

- Primal problem:

$$\begin{aligned}&\text{minimize } f(x)\\&\text{subject to } g(x) \leq 0\\&\text{over } x \in \Omega\end{aligned}$$

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

$$g = (g_1, \ldots, g_m) : \quad g_j : \mathbb{R}^n \to \mathbb{R}$$

Lagrange duality

- Lagrangean: $L(x, \mu) = f(x) + \sum_{j=1}^{m} \mu_j g_j(x)$

$$\sup_{\mu \geq 0} L(x, \mu) = \begin{cases} f(x), & \text{if } x \text{ feasible} \\ \infty, & \text{otherwise.} \end{cases}$$

$$f^\star = \inf_{x \in \Omega} \sup_{\mu \geq 0} L(x, \mu)$$

- Dual function:

$$q(\mu) = \inf_{x \in \Omega} (f(x) + \sum_j \mu_j g_j(x))$$

$q : \mathbb{R}_+^m \to \mathbb{R}$

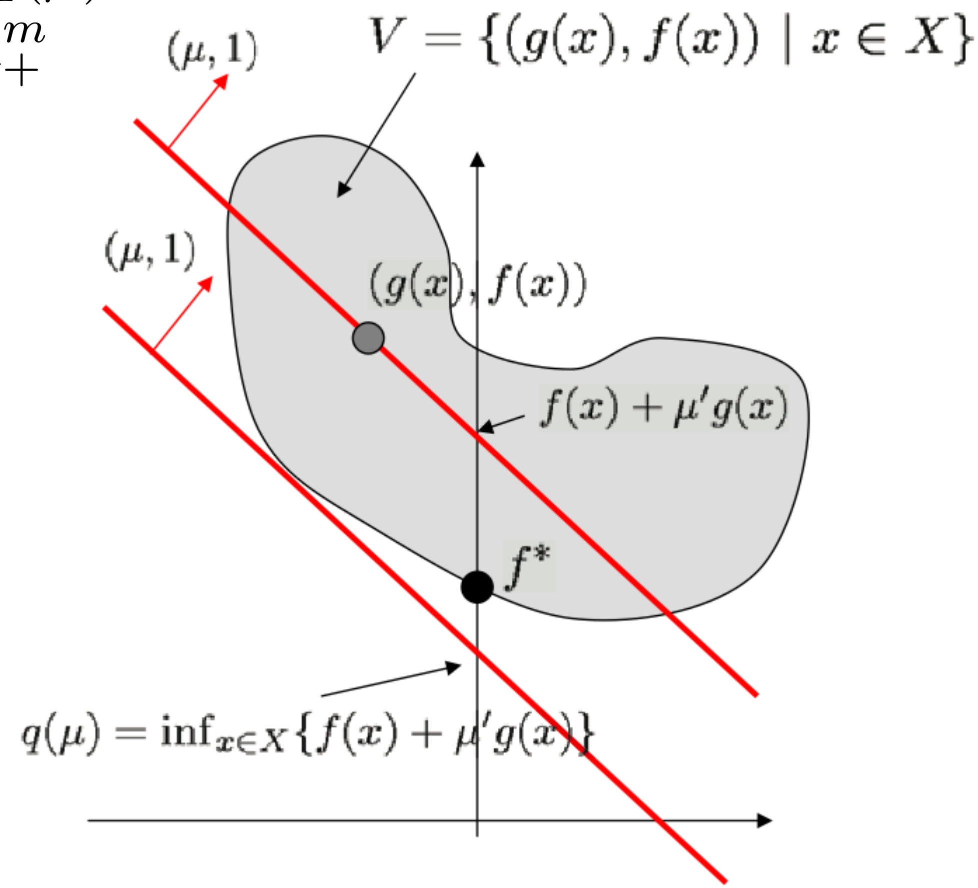
$\mu_j g_j(x)$: cost of of violating the associated constraint

Lagrange duality

- Dual problem: (a convex program)

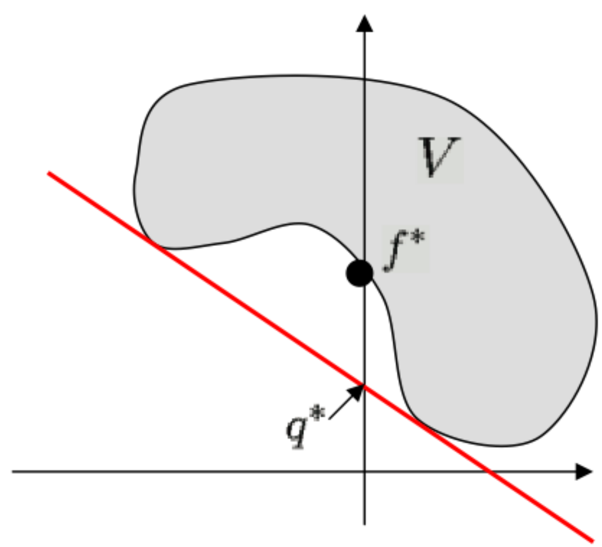
$$\begin{aligned}&\text{maximize } q(\mu)\\&\text{over } \mu \in \mathbb{R}^m_+\end{aligned}$$

$$q^\star = \sup_{\mu \geq 0} q(\mu)$$



Lagrange duality

- Slater condition: $\exists x \in \Omega : g_j(x) < 0, \forall j$
- Weak duality: $q^\star \leq f^\star$
- In case of convex function f and g, if Slater condition is satisfied, strong duality holds: $q^\star = f^\star$
- In absence of convexity, no guarantee on strong duality



Dual gradient algorithm

- In case of strong duality, we may solve the dual problem only, i.e., via GD

Algorithm.
Initialization. Choose $\mu_0 \geq 0$.
For each iteration $k \geq 0$:
$\mu_{k+1} = [\mu_k + \alpha \nabla q(\mu_k)]^+$

Fixed point iterations

- Optimality condition: $\nabla f(x^\star) = 0$
- Iterative methods of the form: $x_{k+1} = F(x_k)$
- For example, GD algorithm is obtained choosing:

$$F(x) = x - \alpha \nabla f(x)$$

whose fixed points are such that $\nabla f(x) = 0$

- Brouwer's fixed point theorem

$X \subset \mathbb{R}^n$ compact convex set

if $F : X \to X$ is continuous, then it has a fixed point

Contraction mappings

- q-contraction mapping:

$$\forall x, y, \quad \|F(y) - F(x)\| \leq q\|y - x\|$$

where we have the choice of the norm, and $q < 1$

- For q-contractions, we have existence and unicity of the fixed point and

$$\|x_k - x^\star\| \leq q^k \|x_0 - x^\star\|$$

Revisiting GD

- GD mapping: $F(x) = x - \alpha \nabla f(x)$

- Assume that *f* is c-strongly convex with L-lipschitz gradient
  then *F* is non-expansive if $0 < \alpha \leq 2/L$
  F is a contraction if $0 < \alpha < 1/c$

(*F* is non-expansive iff $\forall x \neq y, \|F(y) - F(x)\| < \|y - x\|$)