Unconstrained Nonlinear Optimisation

Georgy Gimel'farb
ggim001@cs.auckland.ac.nz

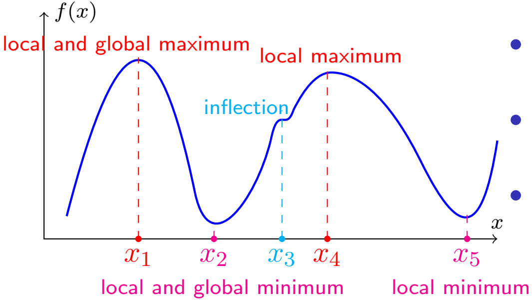
RECOMMENDED READING:

- W. H. Press et al., *Numerical Recipes: The Art of Scientific Computing*. Cambridge Univ. Press, 2007: Section 15.5
- L. R. Foulds: *Optimization Techniques: An Introduction*. Springer-Verlag, 1981: Chapters 7, 8

Extremum of a Function

- One of most important applied problems: to find maximum or minimum value of a function $f(\mathbf{x})$ under constraints $\mathbf{x} \in \mathbf{X}$
  - $f(\mathbf{x}) \equiv f(x_1, \ldots, x_n)$ is a scalar function of n-dimensional vector argument
  - $\mathbf{X}$ is a certain subset of n-dimensional vector space $\mathbf{R}_n$
- Unconstrained optimisation: if $\mathbf{X} = \mathbf{R}_n$

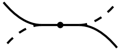
Function of one variable



- Minimum: $\frac{df(x)}{dx} = 0; \; \frac{d^2 f(x)}{dx^2} > 0$

- Maximum: $\frac{df(x)}{dx} = 0; \; \frac{d^2 f(x)}{dx^2} < 0$

- Inflection: $\frac{df(x)}{dx} = 0; \; \frac{d^2 f(x)}{dx^2} = 0$



Functions of Many Variables $f(\mathbf{x})$

- Unconditional local extrema: in these points the gradient of f is equal to zero: $\nabla f(\mathbf{x}) = \left[\frac{\partial f(\mathbf{x})}{\partial x_1}, \ldots, \frac{\partial f(\mathbf{x})}{\partial x_n}\right]^{\mathsf{T}} = \mathbf{0}$
- Whether it is a maximum or a minimum, depends on the matrix of the second derivatives (or Hessian of f):

$$\mathbf{H}(\mathbf{x}) = \begin{bmatrix} \frac{\partial^2 f(\mathbf{x})}{\partial x_1^2} & \frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f(\mathbf{x})}{\partial x_2 \partial x_1} & \frac{\partial^2 f(\mathbf{x})}{\partial x_2^2} & \cdots & \frac{\partial^2 f(\mathbf{x})}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f(\mathbf{x})}{\partial x_n \partial x_1} & \frac{\partial^2 f(\mathbf{x})}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f(\mathbf{x})}{\partial x_n^2} \end{bmatrix}$$

  - Local minimum: if the Hessian is positive definite (the quadratic form $\mathbf{e}^{\mathsf{T}}\mathbf{H}(\mathbf{x})\mathbf{e} > 0$ for any $\mathbf{e} \neq \mathbf{0}$)
  - Local maximum: if the Hessian is negative definite (the quadratic form $\mathbf{e}^{\mathsf{T}}\mathbf{H}(\mathbf{x})\mathbf{e} < 0$ for any $\mathbf{e} \neq \mathbf{0}$)

Quadratic Function $f(\mathbf{x}) = \mathbf{a}^\mathsf{T}\mathbf{x} + \frac{1}{2}\mathbf{x}^\mathsf{T}\mathbf{H}\mathbf{x}$ of n Variables

$$
\begin{aligned}
f(\mathbf{x}) = a_1x_1 + \ldots + a_nx_n + \tfrac{1}{2}\Big(& H_{11}x_1^2 + \widetilde{H}_{12}x_1x_2 + \ldots + \widetilde{H}_{1n}x_1x_n \\
& + \widetilde{H}_{21}x_2x_1 + H_{22}x_2^2 + \ldots + \widetilde{H}_{2n}x_2x_n \\
& \ldots + \widetilde{H}_{n1}x_nx_1 + \widetilde{H}_{n2}x_nx_2 + \ldots + H_{nn}x_n^2 \Big) \\
= \sum_{i=1}^{n} \Big(a_ix_i + \tfrac{H_{ii}}{2}x_i^2 + \sum_{j=i+1}^{n} H_{ij}x_ix_j \Big) & \quad \text{where} \quad H_{ij} = H_{ji} = \frac{\widetilde{H}_{ij} + \widetilde{H}_{ji}}{2}
\end{aligned}
$$

- Gradient $\nabla f(\mathbf{x}) = \mathbf{a} + \mathbf{H}\mathbf{x}$:

$$
\begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{bmatrix} = \begin{bmatrix} a_1 \\ \vdots \\ a_n \end{bmatrix} + \begin{bmatrix} H_{11} & H_{12} & \ldots & H_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ H_{n1} & H_{n2} & \ldots & H_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}
$$

- Hessian $\frac{\partial \nabla f(\mathbf{x})}{\partial \mathbf{x}} \equiv \left[\frac{\partial^2 f}{\partial x_i \partial x_j} \right]_{i,j=1}^{n} \equiv [H_{ij}]_{i,j=1}^{n} \equiv \mathbf{H}$

Useful Definitions of a Positive Definite Matrix

Symmetric $n \times n$ matrix $\mathbf{A}$ is positive definite if one of the following definitions holds:

1. All eigenvalues of $\mathbf{A}$ are positive (> 0)
2. Choleski decomposition $\mathbf{A} = \mathbf{L}\mathbf{L}^\mathsf{T}$ exists
   - Here, $\mathbf{L}$ is a lower triangular matrix with $l_{ii} > 0$
3. Decomposition $\mathbf{A} = \mathbf{L}\mathbf{D}\mathbf{L}^\mathsf{T}$ exists
   - Here, $\mathbf{L}$ is a lower triangular matrix with $l_{ii} = 1$
   - $\mathbf{D}$ is a diagonal matrix with $d_i > 0$
4. All positive pivots (> 0) in Gaussian elimination without pivoting

General conditions 2 or 3 are the most efficient as well as ensure easy solutions to linear systems with coefficients $\mathbf{A}$

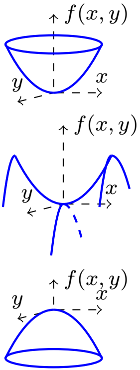
Extrema of $f(x,y) = ax^2 + by^2;\ a \neq 0;\ b \neq 0$

Gradient $\nabla f(x,y) = 0 \Rightarrow \frac{\partial f(x,y)}{\partial x} = 2ax = 0;\ \frac{\partial f(x,y)}{\partial y} = 2by = 0$

$\Rightarrow$ Single extremum at the point $[0,0]^{\mathsf{T}}$

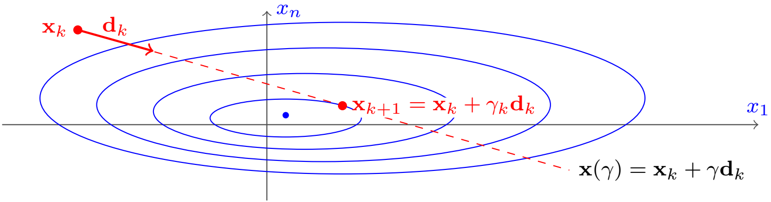
Hessian $\mathbf{H} = \begin{bmatrix} 2a & 0 \\ 0 & 2b \end{bmatrix}$

- Function $f(x,y)$ has the minimum in $[0,0]^{\mathsf{T}}$ if $a > 0$ and $b > 0$: an elliptic paraboloid (the Sylvester's criterion: $2a > 0$ and $4ab > 0$)
- If $a > 0;\ b < 0$ or $a < 0;\ b > 0$, there is no extremum: a hyperbolic paraboloid with a saddle point $[0,0]^{\mathsf{T}}$ (the Sylvester's criterion: $2a > 0$ and $4ab < 0$ or $2a < 0$ and $4ab < 0$)
- Function $f(x,y)$ has the maximum in $[0,0]^{\mathsf{T}}$ if $a < 0$ and $b < 0$: an elliptic paraboloid (the Sylvester's criterion: $2a < 0$ and $4ab > 0$)



Line Search for a Maximal Point

Find a maximiser of $f(\mathbf{x})$ along a direction $\mathbf{d}_k$ from a point $\mathbf{x}_k$



It is used repeatedly in many multivariate search methods

Univariate unimodal functions $u(x)$: properties of the maximiser $x^* = \arg\max_x u(x)$

- If $x_0 < x_1 < x^*$ or $x_0 > x_1 > x^*$, then $u(x_0) < u(x_1) < u(x^*)$
- If $a \leq x^* \leq b$ and $a \leq x_1 < x_2 < b$ or $a < x_1 < x_2 \leq b$, then

$$\left.\begin{array}{lcl} u(x_1) < u(x_2) & \Rightarrow & x_1 < x^* \leq b \\ u(x_1) = u(x_2) & \Rightarrow & x_1 < x^* < x_2 \\ u(x_1) > u(x_2) & \Rightarrow & a \leq x^* < x_2 \end{array}\right\}$$

thus, the search interval $a \leq x^* \leq b$ is reduced

Golden Section Search

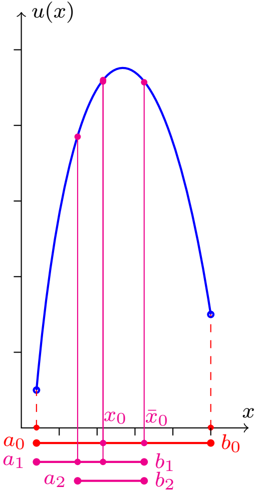
- At each step, reduce an interval $[a_0, b_0]$; $a_0 \leq x^* \leq b_0$, with the maximiser x^* of a unimodal function $u(x)$ by computing symmetric internal points (below: $\tau = (1+\sqrt{5})/2$ is the Greek golden section ratio)

$$\begin{cases} x_i = a_i + (b_i - a_i)(2-\tau) \approx a_i + 0.382(b_i - a_1); & i = 0, 1, 2, \ldots \\ \bar{x}_i = a_i + (b_i - a_i)(\tau - 1) \approx a_i + 0.618(b_i - a_i); & i = 0, 1, 2, \ldots \end{cases}$$

  and evaluating $u(x_i)$ and $u(\bar{x}_i)$:
  - If $u(x_i) > u(\bar{x}_i)$, then set $a_{i+1} \leftarrow a_i$ and $b_{i+1} \leftarrow \bar{x}_i$
  - If $u(x_i) < u(\bar{x}_i)$, then set $a_{i+1} \leftarrow x_i$ and $b_{i+1} \leftarrow b_i$
  - If $u(x_i) = u(\bar{x}_i)$, then set $a_0 \leftarrow x_i$; $b_0 \leftarrow \bar{x}_i$, and start the search again from this new interval $[a_0, b_0]$ and $i = 0$
- Proceed until the interval $[a_0, b_0]$ is sufficiently small, or the next point is within the resolution distance of the last point

Golden Section and Fibonacci Search

Golden section search: an example



- Golden section search is less efficient than the Fibonacci search: for $i = 1, 2, \ldots, n-1$,

$$x_i = a_i + (b_i - a_i)F_{n-i}/F_{n+2-i}$$
$$\bar{x}_i = a_i + (b_i - a_i)F_{n+1-i}/F_{n+2-i}$$

  where F_k is the Fibonacci number: $F_0 = 0$; $F_1 = 1$; $F_k = F_{k-1} + F_{k-2}$, $k = 2, 3, \ldots$

  - Fibonacci search minimises the maximal interval of uncertainty about the maximiser x^* (in that sense it is optimal)
  - But the number of points n to be evaluated in the Fibonacci search has to be prescribed

- Search for the root x^* of the first derivative, $\frac{du}{dx}(x^*) = 0$, be it available, is even more efficient

Gradient Search

- Gradient vector

$$\nabla f(\mathbf{x}) = \left[\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots, \frac{\partial f}{\partial x_n}\right]^\mathsf{T}$$

  is directed to the greatest slope of the function f at any point
- Gradient methods for seeking a maximum (minimum) for f:
  - Evaluate the gradient at an initial point
  - Move along the gradient direction for a computable distance
  - Repeat this process until the maximum (minimum) is found
- If exact partial derivatives are unknown then gradients may be numerically approximated
  - But approximation errors can make the methods less attractive

Basic Gradient Method

Gradient maximisation: the steepest ascent

- Select an initial point $\mathbf{x}_0$ and compute $\nabla f(\mathbf{x})$ at $\mathbf{x}_0$
- Draw a line $\mathbf{x}_0 + t\nabla f(\mathbf{x}_0)$ through $\mathbf{x}_0$ in the gradient direction
- Select the point $\mathbf{x}_1$ on this line yielding the greatest value for f of all points on the line:

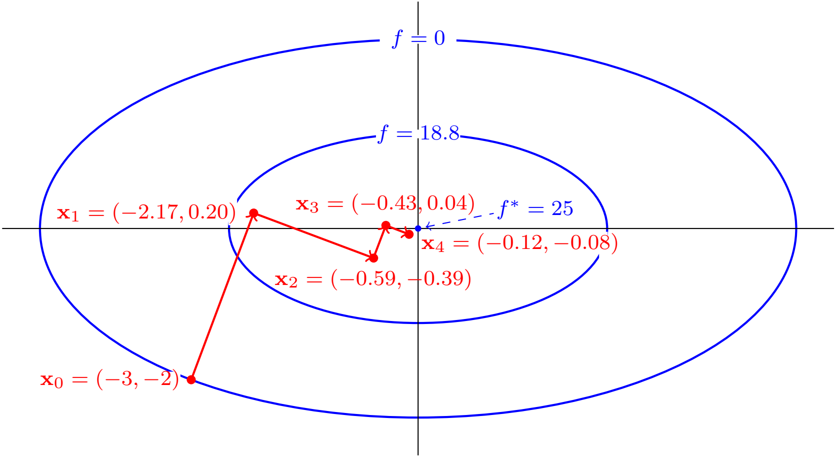
$$f(\mathbf{x}_1) = \max_{t \in (-\infty,\infty)} \{f(\mathbf{x} : \ \mathbf{x} = \mathbf{x}_0 + t\nabla f(\mathbf{x}_0)\}$$

Search for the best point for f along the line:

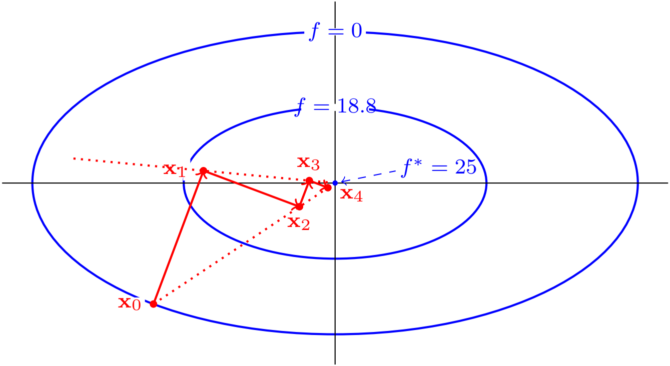
- If computable derivatives and well-behaving f then:
  - Substitute $\mathbf{x}_0 + t\nabla f(\mathbf{x}_0)$ into the equation for f,
  - Differentiate with respect to t, and
  - Set the derivative equal to zero to find t
- Else: any one-dimensional line search

Gradient Maximisation: An Example

$$f(\mathbf{x}) \equiv f(x, y) = 25 - x^2 - 4y^2;\ \nabla f(\mathbf{x}) = (-2x, -8y);\ \mathbf{x}_0 = (-3, -2):$$
$$f(\mathbf{x}_0) = 0;\ f(\mathbf{x}_1) = 20.1;\ f(\mathbf{x}_2) = 24.0;\ f(\mathbf{x}_3) = 24.8;\ f(\mathbf{x}_4) = 24.9;\ \ldots$$



Accelerated Gradient Search



- Once $i > 2$, $\mathbf{x}_i$ for i odd is found by gradient search from $\mathbf{x}_{i-1}$, and $\mathbf{x}_{i+1}$ is found by an accelerated step by maximising over the line through $\mathbf{x}_i$ and $\mathbf{x}_{i-2}$
- Global maximum of a negative definite quadratic function of n variables is provably found after $2n-1$ steps of this procedure

Conjugate Directions

- Producing a sequence of points $\mathbf{x}_0$, $\mathbf{x}_1$, ..., such that each point improves values in maximising a quadratic function $f(\mathbf{x}) = \mathbf{a}^\mathsf{T} + \frac{1}{2}\mathbf{x}^\mathsf{T}\mathbf{H}\mathbf{x}$
- All directions $\mathbf{d}_i$ of search obey the relationship: $\mathbf{d}_i^\mathsf{T}\mathbf{H}\mathbf{d}_j = 0$ for all i, j, $i \neq j$

General method of conjugate directions

- Choose $\mathbf{x}_0$ near an optimal point or randomly
- Carry out a one-dimensional search in the first conjugate direction $\mathbf{d}_1$ to find a new point $\mathbf{x}_1$
- For $i = 2, \ldots, n$, search for a new point $\mathbf{x}_i$ along the next conjugate direction $\mathbf{d}_i$ such that $\mathbf{d}_j^\mathsf{T}\mathbf{H}\mathbf{d}_k = 0$; $j, k \leq i$, $j < k$
- The maximum is located in at most n steps

Conjugate Gradients

Each new conjugate direction – from the gradient at the point concerned

Conjugate gradient method for maximising $f(\mathbf{x})$

- Choose a starting point $\mathbf{x}_0$
- Carry out a one-dimensional search in the gradient direction $\mathbf{d}_1 = \nabla f(\mathbf{x}_0)$ to find the maximum point $\mathbf{x}_1$
- For $i = 2, \ldots, n$, form $\mathbf{d}_i$ from $\nabla f(\mathbf{x}_i)$ to be conjugate to $\mathbf{d}_{i-1}$: $\mathbf{d}_i = \nabla f(\mathbf{x}_i) + \gamma_{i-1}\mathbf{d}_{i-1}$ and $\mathbf{d}_i^{\mathsf{T}}\mathbf{H}\mathbf{d}_{i-1} = 0$

$$\Rightarrow \mathbf{d}_i = \nabla f(\mathbf{x}_i) - \left(\frac{(\nabla f(\mathbf{x}_i))^{\mathsf{T}}\mathbf{H}\mathbf{d}_{i-1}}{\mathbf{d}_{i-1}^{\mathsf{T}}\mathbf{H}\mathbf{d}_{i-1}} \right) \mathbf{d}_{i-1}$$

  - Can be proven by induction: all $\mathbf{d}_i$ are mutually conjugate
  - In actual implementation the directions $\mathbf{d}_i$ can be computed by a simple recurrence relation, and only a few vectors and no matrices need be stored

Direct Search Methods

- If both the gradient and Hessian of $f(\mathbf{x})$ are too complicated to compute but f can be evaluated at any point $\mathbf{x} \in \mathbb{R}_n$

Pattern search of K. Hooke and T. A. Jeeves

- For $i = 1, \ldots, n$ sequentially:
  - If $f(x_1, \ldots, x_i + \varepsilon_i, \ldots, x_n) > f(x_1, \ldots, x_i, \ldots, x_n)$, replace $x_i \leftarrow x_i + \varepsilon$
  - Else if $f(x_1, \ldots, x_i - \varepsilon_i, \ldots, x_n) > f(x_1, \ldots, x_i, \ldots, x_n)$, replace $x_i \leftarrow x_i - \varepsilon$
- Repeat this cycle of perturbations until no perturbations about $\mathbf{x}_j$ bring about an improvement
- Halve the pre-defined perturbation sizes ε_i and repeat the process while the next point brings an improvement over $\mathbf{x}_j$

Sectioning

One-at-a-time search, or sectioning, from an initial point $\mathbf{x}_0$

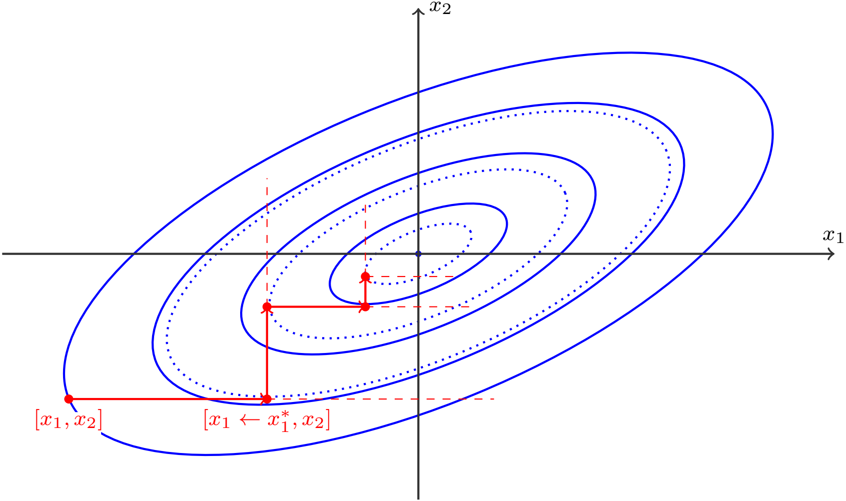
- For $i = 1, \ldots, n$ sequentially, search for the maximum in the direction of the variable x_i by one of the one-dimensional search methods and replace x_i by the maximiser x_i^*: $x_i \leftarrow x_i^*$
- Repeat this cycle of one-dimensional searches until the steps $x_i - x_i^*$; $i = 1, \ldots, n$ become less than a given threshold

Convergence rate is usually too slow and the search may halt far from the optimum

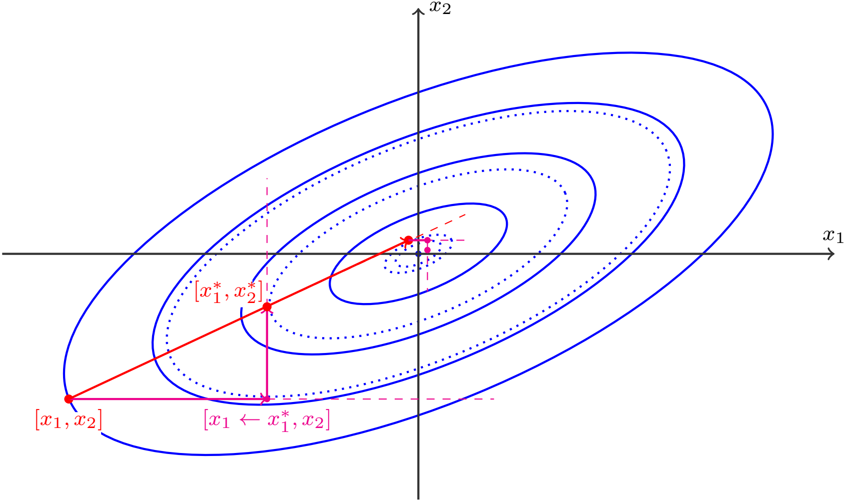
Accelerated search of H. H. Rosenbrock

- Use one-at-a-time search from $\mathbf{x}_0$ to find the next point $\mathbf{x}_1^*$ and the direction $\boldsymbol{\delta}$ with components $\delta_i = x^*_{1:i} - x_{0:i}$
- Search for the maximum in the direction $\boldsymbol{\delta}$ and replace $\mathbf{x}_0$ by the maximiser $\mathbf{x}_1$ found
- Repeat this cycle until $\mathbf{x}_t$ and $\mathbf{x}_{t-1}$ are closer than a threshold

One-at-a-time Search: An Example



Rosenbrock's Search: An Example



Search Method of M. J. D. Powell

- Similar to the method of conjugate gradients, except that derivatives are not required
- Similar to the Rosenbrock's method, except that each search is carried out along a conjugate direction
  - Directions $\mathbf{d}_1, \ldots, \mathbf{d}_n$ become conjugate w.r.t. an approximation of the Hessian matrix

If $\mathbf{x}_0$ is the initial estimate of the maximiser of $f(\mathbf{x})$ then

1. Set the search directions be equal to the coordinate directions
2. For $i = 1, \ldots, n$ sequentially find the maximiser $\mathbf{x}_i$ of f in the the direction $\mathbf{d}_i$ from $\mathbf{x}_{i-1}$
3. Let $\mathbf{d}_i \leftarrow \mathbf{d}_{i+1}$ for $i = 1, \ldots, n-1$ and $\mathbf{d}_n = \mathbf{x}_n - \mathbf{x}_0$
4. Set $\mathbf{x}_0$ be equal to the maximiser of f in the $\mathbf{d}_n$ direction from $\mathbf{x}_n$
5. Return to 2 unless some termination criterion is met