Multidimensional Strategies \((n > 1)\)

Typical classification: According to type of information

- **Direct search methods**
  - only \(f(x)\)
- **Gradient methods**
  - \(f(x)\) and \(\nabla f(x)\)
- **Newton methods**
  - \(f(x)\), \(\nabla f(x)\) and \(\nabla^2 f(x)\)

\[
\nabla f(x) = \begin{pmatrix} \frac{\partial f}{\partial x_1} & \cdots & \frac{\partial f}{\partial x_n} \end{pmatrix}^T \\
\nabla^2 f(x) = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\
& \ddots & \vdots \\
\frac{\partial^2 f}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{pmatrix}
\]

1. Direct Search Strategies

1.1 Grid Search

Initial intervals: \(D_i\), final intervals: \(d_i\)

Number of trials: \[N \sim \prod_{i=1}^{n} \left(\frac{D_i}{d_i}\right)^n\]

If \(\forall i: d_i = d, D_i = D\) \[\Rightarrow N \sim \left(\frac{D^n}{d^n}\right)^n\]

Example: \[\frac{d}{D} = 0.1, \quad n = 17\]

\[\Rightarrow N \sim 10^{17} \sim 3\text{ billion years}\]

“Course of dimensions” (Bellmann).

10^{100} : Practical infinite, because

10^{80} \sim number of elementary particles

10^{40} \sim number of elementary time steps (in the universe, since big bang)

1.2 Monte-Carlo Strategy

Pure random search.

Random samples, \(n\)-dimensionally, uniformly distributed.

Hitting probability, within one trial:

\[n = 1: \quad p_1 = \frac{d}{D}\]
\[n > 1: \quad p_1 = \left(\frac{d}{D}\right)^n\]

Probability of not hitting target region in

1 trial \[p_t = 1 - \left(\frac{d}{D}\right)^n\]

\(N\) trials \[p_t = \left[1 - \left(\frac{d}{D}\right)^n\right]^N\]

At least one hit within \(N\) trials:

\[p^N = 1 - \left[1 - \left(\frac{d}{D}\right)^n\right]^N\]
Grid search better than MC:
MC does not exclude repeated samples or trials very close to each other.

But: Methods using random sampling not in an independent way can be much better.

Improvements of MC:
- Reduce the sampling region during search, to more and more promising regions.
- Adapt probability density of sampling according to successes.

⇒ No longer completely parallel.

Wrong claim of Brooks (1959):
Effort independent of \( n \)

\[ N \sim \frac{V}{v} = \frac{\text{total volume}}{\text{target volume element}} \]

Volume-oriented methods: Time complexity \( O(e^n) \)

1.4 Coordinate Strategy / Gauß-Seidel-Strategy
Basic features:
- Change only one variable at a time.
- For the actual variable:
  Use one-dimensional optimization method to find relative optimum.
- Take each coordinate direction in turn.
- Continue until no improvement possible.

Algorithm:
1. Choose arbitrary \( x_1, i = 1 \).
2. Find search direction \( \vec{v}^{(i)} \) as
   \[ \vec{v}^{(i)} = \vec{e}_i \]
   \[ \vec{e}_i = \left( \begin{array}{c}
   1 \\
   0 \\
   \vdots \\
   0
   \end{array} \right) \text{mod } n \]
   \( \vec{e}_i \text{ cycles sequentially, all directions, in } \vec{e}_i \text{ only position } i \text{ is 1.} \)
3. Determine whether \( a^{(i)} > 0 \) or \( a^{(i)} < 0 \):
   \[ f_i = f(\tilde{x}^{(i)}), \quad f^+ = f(\tilde{x}^{(i)} + \epsilon \vec{v}^{(i)}) \]
   \[ f^- = f(\tilde{x}^{(i)} - \epsilon \vec{v}^{(i)}) \]
   \[ f^+ < f_i \Rightarrow a^{(i)} \text{ correct direction} \]
   \[ f^- < f_i \Rightarrow -a^{(i)} \text{ correct direction} \]

Path-Oriented Methods
Follow some pathline towards optimum.
Distance between two random points within \( n \)-dimensional cube of side-length \( L \):

\[ l = L \cdot \sqrt[2]{n} \]

⇒ Effort \( \sim \sqrt{n} \)

Iteration rule again:

\[ a^{(k)}(\tilde{x}^{(k)}) = a^{(k-1)}(\tilde{x}^{(k-1)}) + a^{(k)} \cdot \vec{e}^{(k)} \]

Direction: E.g. coordinate direction (cyclically)
gradient direction (locally optimal)
random direction (equally distributed)

Step size: E.g. constant maximal (to relative optimum)
random (normally distributed)

Advanced algorithms “learn” \( a \) and \( \vec{v} \)
- adaptive
- learning by doing

Evolution strategies:
- random, equally distributed
- random, normally distributed
- fixed for \( n \gg 1 \)

4. Find optimal step length \( a^{(i)} \), such that
   \[ f(a^{(i)} \pm \vec{v}^{(i)}) = \min f(a^{(i)} \pm \vec{v}^{(i)}) \]

5. Set \( \tilde{x}^{(i+1)} = \tilde{x}^{(i)} + a^{(i)} \vec{v}^{(i)} \)
   Depending on the direction for decreasing \( f \) and
   \( f_{i+1} = f(\tilde{x}^{(i+1)}) \).

6. Set \( i = i + 1 \), go to step 2.
   Continue until no significant change is achieved in the value of \( f \).

Comments:
- Rapid convergence only if \( f(x) \) are approximately concentric surfaces of a hypersphere.
- Inaccurate line searches can be helpful (under- or overrelaxation).
- Deviations from the cyclic scheme possible.
- For badly structured functions, convergence can be extremely slow.
Gauss-Seidel Method:

\[ x_{i+1} = \frac{1}{a_{ii}} \left( b_i - \sum_{j=1}^{i-1} a_{ij} x_{j+1} - \sum_{j=i+1}^{n} a_{ij} x_j \right) \]

Relative bad conditioned!

How to avoid slow convergence?

Change directions of search in favourable manner.

⇒ Lines joining the alternate points of coordinate
Search: → minimum direction

[ if f quadratic, n = 2: All such lines pass minimum ]
not for n > 2.

Pattern directions.

4. If \( \bar{y}_k \) remains same as \( x_k \): (no success)
   Reduce step lengths \( \Delta x_i \), \( i = 1 \); go to 3.
   If \( \bar{y}_k \neq x_k \): (at least one success)
   \( x_{k+1} = \bar{y}_k \); go to 5.

5. Establish a pattern direction \( \vartheta \) as
   \[ \vartheta = \frac{x_{k+1} - x_k}{\|x_{k+1} - x_k\|} \]
   and find a point \( \bar{y}_{k+1,0} \) as
   \[ \bar{y}_{k+1,0} = x_{k+1} + \lambda \vartheta \]
   \( \lambda \) = step length, e.g. \( \lambda = 1 \)
   or from one-dim. opt.

6. \( k = k + 1 \); \( f_k = f(\bar{y}_k) \); \( i = 1 \); repeat step 3.
   if at the end of step 3, \( f(\bar{y}_{k,n}) < f(x_k) \),
   take \( x_{k+1} = \bar{y}_{k,n} \) new base point
   go to 5.
   if \( f(\bar{y}_{k,n}) \geq f(x_k) \)
   \( x_{k+1} = x_k \)
   reduce step length \( \Delta x_i \)
   \( k = k + 1 \); go to step 2.

7. Terminate, when \( \max_i (\Delta x_i) < \varepsilon \).
Example of Hooke and Jeeves

Algorithm Outline:

1. Begin with coordinate directions \( \mathbf{v}_i^{(0)} = \mathbf{e}_i \)
   Fixed step sizes
   If success: Change to new point
     Increased step size \( a_i^{(k+1)} = 3a_i^{(k)} \)
   If failure: No trial into inverse direction
     \( a_i^{(k+1)} = \frac{1}{3} a_i^{(k)} \)

2. As soon as for each direction at least one success and afterwards a failure was achieved
   (usually only after more than one iteration):
   Orthogonalization: New set of directions.

\[
\begin{align*}
\mathbf{d}_i^{(k)} & : \text{Total step covered in direction } \mathbf{v}_i^{(k)} \\
\alpha_i^{(k)} & = \sum_{j=1}^{n} d_j^{(k)} v_j^{(k)} \\
\mathbf{w}_i & = \mathbf{d}_i - \sum_{j=1}^{n} (\alpha_j^{(k)} v_j^{(k+1)}) v_j^{(k+1)} & \forall i = 2, \ldots, n \\
\mathbf{v}_i^{(k+1)} & = \frac{\mathbf{w}_i}{\|\mathbf{w}_i\|} \text{ Gram-Schmidt-Orth. } O(n^2)
\end{align*}
\]

3. Stop, if \( \|\mathbf{w}_0\| < \epsilon \) and \( \|\mathbf{w}_2\| > 0.3\|\mathbf{w}_1\| \).

1.6 Strategy of Rosenbrock: Rotating Coordinates

Idea: Remove limitation on number of search directions. Any rotation coordinate system. One axis to point in most favorable direction.
1.7 Simplex Strategy (Nelder & Mead)

Idea: Reduce, as much as possible, number of simultaneous trials.

- Sequential algorithm
- Adopts variability to topology
- \((n+1)\)-strategy

Reflection:

\[ x_h = \text{vertex with highest } f \text{ among vertices} \]
\[ x_i = (1 + \alpha)x_0 - \alpha x_h \]
\[ x_0 = \frac{1}{n} \sum_{i=1, i \neq h}^{n+1} x_i \text{ centroid} \]
\[ \alpha = \frac{\text{distance }(x_i, x_0)}{\text{distance }(x_h, x_0)} > 0 \text{ reflection coeff. } \alpha = 1.0 \]

Reject \(x_h\), include \(x_i\), if \(f(x_i) < f(x_h) < f(x_i)\).
\(x_i\) is lowest \(f\)-value.

Expansion:

If reflection produces new minimum \(x_i\):
\[ x_c = \gamma x_i + (1 - \gamma)x_0 \]
\[ \gamma = \frac{\text{distance }(x_i, x_0)}{\text{distance }(x_h, x_0)} > 1 \text{ expansion coeff. } \gamma = 2.0 \]

If \(f(x_i) < f(x_i)\):
Replace \(x_h\) by \(x_c\).
Restart reflection
If \(f(x_i) > f(x_i)\):
Replace \(x_h\) by \(x_i\).
Restart reflection

Contraction:

If reflection yields \(x_i\) with:
\(f(x_i) < f(x_i) \land f(x_i) > f(x_i), \forall i \neq h\)
Replace \(x_h\) by \(x_i\) and Contract:
\[ x_c = \beta x_i + (1 - \beta)x_0 \]
\[ 0 \leq \beta = \frac{\text{distance }(x_i, x_0)}{\text{distance }(x_h, x_0)} \leq 1 \text{ contraction coeff. } \beta = 0.5 \]

If \(f(x_i) > f(x_h)\): Also contraction, not changing previous \(x_h\).
Reflection causing oscillation

\[ x_1 \text{ and } x_2 \text{ have identical function value} \]

⇒ Reject vertex corresponding to 2\textsuperscript{nd} worst function value.