9 Ableitungsfreie Optimierung/Direkte Suchverfahren

umfasst Verfahren zu $\min_{x\in\mathbb{R}^n} f(x)$, die nur ein Orakel 0. Ordnung (nur Funktionswerte) benötigen. Sie werden in der Praxis oft eingesetzt, weil

- Funktionen durch Simulationsprogramme ausgewertet werden, die wirklich kein automatisches Differenzieren erlauben,
- Benutzer nicht wissen, was eine Ableitung ist, wozu sie gut ist, und wie man sie bekommt,
- schon bei der Modellbildung auf Optimierungswünsche und Ableitungsinformation geachtet werden müsste (für Optimierung geeignete Modelle zu bauen verlangt oft viel Erfahrung!).

Bei Funktionen, die nicht stückweise glatt sind und erratisch springen, hilft höchstens zufälliges Suchen, sie kommen aber in der Praxis nicht vor.

Meist sind Funktionen stückweise glatt mit Knick- und/oder Sprungstellen.

Für stetige Funktionen mit Knickstellen sollte man Verfahren der nichtglatten Optimierung (nonsmooth optimization) verwenden, die sogenannte Subdifferentialinformation aus Subgradienten nutzen.

Sprungstellen sind nur durch Gebietsunterteilung zu kontrollieren. Auf den stetigen Teilen hilft wieder glatte oder nichtglatte Optimierung.

All das benötigt ein gewisses Problemverständnis. Ohne dieses probiert man zuerst Direkte Suchverfahren.

9 Ableitungsfreie Optimierung/Direkte Suchverfahren

umfasst Verfahren zu $\min_{x\in\mathbb{R}^n} f(x)$, die nur ein Orakel 0. Ordnung (nur Funktionswerte) benötigen. Sie werden in der Praxis oft eingesetzt, weil

- Funktionen durch Simulationsprogramme ausgewertet werden, die wirklich kein automatisches Differenzieren erlauben,
- Benutzer nicht wissen, was eine Ableitung ist, wozu sie gut ist, und wie man sie bekommt,
- schon bei der Modellbildung auf Optimierungswünsche und Ableitungsinformation geachtet werden müsste (für Optimierung geeignete Modelle zu bauen verlangt oft viel Erfahrung!).

Bei Funktionen, die nicht stückweise glatt sind und erratisch springen, hilft höchstens zufälliges Suchen, sie kommen aber in der Praxis nicht vor.

Alle Suchverfahren nehmen an, dass bekannte Funktionswerte Aufschluss über die lokale Funktionsentwicklung geben, haben explizit oder implizit das Ziel, ein lokales Modell der Funktion zu erstellen (Gradienten/ Subdifferentiale tun genau das). Verbreitete/wichtige Verfahren sind

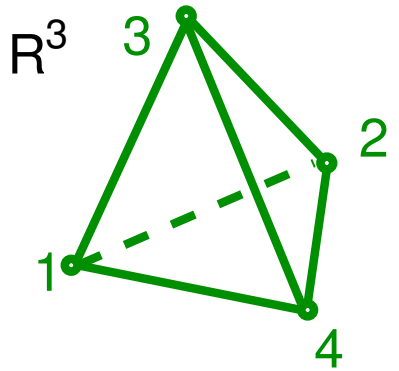
- Das „Simplex“-Verfahren (Nelder-Mead, neuer: Torczon) [verbreitet]
- Das Kriging-Verfahren [für wenige Auswertungen]
- NEWUOA von Powell [wichtig]

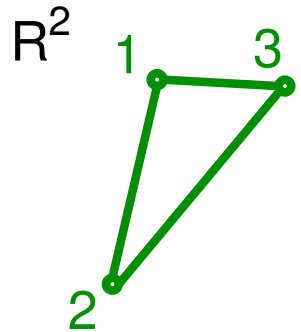
Inhalt

9.1 Das Simplex-Verfahren von Nelder und Mead

. . . hat nichts mit dem Simplex-Verf. der Linearen Optimierung zu tun!

Idee: Um $f : \mathbb{R}^n \to \mathbb{R}$ zu minimieren, wähle $n+1$ affin unabhängige Punkte $x^{(1)}, \ldots, x^{(n+1)} \in \mathbb{R}^n$, diese bilden die Ecken eines n-dimensionalen Simplex.

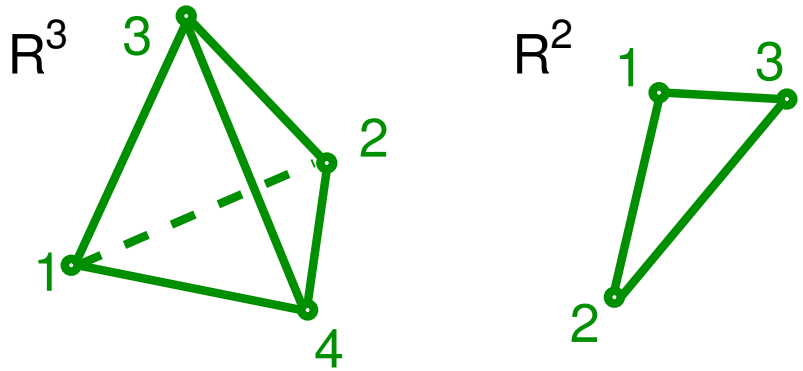




9.1 Das Simplex-Verfahren von Nelder und Mead

. . . hat nichts mit dem Simplex-Verf. der Linearen Optimierung zu tun!

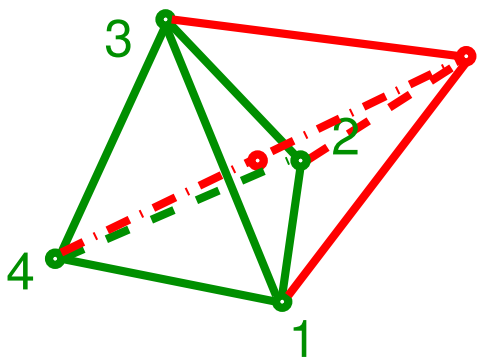
Idee: Um $f : \mathbb{R}^n \to \mathbb{R}$ zu minimieren, wähle $n+1$ affin unabhängige Punkte $x^{(1)}, \ldots, x^{(n+1)} \in \mathbb{R}^n$, diese bilden die Ecken eines n-dimensionalen Simplex.



Sortiere die Punkte so, dass

$$f(x^{(1)}) \leq \cdots \leq f(x^{(n+1)}).$$

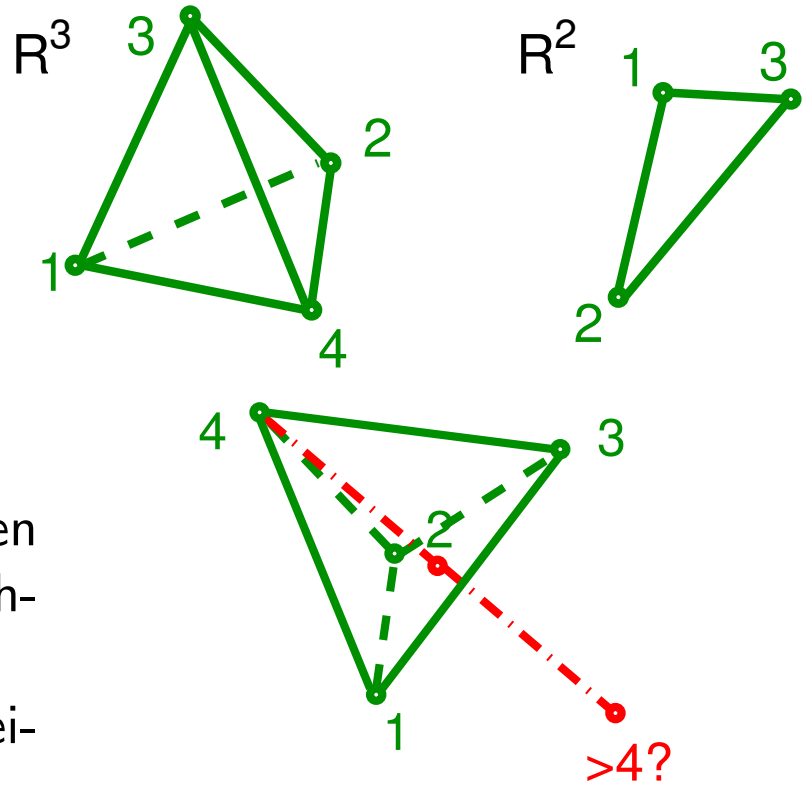
Vorstellung: Die Ebene durch die besseren Punkte $x^{(1)}, \ldots, x^{(n)}$ trennt den schlechten Punkt $x^{(n+1)}$ von den guten.
$\to$ „Klappe“ den Punkt auf die andere Seite (am Baryzentrum spiegeln).



9.1 Das Simplex-Verfahren von Nelder und Mead

. . . hat nichts mit dem Simplex-Verf. der Linearen Optimierung zu tun!

Idee: Um $f : \mathbb{R}^n \to \mathbb{R}$ zu minimieren, wähle $n+1$ affin unabhängige Punkte $x^{(1)}, \ldots, x^{(n+1)} \in \mathbb{R}^n$, diese bilden die Ecken eines n-dimensionalen Simplex.



Sortiere die Punkte so, dass

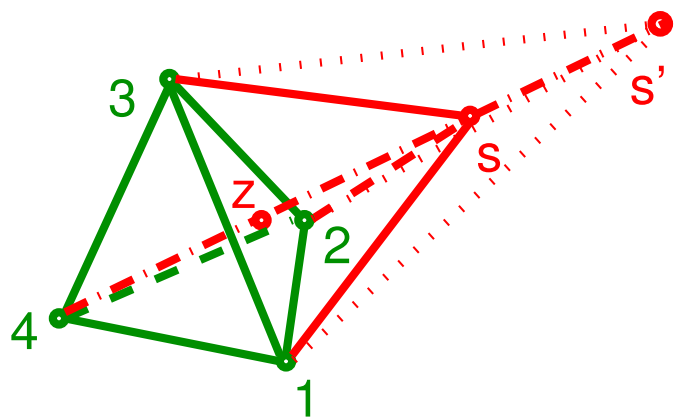
$$f(x^{(1)}) \leq \cdots \leq f(x^{(n+1)}).$$

Vorstellung: Die Ebene durch die besseren Punkte $x^{(1)}, \ldots, x^{(n)}$ trennt den schlechten Punkt $x^{(n+1)}$ von den guten.
$\to$ „Klappe“ den Punkt auf die andere Seite (am Baryzentrum spiegeln).

Ist der Punkt immer noch am schlechtesten, schrumpfe den Simplex in dieser Richtung, ist er sehr gut, strecke die Richtung.

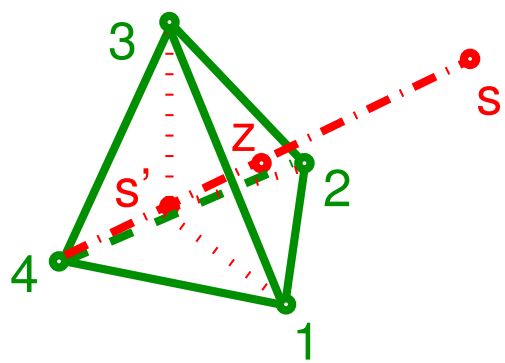
Algorithmisches Schema des Nelder-Mead-Verfahrens:

0. Wähle affin unabh. Punkte $x^{(1)}, \ldots, x^{(n+1)} \in \mathbb{R}^n$ und sortiere sie, $f(x^{(1)}) \leq \cdots \leq f(x^{(n+1)})$, wähle $\alpha > 0$, $\beta > \max\{1, \alpha\}$, $\gamma \in (0, 1)$ [z.B., $\alpha = 1$, $\beta = 2$, $\gamma = \frac{1}{2}$]
1. Setze $z \leftarrow \frac{1}{n} \sum_{i=1}^{n} x_i$ (Baryzentrum der „guten“ Punkte), $s \leftarrow z + \alpha(z - x^{(n+1)})$ (gespiegelter Punkt).
2. Falls $f(x^{(1)}) \leq f(s) \leq f(z)$: **(Spiegelung, reflection)** Setze $x^{(n+1)} \leftarrow s$, sortiere neu, GOTO 1.
3. Falls $f(s) < f(x^{(1)})$: **(Streckung, expansion)** Teste $s' \leftarrow z + \beta(s - z)$. Gilt $f(s') < f(s)$, setze $s \leftarrow s'$. Setze $x^{(n+1)} \leftarrow s$, sortiere neu, GOTO 1.



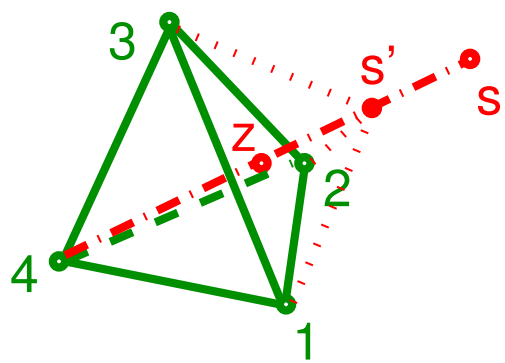
Algorithmisches Schema des Nelder-Mead-Verfahrens:

0. Wähle affin unabh. Punkte $x^{(1)}, \ldots, x^{(n+1)} \in \mathbb{R}^n$ und sortiere sie, $f(x^{(1)}) \leq \cdots \leq f(x^{(n+1)})$, wähle $\alpha > 0$, $\beta > \max\{1, \alpha\}$, $\gamma \in (0,1)$ [z.B., $\alpha = 1$, $\beta = 2$, $\gamma = \frac{1}{2}$]
1. Setze $z \leftarrow \frac{1}{n} \sum_{i=1}^{n} x_i$ (Baryzentrum der „guten“ Punkte), $s \leftarrow z + \alpha(z - x^{(n+1)})$ (gespiegelter Punkt).
2. Falls $f(x^{(1)}) \leq f(s) \leq f(z)$: **(Spiegelung, reflection)** Setze $x^{(n+1)} \leftarrow s$, sortiere neu, GOTO 1.
3. Falls $f(s) < f(x^{(1)})$: **(Streckung, expansion)** Teste $s' \leftarrow z + \beta(s - z)$. Gilt $f(s') < f(s)$, setze $s \leftarrow s'$. Setze $x^{(n+1)} \leftarrow s$, sortiere neu, GOTO 1.
4. Falls $f(x^{(n)}) < f(s)$: **(Schrumpfung, contraction)** Teste $s' \leftarrow \begin{cases} z + \gamma(x^{(n+1)} - z) & \text{falls } f(s) \geq f(x^{(n+1)}), \\ z + \gamma(s - z) & \text{falls } f(s) < f(x^{(n+1)}). \end{cases}$ Gilt $f(s') < \min\{f(x^{(n+1)}), f(s)\}$, setze $x^{(n+1)} \leftarrow s'$,



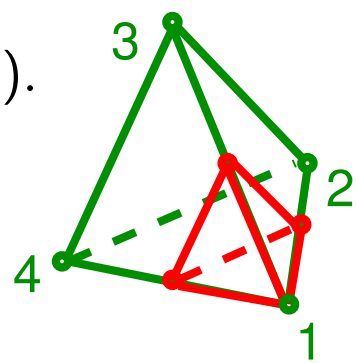
Algorithmisches Schema des Nelder-Mead-Verfahrens:

0. Wähle affin unabh. Punkte $x^{(1)}, \ldots, x^{(n+1)} \in \mathbb{R}^n$ und sortiere sie, $f(x^{(1)}) \leq \cdots \leq f(x^{(n+1)})$, wähle $\alpha > 0$, $\beta > \max\{1, \alpha\}$, $\gamma \in (0,1)$ [z.B., $\alpha = 1$, $\beta = 2$, $\gamma = \frac{1}{2}$]
1. Setze $z \leftarrow \frac{1}{n} \sum_{i=1}^{n} x_i$ (Baryzentrum der „guten“ Punkte), $s \leftarrow z + \alpha(z - x^{(n+1)})$ (gespiegelter Punkt).
2. Falls $f(x^{(1)}) \leq f(s) \leq f(z)$: **(Spiegelung, reflection)**
   Setze $x^{(n+1)} \leftarrow s$, sortiere neu, GOTO 1.
3. Falls $f(s) < f(x^{(1)})$: **(Streckung, expansion)**
   Teste $s' \leftarrow z + \beta(s - z)$. Gilt $f(s') < f(s)$, setze $s \leftarrow s'$.
   Setze $x^{(n+1)} \leftarrow s$, sortiere neu, GOTO 1.
4. Falls $f(x^{(n)}) < f(s)$: **(Schrumpfung, contraction)**
   Teste $s' \leftarrow \begin{cases} z + \gamma(x^{(n+1)} - z) & \text{falls } f(s) \geq f(x^{(n+1)}), \\ z + \gamma(s - z) & \text{falls } f(s) < f(x^{(n+1)}). \end{cases}$
   Gilt $f(s') < \min\{f(x^{(n+1)}), f(s)\}$, setze $x^{(n+1)} \leftarrow s'$,



Algorithmisches Schema des Nelder-Mead-Verfahrens:

0. Wähle affin unabh. Punkte $x^{(1)}, \ldots, x^{(n+1)} \in \mathbb{R}^n$ und sortiere sie, $f(x^{(1)}) \leq \cdots \leq f(x^{(n+1)})$, wähle $\alpha > 0$, $\beta > \max\{1, \alpha\}$, $\gamma \in (0,1)$ [z.B., $\alpha = 1$, $\beta = 2$, $\gamma = \frac{1}{2}$]
1. Setze $z \leftarrow \frac{1}{n} \sum_{i=1}^{n} x_i$ (Baryzentrum der „guten“ Punkte), $s \leftarrow z + \alpha(z - x^{(n+1)})$ (gespiegelter Punkt).
2. Falls $f(x^{(1)}) \leq f(s) \leq f(z)$: **(Spiegelung, reflection)**
   Setze $x^{(n+1)} \leftarrow s$, sortiere neu, GOTO 1.
3. Falls $f(s) < f(x^{(1)})$: **(Streckung, expansion)**
   Teste $s' \leftarrow z + \beta(s - z)$. Gilt $f(s') < f(s)$, setze $s \leftarrow s'$.
   Setze $x^{(n+1)} \leftarrow s$, sortiere neu, GOTO 1.
4. Falls $f(x^{(n)}) < f(s)$: **(Schrumpfung, contraction)**
   Teste $s' \leftarrow \begin{cases} z + \gamma(x^{(n+1)} - z) & \text{falls } f(s) \geq f(x^{(n+1)}), \\ z + \gamma(s - z) & \text{falls } f(s) < f(x^{(n+1)}). \end{cases}$
   Gilt $f(s') < \min\{f(x^{(n+1)}), f(s)\}$, setze $x^{(n+1)} \leftarrow s'$,
   sonst schrumpfe den Simplex hin zu $x^{(1)}$:
   $x^{(i)} \leftarrow \frac{1}{2}(x^{(1)} + x^{(i)})$, $(i = 2, \ldots, n+1)$.
   Sortiere neu, GOTO 1.



Abbruchkriterien?

Vorschläge zu Abbruchkriterien:

1. Nelder und Mead: Setze $\bar{f} := \frac{1}{n+1} \sum_{i=1}^{n+1} f(x^{(i)})$
   Falls $\frac{1}{n+1} \sum_{i=1}^{n+1} [f(x^{(i)}) - \bar{f}]^2 \leq \varepsilon$ STOP (f auf Simplex fast konstant?)
2. Powell: Starte alle k Iteration erneut mit regelmäßigem Simplex um $x^{(1)}$ und Seitenlänge $\|x^{(1)} - x^{(n+1)}\|$ (sonst zu lang und dünn).
   Stoppe, falls keine Verbesserung in $2n$ Iterationen.

Bemerkungen:

- sehr einfach zu implementieren
- benötigt sehr viele Auswertungen, schon $n+1$ um zu starten $\rightarrow$ n klein
- Konvergenz ist nicht garantiert, es gibt streng konvexe Gegenbeispiele!
- sehr skalierungsabhängig!
- Erweiterung auf Nebenbedingungen: Setze $f \gg$ für unzulässige Punkte.
- Torczon: Konvergente Variante für konvexes f.

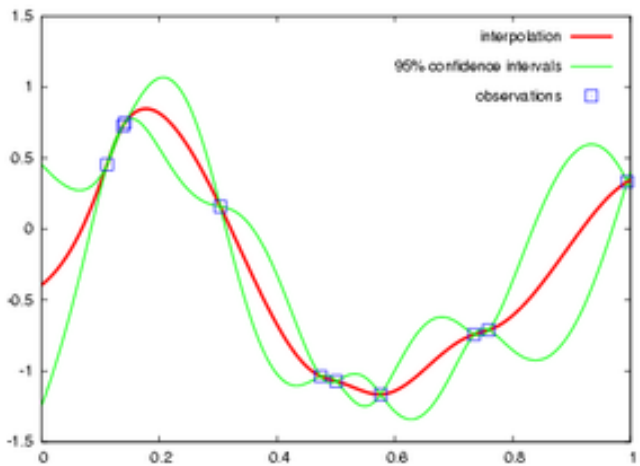
Inhalt

9.2 Das Kriging-Verfahren

Krige war Montaningenieur in Südafrika. Um mit möglichst wenigen Bohrungen Bodenschätze zu finden, orientierte er sich an statistischen Modellen $\rightarrow$ wird gerne verwendet, wenn jede Funktionsauswertung einem realen Experiment oder einer aufwendigen Simulation entspricht.

Idee: Beachte folgende Leitlinien bei Entwurf von Modell und Verfahren:

- Das Modell sollte die Funktion in bekannten Punkten interpolieren.
- Oszillationen sollten im Modell vermieden werden.
- In der Nähe bekannter Punkte sind Modellpunkte vertrauenswürdiger.
- Optimierungsaufwand ist vernachlässigbar gegenüber Auswertungen.



(22.1.2010, http://en.wikipedia.org/wiki/Kriging)

Das Modell der Funktion:

Gegeben seien k Punkte $x^{(1)}, \ldots, x^{(k)} \in \mathbb{R}^n$ mit Werten $f_1, \ldots, f_k \in \mathbb{R}$.
Für von allen $x^{(i)}$ weit entfernte Punkte wird angenommen:

Der Durchschnittswert $\mu = \frac{1}{k} \sum_{i=1}^{k} f_i$ ist guter Schätzer.

Für Punkte nahe an $x^{(i)}$ sollte der Einfluss von f_i groß sein.

$\rightarrow$ Für distanzabhängige Gewichtsfunktionen $b_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $b_i(x) := e^{-\|x - x^{(i)}\|^2}$
bestimme Koeffizienten $\alpha_i \in \mathbb{R}$ so, dass das Modell

$$\hat{f}(x) := \mu + \sum_{i=1}^{k} \alpha_i b_i(x)$$

die Funktionswerte in den $x^{(i)}$ annimmt, $\hat{f}(x^{(i)}) = f_i$ $(i = 1, \ldots, k)$.

$$\rightarrow \text{Löse} \quad \begin{bmatrix} b_1(x^{(1)}) & \ldots & b_k(x^{(1)}) \\ \vdots & \ddots & \vdots \\ b_1(x^{(k)}) & \ldots & b_k(x^{(k)}) \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_k \end{bmatrix} = \begin{bmatrix} f_1 - \mu \\ \vdots \\ f_k - \mu \end{bmatrix}.$$

Aber: Die Koordinaten der $x^{(i)}$ gehören oft zu unterschiedlichen physikalischen Größen, die Euklidische Norm $\|x - x^{(i)}\|$ macht wenig Sinn.

$\rightarrow$ **Skalierung:** Bestimme eine Norm $\|d\|_A^2 := d^T A d$ mit $A \succ 0$ wie folgt:
$\|\cdot\|_A$ ist gut, wenn für jedes $j \in \{1, \ldots, k\}$ das Modell der anderen Punkte

$$\hat{f}_A^j(x) := \mu + \sum_{i=1}^{j-1} \alpha_i^{A,j} b_i^A(x) + \sum_{i=j+1}^{k} \alpha_i^{A,j} b_i^A(x)$$

in $x^{(j)}$ möglichst kleinen Fehler $e_A^j := \hat{f}_A^j(x^{(j)}) - f_j$ hat.

$\rightarrow$ Löse $\min_{A \succ 0} \max_{1 \le j \le k} e_A^j$ (ein nichtglattes Problem) $\rightarrow A_*$, b_i^*, $\hat{f}_*$

Man hofft nun, dass die Fehler $e^j_{A_*}$ auch gute **Fehlerschätzer** für die Abweichung des Modells $\hat{f}_*$ von der Funktion f liefern:

Für j zeigt das Modell $\hat{f}^j_{A^*}$ über die Distanz $d^*_j := \min_{1\le i\le k, i\neq j} \|x^{(j)} - x^{(i)}\|_{A_*}$ einen Fehler von $e^j_{A_*}$. Pro Distanzeinheit von $\|\cdot\|_{A_*}$ weicht damit $\hat{f}_*$ von f hoffentlich nur um

$$\bar{s} := \max_{1\le j\le k} \frac{e^j_{A_*}}{d^*_j},$$

ab, vielleicht noch multipliziert mit einer Konstanten $\nu > 1$, also

hoffentlich $f(x) \ge \hat{f}_*(x) - \nu\bar{s} \min_{0\le i\le k} \|x - x^{(i)}\|_{A_*} =: g(x)$ für $x \in \Omega$,

wobei die Grundmenge $\Omega \subset \mathbb{R}^n$ als kompakt angenommen wird.

Das Optimierungsproblem des Kriging-Verfahrens:

Finde ein globales Minimum zu $\min_{x\in\Omega} g(x)$.

Eine Optimallösung dieses Problems wird als nächster Auswertungspunkt gewählt. Dann wird für die nun $k+1$ Punkte das Modell erneut berechnet, etc.

Inhalt

9.3 NEWUOA (NEW Unconstrained Opt. Alg.)

Interpoliert die „letzten“ $2n+1$ Funktionswerte von $f : \mathbb{R}^n \to \mathbb{R}$ durch ein quadratisches Modell, das sich möglichst wenig vom Vorgänger-Modell unterscheidet (wie bei BFGS), und bestimmt den nächsten Punkt als Näherungslösung eines Trustregion-Problems zu diesem Modell.

Gegeben: Startpunkt $x^{(0)}$ und Intervall $[\underline{\Delta}, \bar{\Delta}]$ für den Trustregion-Radius, wobei $\underline{\Delta}$ auch die Abbruchgenauigkeit für die Lösungskoordinaten sei.

In Iteration k bestimmt NEWUOA ein quadratisches Modell

$$m^{(k)}(x) = \tfrac{1}{2} x^T Q^{(k)} x + (q^{(k)})^T x + c^{(k)},$$

das in Punkten $x^{(i)} \in \mathbb{R}^n$, $(i = 0, \ldots, 2n)$ die Werte f_i annimmt und, für $k > 0$, $\|\nabla^2 m^{(k-1)} - \nabla^2 m^{(k)}\|$ minimiert. Dazu löst es

$$(QP_m^k) \qquad \begin{array}{ll} \min & \sum_{1 \le i,j \le n} (Q_{ij}^{(k-1)} - Q_{ij}^{(k)})^2 \\ \text{s.t.} & \frac{1}{2}(x^{(i)})^T Q^{(k)} x^{(i)} + (q^{(k)})^T x^{(i)} + c^{(k)} = f_i, \quad i = 0, \ldots, 2n, \\ & Q^{(k)} \in S^n, q^{(k)} \in \mathbb{R}^n, c^{(k)} \in \mathbb{R}. \end{array}$$

Pro Iteration ändert sich nur ein $x^{(i)}$, daher lässt sich (QP_m^k) mit vielen Tricks in Konstante mal $(2n+1)^2$ Fließkomma-Operationen lösen.

Die Initialisierung von $m^{(0)}$ verwendet die Punkte $x^{(0)}$, $x^{(2i-1)} := x^{(0)} - \bar{\Delta} e_i$, $x^{(2i)} := x^{(0)} + \bar{\Delta} e_i$ $(i = 1, \ldots, n) \to c^{(0)} = f_0$, $q^{(0)}$, Diagonalmatrix $Q^{(0)}$.

Grober Ablauf von NEWUOA:

0. Initialisierung: Wähle $x^{(0)}$, $[\underline{\Delta}, \bar{\Delta}]$, $\delta \leftarrow \bar{\Delta}$, $\Delta \leftarrow \bar{\Delta}$
   Bestimme $x^{(i)}$, f_i $(i = 1, \ldots, 2n)$ und $m^{(0)}$, setze $k = 0$.
1. Finde $\hat{\imath} \in \{0, \ldots, 2n\}$ mit $f_{\hat{\imath}} \leq f_i$ $(\forall i)$ und löse das Trustregionproblem $\min_{\|d\| \leq \Delta} m^{(k)}(x^{(\hat{\imath})} + d)$ (mit inexaktem konjugierten Gradientenverf.)
2. Falls $\|d\| \leq \frac{1}{2}\rho$: [Modellumgebung verkleinern?]
   Ist zum 3. Mal $|f(x^{(\hat{\imath})} + d) - m^{(k)}(x^{(\hat{\imath})} + d)|$ klein, dann
   (a) falls $\delta = \underline{\Delta}$, STOP,
   (b) sonst setze $\delta \leftarrow \max\{\underline{\Delta}, \delta/10\}$, $\Delta \leftarrow \delta$, GOTO 1.
   Setze $\Delta \leftarrow \max\{\delta, \Delta/10\}$,
   finde Ersatzpunkt für j mit $\|x^{(j)} - x^{(\hat{\imath})}\|$ maximal und GOTO 4.
3. Für $\|d\| \geq \frac{1}{2}\delta$ berechne $\rho = \frac{f_{\hat{\imath}} - f(x^{(\hat{\imath})} + d)}{f_{\hat{\imath}} - m^{(k)}(x^{(\hat{\imath})} + d)}$, passe Δ $(\geq \delta)$ entsprechend an und setze $x^{(j)} \leftarrow x^{(\hat{\imath})} + d$ für geeignet gewähltes j.
4. Bestimme $m^{(k+1)}$ aus (QP_m^{k+1}) mit neuem $x^{(j)}$, $k \leftarrow k + 1$, GOTO 1.

Bemerkungen:

- Der adaptive Parameter δ wird nur verkleinert, wenn Schrittweiten $\geq \delta$ keine Verbesserung mehr zu bringen scheinen.
- Jede Wahl neuer Punkte $x^{(i)}$ soll etwa Abstand δ haben.
- Viele weitere Details für Effizienz, Numerik, etc. ...
- Bester ableitungsfreier Code in Benchmark von Moré und Wild, 2009.