1. Projekt

Ableitungsfreie Methoden

im Fach

Numerische Optimierung

Mai 2020

Maximilian Gaul

Aufgabe 1

Teilintervalle und Funktionswerte des Bisektionsverfahrens aus `Bisektion.m` für das Minimum von

$$h(x) = e^{-x} + 0.5x^2$$

mit dem Startintervall [0, 1] nach 10 Schritten:

Schritt	Intervall	Funktionswert
1	[0.00, 1.00]	h(0.50)=0.7315
2	[0.50, 1.00]	h(0.75)=0.7536
3	[0.50, 0.75]	h(0.63)=0.7306
4	[0.50, 0.63]	h(0.56)=0.7280
5	[0.56, 0.63]	h(0.59)=0.7285
6	[0.56, 0.59]	h(0.58)=0.7281
7	[0.56, 0.58]	h(0.57)=0.7280
8	[0.56, 0.57]	h(0.57)=0.7280
9	[0.57, 0.57]	h(0.57)=0.7280
10	[0.57, 0.57]	h(0.57)=0.7280

Abbildung 1: Die ersten 10 Schritte des Bisektionsverfahrens mit Startintervall [0, 1] für $h(x)$

Die Intervallgrenzen und Funktionsauswertungen sind wie folgt verteilt:

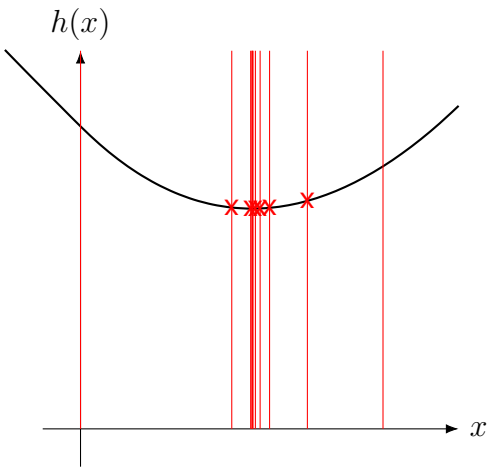

Abbildung 2: $h(x)$ mit Intervallgrenzen und Auswertungspunkten des Bisektionsverfahrens

Aufgabe 2

Siehe `Mutation.m`.

Aufgabe 3

Nachfolgende Erläuterungen beziehen sich auf

$$f(x_1, x_2) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2 \text{ mit Startwert } f_0 = \begin{bmatrix} 2 \\ 4 \end{bmatrix}$$

$$g(x_1, x_2) = 100 \cdot (x_1 - 2)^4 + (x_1 - 2x_2)^2 \text{ mit Startwert } g_0 = \begin{bmatrix} 4 \\ 2 \end{bmatrix}$$

Eine Möglichkeit für ein Abbruchkriterium ist die Anzahl an Iterationen, d.h. der Algorithmus stoppt nach einer bestimmten Anzahl an Durchläufen und gibt den bis dahin berechneten Wert aus. Je nach Funktion und Zufallsvektoren kann die Genauigkeit des Ergebnisses stark schwanken. Während für eine festgelegte obere Grenze an Durchläufen für eine simple Funktion ein Minimum bereits ausreichend genau bestimmt werden kann, benötigt eine komplexere Funktion ggf. weitaus mehr Durchläufe. Die Anzahl an Iterationen ist also kein adaptives Kriterium.

Weiterhin könnte auch der Betrag aus der Differenz des aktuellen Funktionswertes und des letzten Funktionswertes als Abbruchkriterium funktionieren. Dabei aktualisiert man die zu überprüfenden Werte nur, sofern eine Verbesserung eingetreten ist (ansonsten wären letzter und aktueller Funktionswert eventuell gleich). Ist der Differenzbetrag kleiner als eine festgelegte Toleranz, endet der Algorithmus. Typische Werte sind hier z.B. 10^{-4} oder 10^{-8}.

Kriterium	**Funktion**	**Genauigkeit**	**Anzahl Schritte**
Differenzbetrag	f	10^{-6}	4.924.700
Differenzbetrag	f	10^{-6}	282.179
Differenzbetrag	f	10^{-6}	24.907
Differenzbetrag	g	10^{-6}	36.655
Differenzbetrag	g	10^{-6}	9.758
Differenzbetrag	g	10^{-6}	882

Abbildung 3: Einfluss des Zufalls auf die Anzahl an Durchläufen für eine Genauigkeit von 10^{-6} bei festem α

Durch die Wahl des Differenzbetrages kann man vom Zufall profitieren und mit verhältnismäßig wenigen Schritten zu einer guten Genauigkeit kommen. Wählt man dagegen die Anzahl der Iterationen, muss man ggf. neu rechnen wenn das Ergebnis noch zu ungenau ist:

Abbruchkriterium	Funktion	Genauigkeit	Anzahl Schritte
Anzahl Iterationen	f	$1 \cdot 10^{-5}$	225.000
Anzahl Iterationen	f	$7 \cdot 10^{-6}$	225.000
Anzahl Iterationen	f	$2 \cdot 10^{-8}$	225.000
Anzahl Iterationen	g	$2 \cdot 10^{-6}$	15.000
Anzahl Iterationen	g	$6 \cdot 10^{-7}$	15.000
Anzahl Iterationen	g	$1 \cdot 10^{-8}$	15.000

Abbildung 4: Einfluss des Zufalls auf die Genauigkeit bei einer gegebenen Anzahl an Durchläufen bei festem α

Der Parameter α bestimmt den Anteil des Zufallsvektors für den Wert von $\hat{x}$. Kleine Werte von α bedeuten eine höhere Wahrscheinlichkeit, dass sich der Funktionswert verbessert (die Verbesserung fällt aber möglicherweise kleiner aus) da sich besonders in der Nähe eines Minimums, bei quadratischen oder Funktionen 4. Grades der Funktionswert stark ändern und man leicht über das Ziel hinaus schießen kann wenn man zu große Schritte geht (ähnlich wie bei einem Abstiegsverfahren). Falls die Ableitung der Funktion zur Verfügung steht (tut sie hier wahrscheinlich nicht da das Projekt *Ableitungsfreie Methoden* heißt), könnte man ein Verfahren zur Schrittweitensteuerung implementieren (z.B. nach Armijo $\varphi(\alpha) := f(x + \alpha r)$).

Abbruchkriterium	Funktion	Genauigkeit	$\overline{\text{Anzahl Schritte}}_{200}$	α
Differenzbetrag	f	10^{-4}	238.992	1
Differenzbetrag	f	10^{-4}	190.980	0.5
Differenzbetrag	f	10^{-4}	19.557	0.25
Differenzbetrag	f	10^{-4}	9.025	0.125
Differenzbetrag	g	10^{-4}	41.688	1
Differenzbetrag	g	10^{-4}	1.878	0.5
Differenzbetrag	g	10^{-4}	1.101	0.25
Differenzbetrag	g	10^{-4}	383	0.125

Abbildung 5: Einfluss von α auf den Durchschnitt von 200 Durchläufen für eine Genauigkeit von 10^{-4}

Zu Beginn von `Mutation-Selektion` sind die Sprünge der Funktionswerte relativ groß, werden dann aber immer kleiner:

Funktion	x	Funktionswert
f	[2.10 3.82]	101.61
f	[1.85 3.60]	76.25
f	[2.11 3.55]	68.41
f	[2.29 3.37]	50.06
f	[2.02 3.03]	32.96
f	[2.39 2.68]	13.39
f	[2.41 2.24]	9.01
f	[2.42 2.39]	8.99
f	[2.71 2.37]	3.44
f	[2.81 2.32]	2.03
f	[3.20 1.88]	1.35
f	[3.09 2.12]	0.80
f	[2.98 1.79]	0.77

Abbildung 6: Die ersten Schritte eines Aufrufes von `Mutation-Selektion` mit $\alpha = 1$

Eine Strategie könnte daher sein, α erst klein zu wählen (z.B. 0.125) und dann mit jeder Iteration zu vergrößern dabei aber nicht über eine obere Schranke zu kommen (z.B. 0.5). Dies geschieht unter der Annahme, das der Algorithmus zuerst die richtige Richtung finden muss, durch zu große Sprünge werden diese im ungünstigen Fall verpasst d.h. man geht zu weit in eine Richtung.

Funktion	Genauigkeit	$\overline{\text{Anzahl Schritte}}_{600}$	α	Verstärkungsfaktor
f	10^{-4}	5.062	0.1250	1.1000
f	10^{-4}	3.660	0.1250	1.0500
f	10^{-4}	3.850	0.1250	1.0250
f	10^{-4}	6.639	0.1250	1.0125
g	10^{-4}	438	0.1250	1.1000
g	10^{-4}	386	0.1250	1.0500
g	10^{-4}	368	0.1250	1.0250
g	10^{-4}	771	0.1250	1.0125

Abbildung 7: Einfluss von α und einem Verstärkungsfaktor auf den Durchschnitt an Iterationen für eine Genauigkeit von 10^{-4}

Man sieht, dass durch einen Verstärkungsfaktor zwischen in [1.025, 1.05] die Anzahl an Iterationen reduziert werden konnte. Wird der Faktor zu klein gewählt (d.h. α bleibt über einen großen Teil des Durchlaufs relativ klein) verlängert sich die Rechenzeit wieder.

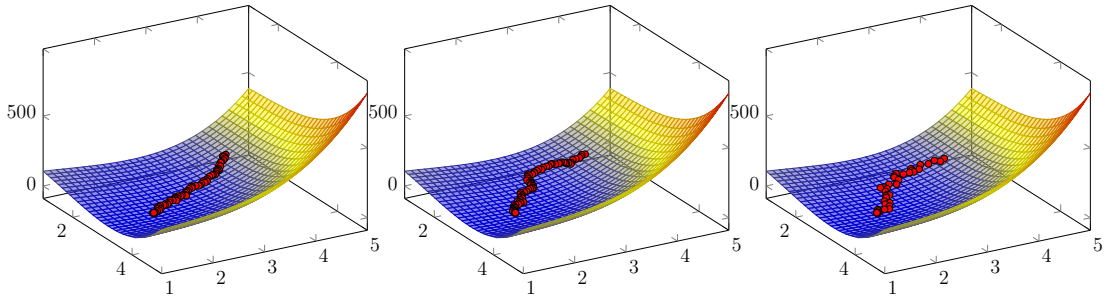

Abbildung 8: $\alpha = 0.125$ mit Verstärkungsfaktor 1.05

Abbildung 9: $\alpha = 0.125$

Abbildung 10: $\alpha = 0.25$

Aufgabe 4

Vergleich der Rechenzeit für die Funktionen f, g, h mit dem Bisektions- und Mutation-Selektion-Verfahren sowie `fminsearch`. Da das Bisektionsverfahren nur im eindimensionalen Fall funktioniert wird es nur auf h angewendet.

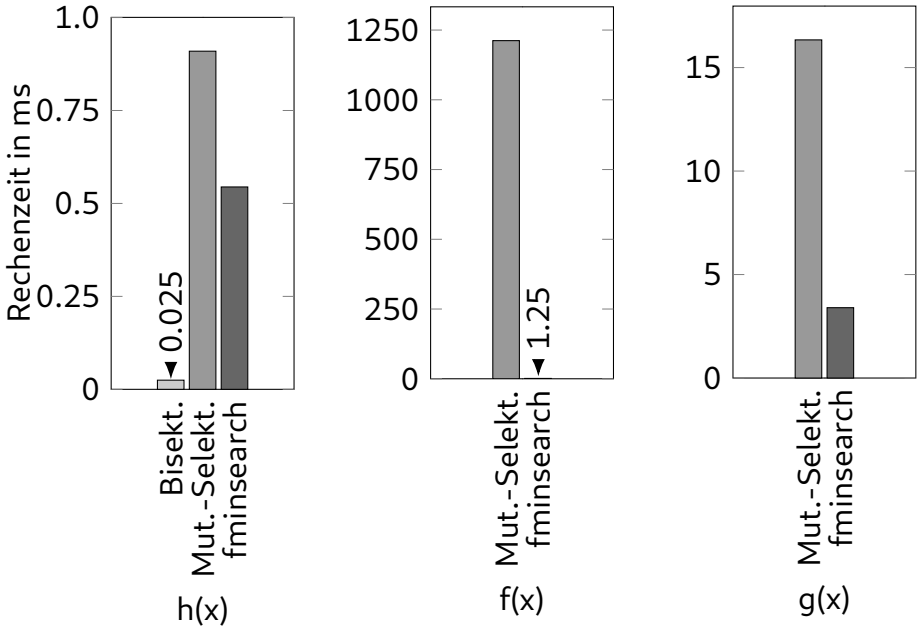

Abbildung 11: Vergleich der Rechenzeit für eine Genauigkeit von 10^{-6} (Durchschnitt aus 1000 Durchläufen)

Aufgabe 5

Die Bedingung der `while`-Schleife bezüglich der Anzahl an Iterationen bleibt erhalten. Direkt danach wird nun aber die Standardabweichung der Funktionswerte an den Ecken berechnet und mit der Toleranz `options.TolX` verglichen (siehe `fminsearch2.m` Zeile 328ff). Dieses Abbruchkriterium ist ungeeignet für Fuktionen, die über einen größeren Wertebereich sehr ähnliche Funktionswerte haben (d.h. sehr flach sind). Daraus folgt, dass der Mittelwert $\bar{f}$ sehr ähnlich

zu den einzelnen Funktionswerten ist, die Standardabweichung ist also niedrig obwohl die Eckpunkte des Simplex weit auseinander liegen können.

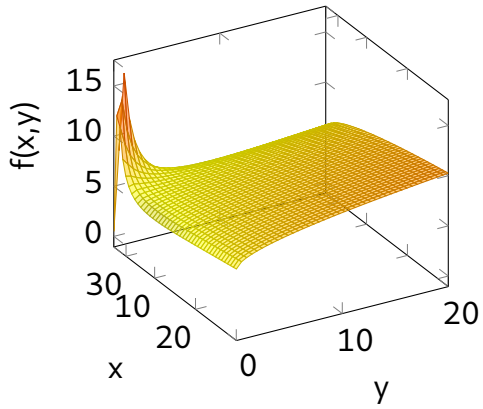

Abbildung 12: Beispiel für eine Funktion, bei der das Standardabweichungs-Kriterium problematisch ist: $f(x,y) = (\sqrt{x} + \sqrt{y}) + 15 \cdot e^{-\sqrt{\ln(x+y)^2}}$

Die Parameter des Nelder-Mead-Verfahrens sind auf $\rho = 1$, $\chi = 2$, $\psi = 0.5$ und $\sigma = 0.5$ in Zeile 204 initialisiert.

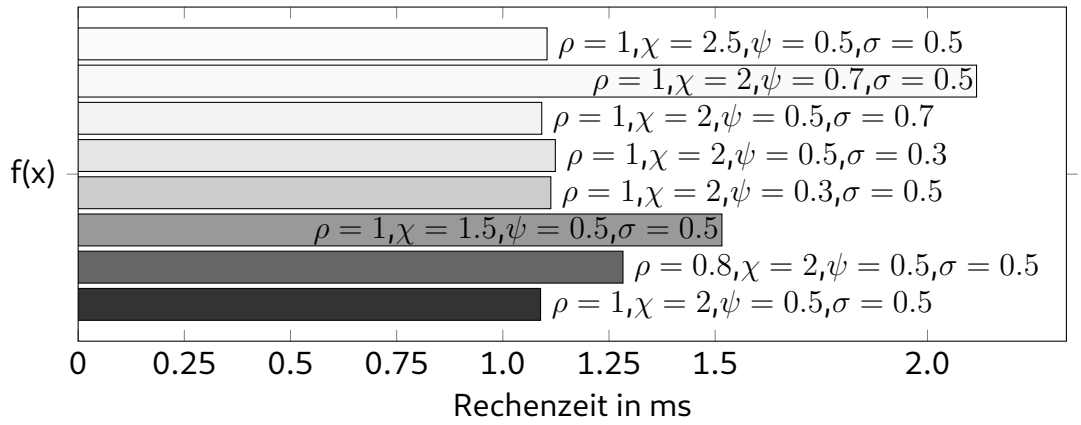

Abbildung 13: Einfluss der Parameter auf die Rechenzeit (Durchschnitt aus 1000 Durchläufen, Genauigkeit 10^{-6})

Die Parameter sind bereits sehr gut gewählt. Eine Änderung einzelner Parameter scheint unterschiedliche Auswirkungen auf die Laufzeit zu haben. Eine Erhöhung von ψ verdoppelte die Rechenzeit gegenüber den ursprünglichen Werten. Besonders schlecht schnitt auch eine Verringerung von χ ab. Die Änderung anderer Parameter hatte keinen besonders großen Einfluss.

Wenn `options.Display='iter'` gesetzt ist, gibt `fminsearch2` nun auch die Eckpunkte des Simplex aus.

Aufgabe 6

Berechnet werden die ersten vier Iterationen des Nelder-Mead-Algorithmus von

$$g(x_1, x_2) = 100 \cdot (x_2 - 2)^4 + (x_1 - 2x_2)^2$$

mit den Parametern $n = 2$, $\alpha = \frac{1}{2}$, $\beta = 2$ und $\gamma = 1$.

$$x^{(0,0)} = \begin{bmatrix} 4 \\ 2 \end{bmatrix}, e_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, e_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Damit erhält man die Punkte $x^{(0,1)} = \begin{bmatrix} 5 \\ 2 \end{bmatrix}, x^{(0,2)} = \begin{bmatrix} 4 \\ 3 \end{bmatrix}$ und den Startsimplex

$$S_0 = (\begin{bmatrix} 4 \\ 2 \end{bmatrix}, \begin{bmatrix} 5 \\ 2 \end{bmatrix}, \begin{bmatrix} 4 \\ 3 \end{bmatrix})$$

k = 0

$$\max\{f(x^{(0,0)}) = 1600, f(x^{(0,1)}) = 8101, f(x^{(0,2)}) = 1604\} = f(x^{(0,1)})$$

$$s_0 = \frac{1}{2} \cdot (\begin{bmatrix} 4 \\ 2 \end{bmatrix} + \begin{bmatrix} 4 \\ 3 \end{bmatrix}) = \begin{bmatrix} 4 \\ \frac{5}{2} \end{bmatrix} \text{ und } x_0 = x^{(0,1)} = \begin{bmatrix} 5 \\ 2 \end{bmatrix}$$

- Reflexion: $\hat{x}_0 = \begin{bmatrix} 4 \\ \frac{5}{2} \end{bmatrix} + 1 \cdot (\begin{bmatrix} 4 \\ \frac{5}{2} \end{bmatrix} - \begin{bmatrix} 5 \\ 2 \end{bmatrix}) = \begin{bmatrix} 3 \\ 3 \end{bmatrix}$ mit $f(\hat{x}_0) = 25$
- Expansion: $\hat{x}_0^* = \begin{bmatrix} 4 \\ \frac{5}{2} \end{bmatrix} + 2 \cdot (\begin{bmatrix} 3 \\ 3 \end{bmatrix} - \begin{bmatrix} 4 \\ \frac{5}{2} \end{bmatrix}) = \begin{bmatrix} 2 \\ \frac{7}{2} \end{bmatrix}$ mit $f(\hat{x}_0^*) = 25$

Nach dem 1. Schritt erhält man den Simplex

$$S_1 = (\begin{bmatrix} 4 \\ 2 \end{bmatrix}, \begin{bmatrix} 2 \\ \frac{7}{2} \end{bmatrix}, \begin{bmatrix} 4 \\ 3 \end{bmatrix})$$

k = 1

$$\max\{f(x^{(1,0)}) = 1600, f(x^{(1,1)}) = 25, f(x^{(1,2)}) = 1604\} = f(x^{(1,2)})$$

$$s_1 = \frac{1}{2} \cdot (\begin{bmatrix} 4 \\ 2 \end{bmatrix} + \begin{bmatrix} 2 \\ \frac{7}{2} \end{bmatrix}) = \begin{bmatrix} 3 \\ \frac{11}{4} \end{bmatrix} \text{ und } x_1 = x^{(1,2)} = \begin{bmatrix} 4 \\ 3 \end{bmatrix}$$

- Reflexion: $\hat{x}_1 = \begin{bmatrix} 3 \\ \frac{11}{4} \end{bmatrix} + 1 \cdot (\begin{bmatrix} 3 \\ \frac{11}{4} \end{bmatrix} - \begin{bmatrix} 4 \\ 3 \end{bmatrix}) = \begin{bmatrix} 2 \\ \frac{5}{2} \end{bmatrix}$ mit $f(\hat{x}_1) = 9$
- Expansion: $\hat{x}_1^* = \begin{bmatrix} 3 \\ \frac{11}{4} \end{bmatrix} + 2 \cdot (\begin{bmatrix} 2 \\ \frac{5}{2} \end{bmatrix} - \begin{bmatrix} 3 \\ \frac{11}{4} \end{bmatrix}) = \begin{bmatrix} 1 \\ \frac{9}{4} \end{bmatrix}$ mit $f(\hat{x}_1^*) = 112.25$

Nach dem 2. Schritt erhält man den Simplex

$$S_2 = (\begin{bmatrix}4\\2\end{bmatrix}, \begin{bmatrix}2\\\frac{7}{2}\end{bmatrix}, \begin{bmatrix}2\\\frac{5}{2}\end{bmatrix})$$

k = 2

$$\max\{f(x^{(2,0)}) = 1600, f(x^{(2,1)}) = 25, f(x^{(2,2)}) = 9\} = f(x^{(2,0)})$$

$$s_2 = \frac{1}{2} \cdot (\begin{bmatrix}2\\\frac{7}{2}\end{bmatrix} + \begin{bmatrix}2\\\frac{5}{2}\end{bmatrix}) = \begin{bmatrix}2\\3\end{bmatrix} \text{ und } x_2 = x^{(2,0)} = \begin{bmatrix}4\\2\end{bmatrix}$$

- Reflexion: $\hat{x}_2 = \begin{bmatrix}2\\3\end{bmatrix} + 1 \cdot (\begin{bmatrix}2\\3\end{bmatrix} - \begin{bmatrix}4\\2\end{bmatrix}) = \begin{bmatrix}0\\4\end{bmatrix}$ mit $f(\hat{x}_2) = 1664$
- Innere Kontraktion: $\hat{x}_2^* = \begin{bmatrix}2\\3\end{bmatrix} + \frac{1}{2} \cdot (\begin{bmatrix}4\\2\end{bmatrix} - \begin{bmatrix}2\\3\end{bmatrix}) = \begin{bmatrix}3\\\frac{5}{2}\end{bmatrix}$ mit $f(\hat{x}_2^*) = 104$

Nach dem 3. Schritt erhält man den Simplex

$$S_3 = (\begin{bmatrix}3\\\frac{5}{2}\end{bmatrix}, \begin{bmatrix}2\\\frac{7}{2}\end{bmatrix}, \begin{bmatrix}2\\\frac{5}{2}\end{bmatrix})$$

k = 3

$$\max\{f(x^{(3,0)}) = 104, f(x^{(3,1)}) = 25, f(x^{(3,2)}) = 9\} = f(x^{(3,0)})$$

$$s_3 = \frac{1}{2}(\begin{bmatrix}2\\\frac{7}{2}\end{bmatrix} + \begin{bmatrix}2\\\frac{5}{2}\end{bmatrix}) = \begin{bmatrix}2\\3\end{bmatrix} \text{ und } x_3 = x^{(3,0)} = \begin{bmatrix}3\\\frac{5}{2}\end{bmatrix}$$

- Reflexion: $\hat{x}_3 = \begin{bmatrix}2\\3\end{bmatrix} + 1 \cdot (\begin{bmatrix}2\\3\end{bmatrix} - \begin{bmatrix}3\\\frac{5}{2}\end{bmatrix}) = \begin{bmatrix}1\\\frac{7}{2}\end{bmatrix}$ mit $f(\hat{x}_3) = 136$
- Innere Kontraktion: $\hat{x}_3^* = \begin{bmatrix}2\\3\end{bmatrix} + \frac{1}{2} \cdot (\begin{bmatrix}3\\\frac{5}{2}\end{bmatrix} - \begin{bmatrix}2\\3\end{bmatrix}) = \begin{bmatrix}\frac{5}{2}\\\frac{11}{4}\end{bmatrix}$ mit $f(\hat{x}_3^*) = 15.25$

Nach dem 4. Schritt erhält man den Simplex

$$S_4 = (\begin{bmatrix}\frac{5}{2}\\\frac{11}{4}\end{bmatrix}, \begin{bmatrix}2\\\frac{7}{2}\end{bmatrix}, \begin{bmatrix}2\\\frac{5}{2}\end{bmatrix})$$

Aufgabe 7

Wie aus Abbildung 3 und Abbildung 11 hervorgeht, ist `Mutation-Selektion` durchschnittlich wesentlich langsamer als `fminsearch` und auch sehr unzuverlässig was die Anzahl der Iterationen angeht, die es zum Finden einer passablen Lösung benötigt. Schuld daran ist natürlich der Zufall, der mit jedem Durchlauf unterschiedliche Richtungsvektoren ausgibt. Wenn man ganz großes Pech hat, erhält man nie eine Abstiegsrichtung und der Algorithmus konvergiert nicht. Deswegen sollte `Mutation-Selektion` mit einer oberen Schranke an möglichen Iterationen versehen werden.

Durch einen anderen Zufallsalgorithmus könnte das Verfahren aber zuverlässiger gemacht werden. In der Computergrafik z.B. werden zum Erstellen von realistischen Bildern Strahlen berechnet die eine zufällige Startrichtung erhalten und in der virtuellen Szene an Objekten abprallen und so die Beleuchtung simulieren (Monte-Carlo Integration). Es konnte gezeigt werden, das durch sog. *Quasirandom number generators* die Bildqualität gesteigert wird da die generierten Zahlen die verfügbare Fläche $[0, 1]^2$ gleichmäßiger ausfüllen und man somit verhindert, dass Zufallszahlen doppelt generiert werden.

Auf `Mutation-Selektion` übertragen bedeutet das, dass die Menge an verfügbaren Richtungen gleichmäßiger abgetastet wird.

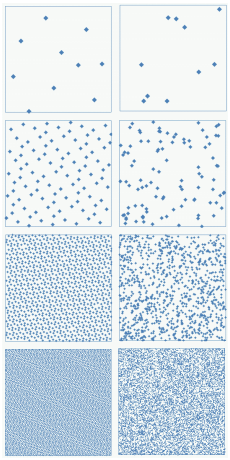

Abbildung 14: Links: *additive recurrent sequence*, Rechts: Zufallszahlen. Aus: `https://en.wikipedia.org/wiki/Low-discrepancy_sequence`

Wie in der Vorlesung besprochen gibt es keinen generellen Konvergenzbeweis für das Nelder-Mead-Verfahren aber es ist wesentlich zuverlässiger in der Vorgehensweise und der Rechenaufwand ist deutlich geringer als im Vergleich zu `Mutation-Selektion`.