2.5 Zufallsgesteuerte Algorithmen

Nachdem wir gesehen haben, dass Erfahrungswerte hilfreich sein können, werden wir nun den *Zufall* nutzen, um den Ablauf eines Verfahrens zu steuern.

2.5.1 Metropolis-Algorithmus und Simulated Annealing

Viele technische Aufgaben laufen auf die Suche nach einem Minimum (oder gleichwertig Maximum) einer Funktion hinaus.

Diese Aufgabe haben wir bereits – wenn auch widerwillig – in der Schule kennen gelernt, allerdings war dort die Welt oft noch sehr einfach:

Wir hatten lineare Funktionen $y=a\cdot x+b$, Parabeln $y=a\cdot x^2+b\cdot x+c$, die Exponentialfunktion $y=e^x$, trigonometrische Funktionen wie $y=\sin(x)$ usw.

Oft war die Situation eine Ähnliche: Wir kannten die Funktion vollständig. Nullstellen ließen sich durch Anwenden fertiger Formeln berechnen; ebenso konnten wir Minima und Maxima durch Ableiten der Funktion ermitteln usw.

Wie aber finden wir das Minimum dieser Funktion, ausgehend vom markierten Startpunkt?

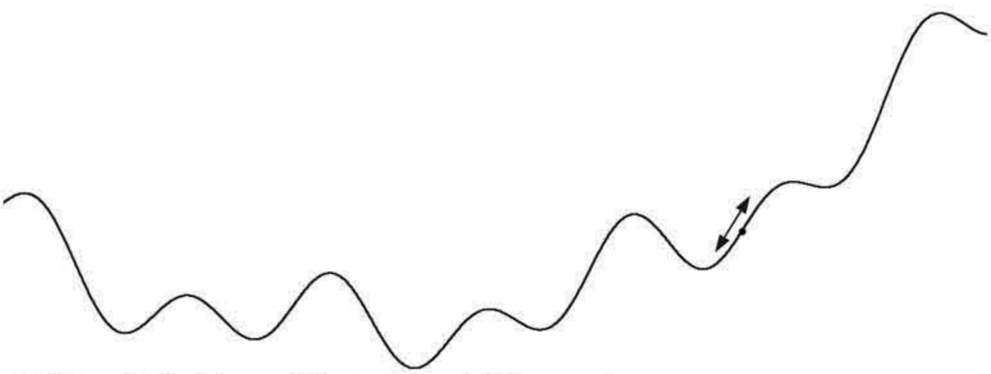

Abbildung 22: Funktion und Startpunkt zur Minimumsuche

Wie wir sehen, ist diese Aufgabe in praktischen Anwendungen leider nicht immer einfach, wir haben keine fertigen Formeln mehr für Nullstellen, evtl. können wir die Funktion auch nicht ohne Weiteres ableiten, weil wir sie gar nicht als Formel beschrieben haben, sondern lediglich als Menge von Messwerten usw. In diesen Fällen können wir beispielsweise das Minimum nicht mehr *berechnen*, wir müssen es *suchen*. Der Unterschied liegt darin, dass wir keine *globalen* Kenntnisse unserer Funktion mehr haben, sondern nur noch über *lokale* Informationen verfügen, einzelne Funktionswerte eben.

Lokale Minimumsuche mit der Methode des steilsten Abstiegs

Ein intuitiv nahe liegender Ansatz für die Minimumsuche ist das sog. Verfahren des steilsten Abstiegs:

Wenn wir ein Tal finden möchten, dann gehen wir bergab.

Das klingt banal, wer würde schon aufwärts klettern oder auf dem gleichen Höhenniveau bleiben, wenn ein Tal gesucht wird.

Mathematisch gesprochen bedeutet das, dass wir bei der Suche nach einem Tal in Richtung des negativen Funktionsgradienten laufen, diese Richtung markiert den steilsten Abstieg.

Im Ergebnis laufen wir zielstrebig ins nächste Tal:

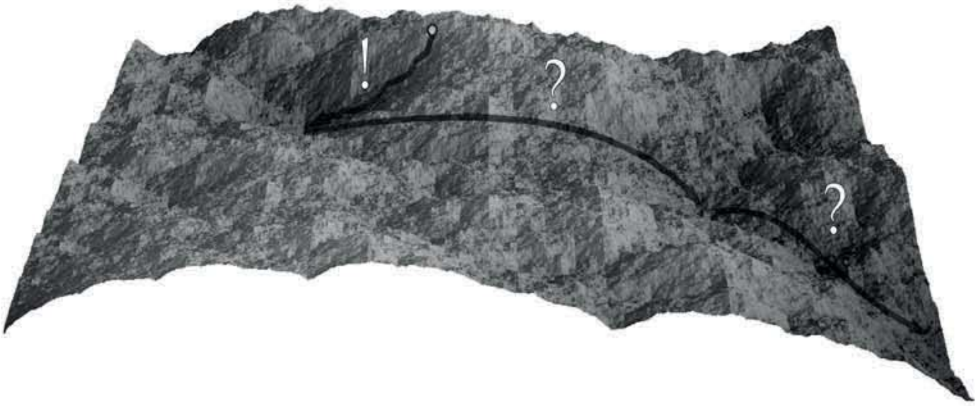

Abbildung 23: Methode des steilsten Abstiegs

Tatsächlich findet die Methode des steilsten Abstiegs zuverlässig dieses Tal, also ein *lokales* Minimum.

Die tiefer gelegenen Täler im rechten Ausschnitt bleiben allerdings ebenso unentdeckt wie das *globale* Minimum, wo auch immer dies liegen mag.

Kann ein zufallsgesteuertes Verfahren tatsächlich besser sein?

Wir sollten uns daran erinnern, dass wir hier ein *globales* Minimum einer Funktion suchen, aber nur *lokale* Kenntnisse haben. Offenbar kann diese Aufgabe also nicht zuverlässig lösbar sein. Wir möchten aber nun eine Strategie kennen lernen, wie wir dieser Herausforderung trotzdem begegnen können.

Der **Metropolis-Algorithmus** beschäftigt sich mit dem Abkühlverhalten (**Simulated Annealing**) von Körpern und wurde 1953 von **N. Metropolis** et al. vorgestellt [Metro53]. „Abkühlung" entspricht dabei dem Übergang in einen energie*ärmeren* Zustand.

Dies kommt unserer Aufgabe nahe und entspricht quasi der Suche nach dem Minimum der Energiefunktion. Interessanterweise hat sich dabei gezeigt, dass ein Körper im Zuge der Abkühlung scheinbar kurzzeitig einen energie*reicheren* Zustand einnehmen kann. Bildlich gesprochen bedeutet das, dass auf der Suche nach einem „Tal" auch „Hügel" überschritten werden.

S. Kirkpatrick et. al. haben 1983 die Idee des Simulated Annealing für kombinatorische Optimierungsaufgaben genutzt [Kirk83].

Das Verfahren zur Suche eines globalen Minimums simuliert diesen Abkühlprozess und geht dabei folgendermaßen vor:

Ausgehend von einem Startwert x_0 suchen wir iterativ immer neue x-Werte, indem wir vom jeweils letzten x-Wert einen Schritt in eine *zufällig gewählte* Richtung gehen.

Wie verhalten wir uns nun?

Wenn $f(x_{neu})<f(x_{alt})$, haben wir offenbar einen Fortschritt bei der Suche nach dem Minimum erzielt und wählen x_{neu} als neuen x-Wert.

Falls aber $f(x_{neu})\geq f(x_{alt})$, haben wir zwar keinen Fortschritt erzielt, trotzdem akzeptieren wir x_{neu} manchmal.

Auf diese Weise schaffen wir es, kleine „Hügel" in der Funktion zu überwinden. Natürlich dürfen wir nicht immer eine Verschlechterung hinnehmen. Wir treffen unsere Auswahl deswegen mit der **Metropolis-Wahrscheinlichkeit** $P_{Metropolis}(T)$:

$$P_{Metropolis}(T)=\begin{cases} 1 & ,\text{falls } f(x_{neu})<f(x_{alt}) \\ e^{-\frac{f(x_{neu})-f(x_{alt})}{T}} & ,\text{falls } f(x_{neu})\geq f(x_{alt}) \end{cases}$$

Formel 8: Metropolis-Wahrscheinlichkeit für die Auswahl des nächsten Lösungskandidaten

$T>0$ ist dabei ein Parameter für das Verfahren, wir wählen hier $T=0.1$.

Ein neuer x-Wert x_{neu} wird mit der Wahrscheinlichkeit $P_{Metropolis}(T)$ akzeptiert. Falls er nicht akzeptiert wird, verwerfen wir den Wert und wiederholen das Verfahren mit einem neuen Kandidaten für x_{neu} .

Nach 1000 Schritten mit Parameter $T=0.1$ haben wir einen Teil der Funktion erforscht. Das globale Minimum wurde leider noch nicht gefunden, bisher wurde nur ein lokales Minimum entdeckt und als bisher beste Lösung vermerkt.

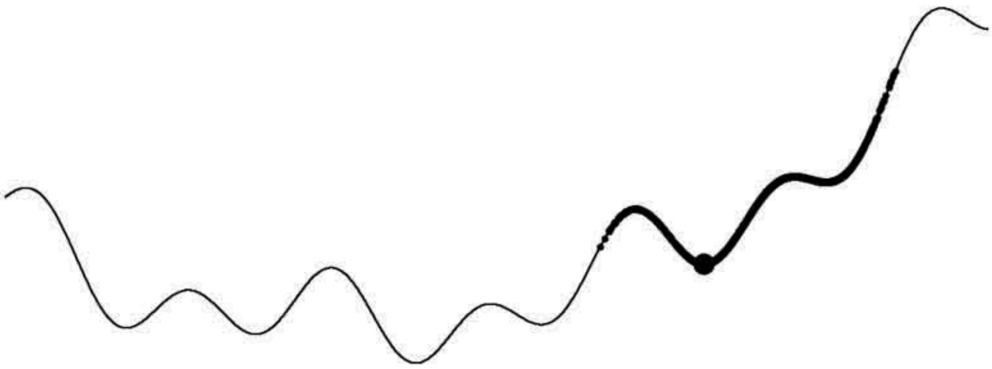

Abbildung 24: Metropolis-Algorithmus, Lokales Minimum nach 1000 Schritten

Nach 2000 Schritten haben wir es aber geschafft. Das globale Minimum ist gefunden, obwohl wir immer nur mit lokaler Information arbeiten konnten.

Der Erfolg kann selbstverständlich nicht garantiert werden – natürlich wäre es möglich, dass unsere Funktion jenseits dieses gedruckten Ausschnitts ein noch viel tiefer gelegenes Minimum hat.

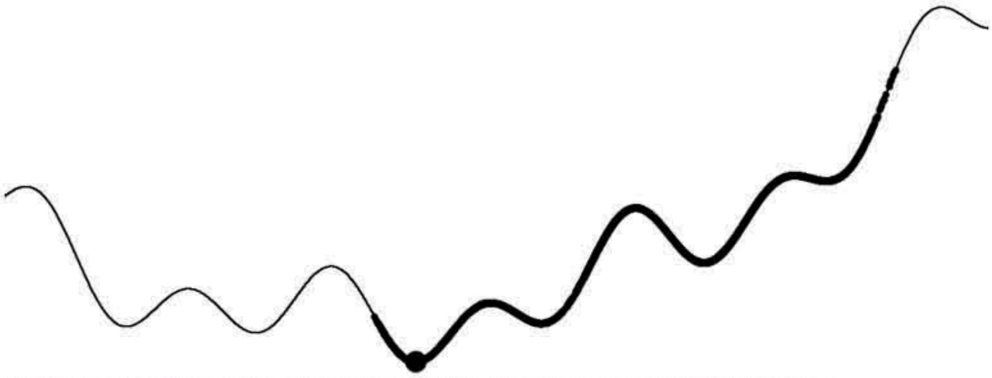

Abbildung 25: Metropolis-Algorithmus, Globales Minimum nach 2000 Schritten

Wir benötigen also zusätzliche Information, wenn wir sichergehen möchten, dass ein Wert tatsächlich das globale Minimum ist. Alternativ benötigen wir statistische Aussagen, um wenigstens zu wissen, mit welcher Wahrscheinlichkeit wir den kleinsten Wert gefunden haben bzw. wie weit wir von diesem noch entfernt sind.

Bei einem zu kleinen Parameter T=0.000001 kommen wir erwartungsgemäß nur bis ins nächste Tal.

Das Verfahren ähnelt der bereits vorgestellten Methode des steilsten Abstiegs, findet also zuverlässig lokale Täler, kann aber keine Hügel mehr überwinden auf der Suche nach einem noch tiefer gelegenen Punkt.

Der Einfluss des Zufalls auf die Suche ist quasi nicht mehr vorhanden.

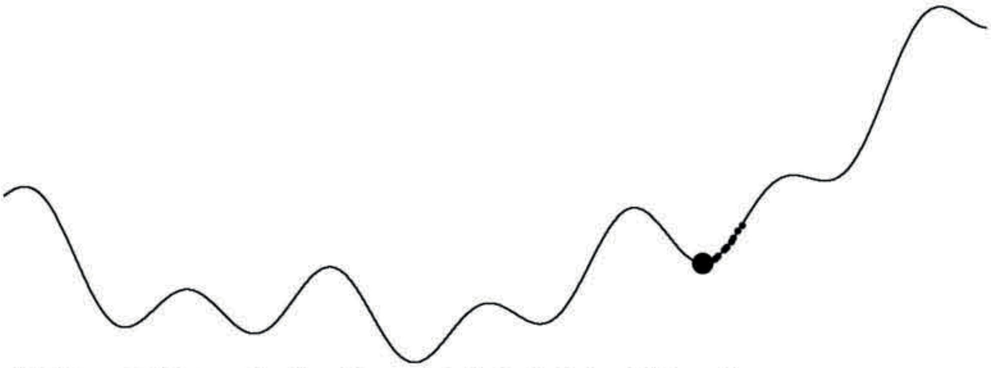

Abbildung 26: Metropolis-Algorithmus, lokale Suche bei zu kleinem T

Bei einem zu großen Parameter T=1000 laufen wir ziellos herum und finden auch nach 2000 Schritten nicht das gesuchte Minimum.

Der Einfluss des Zufalls auf die Suche ist nun zu groß, sie läuft planlos ab und entspricht einem Ratespiel.

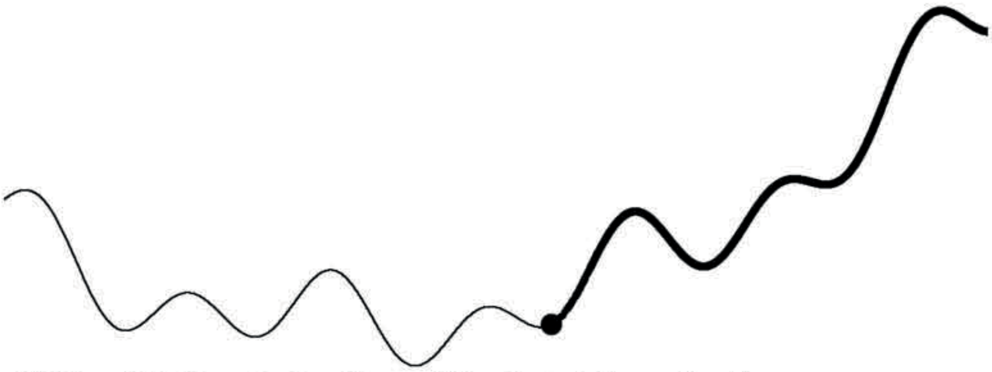

Abbildung 27: Metropolis-Algorithmus, ziellose Suche bei zu großem T

2.5.2 Bewertung

Vorteile:

- Manchmal führt die Entscheidung „steuere das Verhalten zufällig" zu besseren Resultaten als eine einseitige, starre Regel.

Nachteile:

- Durch den Einfluss des Zufalls können wir natürlich nicht immer garantieren, dass ein zufallsgesteuertes Verfahren eine bestimmte Lösung überhaupt findet. Wir können nur statistische Aussagen treffen wenn wir wissen möchten, mit welcher Wahrscheinlichkeit ein bestimmtes Ergebnis gefunden wird und wie lange die Suche dauert.
- Durch den Einfluss des Zufalls müssen wir davon ausgehen, dass eine gute Lösung wieder verloren geht. Wir müssen uns also die jeweils beste bisher gefundene Lösung merken.
- Meistens müssen wir ein Abbruchkriterium wählen, um zu vermeiden, dass das Verfahren unendlich lange läuft.

Nachgefragt ...

1. Muss der Parameter T beim Metropolis-Algorithmus während der Suche konstant bleiben? Können wir das Verfahren verbessern, wenn wir T im Lauf der Zeit anpassen? Wie gehen wir bei der Anpassung vor? Sollte sich eine solche Anpassung an der konkreten Anwendung orientieren?
2. Zeichne ein Bild nach folgender Vorschrift: Ausgehend vom Punkt $(0,0)$ wird der nächste Punkt nach einer der folgenden Formeln berechnet, die zufällig anhand der angegebenen Wahrscheinlichkeiten ausgewählt werden.
   $(x_{neu}, y_{neu}) = (0.85 \cdot x_{alt} + 0.04 \cdot y_{alt}, -0.04 \cdot x_{alt} + 0.85 \cdot y_{alt} + 0.16)$, 85%
   $(x_{neu}, y_{neu}) = (0.20 \cdot x_{alt} - 0.26 \cdot y_{alt}, 0.23 \cdot x_{alt} + 0.22 \cdot y_{alt} + 0.16)$, 7%
   $(x_{neu}, y_{neu}) = (-0.15 \cdot x_{alt} + 0.28 \cdot y_{alt}, 0.26 \cdot x_{alt} + 0.24 \cdot y_{alt} + 0.04)$, 7%
   $(x_{neu}, y_{neu}) = (0.00, 0.16 \cdot y_{alt})$, 1%
   Es entsteht ein Farn nach **M. Barnsley** [Barn88].