

Einführung in den Simulated Annealing Algorithmus

Autor:

Günther M. FOIDL *

Letzte Aktualisierung: 18. November 2009

Zusammenfassung

Simualted Annealing ist ein meta-heuristisches Optimierungsverfahren welches auf kontinuierliche und auf diskrete Probleme angewandt werden kann. Es besitzt den großen Vorteil dass lokale Minima verlassen werden können.

In diesem Artikel werden zuerst die Grundlagen erarbeitet und Beispiele für die Minimierung einer Funktion sowie für das Problem des Handelsreisenden gezeigt.

Inhaltsverzeichnis

| | |
|--|----------|
| 1 Einführung | |
| 1.1 Exkurs Wärmebehandlung | 2 |
| 1.2 Mathematisches Modell | 2 |
| 1.3 Wahl eines Nachbars | 3 |
| 1.3.1 Kontinuierliche Probleme | 3 |
| 1.3.2 Diskrete Probleme | 3 |
| 1.4 Abkühlungskurve | 4 |
| 1.5 Pseudocode | 4 |
| 1.6 Bildliche Vorstellung | 4 |
| 2 Anwendung | 5 |
| 2.1 Minimierung einer reellen Funktion | 5 |
| 2.2 Problem des Handelsreisenden | 6 |
| 3 Verwandte Verfahren | 6 |

1 Einführung

Simualted Annealing ist ein Verfahren zum Auffinden einer Näherungslösung für ein Optimierungsproblem (meist Minimierung) ohne dass alle Lösungen wie bei einem Brute-Force-Ansatz ausprobiert werden müssen. Es ist eine Heuristik, d.h. ein Verfahren das oft sehr gut funktioniert, aber keine Garantie für eine optimale oder auch nur gute Lösung gibt.

Es wird eine *Random Search*-Strategie welche nicht nur Lösungen akzeptiert die einen niedrigeren Funktionswert ergeben (im Falle eines Minimierungsproblems), sondern auch mit einer bestimmten Wahrscheinlichkeit (Boltzmann-Gibbs-Verteilung) Lösungen die einen höheren Funktionswert ergeben. Durch diese Tatsache hat Simualted Annealing den Vorteil dass lokalen Minima wieder verlassen werden können.

Dieses Verfahren kann sowohl für kontinuierliche Probleme der Art $f : \mathbb{R}^n \rightarrow \mathbb{R}$ als auch für diskrete Probleme wie zB das *Traveling Salesman Problem* angewandt werden. In diesem Artikel werden beide Problemtypen einleitend behandelt.

Simualted Annealing ist die Nachbildung eines Wärmebehandlungsverfahrens der Metallurgie, daher auch die englische Bezeichnung *Annealing*.

*✉ gue@korporal.at ☎ +43 664 8918365

1.1 Exkurs Wärmebehandlung

Diese Hintergrundinformation ist hilfreich für das Verständnis.

In der Metallurgie wird ein Metall erhitzt und somit in einen hohen Energiezustand¹ versetzt. Durch langsames Abkühlen² haben die Kristalle ausreichend Zeit sich im Gefüge neu und ohne Verspannungen anzuordnen und ein (zeitlich) stabiles Gefüge zu bilden da sich Atome um ihre Anfangsposition bewegen können. Nach der Abkühlung liegt ein Gefügestand mit minimaler Energie vor, der Nahe am Optimum ist. Aufgrund des niedrigen Energiezustands wird dies auch als *Weichglühen* bezeichnet was sich dadurch äußert dass das Metall weich ist. Am Rande erwähnt wäre das Gegenbeispiel dazu das so genannte *Härten*.

1.2 Mathematisches Modell

Für die Beschreibung des mathematischen Modells wird ausschließlich ein kontinuierliches Minimierungs-Problem betrachtet. Für die Maximierung und für diskrete Problem gilt Analoges. Es wird angenommen dass der Leser mit den grundlegendem mathematischen Zeichen vertraut ist.

Simulted Annealing [1, 2] ist aus dem *Metropolis Algorithmus* hervorgegangen und dieser ist eine *Monte-Carlo-Methode* zur Erzeugung von (thermodynamischen) Systemzuständen entsprechend der Boltzmann-Verteilung. Für weitere Information sei auf die Literatur wie zB auf [3] verwiesen.

Ausgegangen wird von einer so genannten *Fitness-Funktion*

$$f : \mathbf{x} \in D \rightarrow \mathbb{R}, \quad D \subseteq \mathbb{R}^n \quad (1)$$

welche es zu minimieren gilt, also die Lösung für $\min\{f(\mathbf{x})\}$ gesucht wird. Dazu wird ein beliebige Anfangslösung – ein Startvektor – aus

dem Definitionsbereich gewählt $\mathbf{x}(0) \in D$ und der Wert des Temperaturparameters auf die „Glühtemperatur“ erhöht. In einer festgelegten Anzahl $N \in \mathbb{N}$ an Schritten wird die Temperatur gemäß der Abkühlungskurve auf 0 gesenkt.

Die Abkühlungskurve ist kann exponentiell, linear, etc. geartet sein. Ich habe mich jedoch am realen Vorbild orientiert und einen linearen Verlauf gemäß

$$T(t) = T(0) \cdot \left(1 - \frac{t}{N}\right), \quad t = 1, \dots, N \quad (2)$$

gewählt, wobei T für die Temperatur und t für die Zeit steht.

In jedem Abkühlungsschritt, also für die monoton fallende Temperaturkurve $T(t)$, wird aus der Umgebung U des aktuellen Lösungsvektors $\mathbf{x}(t)$ ein Nachbar \mathbf{x}^* gewählt.

$$\mathbf{x}^* \in U(\mathbf{x}(t)), \quad U \subseteq D \quad (3)$$

Die Wahrscheinlichkeit P^* für den Übergang des aktuellen Zustandes \mathbf{x} in den Nachbarzustand \mathbf{x}^* folgt der *Boltzmann-Gibbs-Verteilung* und berechnet sich zu

$$P^* = \begin{cases} 1 & \text{wenn } f(\mathbf{x}^*) < f(\mathbf{x}(t)), \\ e^{-\frac{f(\mathbf{x}^*) - f(\mathbf{x}(t))}{T(t)}} & \text{sonst} \end{cases} \quad (4)$$

D.h. falls der Nachbarschaftszustand eine geringere Energie (Fitness) aufweist wird dieser Zustand mit Sicherheit übernommen. Ist die Energie jedoch höher so wird dieser Zustand mit einer bestimmten Wahrscheinlichkeit übernommen. Dies ist notwendig um lokale Minima³ verlassen zu können. Für den Fall $\lim_{t \rightarrow N} T(t) = 0 \Rightarrow P^* \rightarrow 0$ werden nur Zustände mit niedrigerer Energie übernommen und der Algorithmus entartet in einen *Greedy Algorithmus*. Sonst gilt je höher die Temperatur und je kleiner die Energiedifferenz desto wahrscheinlicher ein Übergang.

¹kinetische Energie der Atome im Kristallgitter

²Stunden, Tage, je nach Größe des Metallstücks

³das ist ein Zustand der schlechter als das globale Minimum ist, aber besser als die Nachbarzustände

Der Lösungsvektor wird gemäß folgender Vorschrift aktualisiert:

$$\mathbf{x}(t+1) = \begin{cases} \mathbf{x}^* & \text{wenn } U(0,1) \leq P^*, \\ \mathbf{x}(t) & \text{sonst} \end{cases} \quad (5)$$

wobei $U(0,1)$ eine gleichverteilte Zufallszahl aus dem Intervall $[0,1]$ ist.

Dieser Zyklus der Erwärmung–Abkühlung wird so lange wiederholt bis die Lösung konvergiert oder sonst ein Abbruchkriterium erreicht wird.

Granville et al. zeigten in [4] dass das globale Minimum mit Sicherheit erreicht wird wenn die Abkühlung unendlich langsam stattfindet. Der Autor kann sich an dieser Stelle die etwas zynische Anmerkung nicht verkneifen dass dieser Beweis auf der Hand liegt denn bei unendlicher Abkühlungszeit wird jeder Punkt im Suchgebiet exploriert. Trotzdem wurde in dieser Arbeit gezeigt dass Simulated Annealing – als Markov-Kette dargestellt – gegen das globale Minimum konvergiert.

1.3 Wahl eines Nachbars

Die Wahl eines Nachbars erfolgt nach einer Heuristik. Der Algorithmus selbst definiert keinerlei Vorgehen für die Wahl von \mathbf{x}^* . Die Heuristik muss an das zu lösende Problem angepasst werden.

Metropolis et al. [3] schlagen eine Heuristik vor welche berücksichtigt dass nach ein paar Iterationen des Algorithmus angenommen werden kann dass der Energiezustand sehr viel niedriger ist als der des zufälligen Anfangszustands. Somit sollten Nachbarn mit ähnlicher Energie wie der Zustand $\mathbf{x}(t)$ gewählt werden. Dadurch werden sowohl die besten als auch die schlechtesten möglichen Nachbarn ausgeschlossen. Letztere sind wesentlich häufiger und somit folgt dass diese Heuristik in der Regel sehr gut funktioniert.

Bei kleineren niedrigdimensionalen Problemen genügt ein zufällige Wahl des Nachbars

aus dem Definitionsgebiet – die Ressourcen heutiger Rechner erlauben dies.

1.3.1 Kontinuierliche Probleme

Für die Wahl eines Nachbars \mathbf{x}^* kann ein neuer Zufallsvektor aus dem Definitionsgebiet D gewählt werden wie in (3) allgemein definiert wurde. Alternativ kann der aktuelle Vektor $\mathbf{x}(t)$ durch Superposition mit statistischem Rauschen in seiner Position verschoben werden [5], also

$$\mathbf{x}^* = \mathbf{x}(t) + \mathbf{D}(t) \cdot \mathbf{r}(t) \quad (6)$$

wobei $\mathbf{r}(t) \sim U(-1,1)^{n_x}$ ein gleichverteilter Zufallsvektor mit der Dimension n_x und \mathbf{D} eine Diagonalmatrix welche die maximal zulässigen Verschiebungen jeder Komponente ξ_i des Vektors $\mathbf{x} = [\xi_1, \dots, \xi_n]^T$ bestimmt. Ergibt die neue Position \mathbf{x}^* eine bessere Lösung, d.h. $f(\mathbf{x}^*) < f(\mathbf{x}(t))$, so kann die Matrix \mathbf{D} wie folgt aktualisiert werden:

$$\mathbf{D}(t+1) = (1 - \alpha)\mathbf{D}(t) + \alpha\omega\mathbf{R}(t) \quad (7)$$

wobei $\mathbf{R}(t) = (\rho_{ii})$ eine Diagonalmatrix ist, deren Elemente ρ_{ii} die Beträge der Verschiebung darstellen $\rho_{ii} = \|\mathbf{x}^* - \mathbf{x}\|$ und α, ω Konstanten sind.

1.3.2 Diskrete Probleme

Für diskrete Probleme ist eine Möglichkeit den obigen Ansatz (6) zu diskretisieren. Somit folgt:

$$\mathbf{x}^* = \mathbf{x}(t) + \mathbf{r}(t) \quad (8)$$

wo jedes Element ρ_i des Vektors $\mathbf{r} = [\rho_1, \dots, \rho_n]^T$ zufällig aus der Menge $\{-1, 0, 1\}$ gewählt wird.

Bei kombinatorischen Problemen – zB beim Problem des Handelsreisenden [6] – kann ein Nachbar dadurch gewählt werden dass in der Permutation der zu bereisenden Städte der Tour zwei folgende Städte vertauscht werden.

Eine komplett neue Permutation der Städte wäre auch möglich – dies empfiehlt sich aber aufgrund oben beschriebener Heuristik nicht und außerdem würde der Aufwand für das neue Erstellen der Tour überproportional ansteigen. Beim Vertauschen von zwei folgenden Städten beträgt der Aufwand maximal $n \cdot \frac{n-1}{2}$, wobei n die Anzahl der Städte ist.

1.4 Abkühlungskurve

Die Wahl des Abkühlungsverlaufs ist ebenfalls ein entscheidendes Kriterium für die Güte des Algorithmus angewandt auf ein konkretes Problem.

Im realen Vorbild des Weichglühens (cf. 1.1) wird die Temperatursenkung im Idealfall so gewählt dass sich das Gefüge stets Nahe dem thermodynamischen Gleichgewicht befindet⁴. Daher ergeben sich kleine Abkühlraten dT und lange Abkühlzeiten.

Bei Simulated Annealing werden die Parameter für die Abkühlung – das sind der Funktionsverlauf $t \mapsto T(t)$, die Starttemperatur $T(0)$ und die Anzahl der Schritte N – daher vom Benutzer definiert um diese an das Problem anpassen zu können.

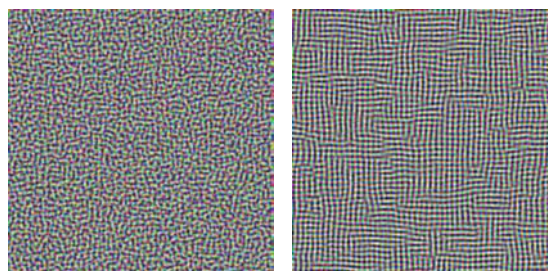
Als gute Wahl für die meisten Probleme hat sich $T(0) = 500$, $N = 100$ und die lineare Abkühlung herausgestellt⁵.

Es sei jedoch angemerkt dass diese Werte an das Problem empirisch angepasst werden müssen.

Abbildung 1 zeigt den Vergleich einer zu schnellen und einer adäquaten Abkühlung. Mit metallurgischem Hintergrund ist deutlich zu sehen dass die schnelle Abkühlung zu einer amorphen Struktur (= schlechte Lösung) und die langsame Abkühlung zu einer kristallinen Struktur (= gute Lösung) führt.

⁴im thermodynamischen Gleichgewicht würde keine Abkühlung erlauben

⁵Motiviert hat mich diese Wahl durch das reale Vorbild bei dem ähnliche Werte verwendet werden



(a) Schnelle Abkühlung (b) Langsame Abkühlung

Abbildung 1: Vergleich von schneller und langsamer Abkühlung. (a) führt zu einer amorphen Struktur während (b) zu einer kristallinen Struktur führt. Für eine Beschreibung des Problems das dargestellt wurde sei auf die Quelle verwiesen. (Quelle: [7])

Wie in Abschnitt 1.2 erwähnt halte ich mich meist an das reale Vorbild und verwende die lineare Abkühlung (2). Eine exponentielle Abkühlung könnte wie folgt definiert werden:

$$T(t) = T(0) \cdot \exp\left(\frac{\log \frac{T(N)}{T(0)}}{N} \cdot t\right) \quad (9)$$

Natürlich kann anstatt der direkten Berechnung der Temperaturwerte $T(t)$ auch eine rekursive Definition verwendet werden.

1.5 Pseudocode

Der Pseudocode wird in Algorithm 1 auf der nächsten Seite gezeigt. Abbildung 2 zeigt das Klassendiagramm der generischen Basisimplementierung von welcher für das konkrete Problem abgeleitet werden kann.

1.6 Bildliche Vorstellung

Verbildlicht kann sich das auch so vorgestellt werden [8]:

Angenommen, man sucht in einer zweidimensionalen Landschaft den (global) tiefsten

Algorithm 1 Simulated Annealing Algorithm

Wähle Anfangslösung $\mathbf{x} \leftarrow \mathbf{x}(0)$
repeat
 Erwärme auf Glühtemperatur $T \leftarrow T(0)$
 for $t \leftarrow 0$ to N **do**
 Wähle einen Nachbar \mathbf{x}^* (3)
 Berechne Energie $f(\mathbf{x}^*)$ (1)
 Berechne Wahrscheinlichkeit P^* (4)
 if $U(0, 1) \leq P^*$ **then**
 $\mathbf{x}(t) \leftarrow \mathbf{x}^*$
 end if
 end for
until Konvergenz oder Abbruchbedingung;
repeat $\mathbf{x}(t)$ als Lösung

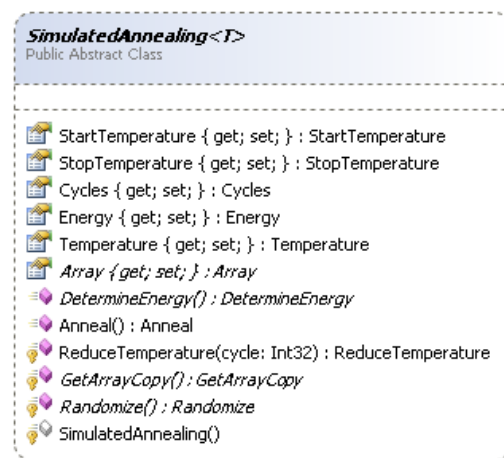


Abbildung 2: Klassendiagramm der Basisimplementierung.

Punkt. Die Landschaft selbst besteht aus vielen unterschiedlich tiefen Dellen. Die einfache Suchstrategie (suche den nächsten tiefsten Punkt) entspricht dem Verhalten einer Kugel, welche in dieser Landschaft ausgesetzt wird. Sie rollt zum nächsten lokalen Minimum und bleibt dort. Bei der simulierten Auskühlung wird der Kugel immer wieder ein Stoß versetzt. Dieser ist stark genug, um die Kugel aus einer flachen Delle (lokales Minimum) zu entfernen, reicht aber nicht aus, um aus dem globalen Minimum zu fliehen.

2 Anwendung

Wie bereits einführend erwähnt kann Simualted Annealing auf kontinuierliche und diskrete Probleme angewandt werden. Ursprünglich wurde Simualted Annealing für die Optimierung des IC-Entwurfs⁶ angewandt [1]. Im Zuge meiner Arbeiten auf diesem Gebiet stellte sich heraus dass Simualted Annealing hervorragend für

- Minimierung von reellen Funktionen $\min\{\mathbb{R}^n \rightarrow \mathbb{R}\}$

⁶Floorplanning

- Training von Feedforwardnetzwerken⁷
- Projektionen eines hochdimensionalen Vektorsets $\{\mathbf{x}_i\} \in \mathbb{R}^n$ auf eine niedrigdimensionale (meist 2) Bildebene $\{\mathbf{x}'_i\} \in \mathbb{R}^m$, $m \ll n$ wie zB durch Anwendung der Sammon's Projection [9, 10]
- Lösung kombinatorischer Optimierungsprobleme wie zB das Problem des Handelsreisenden [6]

eingesetzt werden kann.

Folgend wird die exemplarische Vorgehensweise für je ein kontinuierliches Problem und für eine diskretes Problem gezeigt.

2.1 Minimierung einer reellen Funktion

Eine reelle Funktion $f : \mathbb{R}^n \rightarrow \mathbb{R}$ kann durch Anwendung des bisher erarbeiteten minimiert

⁷bei einer großen Anzahl an Neuronen empfehle ich jedoch *Resilientpropagation* zu verwenden

werden. Da keine zusätzlichen weiteren Definitionen notwendig sind sieht der Autor keinen Grund dieses hier weiter zu verfolgen.

2.2 Problem des Handelsreisenden

Zur Klärung der Problemstellung wird hier verzichtet und auf die Literatur verwiesen, zB [6]. Mich hat immer fasziniert wie schnell Simulated Annealing eine Lösung findet – auch mit 500 zu bereisenden Städten wird eine Lösung binnen weniger Sekunden gefunden. Ein *Brute-Force-Ansatz*, also das ausprobieren aller Kombinationen, würde mit heutigen CPUs Jahre dafür brauchen. Viele andere Lösungsansätze für dieses Problem können praktisch nur für 50...100 Städte angewandt werden.

Das hier gezeigte Vorgehen ist in der grundlegendsten Form und könnte durch Wahl anderer Heuristiken wie zB für die Generierung einer neuen Route noch angepasst werden.

Da es sich um ein kombinatorisches Problem handelt dessen Ziel es ist den Gesamtweg zu minimieren müssen zuerst die mathematischen Rahmenbedingungen erstellt werden.

Darstellung der Städte Städte werden durch Vektoren $\{\mathbf{x}\}$, $\mathbf{x} \in \mathbb{R}^2$ dargestellt.

Distanzmaß Für die Bestimmung des Abstandes $d := d(\mathbf{x}, \mathbf{y})$ zwischen zwei Städten $\mathbf{x} = [\xi_1, \xi_2]^T$ und $\mathbf{y} = [\eta_1, \eta_2]^T$ wird die euklidische Metrik verwendet und somit gilt:

$$d_E(\mathbf{x}, \mathbf{y}) = \sqrt{(\xi_1 - \eta_1)^2 + (\xi_2 - \eta_2)^2} \quad (10)$$

Repräsentation der Route Die Route wird durch einen Indexvektor $\mathbf{i} \in \mathbb{N}^n$ dargestellt, wobei n die Anzahl der Städte ist. Jedes Element dieses Vektors entspricht dem Index einer Stadt in der geordneten Menge der Städte.

Anwendung des Algorithmus Simulated Annealing wird auf den Indexvektor entsprechend bisheriger Erörterungen angewandt.

3 Verwandte Verfahren

- genetische Algorithmen
- Partikelschwarmoptimierung⁸
- Bienenalgorithmus
- und Abwandlungen der genannten

Literatur

- [1] S. Kirkpatrick, C. Gelatt, and M. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983. 2, 5
- [2] S. Kirkpatrick, "Optimization by simulated annealing: Quantitative studies," *Journal of Statistical Physics*, vol. 34, no. 5–6, pp. 975–986, 1984. 2
- [3] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *Journal of Chemical Physics*, vol. 21, no. 6, pp. 1087–1092, 1953. 2, 3
- [4] V. Granville, M. Krivanek, and J.-P. Rason, "Simulated annealing: a proof of convergence," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, pp. 652–656, Jun 1994. 3
- [5] A. Engelbrecht, *Computational Intelligence*. Wiley, 2007. 3
- [6] "Wikipedia: Problem des handelsreisenden." http://de.wikipedia.org/wiki/Problem_des_Handlungsreisenden. 3, 5, 6

⁸siehe auch [11]

- [7] “Wikipedia: Simulated annealing.”
http://en.wikipedia.org/wiki/Simulated_annealing. 4
- [8] M. Gell-Mann, “On getting creative ideas.”
Vortrag, March 2007. Google Tech Talks.
4
- [9] J. W. S. jr., “A nonlinear mapping for data
structure analysis,” *IEEE Transactions on
Computers*, vol. C-18, pp. 401–409, May
1969. 5
- [10] G. M. Foidl, “Sammon projecti-
on.” Codeproject, October 2009.
[http://www.codeproject.com/KB/recipes/
SammonProjection.aspx](http://www.codeproject.com/KB/recipes/SammonProjection.aspx). 5
- [11] G. M. Foidl, “Particle swarm opti-
mization for function optimizati-
on.” Codeproject, September 2009.
[http://www.codeproject.com/KB/recipes/
particleswarmoptimization.aspx](http://www.codeproject.com/KB/recipes/particleswarmoptimization.aspx). 6