Computerphysik WS 2015/2016

Prof. Dr. Roland Netz, FU Berlin

Übungsblatt 14: Optimierung mit Simulated Annealing: Traveling Salesman Problem

Roland Netz

28. Januar 2016

Allgemeine Hinweise

Abgabetermin für die Lösungen ist

- **Sonntag, 07.02.2014, 24:00 Uhr.**

14: Das Problem des Handelsreisenden (20 Punkte)

Wir beschäftigen uns in diesem Übungsblatt mit einem klassischen diskreten Optimierungsproblem, dem "Traveling Salesman Problem". Dazu betrachten wir N Städte, die zufällig auf einer quadratischen Fläche im Bereich $x,y = (0,1)^2$ angeordnet sind. Gesucht ist die kürzeste geschlossene Route auf der jede Stadt genau einmal besucht wird. Ein Algorithmus zur Lösung dieses Problems benötigt eine Zeit, die exponentiell (also nicht-polynomisch!) mit N wächst, was numerisch ein Problem darstellt. Ähnliche Probleme tauchen in der Logistik auf, zum Beispiel bei der Optimierung von Fahrrouten von Paketzustellern oder bei der Optimierung von Fahrplänen und Stundenplänen, aber auch beim Design von Computerchips.

14.1: Implementieren der benötigten Funktionen (12 Punkte)

Im Folgenden sollten Sie ein Programm entwickeln, das eine Optimierung der Route durchführt. Dazu benötigen Sie die folgenden Hilfsfunktionen:

- 14.1.1 (1 Punkt): Erzeugung einer zufälligen Anordnung der N Städte, `posx[i]` und `posy[i]` mit $i = 0, \ldots, N-1$. Dabei sind `posx[i]` und `posy[i]` Zufallszahlen zwischen 0 und 1.
- 14.1.2 (2 Punkte): Berechnung der Distanzmatrix zwischen den Städten, `dist[i,j]`.
- 14.1.3 (1 Punkt): Initialisierung der Reise, also der Reihenfolge in der die Städte besucht werden, `trip[i]`. Die Initialisierung können Sie dabei beliebig durchführen, z.B. `trip[i]=i`.
- 14.1.4 (2 Punkte): Berechnung der Länge der Reise `length[trip]`.
- 14.1.5 (2 Punkte): Modifizierung der Reise indem zwei Städte miteinander vertauscht werden, d.h. Erstellen eines Arrays `tripnew[i]`.

Im Hauptprogramm sollen dann die Längen der Reise `length[trip]` und der modifizierten Reise `length[tripnew]` verglichen werden. Wählen Sie eine effektive Temperatur T und akzeptieren Sie die Modifikation mit einer Wahrscheinlichkeit

$$p = \exp\left(-(\texttt{length}[\texttt{tripnew}] - \texttt{length}[\texttt{trip}])/T\right) \tag{1}$$

Sie sollten die beste Reise `tripopt` (mit der kürzesten Länge `lengthopt`) speichern.

- 14.1.6 (4 Punkte): Implementieren Sie eine Funktion, welche die die Initialisierung ausführt und die Optimierung mittels Gleichung (1) durchführt. Parameter sollten hierzu die Anzahl N der Städte, die Anzahl i_{max} der maximalen Optimierungsschritte und die Temperatur T sein.

14.2: Vergleich von Optimierungsstrategien (8 Punkte)

- 14.2.1 (4 Punkte): Wählen Sie $N = 100$ Städte. Führen Sie $i_{max} = 500.000$ Optimierungsschritte für Temperaturen $T = 10^{-p}$ mit $p = 5,4,3,2,1,0$ durch. Wie hängt die gefundene optimale Reiselänge `lengthopt` von der Temperatur T ab?

  Stellen sie die Start- und Endroute grafisch dar. Plotten Sie außerdem die Länge der Route als Funktion des Optimierungsschritts.

  Wenn Sie mögen können Sie die Reise-Konfigurationen und die Reiselänge während der Optimierung grafisch darstellen (siehe die Beispiele zur Animation aus den früheren Übungsblättern). Interessant ist hierbei insbesondere der Vergleich mit der (bis dahin bestimmten) optimalen Reiseroute.

- 14.2.2 (1 Punkt): Überzeugen Sie sich dass der Optimierungscode für tiefe Temperaturen einem *steepest descent* oder *greedy algorithm* entspricht, bei dem nur Verkürzungen der Reiseroute akzeptiert werden.

  Warum erzielt der Algorithmus für mittlere Temperaturen eine bessere Reiseroute als bei tiefen Temperaturen? Schreiben Sie zur Beantwortung eine kurze Begründung.

- 14.2.3 (3 Punkte): Im sogenannten *simulated annealing* wird die Temperatur schrittweise von einem hohen Anfangswert auf einen niedrigen Endwert gesetzt. Führen Sie wieder 500.000 Optimierungsschritte durch, und kühlen Sie das System linear von $T = 0{,}1$ auf $T = 0$ ab. Welche optimale Reiselänge erhalten Sie während dieser Optimierungsstrategie ?