

Übung 5

Modellierung, Simulation und Optimierungsverfahren II

Dr. Colin W. Glass

Institut für Höchstleistungsrechnen

07.05.2014

Nachbesprechung Übung 4 a: Repräsentation TSP

Ein (symmetrisches) TSP wird durch einen Vektor mit Integeren, z.B. $(3, 15, 2, \dots, 11)$, der Länge n dargestellt, wobei jeder Integer $[1, n]$ genau ein Mal verwendet wird. Der Suchraum dieser Repräsentation hat die Größe $|\mathcal{S}| = n!$, ist also größer als der problemintrinsische Suchraum.

Wie könnte man die Repräsentation anpassen, um $|\mathcal{S}| = \frac{(n-1)!}{2}$ zu erreichen (keine redundanten Lösungen im Suchraum)?

Tipp: Von $n!$ zu $(n-1)!$ und die Halbierung $(\frac{1}{2})$ von $|\mathcal{S}|$ sind zwei separate Schritte.

Nachbesprechung Übung 4 a: Repräsentation TSP

Wie könnte man die Repräsentation anpassen, um $|\mathcal{S}| = \frac{(n-1)!}{2}$ zu erreichen (keine redundanten Lösungen im Suchraum)?

Lösung:

Von $n!$ zu $(n-1)!$ funktioniert ganz leicht: Eine Stadt muss an eine Stelle in der Repräsentation festgenagelt werden. Z.B. Stadt nr. 1 an die erste Stelle. 1 - ... - k

Nachbesprechung Übung 4 a: Repräsentation TSP

Wie könnte man die Repräsentation anpassen, um $|\mathcal{S}| = \frac{(n-1)!}{2}$ zu erreichen (keine redundanten Lösungen im Suchraum)?

Lösung:

Von $n!$ zu $(n-1)!$ funktioniert ganz leicht: Eine Stadt muss an eine Stelle in der Repräsentation festgenagelt werden. Z.B. Stadt nr. 1 an die erste Stelle. 1 - ... - k

Die Halbierung ($\frac{1}{2}$) ist komplizierter: Die Umlaufrichtung muss festgelegt werden. Stadt 1 ist mit zwei anderen Städten verbunden, z.B. 7 und 3. Wir legen fest: an zweiter Stelle der Repräsentation kommt die Stadt mit kleinerer (größerer) Nummer. 1 - 3 - ... - 7

Nachbesprechung Übung 4 b: Repräsentation SAT

Ein SAT wird durch einen Vektor mit Bits, z.B. $(0,1,1,0, \dots, 1)$, der Länge n dargestellt.

Wie könnte man jede Lösung mit nur einer Zahl repräsentieren?

Wie in der Vorlesung schon besprochen: einen Integer $[0, 2^n - 1]$.

Aber: wir müssen wissen wie lang der Bitvektor sein muss!

$(n = 6)$ Lösung 31 $\hat{=}$ 011111

$(n = 9)$ Lösung 31 $\hat{=}$ 000011111

Übung 5: TSP

Wir haben ein asymmetrisches (!) 70-Städte TSP, welches ich zufällig generiert habe ($\text{rand}(70)*10$). Es ist im Wesentlichen einfach eine Tabelle voller Stadt-Stadt Reisekosten. Diese findet ihr auf der IHR Seite (CostTable.mat).

Ihr ladet sie folgendermassen (aus Octave oder Matlab):

```
>> load CostTable.mat
```

Danach habt ihr eine 70x70 Matrix, welche "CostTable" heisst.

P.s: ">>" nicht schreiben.

P.p.s: Octave bzw. Matlab muss CostTable.mat sehen können, also in dem Ordner sein, wo CostTable.mat abgespeichert wurde, oder den entsprechenden Ordner in "path" haben.

Übung 5: TSP

Dann könnt ihr die Evaluierungsfunktion (ebenfalls online) verwenden:

```
function Cost = EvalTSP (Route,CostTable)
    Route(end+1) = Route(1);
    Cost = 0;
    for ind = 1:70
        Cost = Cost + CostTable(Route(ind),Route(ind+1));
    end
```

Übung 5: TSP

>> $\text{RoutenKosten} = \text{EvalTSP}(\text{Route}, \text{CostTable})$

Für eine Route (Vektor Länge 70 mit Stadtnummern) gibt sie Euch die Kosten aus.

Aufgabe: Schreibt einen Hillclimber für dieses TSP.

Bonus: Erweitert den Hillclimber zu Simulated Annealing.