

NP-Vollständigkeit

Im Laufe Ihres Informatiker-Lebens wird Ihnen häufig die Aufgabe gestellt, Entscheidungs- bzw. Planungsprobleme zu automatisieren, die bisher ein Mensch manuell durchgeführt hat. Ideal wäre es hierbei, wenn Sie dabei die menschliche Intuition bzw. Planungserfahrung durch eine von ihrem Programm berechnete optimale Entscheidung bzw. Plan ersetzen. Manchmal ist das effizient (in polynomialer Zeit) möglich wie z.B. beim Navigationsproblem durch den Dijkstra-Algorithmus (Dynamisches Programmieren). Meistens ist das Planungsproblem jedoch NP-Vollständig. Dies bedeutet, dass

- (i) kein effizienter Algorithmus gefunden werden kann ohne damit bereits das berühmte $P=NP$ -Problem zu lösen,
- (ii) bisher nur Algorithmen mit exponentieller Rechenzeit zur Berechnung der optimalen Lösung bekannt sind,
- (ii) für die praktische Software-Implementierung deshalb jeweils nur Näherungslösungen gesucht sind:

Finde in vorgegebener Zeit eine möglichst gute Lösung.

Da der Mensch solch schwierige Entscheidungsproblem genauso wenig in kurzer Zeit optimal lösen kann, ist das Ziel für die Entwicklung von Lösungsverfahren: In wesentlich kürzerer Zeit wesentlich bessere Lösungen als der menschliche Planer zu finden.

Die folgende Aufgabe hilft Ihnen zu entscheiden, ob die Suche nach einem Algorithmus für die optimale Lösung aussichtslos ist, da die NP-Vollständigkeit (leicht) nachgewiesen werden kann.

Aufgabe 1 NP-vollständigkeit von TSP – Varianten in der Anwendung

Zeigen Sie, dass eines der folgenden praxisnahen Auslieferungprobleme NP-Vollständig ist. Gehen Sie in folgenden Schritten vor (gemäß den beiden Folien zur formalen Definition von NP und NPC):

(i) Definieren Sie das zugehörige Entscheidungsproblem A

(Für Optimierungsproblem O mit " ... minimiere $f(z)$ "
lautet das Entscheidungsproblem A " ... gibt es Lösung z mit $f(z) \leq K$ ")

Hinweis:

- Optimierungsproblem entspricht Funktion: Ausgabe der optimalen Lösung z zu Problem x
- Entscheidungsproblem entspricht Prädikat: $1 \Leftrightarrow$ Es gibt Lösung z zu x mit $f(z) \leq K$

(ii) Problem A liegt in NP: Hierzu müssen Sie kurz ein Prüfprogramm P skizzieren, das zu einer Probleminstanz x eine Lösung z rät und dann prüft ob $f(z)$ unter der Schranke K liegt. Überprüfen Sie, ob die Kodierungslänge von z durch ein Polynom p über die Kodierungslänge der vorgegebenen Probleminstanz x beschränkt ist. Meist ist die Kodierungslänge von einer Lösung z kleiner als die Problembeschreibung x und es ist deshalb p aus $O(n)$.

(iii) Zeigen Sie, dass Ihr Problem NP-hart ist. Es genügt hierzu zu zeigen, dass sich ein geeignetes als NP-vollständig bekanntes Problem B auf ihr Problem A reduzieren lässt. Dieser Nachweis ist sehr einfach, wenn P ein Spezialfall von X ist. Dann genügt als Abbildung f für die Reduktion bereits (fast) die Identität (eventuell unterschiedliche Kodierung beider Probleme beachten!). In unseren Beispielen ist das TSP-Problem geeignet.

Sie können sich eines der folgenden Auslieferungsprobleme aussuchen oder ähnliche weitere selber definieren ;-):

(i) **TSPmitLKW**

Gegeben: Entfernungsmatrix E , Standorte der n Kunden k_1, \dots, k_n , m LKWs L_1, \dots, L_m .

Gesucht: m Rundtouren mit minimaler Gesamtlänge, so dass alle n Kunden beliefert werden.

(ii) **TSPmitLKW+Fahrzeit**

Gegeben: Entfernungsmatrix E , Standorte der n Kunden k_1, \dots, k_n , m LKWs L_1, \dots, L_m .

Gesucht: m Rundtouren mit minimaler Gesamtlänge, so dass alle n Kunden beliefert werden und die Fahrzeit von 8 h nicht überschreitet.

(iii) **TSPmit LKW+Kapazität**

Gegeben: Entfernungsmatrix E , Standorte der n Kunden k_1, \dots, k_n , Auslieferungsgewichte der Pakete der n Kunden g_1, \dots, g_n , m LKWs L_1, \dots, L_m mit Zuladekapazität K_1, \dots, K_m .

Gesucht: m Rundtouren (je 1 pro LKW) mit minimaler Gesamtlänge, so dass alle n Kunden beliefert werden und die Zuladungskapazität bei keinem LKW überschritten wird.

(iv) **TSPmitLKW+Fahrzeit+Kapazität**

Gegeben: Entfernungsmatrix E , Standorte der n Kunden k_1, \dots, k_n , Auslieferungsgewichte der Pakete der n Kunden g_1, \dots, g_n , m LKWs L_1, \dots, L_m mit Zuladekapazität K_1, \dots, K_m .

Gesucht: m Rundtouren (je 1 pro LKW) mit minimaler Gesamtlänge, so dass alle n Kunden beliefert werden und weder die Fahrzeit von 8 h noch die Zuladungskapazität bei keinem LKW überschritten wird.

(v) **Detailed Scheduling**

Gegeben: Produktionsaufträge p_1, \dots, p_n , mit Umrüstzeiten von p_i auf p_j in Rüstmatrix $R[i,j]$, m Maschinen M_1, \dots, M_m mit Schichtzeiten T_1, \dots, T_m .

Gesucht: Reihenfolgeplan auf den m Maschinen mit minimaler Gesamtumrüstzeit unter Einhaltung der Schichtzeiten.

Aufgabe 2 TSP Problem – Varianten der Kodierung

Betrachten Sie folgende 3 Mengen:

1. $A_1 = \{ (n, E, k) \mid \text{es gibt Rundtour mit Länge } \leq k \text{ zu } E = \text{Entfernungsmatrix } [n, n] \text{ of integer} \}$
2. $A_2 = \{ 2 \ n \ 2 \ e_{11} \ 2 \ e_{12} \ 2 \dots 2 \ e_{nn} \ 2 \ k \mid \text{es gibt Rundtour mit Länge } \leq \text{Dualzahl } k \text{ zu } E = \text{Entfernungsmatrix } [e_{11} \ e_{12} \dots e_{nn}] \text{ of Dualzahl} \}$
3. $A_3 = \{ \text{val}(2 \ n \ 2 \ e_{11} \ 2 \ e_{12} \ 2 \dots 2 \ e_{nn} \ 2 \ k) \mid \text{es gibt Rundtour mit Länge } \leq \text{Dualzahl } k \text{ zu } E = \text{Entfernungsmatrix } [e_{11} \ e_{12} \dots e_{nn}] \text{ of Dualzahl} \}$
dabei berechnet val den Wert einer Ternärzahl

Beachten Sie hierbei: $A_2 \subset \{0,1,2\}^*$ und $A_3 \subset \mathbb{N}$

Welche der drei Mengen definiert das TSP Problem??

A) Zeigen Sie, dass die 3 Probleme jeweils aufeinander reduzierbar und deshalb gleich schwer zu lösen sind.

B) Welche der drei folgenden Worte x_i liegt in A_2 ?

$$x_1 = 2 \ 10 \ 2 \ 0 \ 2 \ 1 \ 2 \ 01 \ 2 \ 0 \ 2 \ 10,$$

$$x_2 = 2 \ 10 \ 2 \ 0 \ 2 \ 1 \ 2 \ 1 \ 2 \ 0 \ 2 \ 1,$$

$$x_3 = 2 \ 10 \ 2 \ 1 \ 2 \ 1 \ 2 \ 1 \ 2 \ 101.$$

C) Liegt die Zahl 628_{10} in A_3 ?