Übungen zur Algorithmischen Bioinformatik I

Hausarbeit 3

Komplexität schwerer Probleme und Entscheidbarkeit

Xiheng He

Juni 2021

1. Aufgabe: Das Traveling Salesperson Problem (TSP)

(a) Schreiben Sie ein Programm, das die optimale Tour unter allen möglichen Permutationen bestimmt. Testen Sie das Programm fur die Instanz des TSP aus der Vorlesung.
Usage: -mode [TSP|EPP|NN|MST]

(b) **Hamilton vs Euler:** Was macht das TSP bzw. HC Problem schwer? Begründen Sie! Grenzen Sie es dazu vom Euler-Kreis-Problem (EKP) ab. Wie sieht es fur EKP hinsichtlich Entscheidbarkeit und Komplexität aus?
Das HC (Hamilton Kreis) Problem ist zu entscheiden, ob ein solcher Kreis (geschlossener Pfad) in einem gegebenen Graph existiert, der jeden Knoten genau einmal besucht.
Das EKP (Euler-Kreis-Problem) ist zu bestimmen, ob ein solcher Kreis (geschlossener Pfad) in einem gegebenen Graph existiert, der jede genau einmal enthält.
Das TSP (Traveling Salesperson Problem) besteht darin, eine Reihenfolge für den Besuch mehrerer Orte so zu wählen, dass keine Station außer der ersten mehr als einmal besucht wird, die gesamte Reisestrecke des Handlungsreisenden möglichst kurz und die erste Station gleich der letzten Station ist. TSP kann auch so betrachtet werden, ob einen Hamilton Kreis mit Länge $\leq w$ in einem gegebenen Graph existiert da HC auf TSP leicht reduziert werden kann.
Im Einscheidungsproblem HC wollen wir also feststellen, ob ein gegebener Graph einen einfahcne Kreis auf allen Knoten als Teilgraphen enthält. Deshalb müssen wir alle alle Permutationen der Vertices betrachten die insgesamt $|V|!$ sind. Die schwierigkeit liegt daran, dass kein Algorithmus existiert, der das Problem in polynomieller Laufzeit löst.
Aus **Theorem 5.26** Skript S.257: HC ist $\mathcal{NP} - vollstndig$.
Da wir einen Hamilton Kreis mit höchstens w Gewicht in einem gegebenen Graph bestimmen muss um das TSP zu lösen, zeigt dies, dass TSP mindestens so schwierig ist wie das HC Problem. Deshalb existiert auch kein polynomieller Algorithmus für TSP.
Für das EKP (Euler-Kreis-Problem) kann es aber durch Satz von Euler-Hierholze in polynomieller Zeit bestimmt werden, ob ein Euler-Kreis in einem gegebenen Graph existiert.
Satz von Euler:

- Sei G ein ungerichteter Graph dann sind folgende Aussagen äquivalent:
  - G ist eulersch
  - jeder Knoten in G hat geraden Grad

- Sei G ein gerichteter Graph dann sind folgende Aussagen äquivalent:
  - G ist eulersch
  - für jeden Knoten in G sind Eingangsgrad und Ausgangsgrad gleich

Somit ist das Euler-Kreis-Problem entscheidbar und gehört zur Klasse $\mathcal{P}$.

(c) Schreiben Sie ein Programm, das einen bzw. die Eulerkreise in einem gewichteten Graphen bestimmt. Testen Sie das Programm fur die Instanz des TSP aus der Vorlesung und dem Königsberg-Graphen.

(d) Wenden Sie Ihre Programme (mit Bedacht!) auf realistische Probleminstanzen (siehe Aufgabe 4) an. Definieren Sie ggfs. (wenn die bereitgestellten Instanzen zu schwierig/groß sind) geeignete Teilprobleme, um die praktische Anwendbarkeit Ihrer Implementierungen zu untersuchen.

2. Aufgabe: Approximationen des TSP

Implementieren Sie die (einfachen) Approximationen aus der Vorlesung, die auf nächsten Nachbarn (NN, Greedy) und minimalen Spannbäumen (MST) aufbauen. "Nearest Neighbor" reist immer von der aktuellen Stadt zu nächstgelegenen noch nicht besuchten Stadt. "Greedy" fügt immer die kürzeste noch verfügbare Kante hinzu, die zwei Teilpfade verbindet (also Kurzzyklen vermeidet!). MST meint die "twice-around-the-spanning-tree" Approximation aus der Vorlesung. Können Sie Instanzen angeben, die für die jeweiligen Approximationen ungünstig sind? Sind die angegebenen Approximationgüten scharf (d.h. gibt es Instanzen, die nicht zu besseren Lösungen führen wie die Güteschranke)?

Usage: -mode [TSP|EPP|NN|MST] Greedy sucht immer nur den kürzesten Pfad. Eine ungünstige Instanz:

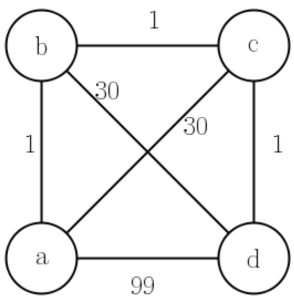

Abbildung 1: Ungünstige Instanz für Greedy

Optimale Tour des TSP ist a, c, d, b, a wobei Der Greedy Algorithmus a, b, c, d, a ausgibt.

Die minimale Spannbäume garantieren nicht die Erzeugung des Hamilton-Rings mit dem kleinsten Gewicht. Und es ist nicht unbedingt, dass die Distanz der jeden Schritt in MST die minimale Distanz im euklidischen Raum ist (d.h. erfüllt die Dreiecksungleichung). Ein gute Bespiele dafür ist die Instanz der TSP aus Vorlesung.

3. Aufgabe: ILP Lösung des TSP

- Diese Ungleichungen definieren gültige TSP Touren (das TSP-Polytop) allerdings werden dafür sehr viele Ungleichungen gebraucht. Wie viele?

  Für "degree constraints" muss für jede Knoten eine Ungleichung paarweise erstellt werden damit ergibt sich insgesamt $n(n-1)/2$ Ungleichungen.

  Für "sub-tour elimination constraints" muss $n-1$ Ungleichungen je ein Knotenpaar erstellt werden damit ergibt sich insgesamt $n(n-1)^2/2$ Ungleichungen.

- Sie verhindern dann auch Kurzzyklen. Wieso?

Gegeben sei die Ungleichung für die Sub-Tour Elimination:

$$t_i - t_j + n \cdot x_{ij} \leq n - 1 \tag{1}$$

Aus (1) ergibt sich:

$$t_i - t_j + n \cdot x_{ij} \leq n - 1 \Longrightarrow t_i - t_j + 1 - n(1 - x_{ij}) \leq 0 \tag{2}$$

$$\forall x_{ij} = 0, t_i \leq n \wedge t_j \geq 0 \Longrightarrow t_i - t_j \leq n - 1 \tag{3}$$

$$\forall x_{ij} = 1, t_i - t_j + 1 - n(1 - x_{ij}) \leq 0 \Longrightarrow t_i - t_j + 1 \leq 0 \tag{4}$$

(3) bedeutet, dass die Sub-Tour Elimination Einschränkung erfüllt wird wenn $x_{ij} = 0$ nämlich Knoten i, j keinen Pfad besitzen.
(4) bedeutet, dass die Sub-Tour Elimination Einschränkung erfüllt wird auch wenn $x_{ij} = 1$ nämlich Knoten i, j einen Pfad besitzen.
Betrachten wir drei Knoten i, j, k. Falls eine Subtour vorhanden ist dann gilt:

$$x_{ij} = x_{jk} = x_{ki} = 1 \tag{5}$$

Einsetzen (5) in (4)

$$\begin{aligned} t_i - t_j + 1 &\leq 0 \\ t_j - t_k + 1 &\leq 0 \\ t_k - t_i + 1 &\leq 0 \end{aligned} \tag{6}$$

Addieren alle Formel aus (6) ergibt sich $\Longrightarrow 3 \leq 0$
Die obige Ungleichung ist offensichtlich nicht wahr, daher existiert die Subtour auch nicht. Somit ist es klar, dass die Ungleichung (1) grantiert dass es keine Subtour in TSP gibt.

4. Aufgabe: Praktische Anwendung und Instanzen des TSP

5. Aufgabe: Anwendung des TSP für Bioinformatikprobleme

Welche Bioinformatikprobleme wurden mittels TSP gelöst bzw. untersucht? Finden und erklären Sie klassische Beispiele! Was sind die zeitlich neuesten Bioinformatik Paper/Publikationen, die das TSP erwähnen und zur Lösung nutzen? Welche neuen Veröffentlichungen gibt es, die versuchen das TSP mit neuen Methoden besser zu lösen?
Sequenzierung, Protein vorhersagen...

Ein klassische Bespiele dafür ist Schrotschuss-Sequenzierung im Sequenzierung. Die Sequenzierung mit dem Schrotschuss-Sequenzierungsverfahren wird in mehrere Phasen eingeteilt:

- Fragmentierung der DNA und Sequenzierung der Fragmente (Fragmentierungs-Phase)
- Feststellung von Überlappungen zwischen den Fragment-Sequenzen (Overlap-Phase)
- Berechnung eines multiplen Alignments der Fragmente (Layout-Phase)
- Ermittlung der Konsensus-Sequenz (Konsensus-Phase)

Nach Fragmentierungs-Phase müssen die Sequenzen wieder richtig zusammengebaut werden. Diese Fragmente stammen alle von der Originalsequenz, daher sollte die Originalsequenz zumindest alle diese Fragmentsequenzen enthalten. Das heißt, die Originalsequenz ist der Superstring dieser fragmentierten Sequenzen und ein "shortest supersequence" muss davon ausgewählt werden. Das Problem kann durch TSP gelöst werden.

Shortest Supersequence Problem:

Gegeben sei eine Buchstabenmenge Menge Σ, eine endliche Stringmenge $A \subseteq \Sigma^*$, findet ein String r, sodass $\forall x \in A$ ein Teilstring in r.

Publikationen und Veröffentlichungen:

Generation of Genetic Maps Using the Travelling Salesman Problem (TSP) Algorithm

https://www.ijser.org/researchpaper/Generation-of-Genetic-Maps-Using-the-Travelling-Salesman-Problem-TSP-Algorithm.pdf

A (Slightly) Improved Approximation Algorithm for Metric TSP

Anna R. Karlin, Nathan Klein, Shayan Oveis Gharan

arXiv.org > cs > arXiv:2007.01409

Diese neue Veröffentlichung biete eine neue Approximation für TSP mit 3/2 ratio