Algorithmen und Datenstrukturen
WS 2025
von Luxburg/Bordt

Übungsblatt 7

Abgabe am 1.12.2025

Aufgabe 1: A* mit verschiedenen Heuristiken (2+1+1 Punkte)

Gegeben sei ein gewichteter Gittergraph $G = (V, E)$ der Größe $n \times n$, wobei $V = \{(i, j) \mid 0 \leq i, j < n\}$ die Menge der Gitterpunkte ist. Jeder Knoten (i, j) ist mit seinen (bis zu 4) direkten Nachbarn $(i \pm 1, j)$ und $(i, j \pm 1)$ verbunden (sofern diese im Gitter liegen). Die Kantengewichte $w : E \to \mathbb{R}_{>0}$ sind positiv. Für zwei Knoten $u = (i_1, j_1)$ und $v = (i_2, j_2)$ definieren wir:

- Manhattan-Distanz: $d_M(u, v) = |i_1 - i_2| + |j_1 - j_2|$
- Gewichtete Manhattan-Heuristik: $h_M(v) = w_{\min} \cdot d_M(v, t)$, wobei $w_{\min} = \min_{e \in E} w(e)$

Aus der Vorlesung kennen Sie die Bedinung für eine zulässige Heuristik h beim A* Suchalgorithmus: Für alle Knoten $v \in V$ gilt: $h(v) \leq d^*(v, t)$, wobei $d^*(v, t)$ die tatsächliche kürzeste Pfad-Distanz von v nach t bezeichnet.

a) Zeigen Sie anhand eines konkreten Beispiels, dass die ungewichtete Manhattan-Distanz $d_M(v, t)$ im Allgemeinen keine zulässige Heuristik für den A* Suchalgorithmus ist. Demonstrieren Sie den Ablauf des Algorithmus an Ihrem Beispiel.

b) Unter welcher Bedingung an die Kantengewichte ist $d_M(v, t)$ dennoch eine zulässige Heuristik? Begründen Sie kurz.

c) Beweisen Sie, dass die gewichtete Manhattan-Heuristik $h_M(v) = w_{\min} \cdot d_M(v, t)$ eine zulässig Heuristik ist.

Aufgabe 2: Zufällige Graphen und Kürzeste-Wege-Statistiken (12 Punkte)

In dieser Aufgabe untersuchen Sie verschiedene Modelle zufälliger Graphen und analysieren deren Struktur mittels Kürzeste-Wege-Algorithmen.

Teil A: Implementierung zufälliger Graphen (4 Punkte)

Implementieren Sie die folgenden drei Modelle für zufällige Graphen mit n Knoten:

1. **Erdős-Rényi Graph $G(n, p)$:** Jede mögliche Kante zwischen zwei Knoten existiert unabhängig mit Wahrscheinlichkeit $p \in [0, 1]$. Alle Kanten erhalten ein zufälliges Gewicht aus dem Intervall $[1, 10]$.
2. **k-Nearest-Neighbor Graph:**
   - Platzieren Sie n Punkte zufällig im d-dimensionalen Einheitswürfel $[0, 1]^d$
   - Verbinden Sie jeden Knoten mit seinen k nächsten Nachbarn (bezüglich euklidischer Distanz)
   - Das Kantengewicht entspricht der euklidischen Distanz zwischen den Punkten
3. **Barabási-Albert Preferential Attachment Graph:**
   - Beginnen Sie mit m_0 vollständig verbundenen Knoten
   - Fügen Sie iterativ neue Knoten hinzu, wobei jeder neue Knoten $m \leq m_0$ Kanten zu existierenden Knoten erhält
   - Die Wahrscheinlichkeit, dass ein neuer Knoten sich mit Knoten i verbindet, ist proportional zum Grad von i: $P(i) = \frac{\deg(i)}{\sum_j \deg(j)}$

- Alle Kanten erhalten ein zufälliges Gewicht aus $[1, 10]$

Teil B: Kürzeste-Wege-Analyse (4 Punkte)

1. Implementieren Sie einen All-Pairs-Shortest-Path Algorithmus (z.B. Floyd-Warshall oder wiederholtes Dijkstra) für gewichtete Graphen.
2. Für jeden Graphtyp und verschiedene Parameterkombinationen:
   - Berechnen Sie alle kürzesten Pfade zwischen allen Knotenpaaren
   - Bestimmen Sie den *Durchmesser* des Graphen (der Durchmesser eines Graphen ist die Länge des längsten kürzesten Weges)
   - Erstellen Sie ein Histogramm der Distanzen aller kürzesten Pfade

Teil C: Visualisierung und Vergleich (4 Punkte)

1. Erstellen Sie für folgende Parameterkombinationen jeweils Visualisierungen:
   - Erdős-Rényi: $n = 100$, $p \in \{0.05, 0.1, 0.2\}$
   - k-NN Graph: $n = 100$, $k \in \{3, 5, 10\}$, $d \in \{2, 3, 5\}$
   - Barabási-Albert: $n = 100$, $m_0 = 5$, $m \in \{2, 3, 5\}$
2. Plotten Sie für jede Konfiguration:
   - Das Histogramm der kürzesten Pfade
   - Optional: Eine Visualisierung des Graphen selbst (für kleine Instanzen)
3. Erstellen Sie eine Vergleichstabelle mit folgenden Metriken für alle Graphtypen:
   - Durchschnittliche kürzeste Pfadlänge
   - Durchmesser
   - Standardabweichung der Pfadlängen
4. Diskutieren Sie Ihre Beobachtungen:
   - Welche charakteristischen Unterschiede zeigen sich in den Distanzverteilungen?
   - Wie verändert sich der Durchmesser in Abhängigkeit von den Parametern?

Hinweise:

- Verwenden Sie z.B. Python mit NetworkX oder einer andere geeigneten Graph-Bibliothek
- Für Visualisierung in Python eignen sich matplotlib oder seaborn
- Achten Sie darauf, dass Ihre Graphen zusammenhängend sind (ggf. nur größte Zusammenhangskomponente betrachten)

Abgabe: Erstellen Sie eine PDF mit den wichtigsten Plots und beschreiben Sie, was in den Plots abgebildet ist. Achten Sie auf eine gute Beschriftung der Achsen und der in den Plots abgebildeten Kurven / Punkten.