

Übungsblatt 5: Genauere Analyse

Ausgegeben am 11.12.2020. Geschätzter Zeitaufwand: 2 Woche

Aufgabe 1: Skalierungsexperimente

Um zu untersuchen, wie die bidirektionale Suche in Abhängigkeit der Graphgröße skaliert, teste sie auf ausreichend vielen unterschiedlich großen generierten Instanzen. Verwendet dazu den Generator¹ für sogenannte GIRGs (geometric inhomogenous random graphs). Der Generator kommt als C++ Library mit Command Line Interface für GIRGs und für Hyperbolische Zufallsgraphen.

Aufgabe 2: Einfluss der Parameter

Das GIRG-Modell hat verschiedene Parameter. Welchen Einfluss haben die Parameter p_{le} und α auf die Performanz der bidirektionalen Suche?

Hinweis 1: p_{le} ist in dem Command Line Interface auf das Intervall $(2, 3]$ beschränkt. Werte die größer als 3 sind, sind aber auch sinnvoll. Um größere Werte zu verwenden musst du ggf. das Command Line Interfaces anpassen, sodass größere Werte nicht mehr abgefangen werden.

Hinweis 2: Der Parameter α ist auf $(1, \infty]$ beschränkt. Wenn du hier verschiedene Werte ausprobierst, dann denk eher über $1/\alpha$ nach. Also: Es ist zunächst hilfreich Werte zu wählen, sodass die zugehörigen $1/\alpha$ halbwegs gleichmäßig in $[0, 1)$ verteilt sind.

Aufgabe 3: Interessante Grapheigenschaften

Kannst du herausfinden, welche Grapheigenschaften von den Parametern p_{le} und α beeinflusst werden?

Schaue dir dazu zum Beispiel Gradverteilungen der generierten Graphen an oder schaue in die Literatur zu GIRGs. Ein guter Startpunkt für letzteres ist ggf. das Papier² zu dem Generator, den du in Aufgabe 1 verwendet hast.

Hinweis: Statt dem Parameter α selbst wird in der Literatur auch oft die sogenannte *Temperatur* $T = 1/\alpha$ verwendet.

¹<https://github.com/chistopher/girgs/>

²<https://arxiv.org/abs/1905.06706>

Aufgabe 4: Echte Graphen

Was sagt uns das jetzt über echte Graphen? Wie verhält sich die Laufzeit der bidirektionalen Suche auf echten Graphen abhängig von den Eigenschaften aus Aufgabe 3?