Extremwertproblem Wegfindung

Vergleich verschiedener Algorithmen

Maximilian Stark

16. Oktober 2015

Inhaltsverzeichnis

1 Einleitung

2 Grundlagen und Terminologie

Zu Beginn werden in diesem Abschnitt die grundlegenden Begriffe der Graphen-Theorie geklärt. Auch Fachbegriffe aus der Implementierung durch die Informatik werden erläutert.
Das Ziel dieser Arbeit ist die Darstellung und der Vergleich verschiedener *Wegfindungs-Algorithmen* in der Anwendung an verschiedenen generierten *Graphen*.

Zugrunde all dem liegt die Graphen-Theorie. Deren Fundament ist der namensgebende *Graph* $G = \{V, E\}$, welcher aus einer Menge von *Knoten* V (von engl. „Vertex") und aus einer Menge *Kanten* E (von engl. „Edge").
Zeichnerisch werden *Knoten* als Punkte oder Kreise dargestellt; *Kanten* als Verbindungslinien zwischen zwei *Knoten*. Jede *Kante* hat einen *Startknoten* und einen *Endknoten*. Wenn von einer *gerichteten Kante* die Rede ist, lässt sich das als Pfeil interpretieren, da die Verbindung unidirektional gilt. Ebenso gibt es die *gewichteten Kanten*, denen nicht nur zwei *Knoten* zugeordnet werden, sondern zusätzlich noch ein Gewicht w (von engl. „Weight"), ein Zahlenwert, der als Kosten der Beziehung zwischen den beiden *Knoten* gesehen werden kann.
In der Wegfindung ist ein *Weg* P (von engl. „Path") als geordnete Abfolge von *Knoten* definiert. Da in der Regel jedes *Knoten*-Paar nur einfach verbunden ist, reicht in der Implementierung dieser Ansatz aus.
Visuell wird der *Graph* durch einen *Layout-Algorithmus* dargestellt, welcher allen *Knoten* durch gewisse Berechnungen Positionen zuteilt. Mehr dazu in Abschnitt 5.
Generell sind *Algorithmen* eine festgelegte Abfolge von Schritten um Daten zu verarbeiten. In der Informatik sind diese einzelnen Schritte Befehle.

3 Aufbau und Bedienung des Programms

Das Programm, im eigentlichen Fokus stehend, fungiert sowohl als visuelle Möglichkeit der Darstellung, als auch als Quelle für Vergleichsdaten und Messungen in selbst erzeugten Szenarien. Geschrieben ist das Programm in der Programmiersprache *Java* unter Verwendung der *JavaFX*-Standardbibliothek [1].

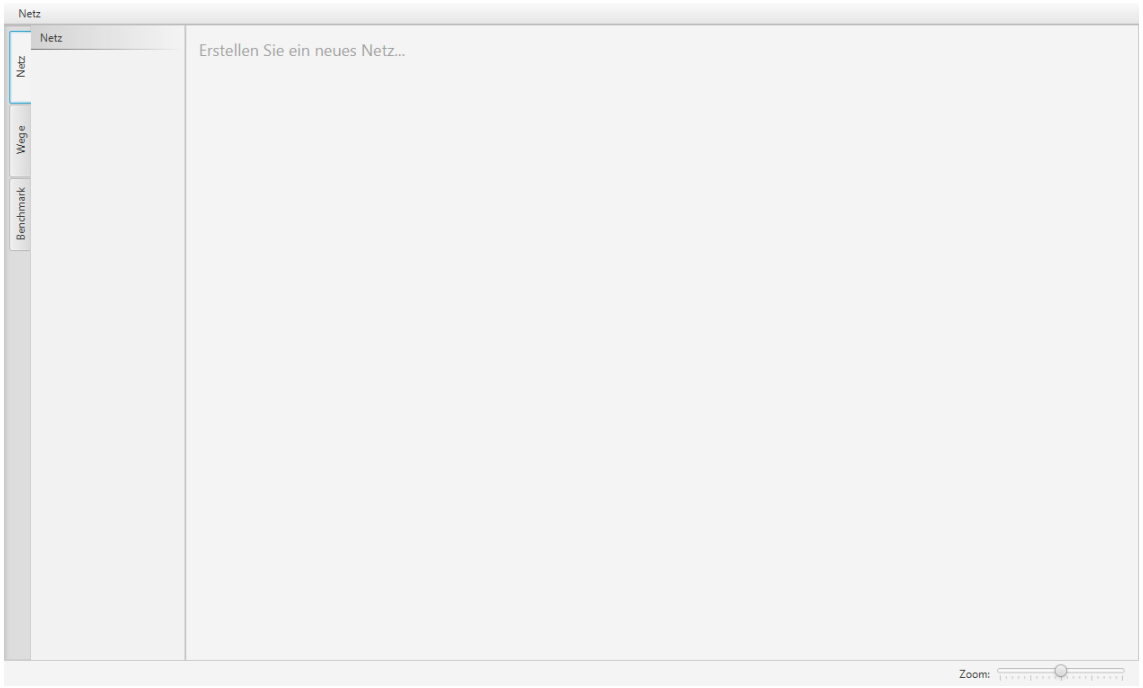

Abbildung 1: Die Start- und Hauptansicht der Applikation

Auf den ersten Blick ist die Anwendungsoberfläche in zwei größere Bereiche aufgeteilt. Im linken, kleineren Seitenbereich werden detaillierte Informationen über den *Graphen*, bereits berechnete *Wege* und die Konfigurationsmöglichkeiten neuer Wege, in mehreren „Tabsünterteilt, angezeigt. Der große rechte Bereich, zu Beginn der Anwendung nur mit „Erstellen Sie ein neues Netz..."(siehe Abb. 1) beschriftet, dient als Hauptansicht von sowohl des Graphen, als auch der Vergleichsstatistiken und Tabellen.

Die Bedienung kann vollständig mit der Maus erfolgen, da sich sämtliche Features visuell intuitiv und minimalistisch präsentieren. Nur vereinzelt führen Tastatureingaben oder „Hotkeysßu mehr Komfort oder Genauigkeit der Anwendung. So kann beispielsweise das *Relayout* (Abschnitt 5) des *Graphen* per „L" Taste, das *Generieren* (Abschnitt 4) eines neuen unter Benutzung von „N".

-

4 Konstruktion des Graphen

Im nächsten Schritt werden wir nun einen Graphen erzeugen lassen und die Funktionsweise des Generators betrachten.

5 Visuelles Layout des Graphen

6 Wegfindungs-Algorithmen

6.1 Gröbste Züge von Intelligenz: Tiefensuche

-

6.2 Heuristik als Mittel zum Ziel: Der Dijkstra-Algorithmus

-

6.3 Der Allstar: Der A*-Algorithmus

-

7 Vergleichsstatistik und Fazit

-

8 Schluss

Literatur

[1] JavaFX-Homepage http://docs.oracle.com/javase/8/javafx/get-started-tutorial/jfx-overview.htm *abgerufen am 14.10.15*