Oberschule
RONZELENSTRASSE
Die Sportbetonte Schule

Oberschule an der RonzelenstraSSe

Mathematik und *Informatik*

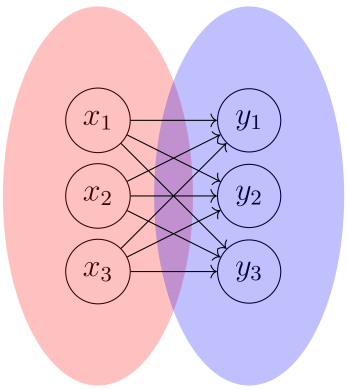
Projektarbeit im Rahmen der Qualifikationsphase

Zeitliche Optimierung der Klausurenplanung an Schulen mithilfe von Graphentheorie eine Web-App-Entwicklung

---

2024/2025

Julius Backes | Tom Kurzke



Betreuende & Prüfende Lehrer:

Hr. V. Wolff (Mathematik LK) & Hr. O. Huras (Informatik GK)

Inhalt

1 Einleitung

1.1 Problemstellung

1.2 Motivation

1.3 Ziel der Arbeit

2 Grundlagen der Graphentheorie

2.1 Grundbegriffe

2.2 Färbungsprobleme

2.3 Komplexität und Lösungsstrategien

In diesem Kapitel werden grundlegende Konzepte der Graphentheorie eingeführt. Diese sind essenziell, um ein tiefes Verständnis von der Funktionsweise der finalen Anwendung zu entwickeln.

Ein Graph aus der Graphentheorie unterscheidet sich signifikant von einem "typischen" Graphen, wie man ihn aus der Schulmathematik oder aus der Analysis kennt.

Ein graphentheoretischer Graph besteht aus einer Sammlung von Knoten (eng. vertices), die je nach Graph durch Kanten (eng. edges) miteinander verbunden sind. Ein Graph G

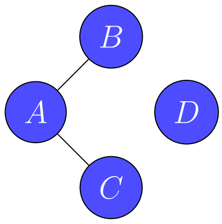


Abb. 1: Ein einfacher ungerichteter Graph

wird als ein Paar $G = (V_G, E_G)$ definiert, wobei V_G eine Menge von Knoten darstellt, z.B. $\{1, 8, 7\}$ oder $\{A, B, C\}$, und E_G eine Menge von Kanten ist, also Verbindungen zwischen je zwei Knoten. Für die Kantenmenge gilt:

$$E_G = \{\{x, y\} \colon x, y \in V \land x \neq y\}$$

Zur einfacheren Darstellung von Graphen verwenden wir folgende Notation:

$$\{\{a, b\}, \{c, d\}\} \Leftrightarrow \{ab, cd\}$$

Demnach wird der Graph in Abbildung 1 1 mit $G = (V, E)$ beschrieben, wobei $V_G = \{A, B, C, D\}$ und $E_G = \{AB, AC\}$ sind.

Bis jetzt haben wir uns nur den ungerichteten Graphen angesehen. Bei einem gerichteten Graphen werden statt Mengen geordnete Paare für die Darstellung von Kanten genutzt.

$$E_G = \{(x, y)\colon\ x, y \in V \wedge x \neq y\}$$

Für das Beispiel in Abb. 2 würde sich dann $G = (V_G, E_G)$ mit $V_G = \{A, B, C, D\}$

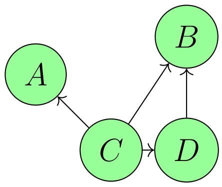


Abb. 2: Ein einfacher gerichteter Graph

und $E_G = \{(C, A), (C, B), (C, D), (D, B)\}$ ergeben. Im Folgenden werden nur noch ungerichtete Graphen von Nöten sein, da gerichtete Graphen nicht nötig für die Anwendung dieser Arbeit sind.

Definition 1. Sei $G = (V_G, E_G)$ ein Graph und $a, b \in V_G$, so nennt man a und b angrenzend gdw. $\{a, b\} \in E_G$ gilt.

Definition 2. Der **Grad** eines Graphen G, ist die Anzahl seiner Knoten.

$$\deg(G) = |V_G|$$

Definition 3. Die **GröSSe** eines Graphen G, ist die Anzahl seiner Kanten; sie ergibt sich aus $|E_G|$.

Definition 4. Als einen **Empty-Graph** bezeichnet man einen Graphen G mit $V_G \neq \{\} \wedge E_G = \{\}$.

Definition 5. Als einen **Null-Graph** bezeichnet man einen Graphen G mit $V_G = \{\} \wedge E_G = \{\}$.

Einen Graphen X mit $V_X = \{\}\ \wedge\ E_X = \{\}$ kann es nicht geben, da $\{a, b\} \in V_X$ gelten muss.

Ein Teilgraph ist ein Graph T eines Graphen G bei dem folgendes gilt:

$$V_T \subseteq V_G \land E_T \subseteq E_G \Rightarrow T \subseteq G$$

3 Klausurenplanung als Optimierungsproblem

3.1 Modellierung

3.2 Zielsetzungen

4 Entwicklung der Web-Applikation

4.1 Anforderungen und Spezifikationen

4.1.1 Funktionale Anforderungen

4.1.2 Nicht-funktionale Anforderungen

4.2 Systemarchitektur

4.2.1 Frontend-Design

4.2.2 Backend-Design

4.2.3 Technologie-Stack

4.3 Implementierung

4.3.1 Algorithmische Umsetzung

4.3.2 Benutzeroberfläche

5 Mathematische Analyse der Optimierung

5.1 Bewertung der Lösungsqualität

5.2 Effizienz und Komplexität

6 Evaluation und Tests

7 Zusammenfassung und Ausblick

7.1 Zusammenfassung

7.2 Ausblick

8 Literaturverzeichnis