Einführung in die Diskrete Mathematik

(Professor Dr. Jens Vygen, Wintersemester 2020/21)

- Vorlesungszeiten: dienstags und donnerstags 16–18 Uhr (c.t.)
- Ort: Gerhard-Konow-Hörsaal (im Arithmeum) oder (wahrscheinlich) auf Zoom
- Übungen: zweistündig nach Vereinbarung (circa fünf Übungsgruppen)
- Webseite: `http://www.or.uni-bonn.de/~vygen/lectures/edmws20.html`
- Ein Vorabtreffen mit aktuellen Informationen und wird Anfang Oktober auf Zoom stattfinden. Details ab Ende September auf der o.g. Webseite bzw. auf e-Campus. Wenn Sie vorher Fragen haben, kontaktieren Sie mich bitte per e-mail.

Voraussetzungen

Die Vorlesung setzt mathematische Grundbegriffe sowie Grundlagen aus *Algorithmische Mathematik I* voraus; insbesondere zu Graphen und elementaren Algorithmen.

Literaturempfehlungen

Die Vorlesung wird größtenteils auf Teilen folgenden Buches basieren, das auch Grundlage weiterer Lehrveranstaltungen der Diskreten Mathematik ist:

- B. Korte, J. Vygen: Kombinatorische Optimierung: Theorie und Algorithmen. Springer, dritte Auflage 2018 (Übersetzung der sechsten englischen Auflage)

Weitere empfehlenswerte Bücher für Teile der Vorlesung:

- R.K. Ahuja, T.L. Magnanti, J.B. Orlin: Network Flows. Prentice-Hall 1993
- W. Cook, W. Cunningham, W. Pulleyblank, A. Schrijver: Combinatorial Optimization. Wiley 1997
- D. Jungnickel: Graphs, Networks and Algorithms. Springer, fourth edition 2013
- A. Schrijver: Combinatorial Optimization: Polyhedra and Efficiency. Springer 2003

Alle genannten Bücher sind z.B. in der Bibliothek des Forschungsinstituts für Diskrete Mathematik (im Arithmeum, Lennéstraße 2) mehrfach vorhanden und auch ausleihbar.

Weitere Lehrveranstaltungen im Bereich C (Diskrete Mathematik):

Lineare und Ganzzahlige Optimierung und *Programmierpraktikum Diskrete Optimierung* jedes Sommersemester. *Kombinatorik, Graphen, Matroide* und *Hauptseminar Diskrete Optimierung* jedes Wintersemester. Weitere Informationen im Internet unter

`http://www.or.uni-bonn.de/teaching.de.html`

Beispiele für Themen

Wir behandeln grundlegende Themen der Diskreten Mathematik, insbesondere algorithmische Probleme. Dazu gehören Eulertouren und Hamiltonkreise, Bäume, Branchings, Netzwerkflüsse, minimale Schnitte, Zusammenhang, kostenminimale Flüsse, bipartites Matching und Anwendungen, Multicommodity flows und disjunkte Wege sowie *NP*-Vollständigkeit.

Eulertouren und Hamiltonkreise

Sei G ein ungerichteter Graph. Ein **Hamiltonkreis** in G ist ein Subgraph, der ein Kreis ist und alle Knoten enthält. Eine **Eulertour** in G ist ein geschlossener Kantenzug, der jede Kante genau einmal enthält. In beiden Fällen durchlaufen wir also den Graphen G, kehren am Ende zum Ausgangspunkt zurück, und besuchen jeden Knoten bzw. jede Kante genau einmal.

Zwei naheliegende Fragen sind, ob ein gegebener Graph eine Eulertour bzw. einen Hamiltonkreis enthält. Die erste Frage ist sehr leicht zu beantworten, die zweite sehr schwierig. Genauer werden wir beweisen:

Satz: (Euler 1736, Hierholzer 1873) Ein ungerichteter Graph enthält genau dann eine Eulertour, wenn er zusammenhängend ist und jeder Knoten geraden Grad hat. Man kann in linearer Zeit eine Eulertour finden oder entscheiden, dass keine existiert.

Satz: (Cook 1971, Karp 1972) Für das Entscheidungsproblem, ob ein gegebener ungerichteter Graph einen Hamiltonkreis besitzt, gibt es genau dann einen polynomiellen Algorithmus, wenn $P = NP$ ist.

Dasselbe gilt für viele andere ("*NP*-vollständige") Probleme, von denen wir einige grundlegende kennen lernen werden. Man vermutet $P \neq NP$, aber diese Frage ist offen.

Netzwerkflüsse

Ein zentrales Problem ist das **Minimum-Cost-Circulation-Problem**: Sei G ein gerichteter Graph mit Kapazitäten $u : E(G) \to \mathbb{R}_{\geq 0}$ und Kosten $c : E(G) \to \mathbb{R}$. Gesucht ist eine Zirkulation $f : E(G) \to \mathbb{R}_{\geq 0}$ in (G, u) (das heißt wir fordern $f(e) \leq u(e)$ für alle $e \in E(G)$ und $\sum_{e \in \delta^+(v)} f(e) = \sum_{e \in \delta^-(v)} f(e)$ für alle $v \in V(G)$), für die $\sum_{e \in E(G)} c(e)f(e)$ minimal ist.

Für dieses Problem wurde 1986 von Éva Tardos der erste streng polynomielle Algorithmus gefunden. Wir serden solche Algorithmen studieren.

Der Spezialfall, in dem $c(e) = 0$ für alle bis auf eine Kante ist, ist äquivalent zum **Maximum-Flow-Problem**. Hierfür gibt es noch effizientere Algorithmen, beispielsweise den von Goldberg und Tarjan (1988).

Netzwerkflüsse spielen eine wichtige Rolle in dieser Vorlesung, in der Diskreten Mathematik allgemein, sowie in zahlreichen Anwendungen. Ein Beispiel aus der Graphentheorie: Ein Graph heißt ***k*-fach zusammenhängend**, wenn er mehr als k Knoten enthält und nach Entfernen von beliebigen $k-1$ Knoten stets ein zusammenhängender Graph übrig bleibt.

Satz: (Whitney 1932) Ein Graph mit mehr als k Knoten ist genau dann k-fach zusammenhängend, wenn es für je zwei Knoten v und w eine Menge von k Wegen von v nach w gibt, die paarweise keinen inneren Knoten gemeinsam haben.