Vorlesung Effiziente Algorithmen und Datenstrukturen 2
Sommersemester 2005
(4-stdg. Vorl. mit 2-stdg. Übung)

Lehrstuhl für Effiziente Algorithmen
Technische Universität München
Dr. Sven Kosub
Dipl.-Inf. Stefan Eckhardt

Inhalt

1. Graphenalgorithmen

1.1 Datenstrukturen für Graphen

1.1.1 Kantenlisten
(Knotenobjekte, Kantenobjekte; Zeitkomplexitäten üblicher Operationen)

1.1.2 Adjazenzlisten
(Knotenobjekte, Kantenobjekte; Zeitkomplexitäten üblicher Operationen)

1.1.3 Adjazenzmatrizen
(Knotenobjekte, Kantenobjekte, Matrixobjekte; Zeitkomplexitäten üblicher Operationen; Platzproblematik)

1.2 Graphentraversierung

1.2.1 Tiefensuche
(Kantenarten; DFS-Algorithmus; Eigenschaften und Komplexität der Tiefensuche; durch Tiefensuche effizient lösbare Probleme)

1.2.2 Breitensuche
(Kantenarten; BFS-Algorithmus; Eigenschaften und Komplexität der Breitensuche; durch Breitensuche effizient lösbare Probleme)

1.3 Erreichbarkeitsprobleme

1.3.1 Zweifach zusammenhängende Teilgraphen
(Definition: Artikulationsknoten, Trennkanten, zweifach zusammenhängender Graph, zweifach zusammenhängende Komponente; Charakterisierung des Zweifach-Zusammenhangs mittels Artikulationsknoten und Trennkanten; Äquivalenzrelation $\equiv$ auf Kantenmenge; Hilfsgraphkonstruktion bzgl. $\equiv$ und Algorithmus mit $O(n+m)$ durch Spannwaldkonstruktion)

1.3.2 Starker Zusammenhang
(Definition: stark zusammenhängend; Tiefensuche in gerichteten Graphen; Linearzeitalgorithmus für Test auf starken Zusammenhang)

1.3.3 Transitive Hülle
(Definition: transitive Hülle; FLOYD-WARSHALL- Algorithmus; Implementierung mit $\Theta(n^3)$)

1.4 Minimale Spannbäume
(Definition: minimaler Spannbaum; „blaue“ und „rote“ Regel)

1.4.1 Algorithmus von Kruskal

(Idee und Algorithmus; Komplexität: $O(m \log n)$)

1.4.2 Algorithmus von Prim und Jarnik

(Idee und Algorithmus; Komplexität: $O(n \log n + m)$; zweiter Algorithmus von Prim und Jarnik mit Komplexität $O(m \log^* n)$)

1.4.3 Algorithmus von Boruvka

(Idee und Algorithmus; Komplexität: $O(m \log n)$)

1.4.4 Algorithmus von Chazelle

(Definition: kontrahierbare Knotenmenge, Kontraktion, (zulässiger) Minor, Minorenhierarchie (mit Beispiel); Postorder-Traversierung der gegebenen Minorenhierarchie; Komplexitätsanalyse: $O(m \cdot \alpha(m, n))$ mit Vorberechnungshypothese; Berechnung der Minorenhierarchie ohne Soft Heaps; Berechnung der Minorenhierarchie mit Soft Heaps; Definition: aktiver Pfad, Schnittkante, kritsche Kante; Invarianten für Soft Heaps; Operationen Retraction und Extension; Gesamtalgorithmus; Korrektheitsbeweis mit Hilfe der starken Kontrahierbarkeit; Komplexitätsanalyse der Berechnung der Minorenhierarchie: $O(m + d^3 \cdot n)$

2. Algorithmen für Flüsse und Matchings

2.1 Flüsse und Schnitte

(Definition: Flussnetzwerk, Quelle, Senke, Kapazität, Fluss, maximaler Fluss, Schnitt, Vorwärts-, Rückwärtskante bzgl. Schnitt, Fluss über Schnitt; Betrag eines Fluss entspricht Fluss über beliebigen Schnitt; Fluss über Schnitt ist höchstens Schnittkapazität)

2.2 Maximaler Fluss

(Definition: Restkapazitäten, augmentierender Pfad; Flussanpassung entlang augmentierender Pfade; Max-Flow, Min-Cut Theorem)

2.2.1 Algorithmus von Ford und Fulkerson

(Algorithmus und Implementierung; Komplexität: $O(|f^*| \cdot m)$; kritisches Beispiel)

2.2.2 Algorithmus von Edmonds und Karp

(Definition: Restdistanz; Monotonie der Restdistanzen bei Augmentierung entlang Pfade minimaler Länge; Schranke für Anzahl der Augmentierungen: $n \cdot m$; Gesamtkomplexität: $O(n \cdot m^2)$)

2.3 Maximale Matchings

(Definition: bipartiter Graph, Matching, maximales Matching; äquivalente Formulierung als Flussproblem; Komplexität von Ford-Fulkerson: $O(n \cdot m)$)

2.4 Flüsse mit minimalen Kosten

(Definition: Kosten eines Flusses, Fluss mit minimalen Kosten, augmentierender Kreis; Kostenanpassung entlang augmentierender Kreise; Charakterisierung von Flüssen mit minimalen Kosten durch Nichtexistenz augmentierender Kreise mit negativen Kosten; Algorithmus basierend auf Ford-Fulkerson und Bellmann-Ford; Komplexität: $O((|f^*| + w^* \cdot n) \cdot m)$; Algorithmus basierend auf iterativer Augmentierung von Pfaden mit minimalen Kosten; Korrektheitsbeweis; Komplexität: $O(|f^*| \cdot n \cdot m)$; Gewichtsrekalibrierung; Algorithmus MinCostFlow; Komplexität: $O(|f^*| \cdot (n \log n + m))$

3. Lineare Programmierung
(einführendes Beispiel)

3.1 Standardformulierungen
(Definition: LP, Dimension, Zielfunktion, Nebenbedingungen, Nichtnegativitätsbedingungen, zulässige und unzulässige Lösung, Zielwert, optimale Lösung, optimaler Zielwert, unerfüllbar und erfüllbar, unbeschränkt und beschränkt; LP-Variationen; Äquivalenz von LP-Formulierungen; Definition: Schlupfvariable, Basisvariable, Nichtbasisvariable, Simplexnormalform; Existenz und Eindeutigkeit der Simplexnormalform; Beispiele für LP- Formulierungen: Kürzeste Wege, Maximaler Fluss)

3.2 Simplex-Algorithmus

3.2.1 Grundlagen
(Geometrische Interpretation; Idee und Beispiel für Simplexalgorithmus)

3.2.2 Formaler Simplexalgorithmus
(Pivotisierung; Simplexalgorithmus ohne Initialisierung)

3.2.3 Terminierung
(Zulässigkeit der gefundenen Lösung; Degenerizität und Zirkulierungen: höchstens $\binom{n+m}{m}$ Vertauschungen; Regel von BLAND)

3.2.4 Optimalität und Dualität
(Definition: duales LP, primales LP; Zielwertungleichung für zulässige Lösungen; hinreichende Bedingung für optimale Lösungen für duales und primales LP; Optimalität der vom Simplexalgorithmus berechneten Lösungen mittels des dualen LP)

3.2.5 Initialisierung
(Hilfs-LP; Charaktierisierung der Erfüllbarkeit von LP; Algorithmus INITIALIZESIMPLEX; Korrektheit; Fundamentalsatz der linearen Programmierung)

Literatur

[Cha01] BERNARD CHAZELLE. A Minimum Spanning Tree Algorithm with Inverse Ackermann-Type Complexity. *Journal of the ACM*, 47(6):1028-1047, 2001.

[CLRS01] THOMAS H. CORMEN, CHARLES E. LEISERSON, RONALD L. RIVEST und CLIFFORD STEIN. *Introduction to Algorithms.* 2. Auflage, MIT Press, Cambirdge, MA, 2001.

[FT87] MICHAEL L. FREDMAN and ROBERT ENDRE TARJAN. Fibonacci Heaps and Their Use in improved Network Optimization Algorithms. *Journal of the ACM*, 34(3):596–615, 1987.

[GT021] MICHAEL T. GOODRICH und ROBERTO TAMASSIA. *Algorithm Design: Foundations, Analysis, and Internet Examples.* John Wiley & Sons, Inc., 2002.

[Sch01] UWE SCHÖNING. *Algorithmik.* Spektrum Akademischer Verlag, Heidelberg, 2001.

[Sch03] ALEXANDER SCHRIJVER. *Combinatorial Optimization: Polyhedra and Efficiency.* Band 24 der Reihe *Algorithms and Combinatorics*, Springer-Verlag, Berlin, 2003.

[Tar83] ROBERT ENDRE TARJAN. *Data Structures and Network Algorithms.* CBMS-NSF Regional Conference Series in Applied Mathematics, SIAM, Philadelphia, PA, 1983.

Literaturhinweise

zu Abschnitt 1.1 bis 1.3: [GT021, S. 287–328].
zu Abschnitt 1.4 (vor 1.4.1): [Tar83, S. 71–72].
zu Abschnitt 1.4.1 bis 1.4.3: [GT021, S. 362–372]; für die Algorithmenvarianten in Abschnitt 1.4.2 siehe [FT87].
zu Abschnitt 1.4.4: [Cha01].
zu Abschnitt 2: [GT021, S. 381–414]; zur Begriffsbildung siehe auch [Sch01, S. 215–228]; für die Beweise in Abschnitt 2.4 siehe auch z.B. [Sch03, Band A, S. 177–179].
zu Abschnitt 3: [CLRS01, S. 770–821].