FSGU®
AKADEMIE

Problemlösen in grafischen Strukturen

Modul 31801 - Zusammenfassung

Demo Version

Impressum

Herausgeber:

FSGU® AKADEMIE - Ein Unternehmen der FSGU® GmbH
Erlenweg 1
D-77948 Friesenheim
info@fernstudium-guide.de | www.fernstudium-guide.de

Dieses Werk ist urheberrechtlich geschützt.
Jegliche unzulässige Form der Entnahme, des Nachdrucks, der Vervielfältigung, Veröffentlichung oder sonstiger Verwertung ist untersagt und wird strafrechtlich verfolgt.

Alle Rechte vorbehalten. © FSGU AKADEMIE

Hinweis des Verfassers

Gerade in den B-Modulen werden an der FernUni Hagen wenige oder gar keine Hilfen zur Bearbeitung und Vorbereitung zur Klausur zur Verfügung gestellt. Die mentorielle Betreuung nimmt stark ab.

Im Falle von Problemlösen in graphischen Strukturen – Modul 31801 gibt es kaum verständliche Unterlagen die einem helfen, schnell in den Stoff hereinzukommen, und mit denen man dann das Vorlesungsheft verstehen kann.

Diese Arbeitsunterlage kann natürlich nicht die Kursmaterialien ersetzen, aber hilft sie doch sehr, die Kursmaterialien zu verstehen, und zeigt einfach auf welche Aufgaben es gibt, und wie kann man diese im Sinne der Klausur bearbeiten und lösen kann. Das Skript wird als Türöffner verstanden um die Aufgaben des Institutes besser bearbeiten zu können. Zu jedem Aufgabentyp gibt es Hinweise, die man sich sonst sehr mühsam erarbeiten müsste, und man spart dadurch sehr viel Zeit.

Diese Zusammenstellung von Kochrezepten soll helfen sich besser und schneller auf die Klausur vor zu bereiten. Einige der hier angesprochenen Themen lassen sich auch mit Hilfe des Internets kaum finden und erfordern einen großen Zeitaufwand um die Verfahren zu verstehen.

Inhaltsverzeichnis

1. Graphen Grundlagen zur Lösung der Ersten Aufgabe 6
1.1. Der gerichtete Graph 6
1.2. Eigenschaften von Graphen 7
1.3. Adjazenzmatrix 8
1.4. Inzidenzmatrix 8
1.5. Richtungen in Graphen 9
1.6. Der bipartite Graph 9
2. Der Kruskal und Prim Algorithmus 11
3. Das Bellmann Verfahren 13
4. Der Dijkstra Algorithmus 18
4.1. Dijkstra mit Tabelle 21
5. Der Ford Fulkerson Algorithmus 24
5.1. Der Rückwärtsfluss 24
5.2. Möglichkeit A über einzelne Wege zu gehen 26
5.3. Möglichkeit B: Alle Wege ausfüllen und dann einen auswählen 29
5.4. Vorschlag zur Bearbeitung des Ford Fulkerson Algorithmus 32
5.5. Minimaler Schnitt 32
5.6. Kapazität 33
6. Der Add Algorithmus 34
7. Die Spalten Minimum Methode 39
8. Die Vogel-Methode 43
9. Die Nordwesteckenregel Methode 49
9.1. Grundlagen der Transportprobleme 51
9.2. Lösungsmethoden für die Transportprobleme 52
10. Der Stepping Stone Algorithmus 53
11. Die Out of Kilter Tabelle 57
12. Der Tripel (Fadenkreuz) Algorithmus 62
13. Die Ungarische Methode 68
14. Die Tiefen- und Breitensuche in Graphen 74
14.1. Die Tiefensuche „First“ in „First Out“ 74
14.2. Breitensuche (merken in einer Liste) 76
15. Das Simulated Annealing (Abkühlen mit e-Funktion) 78

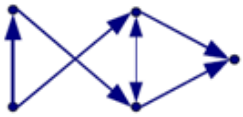
16. Der Genetische Algorithmus **83**
16.1. Der Ein Punkt Cross Over Algorithmus 83
16.2. Der Zwei Punkt Cross Over Algorithmus 84
16.3. Ordnungsbasiertes Cross Over Vertauschen 84
16.4. Inversion Cross Over Algorithmus 85
16.5. N Punkt Cross Over 85
16.6. Uniformes Cross Over 85
16.7. Beispiel Lackieren von Autos 86
17. Das A* Verfahren **88**
18. Das Tabu Search (TS) Verfahren **89**
18.1. Rucksack Tabu 89
18.2. Mathematisch beschriebener Tabu Search 89
18.3. Nachbarschaft Probleme 90
18.4. Tabu To und From Attribute 90
18.5. Darstellung der Nachbarn in einer Tabelle 91
18.6. Die Tabu Liste 91

1. Graphen Grundlagen zur Lösung der Ersten Aufgabe

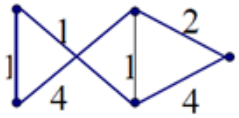
1.1. Der gerichtete Graph

Im gerichteten Graph entspricht N (Knoten = Node) die Menge der Knoten und E (Kanten = Edges) die Menge der Kanten, die die Knoten verbindet.

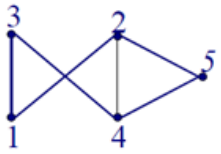
- Das Bild zeigt einen gerichteten Graphen mit gerichteten Kanten (Pfeile die die einzelnen Knoten verbinden)



- Das Bild zeigt einen gewichteten Graphen mit gewichteten Kanten (Zahlen auf den Kanten)

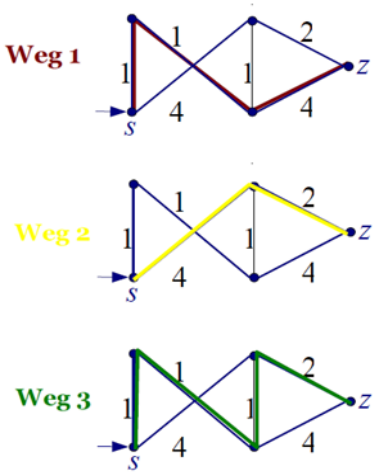


- Das Bild zeigt einen benannten Graphen mit benannten Knoten (die Knoten haben Namen)



- Kurze Wege Graphen. Oft gibt es mehrere Wege zum Ziel. Gesucht ist der kürzeste Weg vom Startknoten „S“ zum Ziel „Z“.

  Weg 1 kostet 6 Einheiten
  Weg 2 kostet 6 Einheiten und
  Weg 3 kostet 5 Einheiten



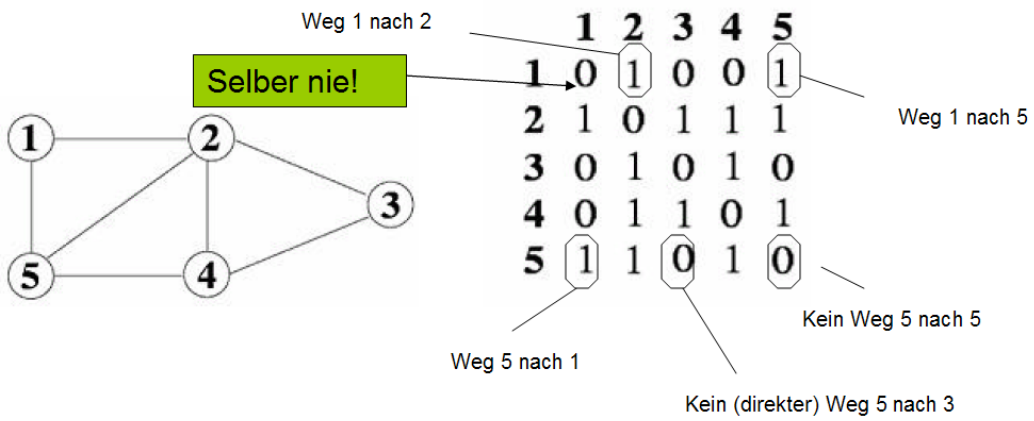
1.2. Eigenschaften von Graphen

Die Eigenschaften von Graphen werden gerne in einer Multiple Choice Matrix abgefragt. Diese Grundlagen helfen einem bei der Beantwortung.

- Wir sprechen von adjazenten Knoten v und v' wenn die Kante (v,v') existiert.
- Der Grad eines Knoten entspricht der Anzahl der Kanten, die diese Knoten mit anderen Knoten verbindet.
- Der Weg in einem Graphen ist eine Liste aufeinanderfolgender Knoten, die über Kanten verbunden werden.
- Ein Graph heißt zusammenhängend, wenn von jedem Knoten zu jedem anderen Knoten ein Weg existiert.
- Ein nicht zusammenhängender Graph besteht aus mehreren Komponenten. Ein zusammenhängender Graph besteht genau aus einer Komponente.
- Ein Weg heißt einfach, wenn jeder Knoten einmal besucht wird.
- Die Weglänge ist die Anzahl der Kanten auf diesem Weg.
- Ein Zyklus ist ein einfacher Weg, bei dem Anfangs- und Endknoten identisch sind.
- Wenn zu jedem Paar von Knoten v und v' ein Weg v nach v* (und von v' nach v für gerichtete Graphen) existiert, dann handelt es sich um einen stark zusammenhängenden Graphen.
- Ein Graph ohne Zyklen ist ein Baum.
- Ein aufgespannter Baum ist ein Teil eines Graphen, der alle Knoten enthält, aber nur so viele Kanten hat, dass er einen Baum bildet.
- Ein Graph, bei dem für alle Knoten v und v' eine Kante existiert heißt vollständig. Ein solcher Graph enthält V*(V-1)/2 Kanten.
- Ein Graph dem nur wenige Kanten zu einem vollständigen Graphen fehlen heißt dicht.
- Ein Graph der nur wenige Kanten besitzt wird als lichter Graph bezeichnet.

1.3. Adjazenzmatrix

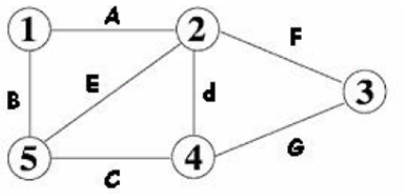
Eine Adjazenzmatrix ist eine Bollsche (n x n) Matrix. A=(aij). Die Einträge aij beschreiben die Kanten von einem Knoten i zu einem Knoten j. Besteht eine Verbindung zwischen den Knoten i und j, so ist aij gleich 1, sonst 0. Handelt es sich bei dem Graphen dabei um einen ungerichteten Graphen, so ist die Matrix unsymmetrisch.



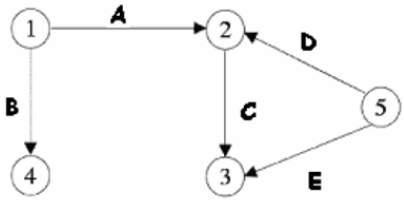
1 für Verknüpfung von Knoten zu Knoten sonst 0, bei Wichtung erhöht sich die 1

1.4. Inzidenzmatrix

Bei einer Inzidenzmatrix A=(aij) wird jeder Knoten durch eine Zeile und jede Kante durch eine Spalte beschreiben. Wenn die Kante j den Knoten i verlässt, ist $a_{ij} = -1$. Wenn die Kante j auf den Knoten i zeigt, ist $a_{ij}=1$, ansonsten $a_{ij}=0$.



	A	B	C	D	E	F	G
1	1	1	0	0	0	0	0
2	1	0	0	1	1	1	0
3	0	0	0	0	0	1	1
4	0	0	1	1	0	0	1
5	0	1	1	0	1	0	0



	A	B	C	D	E
1	-1	-1	0	0	0
2	1	0	-1	1	0
3	0	0	1	0	1
4	0	1	0	0	0
5	0	0	0	-1	-1

Zeigt auf Knoten 3

Geht von Knoten 5 weg

Zeigt auf Knoten 4

Tabelle

Knoten durch Zeile beschrieben, Kante durch Spalte beschrieben

Zeigt Pfeil auf Knoten +1, , geht Pfeil von Knoten weg -1

1.5. Richtungen in Graphen

Der ungerichtete Graph

In einem ungerichteten Graphen ist jede Kante (v1, v2) ein ungeordnetes Paar. Die Kanten verbinden die Knoten ohne eine Richtung vorzuschreiben.
In einer Wanderkarte sind die Knoten Ausgangs- und Zielpunkte, Gipfel und Hütten, und die Kanten sind die Wege.

Der gerichtete Graph

In einem gerichteten Graphen ist jede Kante v1, v2 ein geordnetes Paar unterschiedlicher Knoten v1 und v2, die diese beiden Knoten in einer bestimmten Richtung verbindet. Zum Beispiel ein Produktionsplan (Abfolge von Produktionsschritten). Ein Sonderfall ist der Baum.

Merke: Der gerichtet Graph hat einen Pfeil, der ungerichtet Graph hat keinen Pfeil, kann also in beide Richtungen durchlaufen werden.

Der gewichtete Graph

In einem gewichteten Graphen werden den Kanten Werte zugeordnet. Diese Werte stellen je nach Anwendung Kosten, Zeitdauern oder Entfernungen dar. Diese Werte stehen auf den Kanten.

1.6. Der bipartite Graph

Bipartite Graphen

Ein Graph wird bipartit genannt, wenn seine Ecken in zwei Teilmengen M und N aufgeteilt werden können, so dass er nur Kanten gibt zwischen einer Ecke M und einer Ecke N. Beispiel:

- Tanzpaare
- Arbeitskräfte und deren Aufgaben
- Party wer bringt was zu Essen mit