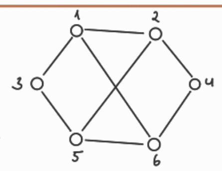
1. Der Netzsimplex-Algorithmus

1.1. Grundlagen der Grafentheorie

Definition 1.1

Sei V eine endliche, nicht-leere Menge und E eine Menge von zweielementigen Teilmengen von V. Dann heißt das Tupel $G=(V,E)$ ungerichteter Teilgraph (oder auch nur Graph). Die Elemente von V heißen Knoten, Ecken oder Punkte, die Elemente von E Kanten. Für die Knoten- bzw. Kantenmenge eines Graphen G schreiben wir auch $V(G)$ bzw. $E(G)$.

Beispiel



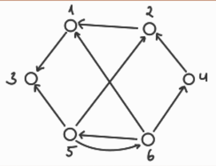
$V:=\{1,2,3,4,5,6\}$

$E:=\{\{1,2\},\{1,3\},\{1,6\},\{2,4\},\{2,5\},\{3,5\},\{4,5\},\{5,6\}\}$

Definition 1.2

Sei V eine endliche nicht-leere Menge und $A\subseteq\{(i,j)\in V\times V : i\neq j\}$ eine (endliche) Menge. Dann heißt das Tupel $D=(V,A)$ gerichteter Graph oder auch Digraph. Die Elemente von V heißen Knoten, die Elemente von A Bögen. Für die Bogenmenge eines Digraphen schreiben wir auch $A(G)$

Beispiel



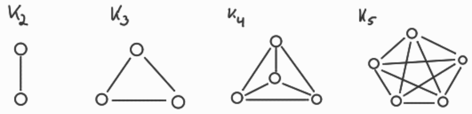
$V:=\{1,2,3,4,5,6\}$

$A:=\{(1,3),(2,1),(4,2),(5,3),(5,2),(5,6),(6,1),$
$(6,4),(6,5)\}$

Def 1.3

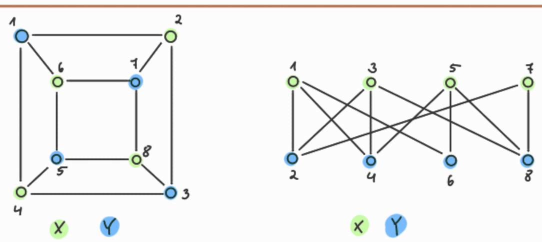
Der vollständige Graph K_n hat als Kanten alle 2-elementigen Teilmengen von $V:=\{1,\dots,n\}$

Beispiel



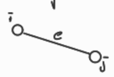
Allgemein ist ein bipartiter Graph (auch paarer Graph) ein Graph $G=(V,E)$, dessen Knotenmenge V in zwei nicht-leere Untermengen X,Y geteilt werden kann, das heißt $X\cup Y=V$ und $X\cap Y=\emptyset$, so dass jede Kante e von der Form $e=\{x,y\}$ mit $x\in X$ und $y\in Y$ ist. Die Zerlegung (X,Y) heißt Zweiteilung oder Bipartition.

Beispiel

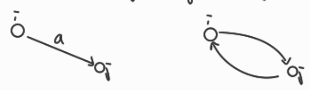


Definition 1.4

Für eine Kante $e=\{i,j\}$ eines Graphen heißen i,j Endpunkte von e. Man sagt i,j sind mit e inzident und i,j sind adjazent oder i,j sind Nachbarn.



Für einen Bogen $a=(i,j)$ eines Digraphen heißt i Anfangsknoten und j Endknoten. i,j sind mit a inzident. i ist Vorgängerknoten von j und j ist Nachfolgerknoten von i. Die Bögen (i,j) und (j,i) heißen gegenläufig oder antiparallel



Definition 1.5

Für einen Knoten i eines Graphen ist der Grad $\deg(i)$ definiert, als die Anzahl der mit i inzidenten Kanten.

Für einen Knoten i eines Digraphen definieren wir den Ausgangsgrad $\deg_{out}(i)$ als die Anzahl der Kanten, die mit i inzident sind und i als Anfangsknoten haben, sowie den Eingangsgrad $\deg_{in}(i)$ als Anzahl der Knoten, die mit i inzident sind und i als Endknoten haben. Der Grad des Knoten ist dann definiert als $\deg(i):=\deg_{out}(i)+\deg_{in}(i)$.

Lemma 1.6

Die Anzahl der Knoten ungeraden Grades in einem (Di-) Graphen ist gerade.

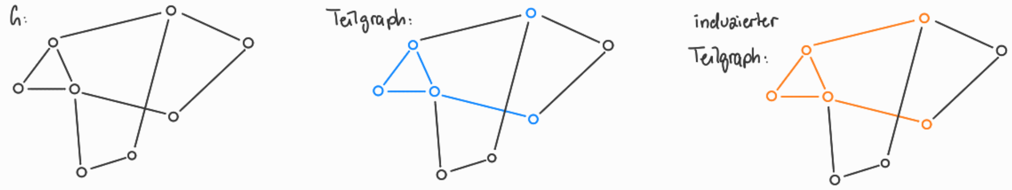
Beweis: Summiert man $\deg(i)$ über alle Knoten, so kommt jede Kante genau zweimal vor. Damit gilt: $\sum_{i\in V}\deg(i)=2\cdot|E|$.

Rechts steht eine gerade Zahl, also ist links die Anzahl ungerader Summanden gerade. $\square$

Def 1.7 Gilt $\deg(i) = \deg(j)$ für alle Knoten $i, j \in V$, so heißt der Graph regulär. Gilt $\deg(i) = k$ $\forall i \in V$, so heißt der Graph k-regulär.

Def 1.8 Seien G und T Graphen. T ist ein Teilgraph (Untergraph, Subgraph) von G, wenn $V(T) \subseteq V(G)$ und $E(T) \subseteq E(G)$. In diesem Fall ist G Obergraph von T. T ist induzierter Teilgraph von G, wenn T Teilgraph von G ist und außerdem gilt: $E(T) = \{\{i,j\} \in E(G) : i, j \in V(T)\}$

Beispiel



Definition 1.9 Sei $(e_1, \dots, e_n)$ eine Sequenz von Kanten in einem Graphen G. Wenn es Knoten $v_0, \dots, v_n$ gibt mit $e_i = \{v_{i-1}, v_i\}$ für alle $i = 1, \dots, n$, so heißt die Sequenz Kantenzug. Im Falle von $v_0 = v_n$ spricht man von einem geschlossenen Kantenzug. Sind die e_i paarweise verschieden, liegt ein Weg (oder Pfad) vor. Ein geschlossener Weg ist ein Kreis.

Ein Weg ist einfach, wenn die v_i paarweise verschieden sind. Ein Kreis ist einfach, wenn die v_i verschieden sind mit Ausnahme von $v_0 = v_n$.

Ein Zyklus ist ein einfacher Kreis. n wird als Länge (des Kantenzugs, des Wegs oder des Kreises) bezeichnet. Ein (un-)gerader Kreis ist ein Kreis, dessen Länge eine (un-)gerade Zahl ist.

Die Knoten v_0, v_n heißen Anfangs- bzw. Endknoten (falls $v_0 \neq v_n$). Ein Graph, der keine Kreise enthält, ist kreisfrei.

<u>Bemerkung</u>: Eine Kantensequenz ist eindeutig abbildbar auf eine entsprechende Knotensequenz.

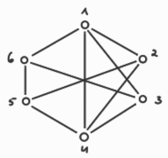
Beispiel

a) $(6, 1, 2, 4, 1, 2)$ – Kantenzug, aber kein Weg

b) $(6, 1, 2, 4, 1, 3)$ – nicht-einfacher Weg

c) $(6, 1, 2, 4, 1, 3, 6)$ – nicht-einfacher Kreis

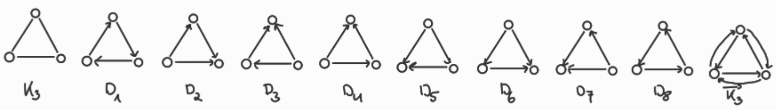
d) $(6, 1, 2, 5, 4, 3, 6)$ – einfacher Weg (Zyklus)



Definition 1.10 Sei $D = (V, A)$ ein Digraph. Setzt man $E := \{\{i,j\} : (i,j) \in A$ und $(j,i) \in A\}$, so bezeichnet man $|D| := (V, E)$ als den zugehörigen Graphen.

Sei $G = (V, E)$ ein Graph. Jeder Digraph D mit $|D| = G$ wird Orientierung von G genannt. Setzt man $A := \{(i,j), (j,i) : \{i,j\} \in E\}$, so ist $\vec{G} := (V, A)$ der zugehörige Digraph, auch vollständige Orientierung von G genannt. Die vollständige Orientierung des K_n wird vollständiger Digraph auf n Knoten genannt.

Beispiel



Definition 1.11 Eine Sequenz von Bögen $(a_1, \dots, a_n)$ in einem Digraph $D = (V, A)$ heißt Kantenzug, (einfacher) Weg oder (einfacher) Kreis, wenn die entsprechende Sequenz in $|D|$ die jeweilige Eigenschaft hat. Ist $(v_0, v_1, \dots, v_n)$ die zugehörige Punktfolge, so ist entweder $a_i = (v_{i-1}, v_i)$ oder $a_i = (v_i, v_{i-1})$. Im ersten Fall spricht man von einer Vorwärtskante, im zweiten von einer Rückwärtskante. Sind alle Kanten des Weges nur Vorwärts- bzw. nur Rückwärtskanten, so ist es ein vorwärts- bzw. rückwärtsgerichteter Kantenzug (oder Weg) bzw. gerichteter Kreis.

Bsp Die Kantenzüge in $D_2, D_3, D_4, D_5, D_6, D_7$ sind einfache Kreise, die Kantenzüge in D_1, D_8 sind einfache gerichtete Kreise.

Definition 1.12 Ein Knoten i eines Graphen heißt verbindbar mit einem Knoten j, wenn es einen Weg im Graphen gibt, der i, j als Endknoten hat.

Ein Graph ist zusammenhängend, wenn je zwei seiner Knoten verbindbar sind. Für einen Knoten i eines Graphen bezeichnet $C(i)$ die Menge aller Knoten, die mit i verbindbar sind. Der durch $C(i)$ induzierte Untergraph heißt Zusammenhangskomponente von i.

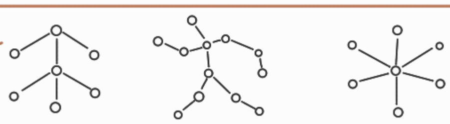
Definition 1.13 Zwei Knoten i, j eines Digraphen heißen verbindbar, wenn es einen vorwärtsgerichteten Weg im Digraphen mit Anfangsknoten i und Endknoten j gibt.

Ein Digraph heißt (schwach) zusammenhängend, wenn Graph $|D|$ zusammenhängend ist. Digraph D heißt stark zusammenhängend, wenn je zwei seiner Knoten verbindbar.

Bsp Die Digraphen $D_2, \dots, D_7$ sind schwach zusammenhängend, D_1 und D_8 sind stark zusammenhängend.

Def 1.14 Ein Baum ist ein zusammenhängender Graph, der keine Kreise enthält.

Beispiel



Satz 1.15 (Charakterisierung von Bäumen)

Sei $G=(V,E)$ ein Graph. Die Folgende Aussagen sind äquivalent für G:

i) G ist ein Baum.

ii) Pfade in G sind eindeutig, d.h. für je zwei Knoten $x,y \in V$ gibt es genau einen Pfad von x nach y.

iii) G ist ein minimaler zusammenhängender Graph, d.h. ist zusammenhängend, aber die Löschung einer beliebigen Kante führt zu einem unzusammenhängenden Graphen.

iv) G ist ein maximaler Graph ohne Kreise, d.h. G enthält keinen Kreis und jeder Graph, der aus G durch Hinzufügen einer weiteren Kante entsteht, enthält einen Kreis.

v) G ist zusammenhängen und es gilt $|V|=|E|+1$.

Beweis: Induktionsbeweis über die Knotenanzahl mittels Baumwachstumslemma (Lemma 19!).

Induktionsanfang: bemerke, dass alle Äquivalenzen für Graph mit 1 Knoten wahr sind. Sei daher G ein Baum mit mindestens 2 Knoten. Dann hat G mindestens 2 Endknoten (Endknotenlemma); sei v einer davon und sei v' ein Nachbar.

Induktionsannahme: Die Aussage ist wahr für $G-v$.

Die Aussagen i) => ii), iii), iv), v) sind dann offensichtlich wahr.

ii) => i): G ist zusammenhängend (es existiert ein Weg zwischen allen Knotenpaaren). G enthält keinen Kreis, da es sonst Knoten x,y mit zwei verschiedenen Wegen gäbe.

iii) => i): G ist zusammenhängend (in iii) gefordert). G ist kreisfrei, da es sonst eine Kante $\{x,y\}$ gibt, ohne die der Graph immer noch zusammenhängend wäre.

iv) => i): G ist kreisfrei (in iv) gefordert). G ist zusammenhängend, da für alle $\{x,y\} \notin E$ der Graph G mit $\{x,y\}$ einen Kreis enthält. Löscht man diese Kante, bleibt ein Pfad von x nach y.

v) => i): Induktion über Knotenanzahl.

Sei G zusammenhängend mit $|V|=|E|+1 \geq 2$. Die Summe aller Knotengrade ist dann $2 \cdot |V|-2$. Also kann nicht jeder Knoten Grad 2 haben. Der Grad jedes Knotens ist aber mindesten 1. Somit gibt es einen Knoten v, dessen Grad genau 1 ist, d.h. v ist Endknoten.

Der Graph $G'=G-v$ ist auch zusammenhängend und erfüllt $|V(G')|=|E(G')|+1$. Nach Induktionsannahme ist G' daher ein Baum. Folglich ist auch G ein Baum (nach Lemma 19). □

Def 1.17 Ein Knoten mit Grad 1 in einem Graphen heißt Endknoten oder Blatt von G

Lemma 1.18 (Endknotenlemma)

Ein Baum T mit mindestens 2 Knoten enthält mindesten 2 Endknoten.

Beweis: Sei $P=(v_0,v_1,\dots,v_k) \triangleq (e_1,\dots,e_k)$ ein Pfad maximaler Länge in T. Die Länge von P ist mindestens 1, also ist $v_0 \neq v_k$.

Behauptung: v_0 und v_k sind beides Endknoten.

Denn ist (o.B.d.A) v_0 kein Endknoten, dann gibt es eine Kante $e=\{v_0,v\}$, die v_0 enthält und von $e_1=\{v_0,v_1\}$ verschieden ist. Dann ist <u>entweder</u> v ein Knoten im Pfad P, d.h. $v=v_i$ für ein $i \geq 2$ und e zusammen mit diesem Teil des Pfades ein Kreis. – Widerspruch!

<u>Oder</u> $v \notin \{v_0,\dots,v_k\}$, dann kann Pfad P durch Hinzufügen von e verlängert werden. – Widerspruch! □

<u>Bemerkung:</u> Lemma 17 gilt nicht in unendlichen Graphen.

Def 1.18 Sei G ein Graph und v ein Knoten von G. Mit $G-v$ bezeichnen wir den Graphen, der entsteht, wenn man von G Knoten v und alle dazugehörigen Kanten entfernt.

Beispiel

Lemma 1.19 (Baumwachstumslemma)

Sei G ein Graph, der einen Endknoten v enthält. Dann sind folgende Aussagen äquivalent:

i) G ist ein Baum

ii) $G-v$ ist ein Baum

Beweis: i) => ii): Betrachte zwei Knoten x,y in $G-v$. Da G zusammenhängt, sind x,y durch einen Pfad verbunden. Dieser Pfad enthält keine Knoten von Grad 1 (außer evtl. x,y). Also enthält er nicht v. Daher ist der Pfad vollständig in $G-v$ enthalten und $G-v$ ist somit zusammenhängend. Da G keinen Kreis enthält, kann auch $G-v$ keinen Kreis enthalten. Damit ist $G-v$ ein Baum.

ii) => i): Durch Hinzufügen eines Endknotens v zu $G-v$ kann kein Kreis entstehen. Es gibt jeweils einen Pfad von v über den (eindeutigen) Nachbarn v' von v zu jedem anderen Knoten in G. Also ist G ein Baum. □

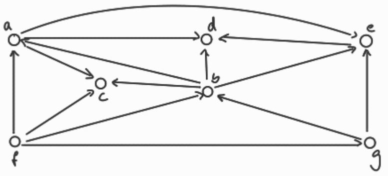
09. April 2024

1.2. Das Warenumschlag-Problem (Minimalkosten-Fluss-Problem)

Das WUP besteht darin, die günstigsten Wege in einem Transportnetz zu finden, um dann eine Menge von Gütern (z.B. Öl, Orangen, Eisenbahnwagen) zwischen einer Menge von Quellen (Startpunkten) und einer Menge von Senken (Zielpunkten) zu befördern.

Beispiel

Im Folgenden betrachten wir diesen Digraphen als Beispiel für ein Transportnetz:



Quellen: f, g
Senken: c, d, e
Zwischenknoten: Rest

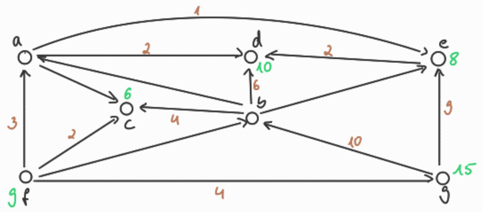
Wir nehmen die folgenden Daten an:

Quellknoten	Menge
f	9
g	15
	24

Senke	Menge
c	6
d	10
e	8
	24

Es ist kein Zufall, dass die Gesamtmenge von Quellen und Senken gleich ist. Wir fordern hier die praktisch unrealistisch, aber theoretisch hilfreiche Bedingung: Der Gesamtbedarf entspricht dem Gesamtangebot.
Um einen solchen Umschlag anzugeben, führen wir Variablen $x_{i,j}$ ein, die angeben, wieviele Güter auf einem Bogen (i,j) transportiert werden. Man beachte, dass dies zu Mehrdeutigkeit führen kann:

Beispiel



gesendete Waren

$x_{f,a} = 3$, $x_{f,c} = 2$, $x_{f,g} = 4$, $x_{g,b} = 10$

$x_{g,e} = 8$, $x_{b,d} = 6$, $x_{b,c} = 4$, $x_{e,d} = 2$

$x_{a,d} = 2$, $x_{a,e} = 1$, $x_{i,j} = 0$ sonst

Die vier Einheiten aus Knoten f über Bogen (f,g) könnten teilweise in Knoten c, d oder e ankommen. Diese Mehrdeutigkeit ist i.d.R. bedeutungslos. Wenn die transportierten Güter nicht unterscheidbar sind, dann ist es an der Senke egal, welches die Quelle war.

Ein Vektor $x \in \mathbb{R}^{|E|}$ beschreibt genau dann einen zulässigen Warenumschlag (wobei $|E|$ = Menge der Bögen des Digraphen), wenn die nachfolgenden Flusserhaltungsbedingung erfüllt sind:

- Zwischenknoten: Die Gütermenge, die ihn den ZK hineinfließt, ist gleich der Menge, die rausfließt.

  Beispiel: Im Zwischenknoten b: $0+10 = 4+0+6+0$

- Senkeknoten: Die Gütermenge, die in eine Senke hineinfließt, ist gleich der Menge, die rausfließt + Bedarf der Senke

  Beispiel: Im Senkeknoten e: $9+1+0 = 2+8$

- Quellknoten: Die Gütermenge, die in eine Quelle hineinfließt + Aufkommensmenge der Quelle ist gleich der Menge, die rausfließt

  Beispiel: Im Quellknoten g: $4+15 = 10+9$

- Nichtnegativität: Transport negativer Gütermengen ist nicht gestattet (Gütermenge $x_{ij} \overset{!}{\geq} 0$)

Algebraisch können wir diese Bedingungen wie folgt beschreiben: $Ax = b$, $x \geq 0$
wobei A eine sog. Knoten-Kanten-Inzidenzmatrix, b die Gütermenge je Knoten beinhaltet:

- Zwischenknoten: Ist i ein ZK, so ist $b_i = 0$
- Senkeknoten: Ist i ein SK, so ist $b_i > 0$ die Bedarfsmenge
- Quellknoten: Ist i ein QK, so ist $b_i < 0$ und $|b_i|$ ist die Aufkommensmenge

Es wird $\sum_{i \in V} b_i = 0$ angenommen.

Die Inzidenzmatrix $A = (a_{ij})_{1 \leq i \leq n, 1 \leq j \leq m} \in \{-1, 0, 1\}^{n \times m}$ mit $n = |V|$, $m = |E|$, hat den Eintrag $a_{ij} = -1$, wenn Knoten i der Anfangsknoten von Bogen j ist und $a_{ij} = +1$, wenn i Endknoten von Bogen j ist.

Unser Beispiel

$$A = \begin{array}{c} \\ a \\ b \\ c \\ d \\ e \\ f \\ g \end{array} \begin{array}{c} \begin{array}{cccccccccccccc} (a,c) & (a,d) & (a,e) & (b,a) & (b,c) & (b,d) & (b,e) & (e,d) & (f,a) & (f,b) & (f,c) & (f,g) & (g,b) & (g,e) \end{array} \\ \begin{pmatrix} -1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & -1 & -1 & -1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & -1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & -1 \end{pmatrix} \end{array}, \text{ sowie } x = \begin{pmatrix} x_{a,c} \\ x_{a,d} \\ \vdots \\ x_{g,b} \\ x_{g,e} \end{pmatrix} \text{ und } \vec{b} = \begin{pmatrix} 0 \\ 0 \\ 6 \\ 10 \\ 8 \\ -9 \\ -15 \end{pmatrix} \begin{array}{c} a \\ b \\ c \\ d \\ e \\ f \\ g \end{array}$$

Für das Versenden von Gütern über einen Bogen fallen jeweils Kosten an, die proportional zur Menge der verschickten Einheiten sind. Diese werden in einem Vektor $c \in \mathbb{R}_+^{|V|}$ ab.

Beispiel: $c = (\underset{c_{a,c}}{53}, \underset{c_{a,d}}{18}, 29, 8, 60, 28, 37, 5, 44, 38, 58, 14, 23, \underset{c_{g,e}}{59})$

Die Gesamtkosten für den Warenumschlag x sind dann gegeben durch: $cx = \sum_{(i,j)\in E} c_{ij} \cdot x_{ij}$.

Definition 1.20

Sei A eine $n\times m$ Inzidenzmatrix eines gewichteten Digraphen $D=(V,E,c)$ und $b: V \to \mathbb{R}$ mit $\sum_{i\in V} b_i = 0$. Dann ist das **Warenumschlagproblem** (transhipment problem) definiert als das Problem: min cx

s.d. $Ax = b$

$x \geq 0$

Ein Vektor $x \in \mathbb{R}_+^{|E|}$ mit $Ax = b$ heißt **zulässige Lösung**.

Bemerkung: Nimmt man beliebige $n-1$ Zeilen des Gleichungssystem $Ax=b$ und summiert diese auf, so erhält man das Negative der weggelassene Zeile. Also kann man eine Zeile streichen, o.B.d.A. entfernen wir die letzte Zeile und erhalten so das Problem: min cx

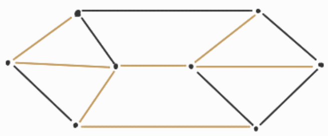
s.d. $\tilde{A}x = \tilde{b}$

$x \geq 0$

Matrix $\tilde{A}$ hat $n-1$ Zeilen und Vektor $\tilde{b}$ hat $n-1$ Komponenten. Wir bezeichnen $\tilde{A}$ als **gekürzte Inzidenzmatrix**.

Definition 1.21

Ein **aufspannender Baum** ist ein Untergraph von einem Graphen G, der ein Baum ist und alle Knoten von G enthält

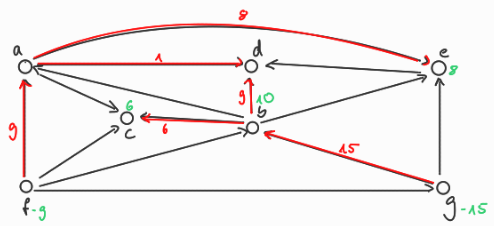


Def. 1.22

Eine zulässige Baumlösung (T, x) ist ein Tupel bestehend aus einem aufspannenden Baums T für D und einer zulässigen Lösung x des Warenumschlagsproblems, wobei $x_{ij} = 0$ für alle Bögen $(i,j) \notin T$ gilt.

Beispiel

Die roten Bögen stellen einen aufspannenden Baum dar, die Werte an den Bögen geben den jeweiligen Flow an.



Aus cx kann für obigen Baum der zugehörige Zielfunktionswert bestimmt werden:

$44 \cdot 9 + 29 \cdot 8 + 18 \cdot 1 + 28 \cdot 9 + 23 \cdot 15 + 60 \cdot 6 = 1603$ Einheiten

Wir wollen zeigen, dass der Wert von x bei einer zulässigen Baumlösung (T,x) eindeutig bestimmt ist.

Lemma 1.23

Sei $T=(V,E)$ ein (gerichteter oder ungerichteter) Baum und $v_1 \in V$ ein beliebiger Knoten. Dann können die Knoten in V so als $v_2,\dots,v_n$ und die Kanten in E so als $e_2,\dots,e_n$ aufgezählt werden, dass es für alle $i=2,\dots,n$ genau eine Kante e_i gibt, deren einer Endknoten v_i und deren anderer Endknoten von $v_1,\dots,v_{i-1}$ ist.

Beweis: (per Induktion über i)

Induktionsanfang: Sei $i=2$. Da $|V|\geq 2$ und T zusammenhängend, also gibt es eine Kante e mit v_1 als Endknoten. Bezeichne diese Kante als e_2 und den anderen Endknoten als v_2.

Induktionsschritt: Sei die Aussage wahr für ein i mit $2\leq i<n$. Da $i<n$, sind noch nicht alle Kanten des Baumes in $e_2,\dots,e_i$. Da T zusammenhängend, gibt es eine Kante $e\in E\setminus\{e_2,\dots,e_i\}$ mit einem Endknoten in $v_1,\dots,v_i$ und der andere Endknoten nicht in $v_1,\dots,v_i$ ist.

Denn: Angenommen es gäbe eine solche Kante nicht.

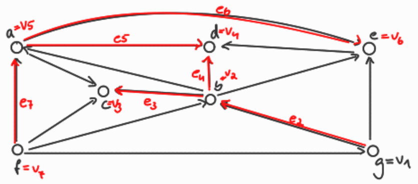
a) Hat e keinen Endknoten in $v_1,\dots,v_i$, dann wäre T nicht zusammenhängend. ↯

b) Hat $e=(a,b)$ beide Endknoten in $v_1,\dots,v_i$, dann gibt es nach Induktionsvoraussetzung, dass es einen Weg P von a zu v_1 und einen Weg Q von b bis v_1 (über Zwischenknoten aus $v_2,\dots,v_i$) gibt. Sei v_j der erste gemeinsame Knoten von P und Q (spätestens v_1). Bezeichne mit P' den Weg von a nach v_j und mit Q' den Weg von v_j nach b. Dann ist (P',Q,e) ein Kreis. Widerspruch da T kreisfrei (da Baum) ist.

Setze $e_{i+1}:=e$, v_{i+1} ist dann der andere Endknoten von e. □

Beispiel

Wir wählen in unserem Beispiel-Baum (rote Bögen) $v_1:=g$.



Hier wurden die weiteren Knoten und die Bögen entsprechend der Aussage von Lemma 1.23 nummeriert.

Betrachte nochmal die Matrix A:

$$A = \begin{array}{cc} & \begin{array}{cccccccccccccc} & e_5 & e_6 & & e_3 & e_4 & & & e_7 & & & & e_2 & \\ (a,c) & (a,d) & (a,e) & (b,a) & (b,c) & (b,d) & (b,e) & (e,d) & (f,a) & (f,b) & (f,c) & (f,g) & (g,b) & (g,e) \end{array} \\ \begin{array}{c} v_5\ a \\ v_2\ b \\ v_3\ c \\ v_4\ d \\ v_6\ e \\ v_7\ f \\ v_1\ g \end{array} & \begin{pmatrix} -1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & -1 & -1 & -1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & -1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & -1 \end{pmatrix} \end{array}$$

Wir ersetzen nun die „alten“ Bezeichnungen der Knoten und Bögen durch die neuen ↑. Nun streichen wir die Zeile v_1 (d.h. wir betrachten die gekürzte Indexmatrix) und alle Spalten, welche nicht zu Baumbögen gehören.

Die übrig gebliebenen Zeilen und Spalten permutieren wir gemäß Index-Reihenfolge.

Beispiel

$$B = \begin{array}{cc} & \begin{array}{cccccc} e_2 & e_3 & e_4 & e_5 & e_6 & e_7 \end{array} \\ \begin{array}{c} v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \end{array} & \begin{pmatrix} 1 & -1 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

Dadurch erhalten wir eine quadratische obere Dreiecksmatrix mit von Null verschiedenen Einträgen auf der Hauptdiagonalen.

Lemma 1.24

Sei $T=(V,E)$ ein aufspannender Baum für einen Digraphen für D und $n:=|V|$. Seien die Knoten und die Bögen wie in Lemma 1.23 aufgezählt. Sei B eine $(n-1)\times(n-1)$ Untermatrix von $\hat{A}$, wobei die Zeilen von B gemäß der Knotenreihenfolge in V permutiert sind. Die Spalten von B entsprechen den Bögen von T in der aufgezählten Reihenfolge. Dann ist B eine obere Dreiecksmatrix mit von Null verschiedenen Einträgen auf der Hauptdiagonalen.

Beweis: s. Übung

Korollar 1.25

Das lineare Gleichungssystem $Bx^*=\tilde{b}$ hat eine eindeutige Lösung

Korollar 1.26 "Rekonstruktionslemma"

Der aufspannende Baum T bestimmt auf eindeutige Weise eine zulässige Lösung x des Warenumschlagsproblems.

Beweis: Setze $x_{u,v}=x^*_{u,v}$ für $(u,v)\in E(T)$ und $x_{u,v}:=0$ sonst. Dann ist (T,x) eine zulässige Baumlösung □

1.3. Die Netzsimplexmethode

Die Netzsimplexmethode arbeitet mit zulässigen Baumlösungen. In jeder Iteration wird entweder der aktuelle Baum durch einen anderen (besseren) Baum ersetzt oder es wird nachgewiesen, dass der aktuelle Baum bereits einer optimalen Lösung entspricht.
Jede Iteration besteht aus drei Schritten:

1. Bestimmung der „fairen Marktpreise" für den aktuellen Baum
2. Suche nach einem eintretenden Bogen
3. Bestimmung einer neuen Flussverteilung (+ austretender Bogen)

Diese 3 Schritte werden wiederholt, bis kein eintretender Bogen mehr gefunden wird. In diesem Fall bricht das Verfahren ab, die letzte gefundene Lösung kann ausgegeben werden.

Ein Unternehmen U verschickt seine Waren so, wie es eine gegebene Baumlösung T vorschreibt. Die Transportkosten werden dann direkt dem Warenwert vor Ort aufgeschlagen. Wenn eine Einheit der Ware am Ort i den Preis y_i hat und das Unternehmen Bogen (i,j) nutzt, um sie nach Ort j zu bringen, dann müsste $y_j = y_i + c_{ij}$ sein. Wird sie für weniger als $y_i + c_{ij}$ verkauft, würden sie Verlust machen, da U einen Teil seiner Transportkosten nicht zurückbekommt. Bei mehr als $y_i + c_{ij}$ könnte ein Konkurrent die Ware in i selber transportieren und in j billiger verkaufen.

Also der erste Schritt wäre die Berechnung "fairer Marktpreise" $y_1, \dots, y_n$, so dass $y_i + c_{ij} = y_j \ \forall (i,j) \in E(T)$. Dieses Gleichungssystem hat $(n-1)$ Gleichungen, eine für jede Baumkante in T. Es gibt aber n Unbekannte (da n Knoten in T). Das System ist nicht eindeutig lösbar! Es gibt <u>einen</u> Freiheitsgrad. Wenn $(y_1, \dots, y_n)$ eine Lösung ist, dann ist $(y_1+d, \dots, y_n+d)$ auch eine Lösung (für beliebiges $d \in \mathbb{R}$).
Die "Fairness" der Preise hängt hier nur von ihrer relativen Differenz $y_j - y_i$ ab, aber nicht von den absoluten Preisen. Die Fairness bleibt auch dann erhalten, wenn alle Preise gleichmäßig fallen/sinken. Insbesondere können wir einen Preis beliebig festlegen, z.B. $y_n = 0$. Es bleibt ein System mit $(n-1)$ vielen Gleichungen in den Variablen $y_1, y_2, \dots, y_{n-1}$ übrig.

Beispiel

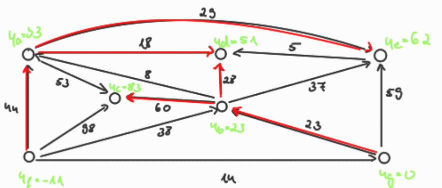
$$
\begin{array}{lrrrrrrl}
e_2 & & y_b & & & & & = 23 \\
e_3 & & -y_b & +y_c & & & & = 60 \\
e_4 & & -y_b & & +y_d & & & = 28 \\
e_5 & -y_a & & & +y_d & & & = 18 \\
e_6 & -y_a & & & & +y_e & & = 29 \\
e_7 & y_a & & & & & -y_f & = 44
\end{array}
$$

Die linke Seite dieses Gleichungssystems entspricht $y \cdot B$, mit B obere bekannte Dreiecksmatrix.
Daher hat es eine eindeutige Lösung: $y_b = 23$, $y_c = 83$, $y_d = 51$, $y_a = 33$, $y_e = 62$, $y_f = -11$ und wir setzen $y_g = 0$.

Damit haben wir die "fairen Marktpreise" bestimmt. (Schritt 1 abgeschlossen)

Schritt 2: Wir betrachten Mitbewerber K von U. Lohnt es sich für K, Waren in i zu kaufen, über (i,j) zu transportieren und in j zu verkaufen?
Die Antwort hängt von Knotenpaar bzw. Bogen (i,j) ab.

Beispiel



- Kaufe Ware am Knoten b für $y_b = 23$ GE
- Verschiebe sie für 37 GE über (b,e)
- Warenwert in Knoten e is $y_b + c_{b,e} = 60$ GE $< y_e$

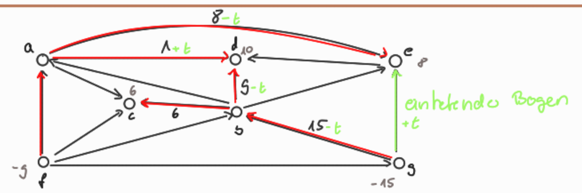
Allgemein: suche Bogen (i,j) mit $y_i + c_{ij} < y_j$. Ein solcher Bogen heißt eintretender Bogen. Gibt es keinen, so ist die Lösung optimal. Bei der Suche nach einem eintretenden Bogen müssen nur die Bögen untersucht werden, die aktuell nicht zur Baumlösung gehört. Gibt es mehrere, wähle z.B. denjenigen mit größter Differenz $y_j - y_i - c_{ij}$.

Bsp: Hier: Bogen (g,e) mit $y_e - y_g - c_{g,e} = 62 - 0 - 59 = 3$.

16. April 2024

Angenommen, Firma U erfährt von den Plänen von K. U möchte diese Info zum eigenen Vorteil ausnutzen. Was tun?
U sendet selber Güter über den eintretenden Bogen. Bezeichne die Anzahl mit t. Wie groß kann t werden?

Beispiel



Ignorieren weiterhin Nicht-Baumbögen und müssen die Transportmengen im Baum anpassen, so dass die Zulässigkeit des Transportplans weiterhin gegeben ist.
Durch Einfügen des eintretenden Bogens in Baum T entsteht ein (ungerichteter) Kreis. Erhöht man den Fluss auf dem eintretenden Bogen, muss entsprechend der Fluss auf den anderen Kreisbögen um t Einheiten erhöht oder erniedrigt werden. Wann was?
Dies ist aus den Flusserhaltungsbedingungen abzuleiten (eindeutig):

- sind an einem Knoten zwei eingehende oder zwei ausgehende Bögen, dann haben sie unterschiedliche Vorzeichen beim t.
- anderenfalls (einer eingehend, einer ausgehend) ⇒ gleiche Vorzeichen.

Starte mit dem Knoten, an dem der eintretende Bogen beginnt und wende diese Regel nacheinander auf alle Bögen des Kreises an (s.o.).
Jede Einheit, die Unternehmen U über den eintretenden Bogen transportiert, senkt die Gesamttransportkosten. Wähle t daher so groß wie möglich. Aber: auf den bisherigen Baumbögen dürfen keine negativen Flusswerte auftauchen.

Beispiel: Es sind also folgende Nebenbedingung zu beachten:

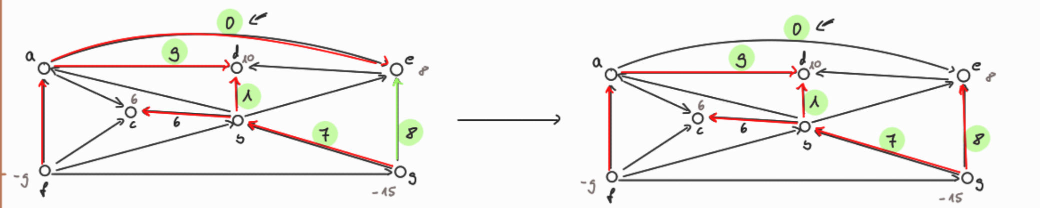
$$x_{a,e} = 8 - t \geq 0,\; x_{f,d} = 9 - t \geq 0 \text{ und } x_{g,b} = 15 - t \geq 0$$

Diese Nebenbedingungen verhindern, dass t beliebig groß werden kann.

Bsp: In unserem Beispiel liefert $x_{a,e}$ die schärfste Begrenzung: $t = 8$ ist der größtmögliche Wert.

Nun passt man die Flusswerte auf den Bögen entsprechend an, wird ein Bogen einen Null-Fluss aufweisen und kann daher aus dem Baum entfernt werden.

Beispiel



So entsteht eine neue Lösung x' mit dem dazugehörigen Baum $T' := T + (g,e) - (a,e)$, mit (T', x') ist wiederum zul. Baumlösung und ihr Zielfunktionswert ist $cx' = 44 \cdot 9 + 18 \cdot 9 + 28 \cdot 1 + 23 \cdot 7 + 60 \cdot 6 + 59 \cdot 8 = 1579$ (vs. 1603 zuvor).

restliche Iterationsdurchläufe

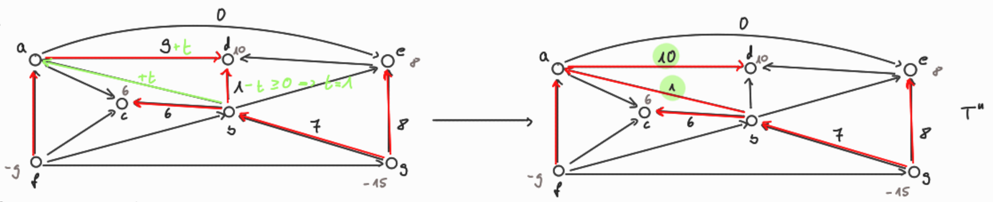
ITERATION 2:

Die nächste Iteration beginnt wieder mit Schritt 1, also der Bestimmung fairer Marktpreise für den neuen Baum T'. Wir stellen das lineare Gleichungssystem $y_i + c_{i,j} = y_j \;\forall (i,j) \in E(T')$ auf und erhalten die eindeutige Lösung $y_g = 0, y_e = 59, y_b = 23, y_c = 83, y_d = 51, y_a = 33, y_f = -11$.

Rückblende: In der vorherigen Iteration hat K versucht, Waren in G einzukaufen, über (g,e) zu versenden und in e zu verkaufen. Dank der Reaktion von U lohnt sich das für K nicht mehr: Die Differenz der Marktpreise in g und e ist $y_e - y_g = 59 - 0 = 59$, was genau den Transportkosten entspricht.

Der eintretende Bogen in Schritt 2 ist (b,a) mit $y_a - y_b - c_{b,a} = 33 - 23 - 8 = 2$.

Schritt 3:



Der Zielfunktionswert dieser Lösung x'' ist $cx'' = 44 \cdot 9 + 18 \cdot 10 + 23 \cdot 7 + 60 \cdot 6 + 59 \cdot 8 + 8 \cdot 1 = 1577$

ITERATION 3:

Das Lösen des Gleichungssystems zur Bestimmung fairer Marktpreise für Baum T'' liefert:

$$y_g = 0,\; y_e = 59,\; y_b = 23,\; y_c = 83,\; y_a = 31,\; y_d = 49,\; y_f = -13$$

Jetzt ist $y_i - y_j - c_{i,j} \leq 0 \;\forall (i,j) \in E(T'')$. Es gibt also keinen weiteren eintretenden Bogen und der Netzsimplex-Algorithmus terminiert.

Warum ist die gefundene Lösung global optimal?

Abbruchbedingung war $y_i - y_j - c_{ij} \leq 0$, also $y_i - y_j \leq c_{ij}$ $\forall (i,j) \in E$. Die über einen Bogen (i,j) transportierte Menge x_{ij} ist nicht-negativ.
Also gilt: $(y_j - y_i) x_{ij} \leq c_{ij} \cdot x_{ij}$ $\forall (i,j) \in E$. Also summiert man über alle Bögen und erhält die Abschätzung

$$yb = y(Ax) = (yA)x = \sum_{(i,j)\in E} (y_j - y_i) \cdot x_{ij} \leq \sum_{(i,j)\in E} c_{ij} \cdot x_{ij} = cx.$$

Also gezeigt: $yb \leq cx$.

Im vorliegenden Fall: $y_6^* \cdot b = 83 \cdot 6 + 49 \cdot 10 + 59 \cdot 8 + (-13) \cdot (-9) + (-15) \cdot 0 = 1577$

Die Marktpreise aus der letzten Runde liefern demnach eine untere Schranke, die den Wert der gefundenen Lösung x^* entspricht. Daher ist die Lösung global optimal

Satz 1.27

Sei x eine zulässige Baumlösung.

1) Wenn kein eintretender Bogen gefunden wird, dann ist x global optimal.
2) Wenn kein austretender Bogen gefunden wird, dann ist die Probleminstanz unbeschränkt.
3) Ist x' eine zulässige Baumlösung nach dem Austausch, dann ist das Netz-Simplex-Verfahren monoton, d.h. $cx \geq cx'$.

Beweis: Jede Iteration beginnt mit einer zulässigen Baumlösung (T,x)

Schritt 1: Wir berechnen den Zeilenvektor $y = (y_1, \dots, y_n)$ mit $y_i + c_{ij} = y_j$ $\forall (i,j) \in T$. Setze: $\bar{c} := c - yA$, dann ist $\bar{c}_{ij} = c_{ij} + y_i - y_j$. Also ist $\bar{c}_{ij} = 0$ $\forall (i,j) \in T$ und $x_{ij} = 0$ $\forall (i,j) \notin T$

(da (T,x) zul. BL). Also gilt $\bar{c}x = 0$.

Sei nun $\bar{x}$ ein beliebiger Vektor, der $A\bar{x} = b$ erfüllt. Dann erfüllt dieser auch $c\bar{x} = c\bar{x} - yA\bar{x} + yA\bar{x} = \underbrace{(c - yA)}_{\bar{c}}\bar{x} + y\underbrace{A\bar{x}}_{b} = \bar{c}\bar{x} + yb$.

Da dies für alle Vektoren $\bar{x}$ mit $A\bar{x} = b$ gilt, gilt es insbesondere für unsere obige Baumlösung x, also: $cx = \underbrace{\bar{c}x}_{0} + yb = yb$.

Es folgt, dass: $c\bar{x} = \bar{c}\bar{x} + xb = \bar{c}\bar{x} + cx$ $\forall \bar{x}$ mit $A\bar{x} = b$

Schritt 2: Wir suchen einen Bogen $e = (u,v)$ mit $y_u + c_{u,v} < y_v$. Wenn es keinen solchen Bogen gibt, dann ist die Lösung x optimal:

Denn wenn $c_{ij} + y_i - y_j \geq 0$ $\forall (i,j) \in T$, so ist $\bar{c} \geq 0$, also $\bar{c}\bar{x} \geq 0$ $\forall \bar{x} \geq 0$. Also können wir abschätzen: $c\bar{x} = \bar{c}\bar{x} + cx \geq cx$ für alle zulässigen Lösungen $\bar{x}$ (d.h. $A\bar{x} = b$, $\bar{x} \geq 0$).

Schritt 3: Da T ein aufspannender Baum ist (zusammenhängend und kreisfrei), enthält er einen eindeutigen Pfad zwischen u und v. Daher enthält $T+e$ einen eindeutigen Kreis. Durchläuft man diesen in Richtung des eintretenden Bogens e, so kann man zwischen Vorwärtsbögen (in Richtung e) und Rückwärtsbögen unterscheiden. Wir setzen:

$$x'_{ij} = \begin{cases} x_{ij} + t, & \text{wenn } (i,j) \text{ Vorwärtsbogen} \\ x_{ij} - t, & \text{wenn } (i,j) \text{ Rückwärtsbogen} \\ x_{ij}, & \text{wenn } (i,j) \text{ nicht in Kreis} \end{cases}$$

für einen Wert t. Man bemerke, dass $Ax' = Ax = b$, da sich die beiden zusätzlichen $\pm t$ an jedem Knoten des Kreises aufheben.

Also erfüllt x' die Gleichung $c\bar{x} = \bar{c}\bar{x} + cx$, hier: $cx' = \bar{c}x' + cx$.

Nun ist $e = (u,v)$ der einzige Bogen (i,j) mit $\bar{c}_{ij} \neq 0$ und $x'_{ij} \neq 0$. Also ist $\bar{c}x' = \bar{c}_e x'_e = \bar{c}_e \cdot t$. Somit ist $cx' = \bar{c}x' + cx = cx + \bar{c}_e \cdot t$. Der Wert t soll so gewählt werden, dass x' zulässig und cx' so klein wie möglich. Da $\bar{c}_e < 0$, wird t maximal gewählt, so dass $x' \geq 0$ ist.

Um das zu erreichen, sucht man einen Rückwärtsbogen f mit $x_f \leq x_{ij}$ für alle Rückwärtsbögen (i,j) und setzt $t := x_f$. Wenn es keinen Rückwärtsbogen gibt, dann liefert jedes positive t ein zulässiges x'. In diesem Fall ist das Problem <u>unbeschränkt</u>, da es dann für jedes $M > 0$ eine zulässige Lösung x' gibt mit $cx' < -M$.

Hinweis: Probleme mit $c \geq 0$ sind niemals unbeschränkt, ihre zulässige Lösung x erfüllt stets $cx \geq 0$.

Mit $t = x_f$ gilt für die neue Lösung x' dass $x'_f = 0$ und $x'_{ij} = 0$ für alle $(i,j) \notin T + e$.

Der Teilgraph $T' := T - e + f$ ist zusammenhängend, kreisfrei und aufspannender Baum, d.h (T', x') ist zulässige Baumlösung.

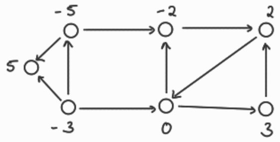
Der dritte Schritt wird auch als Austausch (Pivot) bezeichnet. $\square$

18. April 2024

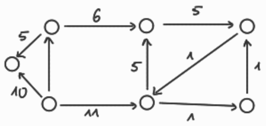
Degeneriertheit: Im einleitenden Beispiel machten wir die Erfahrung, dass sich die Zielfunktionswerte in jeder Iteration verbessern. Dies muss aber nicht immer der Fall sein.

Beispiel

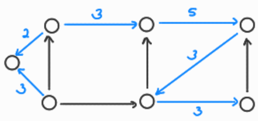
Angebot \ Nachfrage



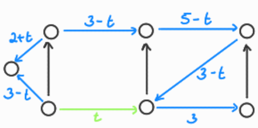
Kosten



initiale zulässige Baumlösung

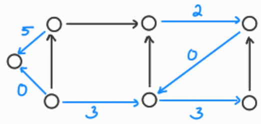


Wir starten das Netzsimplex-Verfahren. In der ersten Runde liefert es:



Fortsetzung Beispiel

Als austretender Bogen kommt jeder mit dem Label 3-t in Betracht, wir wählen wir den oberen. In der nächsten Runde sieht der Baum wie folgt aus:

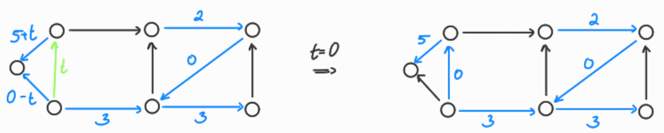


Das Neue an dieser Lösung ist, dass auch Bögen des Baumes den Wert 0 aufweisen.

Solche Lösungen werden als degeneriert bezeichnet. Degeneriertheit an sich ist harmlos, kann aber schwerwiegende Probleme nach sich ziehen:

Fortsetzung Beispiel

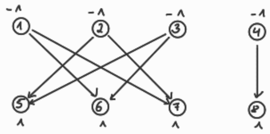
Die nächste Iteration



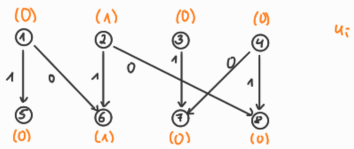
Hier haben wir es mit einem Austausch mit $t=0$ zu tun, solche Austausche ändern zwar den Baum, aber der Lösungsvektor hat sich nicht geändert, $x'=x$, daher auch nicht der Zielfunktionswert $cx=cx'$. Der Netzsimplex scheint 'auf der Stelle zu treten'.
Wenn der Netzsimplex keine Fortschritte macht in der Zielfunktion, kann ein weiteres Problem hinzukommen: Zykeln.

Beispiel nach [3.7 Gassner 1964]

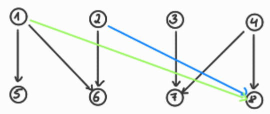
Gegeben sei ein Graph mit 8 Knoten $V=\{1,\dots,8\}$, $S:=\{1,\dots,4\}$ seien Quellen und $T=\{5,\dots,8\}$ die Senken.



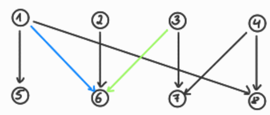
Zwischen jeder Quelle i und jeder Senke j gibt es einen Bogen (i,j) und jede Quelle hat eine Wareneinheit abzugeben und jede Senke nimmt eine auf.
Die Transportkosten sind $c_{16}=c_{17}=c_{25}=c_{27}=c_{35}=c_{36}=c_{48}=1$ und $c_{ij}=0$ sonst. Der Netz-Simplex wird mit folgendem Baum initialisiert:



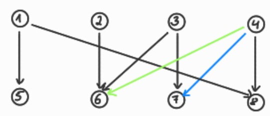
Die Kosten für diese Lösung sind $cx = c_{48} \cdot x_{48} = 1$. Der Algorithmus kann jetzt folgenden Austausch vornehmen:



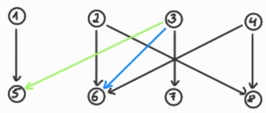
#1: (1,8) rein, (2,8) raus



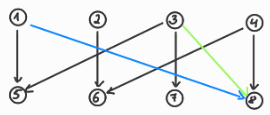
#2: (3,6) rein, (1,6) raus



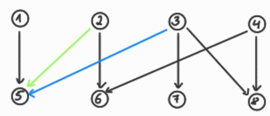
#3: (4,6) rein, (4,7) raus



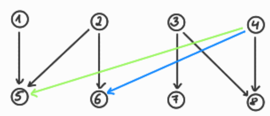
#4: (3,5) rein, (3,6) raus



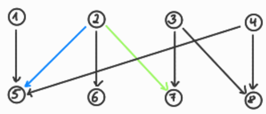
#5: (3,8) rein, (1,8) raus



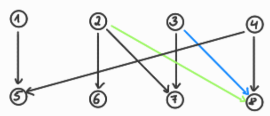
#6: (2,5) rein, (3,5) raus



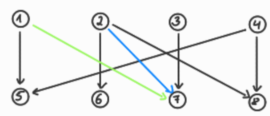
#7: (4,5) rein, (4,6) raus



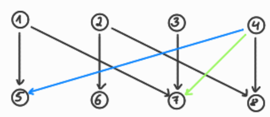
#8: (2,7) rein, (2,5) raus



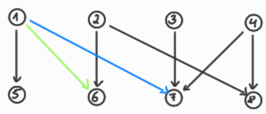
#9: (2,8) rein, (3,8) raus



#10: (1,7) rein, (2,7) raus

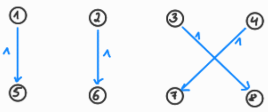


#11: (4,7) rein, (4,5) raus



#12: (1,6) rein, (1,7) raus

Nach 12 Iterationen hat der Netz-Simplex den Initialbaum wieder erreicht. Er könnte jetzt die selben 12 Schritte wiederholen und würde niemals eine Optimallösung finden:



mit Kosten = 0 GE

Dieses Verhalten des Netzsimplex wird als kreiseln oder zykeln bezeichnet. Durch geschickte Wahl der eintretenden und austretenden Bögen kann zykeln vermieden werden. Selbst wenn man diese Auswahlregeln nicht verwendet, kommt dieses Zykeln sehr selten vor. Alle Beispiele sind frei konstruiert. Dazu später mehr...

Nimmt man an, dass es kein Zykeln gibt, folgt daraus, dass jeder während der Ausführung konstruierte Baum nur einmal vorkommt. Also terminiert der Algorithmus nach endlich vielen Schritten, da es nur endlich viele Bäume gibt.

Es bleibt, den Algorithmus mit einer zulässigen Baumlösung zu initialisieren. Eine solche Startlösung zu erzeugen, ist jedoch kein triviales Problem, nicht jedes Warenumschlagproblem hat eine Lösung. Deren Nicht-Existenz ist nicht immer offensichtlich. Um zu einer Startlösung zu gelangen, stellen wir zuerst ein sogenanntes Hilfsproblem $D'=(V, E', c')$ auf, welches neben dem Ausgangsgraphen $D=(V,E)$ noch weitere künstliche Bögen enthält.

Wir wählen einen beliebigen Knoten $w \in V$ des Graphen. Setze $c'_{ij}=0 \ \forall (i,j) \in E$. Für alle Quellen $i \in V \setminus \{w\}$: wenn $(i,w) \notin E$, dann füge ihn zu E' hinzu und setzen $c'_{iw} := 1$. Für alle übrigen Knoten $j \in V \setminus \{w\}$ (Senken/Zwischenknoten): wenn $(w,j) \notin E$, dann füge ihn zu E' hinzu und setze $c'_{wj} := 1$.

Für D' kann sehr einfach eine aufspannende Lösung T_w angegeben werden:

Zum Baum T_w gehören alle $n-1$ Bögen von den Quellen zu w sowie von w zu den Senken bzw. Zwischenknoten. Setze $x_{iw} := -b_i$ für alle Quellen $i \in V \setminus \{w\}$ und $x_{wj} = b_j$ für alle Senken $j \in V \setminus \{w\}$ und $x_{ij} := 0$ sonst.

Der Netzsimplex wird zur Lösung des Hilfsproblems mit Startbaum T_w verwendet. Wir bezeichnen mit (T^*, x^*) die optimale zulässige Baumlösung und unterscheiden drei Fälle:

(1) T^* enthält einen künstlichen Bogen (u,v) mit $x^*_{uv} > 0$

(2) T^* enthält keinen künstlichen Bogen

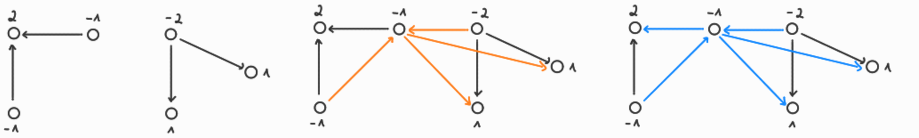
(3) T^* enthält einen künstlichen Bogen und für jeden künstlichen Bogen (i,j) gilt $x^*_{ij} = 0$.

Im Fall (1) hat das Ausgangsproblem hat keine Lösung. Denn: angenommen es gäbe eine zulässige Lösung x. Dann wäre $c'x = 0 < c'x^*$, im Widerspruch zur Optimalität von x^*

Im Fall (2) hat das Ausgangsproblem eine zulässige Lösung und das Netz-Simplex-Verfahren kann mit der zulässigen Baumlösung T^* initialisiert werden.

Im Fall (3) hat das Ausgangsproblem eine zulässige Lösung (angegeben durch x^* auf den Originalbögen in E), aber es existiert keine zulässige Baumlösung.

Beispiel

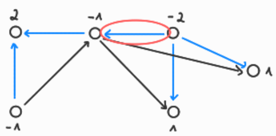


Originalbogen

künstliche Bögen

zulässige optimale Baumlösung

Nach Beendigung des Netz-Simplex auf T_w entsteht folgende Baumlösung:



überflüssige Hilfsbögen

optimale zulässige Baumlösung T^*

Diese enthält weiterhin einen künstlichen Bogen. Man hätte das Ausgangsproblem gleich in zwei unabhängige Teilprobleme aufteilen können. Wir zeigen, dass das in Fall (3) stets möglich ist.

Diese Dekomposition wird uns angezeigt durch die Knotenwerte $y_1, \dots, y_n$ die zur Lösung x^* gehören. Sei (u,v) ein künstlicher Bogen im optimalen Baum T^*. Dann ist $R := \{k \in V \mid y_k \leq y_u\}$ und $S := \{k \in V \mid y_k > y_u\}$ eine disjunkte Menge von V. R ist nicht leer, da $y_u \leq y_u$, also $u \in R$. S ist nicht leer, da $y_v = y_u + c'_{u,v} = y_u + 1 > y_u$, also $v \in S$. Es gilt $y_i + c_{ij} \geq y_j \ \forall (i,j) \in E'$ (Abbruchbedingung des Netzsimplex). Also gibt es keinen Originalbogen $(i,j) \in E$, der R und S verbindet (d.h. $i \in R, j \in S$). 23. April 24

Denn: angenommen es gäbe einen solchen Bogen. Dann wäre $y_i + c'_{ij} = y_i + 0 = y_i \overset{\text{Def. } R}{\leq} y_u \overset{\text{Def. } S}{<} y_j$, im Widerspruch zur Abbruchbedingung (d.h. $y_i + c'_{ij} \geq y_j \ \forall (i,j) \in E'$).

Anschaulich gesprochen: Die Knoten in S erhalten keine Warenlieferungen aus R. Sie brauchen es auch nicht, denn die Quellen in S decken den Bedarf der Senken in S:

$$\sum_{k \in S} b_k \overset{Ax=b}{=} \underbrace{\sum_{\substack{(i,j) \in E' \\ i \in R, j \in S}} x^*_{ij}}_{=0, \text{ wegen } \underbrace{\text{Fall 3}}_{\text{entferne Hilfsbögen wegen Fall 3}} \text{ und keine Originalbögen}} - \sum_{\substack{(i,j) \in E' \\ i \in S, j \in R}} x^*_{ij} = - \sum_{\substack{(i,j) \in E \\ i \in S, j \in R}} x^*_{ij} = 0$$

Für alle Bögen $(i,j) \in E$ mit $i \in S$ und $j \in R$ gilt $x^*_{ij} = 0$, denn: angenommen, es gäbe einen solchen Bogen (i,j) mit $x^*_{ij} > 0$, also $(i,j) \in T^*$. Dann gilt $y_j = y_i + c'_{ij} = y_i$, im Widerspruch zu $y_j \leq y_u < y_i$.

Also ist S "autark" und kann als Teilproblem gesondert vom Rest gelöst werden. Selbiges gilt für R, welches als eigenes Teilproblem gelöst werden kann. Gibt es noch weitere künstliche Bögen der Form (u', v') in T^*, wendet man obige Argumente induktiv an.

Die effiziente Erneuerung der Knotenwerte y

Zur Erinnerung: Die Knotenwerte (faire Marktpreise) $y_1, \dots, y_n$ erfüllen $y_i + c_{ij} = y_j \; \forall (i,j) \in T$. Wenn wir eine Zahl, z.B. $y_n^* := 0$, wählen, dann gibt es eindeutige Werte $y_1^*, \dots, y_{n-1}^*$, so dass $y_i^* + c_{ij} = y_j^* \; \forall (i,j) \in T$. Wählt man für y_n^* eine andere Zahl d, dann erhält man die Knotenwerte $y_1', \dots, y_n'$ mit $y_n' = y_n^* + d$.
Alle möglichen Lösungen für die Knotenwerte zum gegebenen Baum T entstehen auf diese Weise (da es genau eine Variable fixiert wurde, d.h. es gibt genau einen Freiheitsgrad beim Lösen des Gleichungssystems).
Damit ist die Differenz $y_j - y_i$ unabhängig von der Wahl von $y_n^* = 0$ und hängt nur von T ab.

In jeder Iteration müssen die Knotenwerte $y_1, \dots, y_n$ neu ausgerechnet werden. Der zu Grunde liegende Baum ändert sich zwischen zwei aufeinanderfolgenden Iterationen nur wenig. Kann man diese Information nutzbar machen?

Sei T eine zulässige Baumlösung und $y_1, \dots, y_n$ Knotenwerte mit $y_i + c_{ij} = y_j \; \forall (i,j) \in T$. Sei $e = (u,v)$ der eintretende Bogen und f der austretende. In der nächsten Iteration ist dann $T' := T + e - f$ die zulässige Baumlösung. Der Graph $T' - e = (T + e - f) - e = T - f$ besteht aus zwei Zusammenhangskomponenten (Bäumen) T_u und T_v. Es gilt $u \in T_u$ und $v \in T_v$. Setze jetzt $y_h' := y_h \; \forall h \in T_u$ und $y_h' := y_h + c_e + y_u - y_v \; \forall h \in T_v$.

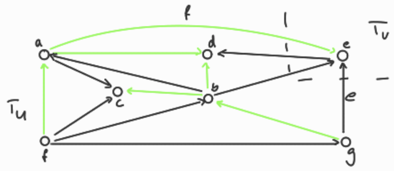
Dann gilt: $y_i' + c_{ij} = y_j' \; \forall (i,j) \in T'$, denn: ein Bogen $(i,j) \neq (u,v)$ ist vollständig entweder in T_u oder T_v enthalten. Für diese Bögen gilt dann $y_j' - y_i' = y_j - y_i = c_{ij}$. Für den Bogen $e = (u,v)$ gilt: $y_u' + c_{u,v} = y_u + c_{u,v} + (y_v - y_u) = y_v + (c_{u,v} + y_u - y_v) = y_v'$

Beispiel

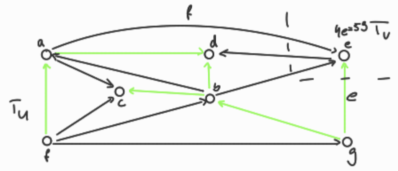
Für unser Eingangsbeispiel und der ersten zulässigen Baumlösung hatten wir folgende Knotenwerte ermittelt.

$y_g = 0,\; y_b = 23,\; y_c = 83,\; y_d = 51,\; y_a = 33,\; y_e = 62,\; y_f = -11.$

Eintretender Bogen ist (g,e), austretender Bogen ist (a,e)

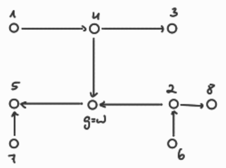


Nach der Updateformel setzen wir $y_h' = y_h$ für fast alle Knoten (alle außer e). Für Knoten e gilt: $y_e' = y_e + c_{ge} + y_g - y_e = c_{ge} + y_g = 59 + 0.$



Wie man zykeln verhindert: Sei T ein Baum und w ein Knoten in T. Anschaulich gesprochen zeigt jeder Bogen von T entweder in Richtung w oder von w weg.

Beispiel

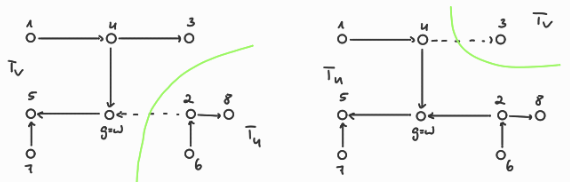


Die Bögen $(1,4), (2,9), (4,9), (6,2), (7,5)$ zeigen in Richtung w und die übrigen zeigen von w weg.

Def 1.28

Sei T ein Baum, w ein Knoten in T und (u,v) ein Bogen in T. Betrachten die Partition $T - (u,v)$ in Bäume T_u mit $u \in T_u$ und T_v mit $v \in T_v$. Wenn $w \in T_v$, dann zeigt (u,v) in Richtung w, wenn $w \in T_u$, dann zeigt (u,v) von w weg.

Beispiel



Wir wählen einen beliebigen Knoten w des Graphens als Wurzel und ändern diesen Wurzelknoten während der Ausführung des Netz-Simplex nicht.

Satz 1.25

Sei T eine zulässige Baumlösung. Sei f eintretender und e austretender Bogen in einem degenerierten Austausch. Wenn der eintretende Bogen e im Graphen $T + e - f$ von der Wurzel wegzeigt, dann wird Netz-Simplex nicht kreiseln.

Beweis: Um das Verhalten des Netz-Simplex besser analysieren zu können, führen wir zwei Kennzahlen $g(T)$ und $h(T)$ für eine zulässige Baumlösung T ein.

Setze $g(T) = c \cdot x$. Diese Setzung ist wohldefiniert, da x eindeutig von T abhängt. Die Knotenwerte $y_1 \dots y_n$ hängen zwar nicht eindeutig von T ab, wohl aber die Differenzen $y_h - y_w \; \forall h = 1, \dots, n$. Also können wir setzen: $h(T) = \sum_{h=1}^{n} y_h - y_w$. Wir betrachten nun zwei Bäume $T_i = T$ und $T_{i+1} = T + e - f$ in der Liste $T_1, T_2, T_3, \dots$ der vom Netz-Simplex während der Ausführung konstruierten Bäume. Nach Satz 1.27 gilt $g(T_{i+1}) \leq g(T_i)$. Falls $g(T_{i+1}) = g(T)$, dann handelt es sich um einen degenerierten Austausch.

Nach Annahme zeigt $e = (u,v)$ in T_{i+1} von Wurzel w weg. Das heißt, $w \in T_u$ (nach Def. 1.28).

Daraus folgt dann:

$$h(T_{i+1}) = \sum_{h=1}^{n} (y_h' - y_w') = \sum_{h \in T_u} (y_h' - y_w') + \sum_{h \in T_v} (y_h' - y_w') \overset{\text{eff. Update}}{=} \sum_{h \in T_u} (y_h - y_w) + \sum_{h \in T_v} (y_h + (c_e + y_u - y_v) - y_w) = \sum_{h=1}^{n} (y_h - y_w) + (c_e + y_u - y_v) \cdot |T_v| = h(T_i) + (c_e + y_u - y_v) \cdot |T_v|$$

Bogen e war so gewählt, dass $c_e + y_u - y_v < 0$ gilt. Zwischenstand: Wenn $g(T_{i+1}) = g(T_i)$, dann $h(T_{i+1}) < h(T_i)$.

Angenommen, der Netz-Simplex zykelt. Dann gibt es Rundenindizes i, j mit $i<j$, so dass $T_i = T_j$. Also gilt $g(T_i) = g(T_j)$. Daraus folgt, $g(T_i) = g(T_{i+1}) = g(T_{i+2}) = \dots = g(T_j)$. Nach dem oben gezeigten gilt dann $h(T_i) > h(T_{i+1}) > h(T_{i+2}) > \dots > h(T_j)$, im Widerspruch zu $h(T_i) = h(T_j)$. $\square$

1976 entwickelte W.H. Cunningham eine Anti-Kreisel Austauschstrategie (Satz 1.25).

Def. 1.30 Eine zulässige Baumlösung (T, x) heißt **stark zulässig**, wenn jeder Bogen $(i,j) \in T$ mit $x_{ij} = 0$ von der Wurzel wegzeigt.

Die Idee ist nun, nur Austausche zu zulassen, bei denen die Sequenz $T_1, T_2, T_3, \dots$ aus stark zulässigen Baumlösungen besteht. Wenn e und f eintretenden bzw. austretenden Bogen in einem Austauschschritt sind, der von T_i zu $T_{i+1} = T_i + e - f$ führt, dann muss e (wegen $x_e = 0$) von der Wurzel weg zeigen, da T_i und T_{i+1} stark zulässig sind.
Aus Satz 1.29 folgt dann, dass der Netz-Simplex nicht kreiselt. Es bleibt zu zeigen:

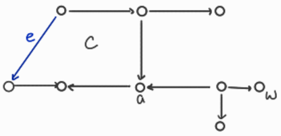
i) Wir können das Verfahren mit einer starken zulässigen Lösung initialisieren

ii) Wir können einen Austausch durchführen, der aus einer stark zulässigen Lösung wieder eine stark zulässige Lösung macht.

25. April 2024

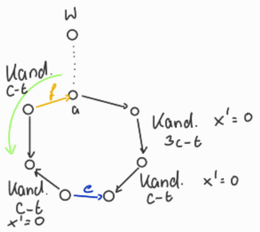
zu i) Um überhaupt mit einem Baum starten zu können, stellen wir ein Hilfsproblem auf. Dazu wählen wir einen beliebigen Knoten w als Wurzelknoten. Es zeigen Bögen (i, w) von allen Quellen zur Wurzel. Für diese gilt $x_{iw} > 0$. Es zeigen Bögen (w, j) von der Wurzel zu allen Senken. Für diese gilt ebenfalls $x_{jw} > 0$. Es zeigen Bögen (w, j) von der Wurzel zu allen Zwischenknoten. Für diese gilt $x_{wj} = 0$. Also ist der initiale Baum des Hilfsproblems stark zulässig.

zu ii) Es ist zu zeigen, dass ein stark zulässiger Baum zu einem wiederum stark zulässigen Baum $T + e - f$ führt, wenn f geeignet gewählt wird.
(Hinweis: Wahl von e für starke Zulässigkeit irrelevant)
Durch Hinzufügen von e zu T entsteht ein eindeutiger Kreis C:



Wir sagen, der Knoten a ist der Anker im Kreis C von $T + e$, wenn a der erste Knoten ist, der auf einem Pfad von Wurzel w zum Kreis C liegt.
Wähle als austretenden Bogen f den ersten in Frage kommenden Bogen, der von a ausgehend auf den Kreis C in Richtung e erreicht wird.
Um zu zeigen, dass $T + e - f$ unter dieser Austauschregel stark zulässig ist, genügt es, die Bögen von C zu betrachten, alle übrigen haben weder ihre Ausrichtung bzgl. w noch ihren Wert geändert.

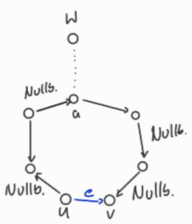
Fall 1: Der Austausch ist nicht-degeneriert

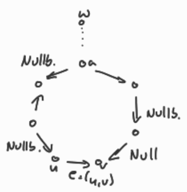


Dann erhöht sich also der Wert von x_{ij} auf allen Vorwärtsbögen (bzgl. e) und erniedrigt sich auf den Rückwärtsbögen (bzgl. e) im Kreis. Die Bögen $(i,j) \in C \setminus e$ mit $x'_{ij} = 0$ sind Kandidaten für den austretenden Bogen f.
Die obige Auswahlregel garantiert, dass alle übrigen Kandidaten, die in x' jetzt Nullfluss haben, von der Wurzel wegzeigen in $T' = T + e - f$.

Fall 2: Der Austausch ist degenerierter Austausch.

In diesem Fall ändert sich die Lösung x nicht. Alle Bögen $(i,j) \in C$ mit $x_{ij} = 0$ bezeichnen wir als **Nullbögen**.





Der eintretende Bogen zerlegt den Kreis in 3 Teile

1) von a bis u

2) e

3) von v bis a

Da T stark zulässig ist, sind alle Nullbögen im ersten Bereich Vorwärtsbögen (bzgl. e) und im dritten Bereich Rückwärtsbögen (bzgl. e).
Die Nullbögen im dritten Bereich sind Kandidaten für den austretenden Bogen. Der erste Kandidat nach v wird als austretender Bogen gewählt.