Kapitel 3

Graphen und Netzpläne

3.1 Spezifikation und innere Struktur von Graphen

Anwendungen der Graphentheorie: Halbautomaten, Zustandsbäume, Wegnetze (Standortbestimmung, Verkehrsleitsysteme, Transport, maximale Flüsse in Leitungsnetzen), Netzpläne, Ablaufplanung, Diagramme, Datenstrukturen, Soziologische Netzwerke (Beziehungsstrukturen, hierarchische Strukturen, Zuordnung von Angestellten zu Jobs), Sequenzanalysen (Spracherkennung, DNA), chemische Verbindungen, Färben von Landkarten, ...

Sei $G = \langle V, E, f \rangle$ ein gerichteter Graph (*Digraph*). Gibt es zu jedem $e_1 \in E$ mit $f(e_1) = (v, w)$ ein $e_2 \in E$ mit $f(e_2) = (w, v)$, so identifizieren wir e_1 und e_2 mit einer ***ungerichteten Kante*** $e = [v, w]$ (e ***inzidiert mit*** v, w). $\Rightarrow$ ***ungerichteter*** *Graph*

Ein Graph kann gerichtete und ungerichtete Kanten enthalten.

Sei $G = \langle V, E, f \rangle$ ein endlicher gerichteter Graph ohne Mehrfachkanten. Dieser kann angegeben werden durch:

a. Angabe aller Paare $(v, w) \in V \times V$, für die $\exists e \in E$ mit $f(e) = (v, w)$ (analog für ungerichtete Graphen)

b. Angabe der Menge $P(v)$ ($S(v)$) aller Vorgänger (Nachfolger) von v für alle Knoten $v \in V$; dabei heißt $u \in V$ *Vorgänger (predecessor)* von $v \in V$, falls $(u, v) \in E$, und $w \in V$ heißt *Nachfolger (successor)* von v, falls $(v, w) \in E$. G ungerichtet: Angabe der Menge $N(v)$ aller Nachbarn von v; dabei heißt n *Nachbar* von v, falls $[n, v] \in E$.

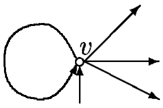
Quelle: Knoten $v \in V$ mit $P(v) = \emptyset$

Senke: Knoten $v \in V$ mit $S(v) = \emptyset$

isolierter Knoten: Knoten $v \in V$ mit $P(v) = S(v) = \emptyset$

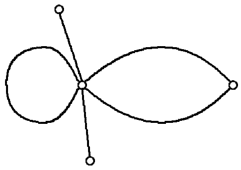
Hingrad von v: $\delta_h(v) = |P(v)|$

Weggrad von v: $\delta_w(v) = |S(v)|$

Beispiel  $\delta_h(v) = 2,\ \delta_w(v) = 4$

G ungerichtet: $\delta(v) = |N(v)| =$ *Grad* von v

Multigraph: Graph mit Mehrfachkanten, z.B.



Bei gerichteten Multigraphen ist $\sum_{v \in V} \delta_h(v) = \sum_{v \in V} \delta_w(v) = |E|$. Bei ungerichteten Multigraphen gilt $\sum_{v \in V} \delta(v) = 2|E|$, woraus folgt, dass die Anzahl der Knoten ungeraden Grades gerade ist.

Ein Graph ohne Mehrfachkanten und ohne Schlingen heißt *schlicht*. Ein *trivialer* Graph besteht aus einem Knoten und keiner Kante.

Satz 1 Ein nicht–trivialer schlichter ungerichteter endlicher Graph besitzt mindestens ein Paar von Knoten, welche denselben Grad haben.

Beweis Sei $V = \{v_1, \dots, v_n\}$. $\Rightarrow 0 \le \delta(v) \le n-1\ \forall v \in V$.
Es kann nicht $v, w \in V$ mit $\delta(v) = 0$ und $\delta(w) = n-1$ geben, da $\delta(v) = 0 \Rightarrow$ jeder der übrigen $n-1$ Knoten kann jeweils mit höchstens $n-2$ weiteren Knoten verbunden sein.
Also kommen für $v_1, \dots, v_n$ als Grade entweder nur $0, 1, \dots, n-2$ oder $1, 2, \dots, n-1$ in Frage. Nach dem Schubfachprinzip (legt man n Dinge in $n-1$ Laden, so müssen in einer Lade mindestens zwei Dinge liegen) folgt daher $\exists v, w \in V$ mit $\delta(v) = \delta(w)$.

c. Ein endlicher Graph $G = \langle V, E \rangle$ ohne Mehrfachkanten mit $V = \{v_1, \dots, v_n\}$ kann angegeben werden durch seine *Adjazenzmatrix* $A = (a_{ij})$:

$$a_{ij} = \begin{cases} 1, & \text{falls es eine Kante von } v_i \text{ nach } v_j \text{ gibt} \\ 0 & \text{sonst} \end{cases}$$

$$v_i \text{ ist eine } \begin{cases} \text{Quelle} & \Leftrightarrow \quad i\text{–te Spalte ist } \vec{0} \\ \text{Senke} & \Leftrightarrow \quad i\text{–te Zeile ist } \vec{0} \end{cases}$$

G ungerichtet $\Leftrightarrow$ A symmetrisch

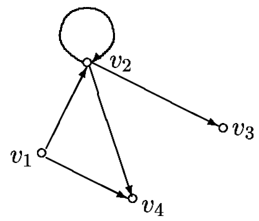
Ein Graph heißt *vollständig*, wenn jeder Knoten mit jedem anderen Knoten durch eine Kante verbunden ist.

Beispiel

a. Sei G ein vollständiger endlicher Graph ohne Mehrfachkanten

$$\Rightarrow \text{zugehörige Adjazenzmatrix: } A = \begin{pmatrix} 0 & 1 & 1 & \dots & 1 & 1 \\ 1 & 0 & 1 & \dots & 1 & 1 \\ \vdots & & & \ddots & & \vdots \\ 1 & 1 & 1 & \dots & 1 & 0 \end{pmatrix}$$

b. Adjazenzmatrix $A = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \begin{array}{c} \begin{array}{cccc} v_1 & v_2 & v_3 & v_4 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \end{array} \Rightarrow$ Graph



Sei $G = \langle V, E, f\rangle$ ein beliebiger Graph. Eine *Bewertung* (der Kanten) von G ist eine Abbildung $\psi : E \to \mathbb{R}$. $\Rightarrow$ bewerteter Graph $G = \langle V, E, f, \psi\rangle$

Beispiele für Bewertungen sind Weglängen, Fahrzeiten bei Straßennetzen, Kapazitäten bei Leitungsnetzen.

Angabe der Bewertung: Wert $\psi(e)$ wird oft neben Kante e geschrieben; bei Graphen ohne Mehrfachkanten:

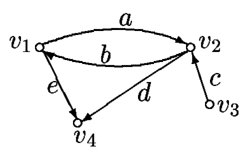
$$b_{ij} = \begin{cases} \psi(i,j) & \text{für } (i,j) \in E \\ \text{ein von der Problemstellung abhängiger Wert für } (i,j) \notin E, & \text{z.B. } 0 \text{ oder } \infty \end{cases}$$

Netzwerk: (zumeist) bewerteter Graph ohne isolierte Knoten

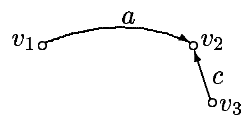
Ein Graph $T = \langle V(T), E(T), f_T\rangle$ heißt *Teilgraph* eines Graphen $G = \langle V(G), E(G), f_G\rangle$, falls $V(T) \subseteq V(G), E(T) \subseteq E(G)$ und $f_T = f_G|E(T)$ (Einschränkung von f_G auf $E(T)$).

Beispiel

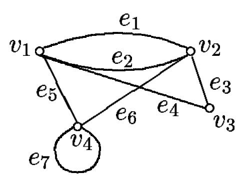
a.



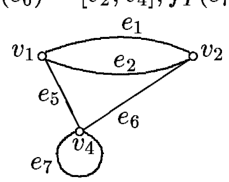
Teilgraph:
$T = \langle \{v_1, v_2, v_3\}, \{a, c\}, f_T\rangle$ mit
$f_T(a) = (v_1, v_2), f_T(c) = (v_3, v_2)$



b.



vollständiger Teilgraph:
$T = \langle \{v_1, v_2, v_4\}, \{e_1, e_2, e_5, e_6, e_7\}, f_T\rangle$ mit
$f_T(e_1) = f_T(e_2) = [v_1, v_2], f_T(e_5) = [v_1, v_4]$,
$f_T(e_6) = [v_2, v_4], f_T(e_7) = [v_4, v_4]$



Sei $G = \langle V, E, f\rangle$ ungerichtet. Eine *Kantenfolge* von G ist eine Folge $K : v_1, e_1, v_2, e_2, \ldots, e_m, v_{m+1},\ v_i \in V, e_i \in E$ mit $f(e_i) = [v_i, v_{i+1}]$. K heißt *geschlossen*, falls $v_1 = v_{m+1}$, sonst *offen*.

Kantenzug: Kantenfolge mit paarweise verschiedenen Kanten
Weg (von v_1 nach v_{m+1}): offener Kantenzug mit paarweise verschiedenen Knoten
Kreis: geschlossener Kantenzug mit paarweise verschiedenen Knoten (bis auf $v_1 = v_{m+1}$)
G gerichteter Graph $\Rightarrow$ Analoga (mit $f(e_i) = (v_i, v_{i+1})$): gerichtete Kantenfolge (*Pfeilfolge*), gerichteter Kantenzug, gerichteter Weg (*Pfad*), gerichteter Kreis (*Zyklus*)

Satz 2 Sei $G = \langle V, E, f\rangle$ ein endlicher, schlichter, gerichteter Graph, der keinen Zyklus enthält. Dann hat G mindestens eine Quelle und eine Senke.

Beweis Konstruiere ausgehend von einem beliebigen Knoten v einen beliebigen Pfad. Da G schlicht, zyklenfrei und endlich ist, muss man einmal zu einem Knoten q kommen, von dem aus keine Kante mehr wegführt. $\Rightarrow$ Senke
Ändere die Richtungen sämtlicher Kanten. $\Rightarrow$ Quelle

Ein ungerichteter Graph G heißt *zusammenhängend*, falls es von jedem Knoten von G zu jedem anderen Knoten von G einen Weg gibt.

Ein gerichteter Graph G heißt *schwach zusammenhängend*, falls G als ungerichteter Graph (vernachlässige Richtungen der Kanten) zusammenhängend ist. G heißt *stark zusammenhängend*, falls es von jedem Knoten von G zu jedem anderen Knoten von G eine Pfeilfolge gibt.

Baum: kreisloser, zusammenhängender, ungerichteter Graph
Wurzelbaum: Baum, in dem ein Knoten als *Wurzel* ausgezeichnet ist

3.2 Klassen von Graphen und deren Anwendungen

3.2.1 Bäume

Satz 3 In jedem Baum B gibt es zwischen zwei Knoten v und w genau einen Weg, der v und w verbindet.

Beweis B zusammenhängend $\Rightarrow$ Es gibt mindestens einen Weg von v nach w. B kreisfrei $\Rightarrow$ Es gibt keinen zweiten solchen Weg.

Satz 4 Jeder endliche Baum mit mehr als einem Knoten hat mindestens zwei Knoten vom Grad 1 (=*Endknoten*).

Beweis Angenommen, die Behauptung ist falsch. $\Rightarrow$ Es gibt einen Baum

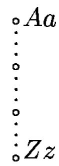
a. mit genau einem Endknoten p: Wir beginnen von p aus einen beliebigen Weg. Da $\delta(v) \geq 2 \ \forall v \in V, v \neq p$ können wir jeden Knoten, den wir über eine Kante erreicht haben, auf einer anderen Kante verlassen. $\Rightarrow B$ besitzt einen unendlichen Weg, Widerspruch.

b. mit keinem Endknoten: analog mit einem beliebigen Knoten p

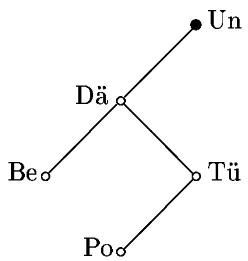
Satz 5 Ein Baum mit n Knoten hat genau $n-1$ Kanten.

Beweis durch vollständige Induktion nach n
Für $n = 1$ ist die Behauptung richtig.
Induktionsvoraussetzung: Behauptung gilt für ein $n \in \mathbb{N}$.
Sei B ein Baum mit $n+1$ Knoten. Dann hat B nach Satz 4 mindestens einen Endknoten v_e, welchen wir (samt damit inzidenter Kante e) entfernen. Dadurch wird aus B ein Graph B' mit n Knoten. Induktionsvoraussetzung $\Rightarrow$ B' hat $n-1$ Kanten. Fügen v_e (und die Kante e) hinzu. $\Rightarrow$ B hat n Kanten.

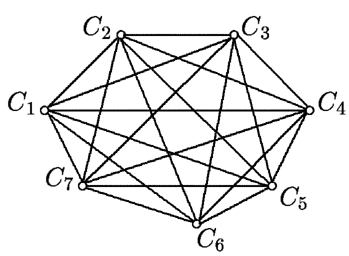
Folgerung Ist G ein ungerichteter, zusammenhängender Graph mit $|V(G)| = n$ und $|E(G)| = n-1$, dann ist G ein Baum.



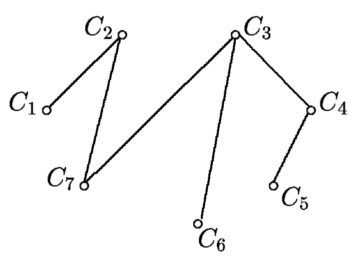
Beispiel Binäre Suchbäume
Ablage von Datensätzen; Datensatz = (Schlüssel, Information);
Schlüssel seien lexikographisch geordnet:
Ablageprinzip:
Der Schlüssel des ersten Datensatzes wird als Wurzel des binären Suchbaums (Wurzelbaum) ausgezeichnet. Unter einem Knoten (Schlüssel) werden der Reihe nach "links" nur kleinere, "rechts" nur Datensätze mit größeren Schlüsseln abgelegt.
Beispiel: Ablage der Ländernamen **Un**garn, **Dä**nemark, **Tü**rkei, **Po**len, **Be**lgien (Schlüssel fett) $\Rightarrow$ binärer Suchbaum:



Beispiel Minimale Vernetzung von Computern
n Computer $C_1, \ldots, C_n$ sollen so vernetzt werden, dass je zwei untereinander kommunizieren können. Verbindung von je zwei Computern $\Rightarrow$ vollständiger Graph K_n mit $\binom{n}{2} = \frac{n^2-n}{2}$ Leitungen
Problem: Suche *spannenden Baum* von K_n, d.h. einen Baum mit Knotenmenge $C_1, \ldots, C_n$.
Satz 3, Satz 5 $\Rightarrow$ Dieser spannende Baum hat $n-1$ Kanten, und je zwei C_i sind durch genau einen Weg verbunden.



vollständiger Graph K_7



spannender Baum

3.2.2 Eulersche und Hamiltonsche Graphen

Eulersche Linie: geschlossener Kantenzug in einem ungerichteten Graphen G, der jede Kante von G genau einmal enthält

Eulerscher Graph: Graph, der eine Eulersche Linie enthält

Satz 6 Ein endlicher, zusammenhängender, ungerichteter Graph ist genau dann ein Eulerscher Graph, wenn alle Knoten von G geraden Grad haben.

Beweis "$\Rightarrow$": Jedes Mal, wenn man in einer Eulerschen Linie über eine Kante zu einem Knoten v kommt, verlässt man v über eine andere Kante. $\Rightarrow \delta(v) = 2t$, wobei t angibt, wie oft man v passiert.
"$\Leftarrow$": ergibt sich auf Grund der Algorithmen zur Bestimmung Eulerscher Linien (Kapitel 4)

Hamiltonscher Kreis (Weg): Kreis (Weg) eines ungerichteten Graphen G, der alle Knoten von G enthält

Hamiltonscher Graph: ungerichteter Graph, der einen Hamiltonschen Kreis enthält

Jeder vollständige Graph K_n ist ein Hamiltonscher Graph.

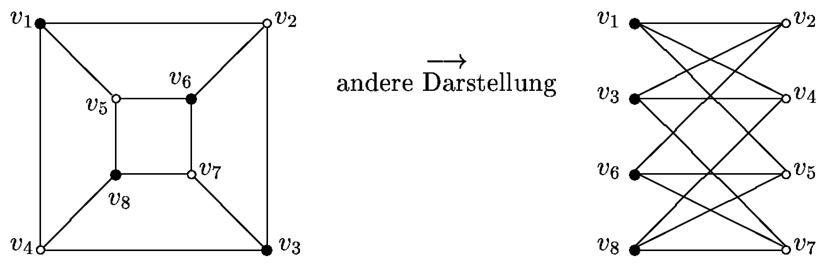
Problem des Handlungsreisenden: Existiert in einem Wegenetz eine Rundreise, d.h. ein Hamiltonscher Kreis?

Satz 7 Sei G ein ungerichteter, schlichter Graph mit $n \geq 3$ Knoten. Ist $\delta(v) \geq \frac{n}{2}$ für alle $v \in V$, dann ist G ein Hamiltonscher Graph. (ohne Beweis)

3.2.3 Bipartite (paare) Graphen

$G = \langle V, E, f \rangle$ ungerichtet. Kann V in zwei disjunkte Mengen V_1, V_2 derart zerlegt werden, dass es nur Kanten zwischen V_1 und V_2 gibt, so heißt G ***bipartit*** (*paarer Graph*).

Beispiel Kantengraph des Würfels



G ist *k–regulär*, falls $\delta(v) = k$ für alle $v \in V$.

Anwendung bipartiter Graphen: Lösen von Zuordnungsproblemen, insbesondere: Jedem $v \in V_1$ soll genau ein $w \in V_2$ zugeordnet werden (***Heiratsproblem***, *Matching–Problem*).

Matching M: Teilmenge der Kantemenge E eines bipartiten Graphen, so dass keine zwei Kanten aus M einen Knoten aus V gemeinsam haben.

maximales Matching: Matching mit möglichst vielen Kanten; $max_M|M|$ (für endliche bipartite Graphen)

Beispiel Mitarbeiter–Zuordnungsproblem
In einem Unternehmen ist für jede Arbeit J_i eine Menge $\{W_{i1}, \ldots, W_{ik_i}\}$ von Mitarbeitern geeignet. Wie groß ist die größte Anzahl von Jobs, die gleichzeitig ausgeführt werden können? $\Rightarrow$ Problem: Suche eines maximalen Matchings

Netzwerk: bewerteter, zusammenhängender, gerichteter, endlicher Graph $G = \langle V, E, f, \psi \rangle$ mit genau einer Quelle q und genau einer Senke s, sowie $\psi(e) \geq 0$ $\forall e \in E$.

Fluss F in einem Netzwerk G ist eine Abbildung $F : E \rightarrow \mathbb{R}^+$, so dass:

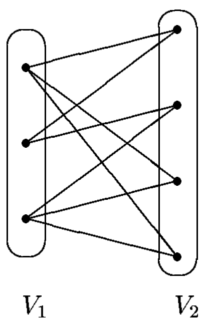
(1) $F(e) \leq \psi(e)\ \forall e \in E$

(2) $\forall v \in V$ gilt: $\sum_{u \in P(v)} F(u, v) = \sum_{w \in S(v)} F(v, w)$

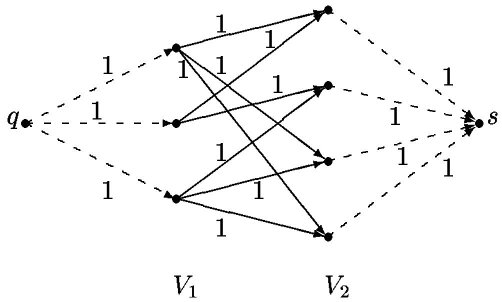
Wert des Flusses $W(F) := \sum_{w \in S(q)} F(q, w) = \sum_{u \in P(s)} F(u, s)$ (q, s Quelle bzw. Senke des Netzwerks)

Problem, ein maximales Matching zu finden, kann auf das Problem des *maximalen Flusses* F^* (=Fluß mit maximalem Wert $W(F^*) := max_F W(F)$) in einem Netzwerk zurückgeführt werden. (Algorithmen zur Bestimmung eines maximalen Flusses siehe Kapitel 4.)

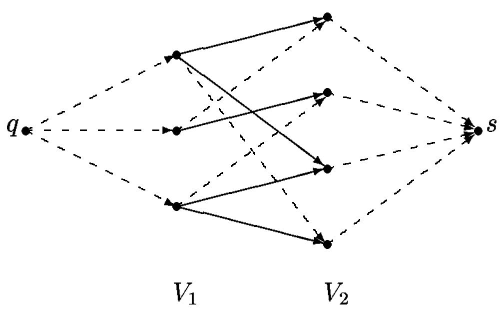
Vorgangsweise: Hinzufügen von Quelle q und Senke s und Orientierung der Kanten so, wie aus den nachstehenden Abbildungen hervorgeht; Bewertung aller Kanten mit 1 $\Rightarrow$ Netzwerk



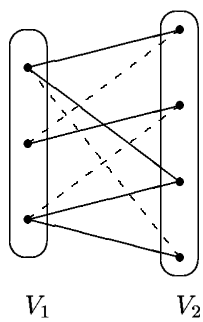
$\longrightarrow$



maximaler Fluß $\longrightarrow$ maximales Matching



strichliert: Kanten mit $F(e) = 1$,
übrige Kanten: $F(e) = 0$



Matching: strichlierte Kanten
zwischen V_1 und V_2

Ein endlicher Graph heißt *k–(Knoten–)färbbar*, falls man jedem Knoten eine von k Farben so zuordnen kann, daß die beiden Endknoten jeder Kante verschieden gefärbt sind. ($\Rightarrow$ Bipartite Graphen sind 2–Knoten–färbbar.) Die kleinste Anzahl k von Farben, mit der G färbbar ist, heißt *chromatische Zahl* $\chi(G)$ *von* G.

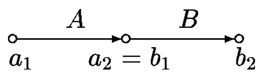
Beispiel Färbung von Landkarten
Länder, die eine gemeinsame Grenze haben, sollen verschieden gefärbt sein. Mit wie vielen Farben kommt man aus?
Vierfarbenvermutung seit 1852: Man kommt stets mit vier Farben aus. (1976 bewiesen von K. Appel und W. Haken)
Ordne den Ländern Knoten zu, und verbinde zwei Knoten, wenn die entsprechenden Länder eine gemeinsame Grenze haben (nicht nur in einem Eck zusammentreffen). $\Rightarrow$ *planarer (plättbarer)* Graph (= ein Graph, den man ohne einander überschneidende Kanten in der Ebene zeichnen kann)

Vierfarbensatz $\chi(G) \leq 4$ für jeden planaren Graphen G.

3.3 CPM– und MPM–Netzpläne

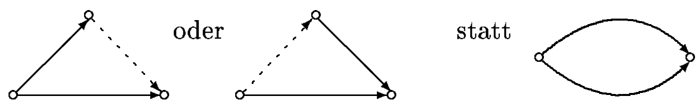
Projekt: Zeit und Ressourcen beanspruchende Tätigkeiten bzw. Arbeitsvorgänge (*Vorgänge*), zwischen denen gewisse *Anordnungsbeziehungen* bestehen (z.B. Bauvorhaben, Fertigungsvorhaben, Beschaffungsauftrag, Entwicklungsprojekt)
Netzplantechnik: optimale Planung und Überwachung der Ausführung eines Projekts

(1) *Vorgangspfeilnetz*: Vorgang $A \to$ gerichtete Kante (a_1, a_2); a_1, a_2, Ereignisse (Beginn bzw. Ende von A)

Ende–Start–Beziehung: 

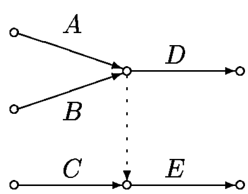
Vorgang B kann frühestens beginnen, wenn Vorgang A abgeschlossen ist, und B folgt als Vorgang unmittelbar (jedoch im allgemeinen nicht unmittelbar im zeitlichen Sinn) auf A.

Modellierung vielfach erst möglich durch Einführung von *Scheinvorgängen* $\dashrightarrow$ (Zeitdauer 0), z.B.



Beispiel

D, E folgen unmittelbar A, B nach, E ist auch unmittelbarer Nachfolger von C, C ist kein Vorgänger von D



Start–Start–Beziehung: $\circ \xrightarrow{A} \circ \xrightarrow{B} \circ$

B kann frühestens beginnen, wenn A bereits begonnen hat, und B folgt (als Vorgang) unmittelbar auf A.

Analog: *Start–Ende–*, *Ende–Ende–Beziehung*

(2) *Ereignisknotennetz*: Ereignis → Knoten a; Ereignisse müssen nicht notwendigerweise Anfang oder Ende einer Tätigkeit sein (z.B. Ereignis "Dachgleiche"); Ereignis a folgt unmittelbar nach b: $\underset{a}{\circ} \longrightarrow \underset{b}{\circ}$

(3) *Vorgangsknotennetz*: Vorgang → Knoten A; Start–Start–Beziehung

$A \xrightarrow{t} B$ Vorgang B kann frühestens t Zeiteinheiten nach Beginn des Vorgangs A anfangen.

$A \xleftarrow{-T} B$ Vorgang B muß spätestens T Zeiteinheiten nach Beginn des Vorgangs A beginnen.

$A \underset{-t}{\overset{t}{\rightleftarrows}} B$ B muß genau t Zeiteinheiten nach dem Beginn von A anfangen.

Wir unterscheiden: Zeit– und Terminplanung, Kostenplanung, Kapazitätsplanung. Dabei können Anordnungsbeziehungen, Dauern, Kosten, Kapazitäten mit gewissen Wahrscheinlichkeiten behaftet sein:
Projektablauf nicht determiniert → Entscheidungsnetzplan (z.B. GERT: Graphical Evaluation and Review Technique)

Projektablauf determiniert, Zeitdauern Zufallsvariable → Planung meist mit PERT (Program Evaluation and Review Technique)

Projektablauf und Zeitdauern der Vorgänge determiniert → *CPM* (*Critical Path Method*), *MPM* (*Metra Potential Method*)

Gesucht: kürzeste Gesamtdauer eines Projekts, frühest und spätest mögliche Anfangs– und Endzeitpunkte der einzelnen Vorgänge, Pufferzeiten für die Vorgänge, *kritische Vorgänge* (Vorgänge, deren Verlängerung/Verschiebung die kürzest mögliche Gesamtprojektdauer verlängert)

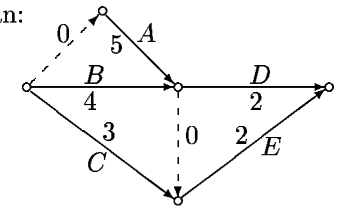
CPM–Netzplan: Vorgangspfeilnetz, Ende–Start–Beziehung; gegeben: Zeitdauern D_{ij} der Vorgänge (i, j) → Kantenbewertung;
Netzplan wird so konstruiert, daß ein zyklenfreier, schwach zusammenhängender Graph mit genau einer Quelle und genau einer Senke entsteht.

MPM–Netzplan: Vorgangsknotennetz, Start–Start–Beziehung;
Netzplan ist schwach zusammenhängend, hat positive und negative Bewertungen, beinhaltet Zyklen (nie mit positiver Länge).

Beispiel CPM / MPM–Netzplan
Die Vorgänge bei einem Projekt seien A, B, C, D, E und sollen die Zeitdauern 5,4,3,2 und 2 Wochen haben. Den Aktivitäten A, B und C gehe keine (Zeit

beanspruchende) Tätigkeit voraus (*Startvorgänge*), und auf D und E folge kein weiterer Vorgang (*Zielvorgänge*). D und E sollen unmittelbar nach A und B folgen; der Vorgang E sei darüber hinaus auch ein unmittelbarer Nachfolger von C, der aber kein Vorgänger von D ist.

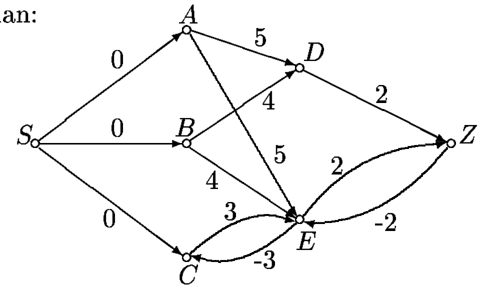
CPM–Netzplan:



Nun verlangen wir zusätzlich, daß die Vorgänge C und E möglichst spät, d.h., soweit dies verschoben in Richtung des Gesamtabschlusses nur geht, beginnen sollen. Dies bedeutet, daß das Projektende *zeitlich* unmittelbar auf E und E *zeitlich* unmittelbar auf C folgen soll.

→ MPM–Netzplan: Vorgänge sind nun Knoten, Start–Start–Beziehung, wobei Anfangszeit des Nachfolgers = frühestens Anfangszeit des Vorgängers + Vorgangsdauer des Vorgängers

MPM–Netzplan:



Gesucht bei CPM mit Knoten $1, 2, \ldots, n$ (Ereignisse), Quelle 1, Senke n, $D_{ij} :=$ Dauer des Vorgangs (i, j):

- FAZ_{ij}, FEZ_{ij}: frühest mögliche Anfangs– bzw. Endzeitpunkte
- SAZ_{ij}, SEZ_{ij}: spätest mögliche Anfangs– bzw. Endzeitpunkte

Hilfsgrößen:

- FZ_i: frühest möglicher Zeitpunkt für den Eintritt des Ereignisses $i \Rightarrow$ $FZ_i = FAZ_{ij}$ und $FEZ_{ij} = FZ_i + D_{ij}$.
- SZ_j: spätest möglicher Zeitpunkt für den Eintritt des Ereignisses $j \Rightarrow$ $SZ_j = SEZ_{ij}$ und $SEZ_{ij} = SZ_j - D_{ij}$

Gesamtpuffer (eines Vorgangs (i, j)): $GP_{ij} := SZ_j - FZ_i - D_{ij}$
Es gilt unter der Annahme $FZ_1 = 0$:
$FZ_i =$ Länge eines längsten Weges von 1 nach i
$SZ_n - SZ_j =$ Länge eines längsten Weges von j nach n
$FZ_n =$ kürzeste Gesamtprojektdauer
Ein Vorgang (i, j) ist ***kritisch***, falls $GP_{ij} = 0$.
Länge eines Weges = Summe der Kantenbewertungen
Jede Kante (i, j) entlang eines längsten Pfades von 1 nach n ist kritisch. → ***kritischer Pfad***

Gesucht bei MPM mit Knoten $1, ..., n$ (Vorgänge):

- FAZ_i, FEZ_i: frühest mögliche Anfangs– bzw. Endzeitpunkte
- SAZ_i, SEZ_i: spätest mögliche Anfangs– bzw. Endzeitpunkte

Es gilt unter der Annahme $FAZ_1 = 0$ (analog zu CPM):
$FAZ_i =$ Länge eines längsten Weges von 1 nach i
$SAZ_n - SAZ_j =$ Länge eines längsten Weges von j nach n

Weglängen können auch negativ sein! $\rightarrow$ Man benötigt Algorithmen für längste Pfade in – auch negativ – bewerteten Netzwerken.