

**GRAFENTHEORIE**

**L.C.M. KALLENBERG**

**UNIVERSITEIT LEIDEN**

**VOORJAAR 2002**

# Inhoudsopgave

<b>1 Grafen en matrices</b>	<b>1</b>
1.1 Grafen en vectorruimtes . . . . .	1
1.2 De incidentiematrix . . . . .	2
1.3 De kringenmatrix . . . . .	6
1.4 De snedenmatrix . . . . .	9
1.5 De padenmatrix . . . . .	11
1.6 De structuurmatrix . . . . .	12
1.7 Opgaven . . . . .	14
<b>2 Standaard algoritmen</b>	<b>18</b>
2.1 Voorwaarts zoeken . . . . .	18
2.2 Zijwaarts zoeken . . . . .	23
2.3 Streng samenhangende componenten . . . . .	24
2.4 Opgaven . . . . .	28
<b>3 Kortste paden</b>	<b>30</b>
3.1 Inleiding . . . . .	30
3.2 Methode van Bellman en Ford . . . . .	31
3.3 Methode van Floyd en Warshall . . . . .	32
3.4 Enkele toepassingen . . . . .	34
3.5 Opgaven . . . . .	37
<b>4 Netwerkstromen</b>	<b>39</b>
4.1 Inleiding . . . . .	39
4.2 Methode van Ford en Fulkerson . . . . .	41
4.3 Methode van Dinic, Malhotra, Kumar en Maheshwari . . . . .	44
4.4 Stromen met onder- en bovengrenzen . . . . .	48
4.5 Minimale kostenstromen . . . . .	50
4.6 Minimale circulatiestromen in netwerk met onder- en bovengrenzen . . . . .	53
4.7 Enkele toepassingen . . . . .	59
4.8 Opgaven . . . . .	62

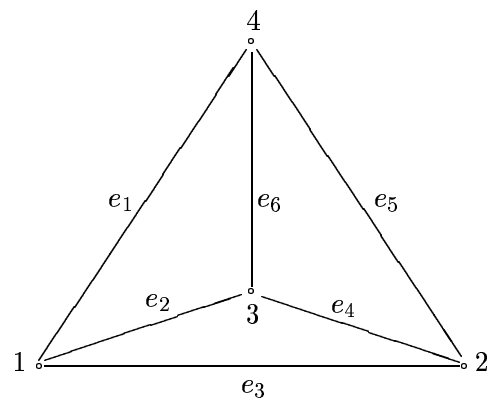
<b>5</b>	<b>Netwerk simplex methode</b>	<b>68</b>
5.1	Inleiding . . . . .	68
5.2	Bases en opspannende bomen . . . . .	70
5.3	Algoritme voor problemen zonder capaciteiten . . . . .	72
5.4	Problemen met onder- en bovengrenzen . . . . .	77
5.5	Maximale stroom probleem . . . . .	80
5.6	Opgaven . . . . .	83
<b>6</b>	<b>Koppelingen</b>	<b>86</b>
6.1	Inleiding . . . . .	86
6.2	Equivalente combinatorische resultaten . . . . .	92
6.3	Koppeling met maximale cardinaliteit in bipartiete graaf . . . . .	97
6.4	Koppeling met maximale gewicht in bipartiete graaf . . . . .	102
6.5	Gilmore-Gomory en Gale-Shapley koppelingen . . . . .	105
6.6	Koppeling met maximale cardinaliteit in een willekeurige graaf . . . . .	108
6.7	Koppeling met maximaal gewicht in een willekeurige graaf . . . . .	115
6.8	Opgaven . . . . .	120
<b>7</b>	<b>Literatuur</b>	<b>125</b>

# Hoofdstuk 1

## Grafen en matrices

### 1.1 Grafen en vectorruimtes

Indien men een bepaalde graaf  $G = (V, E)$  wil aanduiden, dan gebeurt dat vaak door een plaatje, zoals hiernaast, of door een bepaalde notatie ( $K_4$ ). Indien de computer wordt gebruikt, bijv. als een algoritme wordt gebruikt om de kortste paden in een netwerk te bepalen, dan wordt de graaf gewoonlijk gepresenteerd met een bepaalde *datastructuur*, zoals een *lijst* of een *matrix*.



Een graaf kan bijvoorbeeld worden gerepresenteerd door een matrix die voor iedere knooppunt een rij heeft en voor iedere tak een kolom. De kolom behorende bij een zekere tak heeft 1'en in de twee rijen behorende bij de eindpunten van de tak en verder 0'en. Voor de graaf  $K_4$  staat deze matrix hiernaast.

$$\begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

Laat ons wederom de graaf  $K_4$  beschouwen. Iedere deelverzameling van de zes takken kan worden gerepresenteerd door een zestal  $(x_1, x_2, x_3, x_4, x_5, x_6)$  met  $x_i = \begin{cases} 1 & \text{als } e_i \text{ tot de deelverz. behoort} \\ 0 & \text{anders} \end{cases}$

Zo wordt de deelverz.  $\{e_1, e_3, e_4\}$  gerepresenteerd door  $(1, 0, 1, 1, 0, 0)$ . Voor de  $K_4$  zijn er  $2^6$  van dergelijke zestallen.

In het algemeen hebben we bij een graaf met  $m$  takken  $2^m$   $m$ -tallen bestaande uit 0'en en 1'en. Zij  $W(G)$  de verz. van deze  $m$ -tallen. Laat  $\mathbb{Z}_2 = \{0, 1\}$  met optelling en vermenigvuldiging modulo 2; we noteren deze optelling en vermenigvuldiging met  $\oplus$  resp.  $\otimes$ . Deze optelling en vermenigvuldiging induceert een optelling op  $W(G)$  en een vermenigvuldiging van de elementen van  $W(G)$  met een  $\alpha \in \mathbb{Z}_2$  door:

$$x + y = (x_1 \oplus y_1, x_2 \oplus y_2, \dots, x_m \oplus y_m) \text{ en } \alpha \cdot x = (\alpha \otimes x_1, \alpha \otimes x_2, \dots, \alpha \otimes x_m).$$

**Stelling 1.1**  $W(G)$  is een  $m$ -dimensionale vectorruimte over het lichaam  $\mathbb{Z}_2$ .

### Bewijs

De gewenste eigenschappen zijn eenvoudig te verifiëren (ga dit zelf na). ■

We introduceren de volgende deelverz. van  $W(G)$ :

$W_C(G)$ : de  $m$ -tallen van  $W(G)$  die behoren bij de nulgraaf, de enkelvoudige kringen en de vereniging van tak-disjuncte kringen.

$W_S(G)$ : de  $m$ -tallen van  $W(G)$  die behoren bij de nulgraaf, de minimale sneden en de vereniging van tak-disjuncte sneden.

**Stelling 1.2**  $W_C(G)$  en  $W_S(G)$  zijn lineaire deelruimtes van  $W(G)$ .

### Bewijs

Uit de axioma's volgt dat het voldoende is om aan te tonen dat de som van twee elementen van  $W_C(G)$  of van  $W_S(G)$  weer tot  $W_C(G)$  resp.  $W_S(G)$  behoort.

Merk op dat een element tot  $W_C(G)$  behoort d.e.s.d. als ieder knooppunt in de bijbehorende graaf een even graad heeft. Vanwege de optelling modulo twee volgt hieruit direct dat als  $x, y \in W_C(G)$ , ook  $x + y \in W_C(G)$ .

Beschouw vervolgens twee sneden:  $S_1 = (V_1, V_2)$  en  $S_2 = (V_3, V_4)$ . Laat  $A = V_1 \cap V_3, B = V_1 \cap V_4, C = V_2 \cap V_3, D = V_2 \cap V_4, V_5 = A \cup D$  en  $V_6 = B \cup C$ . Zij  $x$  en  $y$  de vectoren behorende bij  $S_1$  resp.  $S_2$ , en laat  $z = x + y$ . De bij  $z$  behorende takkenverz. bestaat uit de takken die lopen tussen  $V_5$  en  $V_6$  (ga dit zelf na):  $z$  behoort dus tot  $W_S(G)$ . ■

## 1.2 De incidentiematrix

Zij  $G = (V, E)$  een graaf met  $n$  knooppunten,  $m$  takken en geen lussen. De *incidentiematrix*  $A(G)$  is een  $n \times m$ -matrix die het verband aangeeft tussen de knooppunten en de takken, en als volgt is gedefinieerd:

niet-gerichte graaf:  $a_{ij} = \begin{cases} 1 & \text{als tak } j \text{ knooppunt } i \text{ als eindpunt heeft} \\ 0 & \text{anders} \end{cases}$

gerichte graaf:  $a_{ij} = \begin{cases} +1 & \text{als pijl } j \text{ knooppunt } i \text{ als beginpunt heeft} \\ -1 & \text{als pijl } j \text{ knooppunt } i \text{ als eindpunt heeft} \\ 0 & \text{anders} \end{cases}$

De matrix op pagina 1 is de incidentiematrix van de  $K_4$ .

Een matrix heet *totaal unimodulair* als de determinant van iedere vierkante deelmatrix  $+1, -1$  of  $0$  is. Dus ook alle elementen van een totaal unimodulaire matrix zijn  $+1, -1$  of  $0$ .

**Stelling 1.3** *De incidentiematrix  $A(G)$  van een niet-gerichte graaf  $G$  is totaal unimodulair d.e.s.d. als  $G$  bipartiet is.*

**Bewijs**

$\Rightarrow$  Veronderstel eerst dat  $A(G)$  totaal unimodulair is en  $G$  niet bipartiet. Dan bevat  $G$  een kring van oneven lengte, zeg  $[v_1, v_2, \dots, v_{2p+1}, v_1]$ . Beschouw de vierkante deelmatrix van  $A(G)$  voortgebracht door deze kring:

$$\begin{pmatrix} 1 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & \dots & 0 & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & \dots & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 & 1 & 1 \end{pmatrix}$$

Om de determinant van deze deelmatrix te bepalen ontwikkelen we hem naar de eerste rij: waarde determinant =  $1 + (-1)^{2p} \cdot 1 = 2$ . De matrix  $A(G)$  is dus niet totaal unimodulair.

$\Leftarrow$  Veronderstel vervolgens dat  $G$  bipartiet is, en dat de takken tussen  $V_1$  en  $V_2$  lopen. Neem een willekeurige  $k \times k$ -deelmatrix  $A_k$  van  $A(G)$ . We bewijzen de bewering met inductie naar  $k$ . Merk op dat het klopt voor  $k = 1$ . Neem nu  $k$  algemeen en kies een willekeurige kolom van  $A_k$ . Het aantal 1'en in deze kolom is 0, 1 of 2. Als de kolom geen 1'en bevat, dan is  $\det(A_k) = 0$ . Als de kolom één 1 bevat, dan ontwikkelen we de determinant naar deze kolom: de determinant is dan  $\pm \det(A_{k-1})$  voor de matrix  $A_{k-1}$  die uit  $A_k$  ontstaat door de rij en kolom die behoort bij de 1 uit de gekozen kolom weg te laten. Volgens de inductieveronderstelling klopt het gestelde. Op deze wijze is de stelling te bewijzen voor iedere deelmatrix  $A_k$  waarin niet alle kolommen twee 1'en bevatten. Beschouw tenslotte een  $A_k$  met in iedere kolom precies twee 1'en. Eén van deze hoort dus bij  $V_1$  en de andere bij  $V_2$ . Maar dan is de som van de rijen van  $A_k$  die behoren bij  $V_1$  gelijk aan de som van de rijen van  $A_k$  die behoren bij  $V_2$ , namelijk een vector met overal 1. Er is dus afhankelijkheid:  $\det(A_k) = 0$ . ■

**Stelling 1.4** *Voor een niet-gerichte graaf  $G$  geldt: een verz. kolomvectoren van  $A(G)$  is lineair onafhankelijk d.e.s.d. als de bijbehorende takken geen kring bevatten.*

**Bewijs**

$\Rightarrow$  Stel dat de bijbehorende takken een kring bevatten. Als we de kolommen van de kring bij elkaar optellen (modulo 2), dan geeft dit de 0-vector omdat ieder knooppunt incident is met een even aantal takken van de kring.

$\Leftarrow$  Stel dat de som van een aantal kolomvectoren de 0-vector oplevert. Dan vormen de bijbehorende takken een deelgraaf zdd. ieder knooppunt incident is met een even aantal takken. Dit is een Eulergraaf en deze bevat een kring; tegenspraak. ■

**Stelling 1.5** *De incidentiematrix  $A(G)$  van een gerichte graaf  $G$  is totaal unimodulair.*

### Bewijs

Neem een vierkante  $k \times k$ -deelmatrix van  $A(G)$ . We passen weer inductie toe naar  $k$  en analoog aan het bewijs van de vorige stelling is het in te zien dat het klopt, maar alleen het geval dat iedere kolom van  $A_k$  precies twee niet-nul elementen bevat moeten we nog bekijken. Echter, in dat geval heeft iedere kolom precies één  $+1$  en één  $-1$ , zodat het optellen van alle rijen de nulvector oplevert: afhankelijkheid, zodat  $\det(A_k) = 0$ . ■

**Stelling 1.6** *Voor een gerichte graaf  $G$  geldt: een verz. kolomvectoren van  $A(G)$  is lineair onafhankelijk d.e.s.d. als de bijbehorende pijlen geen kring bevatten.*

### Bewijs

$\Rightarrow$  Stel dat de bijbehorende pijlen een kring bevatten (we mogen wel aannemen dat de kring enkelvoudige kring is met knooppunten  $v_1, v_2, \dots, v_k, v_1$ ). Doorloop deze kring in de richting  $v_1 \rightarrow v_2 \rightarrow \dots \rightarrow v_k \rightarrow v_1$  en geef de  $i$ -de pijl gewicht  $\lambda_i = +1$  als de richting van de pijl overeenkomt met de richting waarin de kring wordt doorlopen; in het andere geval geven we de  $i$ -de pijl gewicht  $\lambda_i = -1$ . De matrix van de kolommen  $\lambda_i a_i$ , met  $a_i$  de kolom van de incidentiematrix behorende bij de  $i$ -de pijl van de kring, heeft de volgende gedaante:

$$\begin{pmatrix} \lambda_1 a_1 & \lambda_2 a_2 & \cdots & \lambda_{k-1} a_{k-1} & \lambda_k a_k \\ 1 & & & & -1 \\ -1 & 1 & & & \\ & -1 & & & \\ & & & 1 & \\ & & & -1 & 1 \\ 0 & 0 & \cdots & 0 & 0 \\ \cdot & \cdot & \cdots & \cdot & \cdot \\ \cdot & \cdot & \cdots & \cdot & \cdot \\ 0 & 0 & \cdots & 0 & 0 \end{pmatrix}$$

Als we de kolommen van de kring bij elkaar optellen, dan geeft dit de 0-vector, wat in strijd is met de lineaire onafhankelijkheid.

$\Leftarrow$  Stel dat er een lineaire combinatie van een aantal kolomvectoren is die de 0-vector oplevert. Als een pijl het gewicht  $\lambda_i$  heeft gekregen, dan dupliceren we deze pijl  $|\lambda_i|$  keer en we draaien de richting om als  $\lambda_i < 0$ . Omdat in ieder knooppunt het aantal positieve 1'en moet worden gecompenseerd met evenveel -1'en, heeft de graaf in ieder knooppunt evenveel uitgaande als binnekomende pijlen. Dus vormen deze pijlen een gerichte Eulergraaf en deze bevat een ronde. Dus de oorspronkelijke graaf (zonder het omdraaien van pijlen) bevat een kring: tegenspraak. ■

Veronderstel dat  $G$  een niet-gerichte graaf is met  $p$  componenten. Door de knooppunten en de takken geschikt, in volgorde van de componenten, te nummeren kan  $A(G)$  in blokvorm worden geschreven, met  $A_i$  de incidentiematrix van de  $i$ -de component:

$$A(G) = \begin{pmatrix} A_1 & 0 & 0 & \dots & 0 & 0 \\ 0 & A_2 & 0 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & A_{p-1} & 0 \\ 0 & 0 & 0 & \dots & 0 & A_p \end{pmatrix} \quad (1.1)$$

**Stelling 1.7** *Als  $G$  een niet-gerichte graaf is met  $n$  knooppunten en  $p$  componenten, dan is in de vectorruimte  $W(G)$  de rang van  $A(G)$  gelijk aan  $n - p$ .*

**Bewijs**

Op grond van de schrijfwijze (1.1) is het voldoende om te bewijzen dat de rang  $n - 1$  is voor een samenhangende graaf. Stel dus dat  $G$  samenhangend is. Omdat iedere kolom twee 1'en heeft, is de som van alle rijen 0 (de optelling in  $W(G)$  is modulo 2), zodat de rang hoogstens  $n - 1$  is. Stel vervolgens dat er  $k$  rijen zijn, met  $k < n$ , die optellen tot nul. Dan geldt voor iedere tak: ofwel beide uiteinden behoren bij deze  $k$  rijen, ofwel geen der uiteinden behoort bij deze  $k$  rijen. De knooppunten bij deze  $k$  rijen genereren dus een deelgraaf met één of meer componenten: tegenspraak met de samenhang van  $G$ . ■

**Stelling 1.8** *Zij  $G$  een niet-gerichte graaf met  $n$  knooppunten en laat  $A_1$  een  $(n - 1) \times (n - 1)$  deelmatrix van  $A(G)$  zijn. Dan geldt:  $A_1$  heeft onafhankelijke rijen d.e.s.d. als de  $n - 1$  kolommen van  $A(G)$  die behoren bij  $A_1$  corresponderen met de takken van een opspannende boom van  $G$ .*

**Bewijs**

Zij  $G_1$  de deelgraaf van  $G$  voortgebracht door de  $n - 1$  takken die bij  $A_1$  behoren.

⇒ Omdat  $n - 1 \geq \text{rang } A(G_1) \geq \text{rang } A_1 = n - 1$ , geldt dat  $\text{rang } A(G_1) = n - 1$ , zodat - op grond van Stelling 1.7 -  $G_1$  samenhangend is. Een samenhangende graaf met  $n - 1$  takken is een boom.

⇐ Omdat  $G_1$  een boom is, is  $G_1$  samenhangend, zodat - weer op grond van Stelling 1.7 -  $\text{rang } A(G_1) = n - 1$ . Uit het bewijs van Stelling 1.7 volgt tevens dat iedere deelmatrix van  $n - 1$  rijen lineair onafhankelijk is. ■

Veronderstel vervolgens dat  $G$  een gerichte graaf is. De rijen van  $A(G)$  zullen we nu beschouwen in de vectorruimte  $\mathbb{R}^m$  met de gebruikelijke optelling en vermenigvuldiging.

**Stelling 1.9** *Als  $G$  een gerichte graaf is met  $n$  knooppunten,  $m$  pijlen en  $p$  componenten, dan is in de vectorruimte  $\mathbb{R}^m$  de rang van  $A(G)$  gelijk aan  $n - p$ .*



## Bewijs

Analoog aan het bewijs van Stelling 1.7 is het voldoende om het te bewijzen voor een samenhangende graaf. Omdat iedere kolom één  $+1$  en één  $-1$  bevat, tellen de rijen op tot de 0-vector:  $\text{rang } A(G) \leq n - 1$ . Veronderstel vervolgens dat er een lineaire combinatie is van  $k$  rijen, met  $k < n$ , die de 0-vector opleveren. Dan geldt voor iedere pijl: òfwel beide uiteinden behoren (met dezelfde positieve factor) bij deze  $k$  rijen òfwel geen der uiteinden behoort bij deze  $k$  rijen. De knooppunten bij de  $k$  rijen genereren dus een deelgraaf met één of meer componenten: tegenspraak met de samenhang van  $G$ . ■

## 1.3 De kringenmatrix

Zij  $G$  een samenhangende, niet-gerichte graaf met  $n$  knooppunten en  $m$  takken. Laat  $T$  een willekeurige opspannende boom zijn.  $T$  bestaat dus uit  $n - 1$  takken, zeg  $e_{m-n+2}, e_{m-n+3}, \dots, e_m$ . Door aan  $T$  de tak  $e_k (1 \leq k \leq m - n + 1)$  toe te voegen ontstaat precies één kring. De aldus te construeren  $m - n + 1$  kringen hebben lineair onafhankelijke vectoren in  $W_C(G)$ , namelijk:  $e_k$  komt alleen voor in de  $k$ -de kring,  $k = 1, 2, \dots, m - n + 1$ .

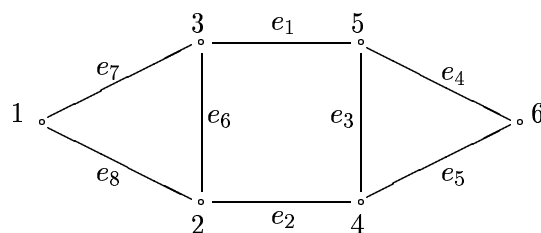
De bij  $T$  behorende *kringenmatrix*  $C_T(G)$  is de matrix met als rijen deze  $m - n + 1$  vectoren.  $C_T(G)$  heeft dus de gedaante:

$$C_T(G) = \left( \begin{array}{cccc|cccc} e_1 & e_2 & \cdots & e_{m-n+1} & e_{m-n+2} & \cdots & e_m & \\ \hline & & & I_{m-n+1} & & & D_T(G) & \end{array} \right) \quad (1.2)$$

### Voorbeeld 1.1

Beschouw nevenstaande graaf en neem als opspannende boom  $T = \{e_2, e_4, e_5, e_7, e_8\}$ . Dit geeft als kringenmatrix (de kringen ontstaan door achtereenvolgens  $e_1, e_3$  en  $e_6$  aan  $T$  toe te voegen):

$$C_T(G) = \left( \begin{array}{ccc|cccc} e_1 & e_3 & e_6 & e_2 & e_4 & e_5 & e_7 & e_8 \\ \hline 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \end{array} \right)$$



De vector van de kring  $[e_1, e_3, e_2, e_8, e_7]$  wordt verkregen door de kringvectoren van  $e_1$  en  $e_3$  (de niet tot  $T$  behorende takken van de kring) bij elkaar op te tellen (modulo 2). De volgende stelling toont aan dat dit algemeen geldt.

**Stelling 1.10** *Zij  $C$  een (som van tak-disjuncte) kring(en), bestaande uit de takken  $\{e_j \mid j \in J\}$ , en laat  $c$  de bijbehorende vector van  $W_C(G)$  zijn. Zij  $J^* = \{j \in J \mid e_j \notin T\} = \{j_1, j_2, \dots, j_k\}$  met  $j_i$  de index van de rij van  $C_T(G)$  behorende bij het  $i$ -de element van  $J^*$ . Dan geldt:  $c = \sum_{i=1}^k [C_T(G)]_{j_i, \bullet}$ , waarbij  $[C_T(G)]_{j_i, \bullet}$  de  $j_i$ -de rij van  $C_T(G)$  is.*

## Bewijs

Beschouw de vector  $c + \sum_{i=1}^k [C_T(G)]_{j_i \bullet}$ . Dit is weer een element van  $W_C(G)$ , dus de corresponderende takkenverz. is òf leeg, òf bevat een kring. Kies een  $p$  zdd.  $e_p \notin T$ .

Als  $e_p \in C : p \in J^*$ , zeg  $p$  behoort bij  $j_s : \{c + \sum_{i=1}^k [C_T(G)]_{j_i \bullet}\}_p = c_p \oplus [C_T(G)]_{j_s p} = 1 \oplus 1 = 0$ .

Als  $e_p \notin C : p \notin J^* : \{c + \sum_{i=1}^k [C_T(G)]_{j_i \bullet}\}_p = c_p = 0$ .

Hieruit volgt:  $\{c + \sum_{i=1}^k [C_T(G)]_{j_i \bullet}\}_p = 0$  voor alle  $p$  met  $e_p \notin T$ . De takken bij de niet-nul elementen van  $c + \sum_{i=1}^k [C_T(G)]_{j_i \bullet}$  zitten in  $T$ .  $T$  bevat geen kring, zodat ook hiervoor  $c + \sum_{i=1}^k [C_T(G)]_{j_i \bullet} = 0$ . Omdat  $c + \sum_{i=1}^k [C_T(G)]_{j_i \bullet}$  een  $(0,1)$ -vector is geldt dus :  
 $c = \sum_{i=1}^k [C_T(G)]_{j_i \bullet}$ . ■

## Gevolg

De rijen van  $C_T(G)$  vormen een basis voor  $W_C(G)$  en de vectorruimte  $W_C(G)$  heeft dimensie  $m - n + 1$ .

**Stelling 1.11** *Als  $c \in W_C(G)$ , dan is  $A(G) \cdot c = 0$  (modulo 2).*

## Bewijs

$[A(G) \cdot c]_i = \sum_j a_{ij} c_j =$  het aantal takken op  $C$  dat met  $v_i$  incident is  $= 0 \pmod{2}$ ,  $1 \leq i \leq n$ , want een kring heeft een even aantal gemeen met ieder knooppunt. ■

## Gevolg

$A(G) \cdot [C_T(G)]^t = 0$ , waarbij  $[C_T(G)]^t$  de getransponeerde matrix van  $[C_T(G)]$  is.

Tenslotte zullen we laten zien hoe de kringenmatrix uit de incidentiematrix kan worden bepaald. Neem daartoe een voortbrengende boom  $T$ . In Stelling 1.8 is aangetoond dat de bij  $T$  behorende takken een  $(n-1) \times (n-1)$  deelmatrix van onafhankelijke rijen opleveren. Na henummering (zet de takken van de boom weer achteraan) kunnen we schrijven:

$A(G) = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}$  en  $C_T(G) = \begin{pmatrix} I_{m-n+1} & D_T(G) \end{pmatrix}$ , met  $A_{12}$  de niet-singuliere  $(n-1) \times (n-1)$  deelmatrix behorende bij de takken van  $T$ . Omdat  $A(G) \cdot [C_T(G)]^t = 0$  (zie het gevolg van Stelling 1.11) kunnen we schrijven:

$$A_{11} I_{m-n+1} + A_{12} [D_T(G)]^t = 0 \rightarrow [D_T(G)]^t = -A_{12}^{-1} A_{11} = A_{12}^{-1} A_{11} \pmod{2}.$$

De inverse  $A_{12}^{-1}$  kan via een speciale methode snel worden bepaald. Deze methode maakt gebruik van de speciale structuur van  $A_{12}$ . Als  $X = A_{12}^{-1}$ , dan is  $X$  de unieke oplossing van  $X A_{12} = I$ , d.w.z.  $X [A_{12}]_{\bullet i} = e_i$ , voor alle  $i$ . Omdat  $A_{12}$  één rij minder heeft dan  $A(G)$ , is er minstens één kolom van  $A_{12}$  met precies één 1 en verder 0'en. Kies zo'n kolom, zeg  $[A_{12}]_{\bullet i} = e_j$ . Dan geldt:  $e_i = X [A_{12}]_{\bullet i} = X e_j = X_{\bullet j}$ . De  $j$ -de kolom van  $A_{12}^{-1}$  is dus  $e_i$ . Aldus zijn alle kolommen van  $A_{12}^{-1}$  te bepalen die in  $A_{12}$  corresponderen met een eenheidskolom. Via de 'verbindingen' met deze kolommen, d.w.z. kolommen die in dezelfde rij een 1 hebben als de rij van de eenheidsvector waarin de 1 staat, kunnen we volgende kolommen van de inverse bepalen. Stel  $[A_{12}]_{\bullet i} = e_j + e_k$

en  $X_{\bullet k}$  is bekend. Dan geldt:  $e_i = X[A_{12}]_{\bullet i} = X e_j + X e_k = X_{\bullet j} + X_{\bullet k}$ , zodat  $X_{\bullet j} = e_i + X_{\bullet k}$  en deze vectoren zijn bekend. Zo doorgaande kan  $A_{12}^{-1}$  efficiënt worden bepaald.

### Voorbeeld 1.1 (vervolg)

$$A_{12} = \begin{pmatrix} e_2 & e_4 & e_5 & e_7 & e_8 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix} \text{ en } A_{11} = \begin{pmatrix} e_1 & e_3 & e_6 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \end{pmatrix}.$$

$$[A_{12}]_{\bullet 2} = e_5 \rightarrow e_2 = [A_{12}^{-1}]_{\bullet 5}.$$

$$[A_{12}]_{\bullet 3} = e_4 \rightarrow e_3 = [A_{12}^{-1}]_{\bullet 4}.$$

$$[A_{12}]_{\bullet 1} = e_4 + e_2 \rightarrow e_1 = [A_{12}^{-1}]_{\bullet 4} + [A_{12}^{-1}]_{\bullet 2} \rightarrow e_1 = e_3 + [A_{12}^{-1}]_{\bullet 2}.$$

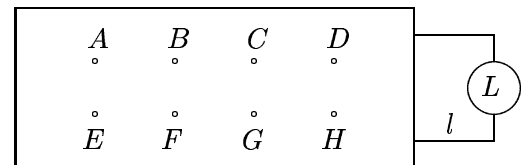
$$[A_{12}]_{\bullet 5} = e_2 + e_1 \rightarrow e_5 = [A_{12}^{-1}]_{\bullet 2} + [A_{12}^{-1}]_{\bullet 1} \rightarrow e_5 = e_3 + e_1 + [A_{12}^{-1}]_{\bullet 1}.$$

$$[A_{12}]_{\bullet 4} = e_1 + e_3 \rightarrow e_4 = [A_{12}^{-1}]_{\bullet 1} + [A_{12}^{-1}]_{\bullet 3} \rightarrow e_4 = e_5 + e_3 + e_1 + [A_{12}^{-1}]_{\bullet 3}.$$

$$\text{Dus: } A_{12}^{-1} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \end{pmatrix} \rightarrow D_T(G) = [A_{12}^{-1} A_{11}]^t = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix}.$$

### Toepassing 1.1

Beschouw nevenstaand apparaat met acht schakelaars ( $A$  t/m  $H$ ) die ieder een stuk stroomdraad, genoteerd met  $a$  t/m  $h$ , kunnen onderbreken. De stroomdraden zijn op een bepaalde manier met elkaar verbonden en zijn ook verbonden met een draad  $l$  die een lamp  $L$  bevat.



Als er met  $l$  een stroomkring tot stand komt, gaat de lamp  $L$  branden. Door proberen zijn we te weten gekomen dat de lamp brandt via de volgende stroomkringen:

$$C_1 = [a, b, f, h, l], \quad C_2 = [a, b, g, l], \quad C_3 = [a, e, f, g, l], \quad C_4 = [a, e, h, l], \quad C_5 = [b, c, e, h, l],$$

$$C_6 = [c, f, h, l], \quad C_7 = [c, g, l] \text{ en } C_8 = [d, l]. \text{ Hoe ziet de schakeling er uit?}$$

We gaan eerst de afhankelijke kringen verwijderen. Dat blijken  $C_1, C_3$  en  $C_5$  te zijn, immers:  $C_1 = C_2 + C_6 + C_7$ ,  $C_3 = C_4 + C_6 + C_7$  en  $C_5 = C_2 + C_4 + C_7$ . Dit geeft de volgende kringenmatrix, met als kringen  $C_2, C_4, C_6, C_7$  en  $C_8$ :

$$C = \begin{pmatrix} a & b & c & d & e & f & g & h & l \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

$$C = \left( \begin{array}{ccccc|cccc} b & e & f & g & d & a & c & h & l \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{array} \right).$$

$$A(G) = \left( \begin{array}{ccccc|cccc} b & e & f & g & d & a & c & h & l \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{array} \right).$$

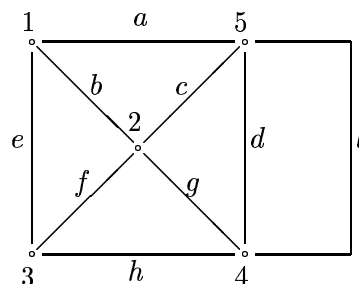
$$A(G) = \left( \begin{array}{ccccc|cccc} b & e & f & g & d & a & c & h & l \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{array} \right).$$

We proberen deze  $C$  in de standaardvorm  $C_T(G) = (I_{m-n+1} \mid D_T(G))$  te schrijven. Hiervoor kunnen we de kolommen van  $b, e, f$  en  $d$  gebruiken, en wel voor de eerste, tweede, derde resp. vijfde eenheidsvector. Voor de vierde eenheidsvector vervangen we de eerste kring door de som van de eerste ( $C_2$ ) en de vierde ( $C_7$ ); de vector bij  $g$  geeft dan de vierde eenheidsvector. Dit geeft de volgende matrix:

Als dit de kringenmatrix van een graaf  $G$  met 9 takken is, dan heeft  $G$  5 knooppunten (want  $m-n+1 = 5$ ). Vervolgens gaan we de incidentiematrix  $A(G)$  trachten te vinden. We gebruiken hierbij de eigenschap dat  $A(G)C^t = 0$ . Hieruit volgt dat  $A(G) = ([D_T(G)]^t \mid I_{n-1})$  voldoet, d.w.z.

Dit is nog geen goede incidentiematrix, want niet iedere rij bevat twee 1'en. Vervang daartoe de vierde rij door de som van de rijen drie en vier; voeg tenslotte nog een vijfde rij toe:

Dit geeft onderstaande schakeling:



## 1.4 De snedenmatrix

Zij  $G$  een samenhangende, niet-gerichte graaf met  $n$  knooppunten en  $m$  takken. Laat  $T$  een willekeurige opspannende boom zijn, bestaande uit de takken  $e_{m-n+2}, e_{m-n+3}, \dots, e_m$ . Door een tak van  $T$  weg te laten is de resterende graaf niet meer samenhangend. Dus iedere tak  $e_k$  ( $m-n+2 \leq k \leq m$ ) genereert een minimale snede. De  $n-1$  aldus geconstrueerde sneden geven  $n-1$  onafhankelijke vectoren in  $W_S(G)$ :  $e_k$  komt namelijk alleen voor in de  $(k-m+n-1)$ -de snede,  $k = m-n+2, m-n+3, \dots, m$ .

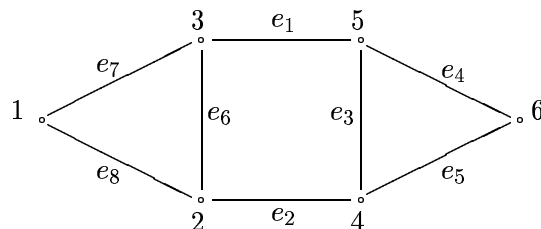
De bij  $T$  behorende *snedenmatrix*  $S_T(G)$  is de matrix met als rijen deze  $n-1$  vectoren.  $S_T(G)$  heeft dus de gedaante:

$$S_T(G) = \left( \begin{array}{cccc|cccc} e_1 & e_2 & \cdots & e_{m-n+1} & e_{m-n+2} & \cdots & e_m & \\ & & & F_T(G) & & & I_{n-1} & \end{array} \right) \quad (1.3)$$

### Voorbeeld 1.1 (vervolg)

Beschouw weer nevenstaande graaf en neem weer als opspannende boom  $T = \{e_2, e_4, e_5, e_7, e_8\}$ . Dit geeft de volgende snedenmatrix door achtereenvolgens  $e_2, e_4, e_5, e_7$  en  $e_8$  uit  $T$  te verwijderen:

$$S_T(G) = \left( \begin{array}{ccc|cccc} e_1 & e_3 & e_6 & e_2 & e_4 & e_5 & e_7 & e_8 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{array} \right)$$



De vector van de snede  $[e_2, e_6, e_7]$  wordt verkregen door de snedenvectoren van  $e_2$  en  $e_7$  (de takken van de snede die tot  $T$  behoren) bij elkaar op te tellen (modulo 2). De volgende stelling toont aan dat dit algemeen geldt.

**Stelling 1.12** *Zij  $S$  een (som van tak-disjuncte) snede( $n$ ), bestaande uit de takken  $\{e_j \mid j \in J\}$ , en laat  $s$  de bijbehorende vector van  $W_S(G)$  zijn. Zij  $J^* = \{j \in J \mid e_j \in T\} = \{j_1, j_2, \dots, j_k\}$  met  $j_i$  de index van de rij van  $S_T(G)$  behorende bij het  $i$ -de element van  $J^*$ . Dan geldt:  $s = \sum_{i=1}^k [S_T(G)]_{j_i \bullet}$ , waarbij  $[S_T(G)]_{j_i \bullet}$  de  $j_i$ -de rij van  $S_T(G)$  is.*

### Bewijs

Beschouw de vector  $s + \sum_{i=1}^k [S_T(G)]_{j_i \bullet}$ . Dit is weer een element van  $W_S(G)$ , dus de corresponderende takkenverz. is òf leeg, òf bevat een snede. Kies een  $p$  zdd.  $e_p \in T$ .

Als  $e_p \in S : p \in J^*$ , zeg  $p$  behoort bij  $j_s : \{s + \sum_{i=1}^k [S_T(G)]_{j_i \bullet}\}_p = s_p \oplus [S_T(G)]_{j_s p} = 1 \oplus 1 = 0$ .

Als  $e_p \notin S : p \notin J^* : \{s + \sum_{i=1}^k [S_T(G)]_{j_i \bullet}\}_p = s_p = 0$ .

Hieruit volgt:  $\{s + \sum_{i=1}^k [S_T(G)]_{j_i \bullet}\}_p = 0$  voor alle  $p$  met  $e_p \in T$ . De takken bij de niet-nul elementen van  $s + \sum_{i=1}^k [S_T(G)]_{j_i \bullet}$  zitten dus in  $E \setminus T$ .  $E \setminus T$  bevat echter geen snede, zodat ook buiten  $T$  de elementen van  $s + \sum_{i=1}^k [S_T(G)]_{j_i \bullet} = 0$  zijn. Omdat  $s + \sum_{i=1}^k [S_T(G)]_{j_i \bullet}$  een  $(0,1)$ -vector is geldt dus :  $s = \sum_{i=1}^k [S_T(G)]_{j_i \bullet}$ . ■

### Gevolg

De rijen van  $S_T(G)$  vormen een basis voor  $W_S(G)$  en de vectorruimte  $W_S(G)$  heeft dimensie  $n - 1$ .

**Stelling 1.13**  $F_T(G) = [D_T(G)]^t$ .

**Bewijs**

Omdat iedere snede een even aantal takken gemeen heeft met iedere kring, geldt:

$$\sum_j [S_T(G)]_{ij} \cdot [C_T(G)]_{kj} = \text{het aantal takken in de doorsnede van de } i\text{-de snede en de } k\text{-de kring} \\ \equiv 0 \pmod{2} \text{ voor iedere } i \text{ en } k.$$

Dus:  $0 = [S_T(G)] \cdot [C_T(G)]^t = \left( F_T(G), I_{n-1} \right) \begin{pmatrix} I_{m-n+1} \\ [D_T(G)]^t \end{pmatrix} = F_T(G) + [D_T(G)]^t,$

d.w.z.  $F_T(G) = [D_T(G)]^t$ . ■

**Gevolgen**

1. De matrix  $S_T(G)$  volgt direct uit  $C_T(G)$  en omgekeerd.
2. De rijen van  $C_T(G)$  vormen een basis voor de nulruimte van  $S_T(G)$ , immers:  
 Uit het bewijs van Stelling 1.13 volgt dat iedere rij van  $C_T(G)$  in de nulruimte van  $S_T(G)$  zit. Omdat de  $m - n + 1$  rijen van  $C_T(G)$  lineair onafhankelijk zijn, en omdat *dimensie nulruimte*  $S_T(G) = m - \text{dimensie } S_T(G) = m - n + 1$ , is  $C_T(G)$  een basis voor de nulruimte van  $S_T(G)$ . Omgekeerd vormen (analoog) de rijen van  $S_T(G)$  een basis voor de nulruimte van  $C_T(G)$ .

**Stelling 1.14** *S is een minimale snede van een samenhangende graaf G d.e.s.d. als S een verz. takken is die ten minste één tak gemeen heeft met iedere opspannende boom van G en minimaal is t.a.v. deze eigenschap.*

**Bewijs**

⇐ De takken van  $G$  die niet tot  $S$  behoren bevatten geen enkele opspannende boom. Deze takken vormen dus geen samenhangende deelgraaf meer.  $S$  is dus een splitsende verz. en omdat  $S$  minimaal is, is  $S$  een minimale snede.

⇒ Omdat  $S$  een snede is, is  $S \cap T \neq \emptyset$  voor iedere opspannende boom  $T$ . Als dit ook geldt voor een deelverz. van  $s$ , dan is deze deelverz. van  $S$  een splitsende verz. en dit is in tegenspraak met de minimaliteit van  $S$ . ■

## 1.5 De padenmatrix

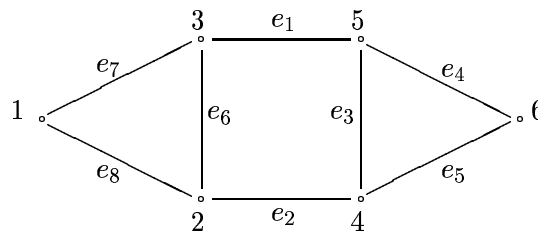
De *padenmatrix* is een matrix die gedefinieerd is voor een tweetal knooppunten, zeg  $v$  en  $w$ . De rijen van de matrix corresponderen met de paden (in een gerichte graaf) of de ketens (in een niet-gerichte graaf) die van  $v$  naar  $w$  lopen:

Gerichte graaf:  $p_{ij} = \begin{cases} 1 & \text{als de } j\text{-de pijl op het } i\text{-de pad van } v \text{ naar } w \text{ ligt} \\ 0 & \text{anders} \end{cases}$

Niet-gerichte graaf:  $p_{ij} = \begin{cases} 1 & \text{als de } j\text{-de tak op de } i\text{-de keten van } v \text{ naar } w \text{ ligt} \\ 0 & \text{anders} \end{cases}$

Voorbeeld 1.1 (vervolg)

Beschouw weer nevenstaande graaf en neem beschouw de ketens tussen  $s$  en  $t$  met  $s$  knooppunt 1 en  $t$  knooppunt 6. Deze ketens zijn:  $[e_7, e_1, e_4]$ ,  $[e_7, e_1, e_3, e_5]$ ,  $[e_7, e_6, e_2, e_5]$ ,  $[e_7, e_6, e_2, e_3, e_4]$ ,  $[e_8, e_2, e_5]$ ,  $[e_8, e_2, e_3, e_4]$ ,  $[e_8, e_6, e_1, e_4]$  en  $[e_8, e_6, e_1, e_3, e_5]$ . Dit geeft onderstaande padenmatrix.



$$P = \begin{pmatrix} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}$$

**Stelling 1.15** Voor een tweetal knooppunten  $v$  en  $w$ , met bijbehorende padenmatrix  $P$ , van een niet-gerichte graaf  $G$  geldt (bij optelling modulo 2): (i) de rijen van de matrix  $A(G) \cdot P^t$ , die niet bij  $v$  of  $w$  behoren, zijn 0-vectoren; (ii) de rijen van de matrix  $A(G) \cdot P^t$ , die bij  $v$  of  $w$  behoren, zijn 1-vectoren.

**Bewijs**

$$\begin{aligned} [A(G) \cdot P^t]_{ij} &= \sum_k a_{ik} p_{jk} \\ &= \text{het aantal takken } e_k \text{ incident met } v_i \text{ en op de } j\text{-de keten tussen } v \text{ en } w \\ &= \begin{cases} 2 & \text{als } v_i \text{ op de } j\text{-de keten tussen } v \text{ en } w \text{ ligt en } i \neq v, w \\ 0 & \text{als } v_i \text{ niet op de } j\text{-de keten tussen } v \text{ en } w \text{ ligt en } i \neq v, w \\ 1 & \text{als } i = v \text{ of } w \end{cases} \quad \blacksquare \end{aligned}$$

## 1.6 De structuurmatrix

De *structuurmatrix*  $Q(G) = (q_{ij})$  van een graaf  $G$  is een  $n \times n$  matrix die het verband aangeeft tussen de knooppunten:

$$q_{ij} = \begin{cases} \text{het aantal takken tussen } v_i \text{ en } v_j \text{ als } G \text{ niet-gericht is} \\ \text{het aantal pijlen van } v_i \text{ naar } v_j \text{ als } G \text{ gericht is} \end{cases}$$

**Stelling 1.16** Voor  $k = 1, 2, \dots$  geldt dat  $[Q(G)]^k$  het aantal takken(pijlen)reeksen van de lengte  $k$  aangeeft.

## Bewijs

We passen inductie naar  $k$  tot (merk op dat het klopt voor  $k = 1$ ).

$$\begin{aligned} \{[Q(G)]^k\}_{ij} &= \sum_{p=1}^n \{[Q(G)]^{k-1}\}_{ip} \cdot q_{pj} \\ &= \sum_{p=1}^n \text{aantal takken(pijlen)reeksen van de lengte } k-1 \text{ van } v_i \text{ naar } v_p, \\ &\quad \text{vermenigvuldigd met het aantal takken(pijlen) van } v_p \text{ naar } v_j \\ &= \sum_{p=1}^n \text{aantal takken(pijlen)reeksen van de lengte } k \text{ van } v_i \text{ naar } v_j \text{ met } v_p \\ &\quad \text{direct vóór } v_j \\ &= \text{aantal takken(pijlen)reeksen van de lengte } k \text{ van } v_i \text{ naar } v_j. \quad \blacksquare \end{aligned}$$

## Gevolg

Voor een gerichte graaf  $G$  geldt:  $G$  heeft geen rondes d.e.s.d. als  $[Q(G)]^k = 0$  voor zekere  $1 \leq k \leq n$ .

Vervolgens zullen we laten zien dat m.b.v. de structuurmatrix in een streng samenhangende graaf een natuurlijke *klassenindeling* kan worden gemaakt. Tevens kan de zogenaamde *periode* met een elegant algoritme berekend worden.

Zij  $G = (V, A)$  een gerichte graaf met structuurmatrix  $Q(G)$  en laat  $q_{ij}^{(k)}$  het  $(i, j)$ -de element van  $[Q(G)]^k$  zijn. Definieer voor iedere  $v_i \in V$  de indexverz.  $I(v_i)$  en de periode  $d(v_i)$  door:

$$I(v_i) = \{k \in \mathbb{N} \mid q_{ii}^{(k)} > 0\} \text{ en } d(v_i) = \text{g.g.d. } \{k \mid k \in I(v_i)\}.$$

In feite gaat het hier niet om de exacte waarde van de matrixelementen, maar om het wel of niet 0 zijn. Hetzelfde hebben we gezien in het hoofdstuk over Markov ketens in het dictaat Inleiding Besliskunde. Ook daar werd gesproken over de periode en een klasse-indeling (daar noemden we de klassen de cyclische verz). De bewijzen zijn identiek, vandaar dat we ze hier weglaten en alleen de resultaten geven.

**Stelling 1.17** *In een streng samenhangende graaf hebben alle knooppunten dezelfde periode.*

## Gevolg

We kunnen dus spreken over de periode  $d$  van een streng samenhangende graaf  $G$ .

**Stelling 1.18** *Zij  $G = (V, A)$  een streng samenhangende graaf met periode  $d$ . Laat  $T = (V, B)$  een opspannende gerichte boom zijn met wortel  $v_1$ , en laat  $k(i)$  het aantal pijlen in  $T$  zijn van  $v_1$  naar  $v_i$ ,  $i = 1, 2, \dots, n$ . Dan geldt:  $d = \text{g.g.d. } \{k(i) + 1 - k(j) \mid (v_i, v_j) \in A \setminus B\}$ .*

**Stelling 1.19** *Zij  $G = (V, A)$  een streng samenhangende graaf met periode  $d$ . Dan is er een partitie  $V_1, V_2, \dots, V_d$  van  $V$  zdd. als  $(v_i, v_j) \in A$  en  $v_i \in V_k$ , dan  $v_j \in V_{k+1}$ , waarbij  $V_{d+1} \equiv V_1$ .*



Bovenstaande stellingen garanderen de correctheid van het volgende algoritme.

*Algoritme 1.1: Bepaling van de perioden en de partities in een gerichte graaf  $G = (V, A)$ .*

1. Bepaal de streng samenhangende componenten van  $G$ .
2. Doe voor ieder van de streng samenhangende componenten het volgende:
  - a. Kies een knooppunt  $v_1$  en een opspannende gerichte boom  $T = (V, B)$  met wortel  $v_1$ .
  - b. Bereken voor iedere  $v_i$  het aantal pijlen  $k(i)$  om in  $T$  van  $v_1$  naar  $v_i$  te gaan.
  - c.  $d = g.g.d. \{k(i) + 1 - k(j) \mid (v_i, v_j) \in A \setminus B\}$ .
  - d.  $V_k = \{v_i \mid k(i) = k - 1 \pmod{d}\}$ ,  $k = 1, 2, \dots, d$ .

### Opmerkingen

1. In het volgende hoofdstuk zullen we een efficiënt algoritme bespreken om de streng samenhangende componenten van een gerichte graaf te bepalen. Het zal blijken dat dit algoritme tegelijk ook in iedere component een opspannende boom oplevert.
2. De  $g.g.d.$  van een aantal getallen kan worden bepaald met het *algoritme van Euclides*. Dit algoritme berekent de  $g.g.d.$  van twee gehele getallen  $a$  en  $b$  en is gebaseerd op de eigenschap dat  $g.g.d. \{a, b\} = g.g.d. \{a - b, b\}$ . We geven het algoritme hieronder.

*Algoritme 1.2: Bepaling van de  $g.g.d.$  van twee getallen  $a, b \in \mathbb{N}$  met  $a \geq b$ .*

1. Als  $b = 0$ , dan  $g.g.d. = a$  en stop.  
Anders:  $a := b$  en  $b := a \pmod{b}$ .
2. Ga naar stap 1.

### Voorbeeld 1.2

Beschouw nevenstaande streng samenhangende graaf.

Neem als opspannende boom vanuit  $v_1$ :

$$T = \{(1, 2), (2, 3), (3, 4), (4, 5), (5, 6)\}.$$

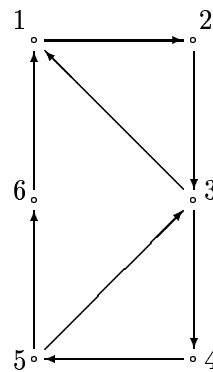
$$\text{Dan is } A \setminus B = \{(6, 1), (3, 1), (5, 3)\}.$$

Voor de boomafstanden  $k(i)$  geldt:

$$k(i) = i - 1, \quad i = 1, 2, \dots, 6.$$

$$d = g.g.d. \{6, 3, 3\} = 3. \quad V_1 = \{1, 4\},$$

$$V_2 = \{2, 5\}, \quad V_3 = \{3, 6\}.$$



## 1.7 Opgaven

### Opgave 1.1

Als  $x$  behoort tot  $W_C(G) \cap W_S(G)$ , dan bevat  $x$  een even aantal 1'en. Toon dit aan.

### Opgave 1.2

Laat  $W_C(G) \cup W_S(G) = \{x + y \mid x \in W_C(G), y \in W_S(G)\}$ .

a. Toon aan:  $W_C(G) \cup W_S(G) = W(G)$  d.e.s.d. als  $W_C(G) \cap W_S(G) = \{0\}$ .

*Aanwijzing:* Gebruik de eigenschap dat voor lineaire vectorruimtes  $X$  en  $Y$  geldt:

$$\dim(X \cup Y) + \dim(X \cap Y) = \dim(X) + \dim(Y).$$

b. Ga na of  $W_C(G) \cup W_S(G) = W(G)$  voor de graaf uit Voorbeeld 1.1.

c. Bepaal een  $c \in W_C(G)$  en een  $s \in W_S(G)$  zdd.  $c + s = (1, 1, \dots, 1)$  voor de graaf uit Voorbeeld 1.1.

### Opgave 1.3

Zij  $G$  een boom en laat  $A_1$  uit  $A(G)$  verkregen worden door een willekeurige rij weg te laten.

Toon aan dat  $\det(A_1) = \pm 1$ .

### Opgave 1.4

Zij  $G$  een in het platte vlak getekende vlakke, samenhangende graaf, en laat  $C$  de matrix zijn met als rijen de vectoren van de  $m - n + 2$  kringen die behoren bij de  $m - n + 2$  gebieden van  $G$ .

a. Geef een voorbeeld van een normale graaf  $G$  waarvoor de bijbehorende matrix  $C$  niet totaal unimodulair is.

b. Toon aan:  $C$  is totaal unimodulair d.e.s.d. als de duale graaf  $G^*$  bipartiet is.

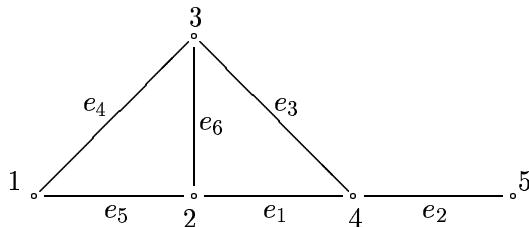
### Opgave 1.5

Beschouw nevenstaande graaf en

neem  $T = \{e_2, e_3, e_5, e_6\}$ .

a. Bepaal  $C_T(G)$  door de onafhankelijke kringen op te sporen.

b. Bepaal  $C_T(G)$  via  $A(G)$ .



### Opgave 1.6

Zij  $G$  de graaf uit voorbeeld 1.1 en laat  $T = \{e_2, e_4, e_5, e_7, e_8\}$ .

Bepaal een graaf  $G_1$  met 8 takken zdd.  $C_T(G) = C_T(G_1)$  (ook in  $G_1$  nemen we voor  $T$  de 2e, 4e, 5e, 7e en 8-ste tak), terwijl  $G$  en  $G_1$  niet isomorf zijn.

### Opgave 1.7

Zij  $G$  een samenhangende graaf. Laat  $S_i$  de snede zijn bestaande uit de takken incident met het  $i$ -de knooppunt,  $i = 1, 2, \dots, n$ . Toon aan dat de vectoren die behoren bij  $S_1, S_2, \dots, S_{n-1}$  een basis vormen voor  $W_S(G)$ .

### Opgave 1.8

De verz. van alle opspannende bomen van een graaf  $G$  bestaat uit:

$\{e_2, e_3, e_5\}, \{e_1, e_4, e_5\}, \{e_1, e_3, e_4\}, \{e_2, e_3, e_4\}, \{e_1, e_2, e_3\}, \{e_1, e_2, e_4\}, \{e_2, e_4, e_5\}$  en  $\{e_1, e_3, e_5\}$ .

Bepaal voor  $T = \{e_2, e_3, e_5\}$  de snedenmatrix  $S_T(G)$ .

### Opgave 1.9

Toon aan dat er voor iedere graaf  $G$  vectoren  $c \in W_C(G)$  en  $s \in W_S(G)$  zijn zdd.  $c + s = (1, 1, \dots, 1)$ .

*Aanwijzing:* Toon aan dat de matrices  $(C_T(G), S_T(G), e)^t$  en  $(C_T(G), S_T(G))^t$ , waarbij  $e = (1, 1, \dots, 1)$ , dezelfde dimensie van de nulruimte hebben en dus ook dezelfde rang.

### Opgave 1.10\* (de opgaven met een \* hebben een hogere moeilijkheidsgraad)

a. Zij  $C_1$  een  $(m - n + 1) \times (m - n + 1)$  deelmatrix van  $C_T(G)$ . Toon aan:

$C_1$  is niet-singulier d.e.s.d. als de bij  $E \setminus C_1$  behorende takken een opspannende boom vormen.

b. Zij  $S_1$  een  $(n - 1) \times (n - 1)$  deelmatrix van  $S_T(G)$ . Toon aan:

$S_1$  is niet-singulier d.e.s.d. als de bij  $S_1$  behorende takken een opspannende boom vormen.

### Opgave 1.11\*

Toon via de volgende stappen aan dat de snedenmatrix en de kringenmatrix van een bipartiete graaf totaal unimodulair zijn.

a. Beschouw een  $(n - 1) \times (n - 1)$  deelmatrix  $S_1$  van  $S_T(G)$ . Bewijs dat  $\det(S_1) \in \{0, +1, -1\}$ .

b. Beschouw een  $k \times m$  deelmatrix  $S_2$  van  $S_T(G)$ . Laat zien dat er een graaf  $G_1$  is met eveneens  $m$  takken en waarvoor geldt dat  $S_{T_1}(G_1) = S_2$  voor een zekere opspannende boom  $T_1$  van  $G_1$ .

c. Toon aan dat  $S_T(G)$  totaal unimodulair is.

d. Toon aan dat  $C_T(G)$  totaal unimodulair is.

### Opgave 1.12

Toon het volgende aan:

Voor een tweetal knooppunten  $s$  en  $t$ , met bijbehorende padenmatrix  $P$ , van een gerichte graaf  $G$  geldt:

a. de rijen van de matrix  $A(G) \cdot P^t$ , die niet bij  $s$  of  $t$  behoren, zijn 0-vectoren;

b. de rij van de matrix  $A(G) \cdot P^t$ , die bij  $s$  behoort, is een 1-vector;

c. de rij van de matrix  $A(G) \cdot P^t$ , die bij  $t$  behoort, is een (-1)-vector.

### Opgave 1.13

Is de padenmatrix  $P$  van een gerichte graaf totaal unimodulair? Verklaar uw antwoord.

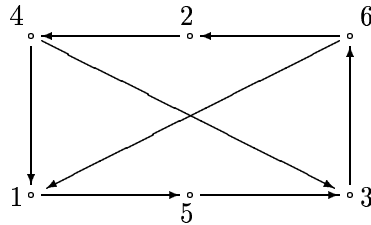
**Opgave 1.14**

Zij  $G$  een normale, niet-gerichte graaf. toon het volgende aan:

- (i) Het aantal kringen in  $G$  bestaande uit 3 takken is  $\frac{1}{6} \sum_{i=1}^n q_{ii}^{(3)}$ ;
- (ii) Het aantal kringen in  $G$  bestaande uit 4 takken is  $\frac{1}{8} [\sum_{i=1}^n q_{ii}^{(4)} - 2m - 2 \cdot \sum_{i=1}^n \sum_{j \neq i} q_{ij}^{(2)}]$ .

**Opgave 1.15**

Bepaal met Algoritme 1.1 de periode en de partitie voor onderstaande streng samenhangende graaf.



# Hoofdstuk 2

## Standaard algoritmen

### 2.1 Voorwaarts zoeken

We zullen de techniek van het *voorwaarts zoeken* (Engels: depth-first search) allereerst voor een *niet-gerichte graaf*  $G = (V, E)$  bespreken. Behalve de structuurmatrix wordt voor de representatie van een normale graaf ook vaak een *structuurlijst* gebruikt. Bij een structuurlijst wordt voor ieder knooppunt  $v$  bijgehouden met welke andere knooppunten  $v$  incident is:

$$L[v] = \begin{cases} \{w \mid v \text{ en } w \text{ verbonden door een tak}\} & \text{als } G \text{ niet-gericht is} \\ \{w \mid \text{er is een pijl van } v \text{ naar } w\} & \text{als } G \text{ gericht is} \end{cases}$$

We veronderstellen dat  $G$  wordt gerepresenteerd door structuurlijsten  $L[v_i]$ ,  $i = 1, 2, \dots, n$ .

We beschrijven een systematische methode om "door een graaf te lopen". We houden bij welke knooppunten we tegenkomen, in welke volgorde, en hoe de verbindingen lopen. Daarmee creëren we in iedere component een gerichte opspannende boom en de andere takken worden zó gericht dat ze met de bijbehorende kring een ronde vormen (dit zullen we bewijzen in Stelling 2.3). Hiermee kunnen we o.a. vragen over de samenhang beantwoorden.

De procedure van het voorwaarts zoeken in een niet-gerichte graaf laat zich als volgt beschrijven. Start in een willekeurig knooppunt, zeg  $v$ , en begin het zoeken vanuit  $v$ . Dit knooppunt noemen we de *wortel*, en we zeggen dat  $v$  *bezocht* is. Het bezocht zijn houden we bij door de knooppunten in de volgorde waarin ze bezocht zijn te nummeren:  $N[w]$ ,  $w \in V$  geeft aan als hoeveelste het knooppunt  $w$  wordt bezocht (knooppunten die nog niet bezocht zijn krijgen  $N$ -waarde 0). Vervolgens kiezen we het eerste knooppunt van de lijst  $L[v]$ , zeg knooppunt  $w$ , en we bekijken de tak  $(v \& w)$ . We richten de tak van  $v$  naar  $w$  als de pijl  $(v, w)$  en stoppen deze pijl in de verz.  $F$  van *voorwaartse pijlen* (Engels: *forward*) en noteren dat  $v$  de *vader* is van  $w$  door  $v = PRED[w]$ . We zetten het zoeken nu in knooppunt  $w$  voort.

In het algemeen, als we in knooppunt  $v$  zijn, dan zijn er de volgende mogelijkheden:

1. We bekijken  $(v \& w)$ , waarbij  $w$  nog niet eerder bezocht is (d.w.z.  $N[w] = 0$ ):  
geef  $w$  het volgende nummer, voeg  $(v, w)$  aan  $F$  toe, maak  $v = PRED[w]$  en ga verder in  $w$ .

- 2 We bekijken  $(v&w)$ , waarbij  $w$  wel eerder bezocht is (d.w.z.  $1 \leq N[w] \leq N[v]$ ):  
indien  $(v&w)$  al eerder is bekeken, wat dan vanuit  $w$  is gedaan, doe niets;  
indien  $(v&w)$  niet eerder is bekeken, voeg  $(v, w)$  aan een verz.  $B$  toe, de *achterwaartse pijlen*,  
(Engels: *backward*) en ga door met het zoeken vanuit  $v$ .
3. Indien alle met  $v$  incidente takken onderzocht zijn, dan heet  $v$  *afgehandeld* en zetten we het zoeken vanuit de vader van  $v$  (d.w.z. vanuit  $PRED[v]$ ) voort. Indien  $v$  geen vader heeft (d.w.z.  $v$  is de wortel van de boom) en er zijn nog niet-bezochte knooppunten, dan gaan we in zo'n nog niet bezocht knooppunt weer opnieuw beginnen.

Het nagaan of een tak  $(v&w)$  al eens eerder is bekeken, dus vanuit  $w$ , gebeurt als volgt:

- we kunnen nagaan of  $(w, v) \in F$  door te controleren of  $w = PRED[v]$ ;
- we kunnen nagaan of  $(w, v) \in B$  door te controleren of  $w \neq PRED[v]$  en  $N[w] > N[v]$ .

Het bovenstaande leidt tot het volgende algoritme ( $j$  houdt de nummering bij en  $k$  het aantal componenten).

*Algoritme 2.1: Voorwaarts zoeken in een niet-gerichte graaf*

1.  $N[v] = PRED[v] = 0$ ,  $v = 1, 2, \dots, n$ ;  $F = B = \emptyset$ ;  $j = 0$ ;  $k = 1$ .
2. Kies een knooppunt  $v$  met  $N[v] = 0$ .
3.  $j := j + 1$ ;  $N[v] = j$ .
4. Doe voor iedere  $w \in L[v]$  het volgende:
  - a. Als  $N[w] = 0$ :  $F := F \cup (v, w)$ ;  $PRED[w] = v$  en voer de stappen 3 en 4 uit met  $v = w$ .
  - b. Als  $N[w] \neq 0$ : als  $w \neq PRED[v]$  en  $N[w] \leq N[v]$ :  $B := B \cup (v, w)$ .
5. Als er nog knooppunt  $v$  is met  $N[v] = 0$ :  $k := k + 1$  en ga naar stap 2.

### Voorbeeld 2.1

Beschouw nevenstaande graaf. Het algoritme verloopt als volgt.

$N[v] = PRED[v] = 0$ ,  $v = 1, 2, \dots, 6$ ;  $F = B = \emptyset$ ;  $j = 0$ ;  $k = 1$ .

$v = 1$ ;  $j = 1$ ;  $N[1] = 1$ ;  $w = 2$ :  $(1, 2) \in F$ ;  $PRED[2] = 1$ .

$v = 2$ ;  $j = 2$ ;  $N[2] = 2$ ;  $w = 1$ : -;  $w = 4$ :  $(2, 4) \in F$ ;  $PRED[4] = 2$ .

$v = 4$ ;  $j = 3$ ;  $N[4] = 3$ ;  $w = 1$ :  $(4, 1) \in B$ ;  $w = 2$ : -;  $w = 6$ :  
 $(4, 6) \in F$ ;  $PRED[6] = 4$ .

$v = 6$ ;  $j = 4$ ;  $N[6] = 4$ ;  $w = 3$ :  $(6, 3) \in F$ ;  $PRED[3] = 6$ .

$v = 3$ ;  $j = 5$ ;  $N[3] = 5$ ;  $w = 1$ :  $(3, 1) \in B$ ;  $w = 5$ :  $(3, 5) \in F$ ;  
 $PRED[5] = 3$ .

$v = 5$ ;  $j = 6$ ;  $N[5] = 6$ ;  $w = 2$ :  $(5, 2) \in B$ ;  $w = 3$ : -;  $w = 6$ :  $(5, 6) \in B$ .

$v = 5$  is nu afgehandeld en we gaan verder bij  $v = 3$ :

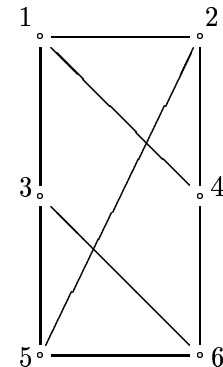
$v = 3$ ;  $w = 6$ : -.  $v = 3$  is nu afgehandeld en we gaan verder bij  $v = 6$ :

$v = 6$ ;  $w = 4$ : -;  $w = 5$ : -.  $v = 6$  is nu afgehandeld en we gaan verder bij  $v = 4$ :

$v = 4$  is reeds afgehandeld en we gaan verder bij  $v = 2$ :

$v = 2$ ;  $w = 5$ : -.  $v = 2$  is nu afgehandeld en we gaan verder bij  $v = 1$ :

$v = 1$ ;  $w = 3$ : -;  $w = 4$ : -.  $v = 1$  is nu afgehandeld en zijn nu klaar.



We eindigen dus met één component ( $k = 1$ ) en met:

$$N[1] = 1; N[2] = 2; N[3] = 5; N[4] = 3; N[5] = 6; N[6] = 4.$$

$$F = \{(1, 2), (2, 4), (4, 6), (6, 3), (3, 5)\} \text{ en } B = \{(4, 1), (3, 1), (5, 2), (5, 6)\}.$$

**Stelling 2.1** *Het algoritme van het voorwaarts zoeken in een niet-gerichte graaf heeft complexiteit  $\mathcal{O}(p)$ , waarbij  $p = \max(n, m)$ .*

### Bewijs

Stap 1 van het algoritme wordt éénmaal doorlopen en vereist  $\mathcal{O}(n)$  stappen. Verder wordt ieder knooppunt  $v$  éénmaal voor de eerste keer bezocht. Tel het werk dat in de stappen 3 en 4 voor deze  $v$  wordt gedaan bij elkaar: dit heeft als orde van complexiteit het aantal elementen in  $L[v]$ . De totale complexiteit van de stappen 3 en 4 is dus van de orde  $\sum_{v \in V} \#L[v] = 2m$ . De stappen 2 en 5 hebben complexiteit  $\mathcal{O}(1)$  en worden hoogstens  $n$  keer uitgevoerd. Voor de totale complexiteit van het algoritme geldt dus:  $\mathcal{O}(n) + \mathcal{O}(m) = \mathcal{O}(p)$ . ■

**Stelling 2.2** *(i) de pijlen van  $F$  vormen een opspannend gericht bos met als wortels de knooppunten  $v$  met  $PRED[v] = 0$ ; (ii) de met  $F$  geassocieerde kringen zijn rondes, d.w.z. de enige pijl van  $B$  in zo'n kring loopt "de goede kant" op; (iii) iedere enkelvoudige ronde in de gerichte graaf is een met  $F$  geassocieerde kring.*

### Bewijs

Voor alle onderdelen van het bewijs mogen we wel aannemen dat de graaf samenhangend is.

(i) Behalve de wortel wordt ieder knooppunt één keer bezocht op het moment dat zijn  $N$ -waarde 0 is en op dat moment wordt een in dat knooppunt binnenkomende pijl aan  $F$  toegevoegd: ieder knooppunt, behalve de wortel, heeft in  $F$  precies één binnenkomende pijl.  $F$  bevat dus  $n - 1$  pijlen en door vanuit een knooppunt terug te lopen wordt de wortel bereikt: de pijlen van  $F$  vormen een opspannende gerichte boom met als wortel het knooppunt  $v$  met  $PRED[v] = 0$ .

(ii) Beschouw een tak  $(v \& w)$  die niet tot  $F$  behoort en veronderstel dat  $N[w] > N[v]$ . De tak  $(v \& w)$  is dus niet vanuit  $v$  voor het eerst bekeken (anders was deze in  $F$  opgenomen), maar vanuit  $w$ . Er geldt:  $(w, v) \in B$  en  $w$  is eerder afgehandeld dan  $v$  (het afhandelen gebeurt van hogere naar lagere  $N$ -waarden). Omdat vanuit een nog niet afgehandeld knooppunt alle knooppunten met hogere  $N$ -waarde via pijlen van  $F$  bereikbaar zijn, is er in  $F$  een pad van  $v$  naar  $w$ . Dit geeft tezamen met de pijl  $(w, v)$  de gewenste ronde.

(iii) Laat  $C$  een ronde zijn in de gerichte graaf met  $q \geq 2$  pijlen van  $B$  en stel dat dit de pijlen  $(w_i, x_i)$ ,  $1 \leq i \leq q$  zijn en dat  $C = [w_1, x_1, \dots, w_2, x_2, \dots, w_q, x_q, \dots, w_1]$ . Volgens onderdeel (ii) is er voor iedere  $1 \leq i \leq q$  een pad met pijlen van  $F$  van  $x_i$  naar  $w_i$ . Omdat ieder knooppunt hoogstens één binnenkomende pijl van  $F$  heeft, komt het pad van  $x_i$  naar  $w_i$  bij  $w_i$  aan met de pijlen van  $C$  vanaf  $x_{i-1}$ : er is dus een pad van  $x_i$  naar  $x_{i-1}$  met louter pijlen van  $F$ ,  $1 \leq i \leq q$ , waarbij  $x_0 \equiv x_q$ . Dit geeft echter een ronde binnen de boom  $F$ :  $x_1 \rightarrow x_q \rightarrow x_{q-1} \rightarrow \dots \rightarrow x_1$ . Dit is in een boom echter onmogelijk: tegenspraak. ■

Vervolgens bespreken we het voorwaarts zoeken in een *gerichte graaf*. De lijst  $L[v]$  bevat nu de knooppunten  $w$  die een pijl van  $v$  naar  $w$  hebben. De procedure voor een gerichte graaf is analoog aan die van een niet-gerichte graaf, alleen worden de pijlen die niet tot  $F$  behoren wat anders ingedeeld: in plaats van  $B$  hebben we nu drie verz.:  $I, D$  en  $C$  (afkortingen voor de Engelse termen *increasing*, *decreasing* en *crossing*). Als we vanuit  $v$  de pijl  $(v, w)$  bekijken en  $w$  is al eerder bezocht, dan doen we het volgende: als  $N[w] > N[v]$  (dan is er in  $F$  dus een pad van  $v$  naar  $w$ ), dan wordt  $(v, w)$  aan de verz.  $I$  toegevoegd; als  $N[w] \leq N[v]$ , dan maken we onderscheid tussen de gevallen dat  $w$  nog niet of reeds wel is afgehandeld. Als  $w$  nog niet is afgehandeld, dan voegen we  $(v, w)$  aan  $D$  toe en als  $w$  reeds is afgehandeld, dan voegen we  $(v, w)$  aan  $C$  toe. We houden via de getallen  $R[v]$  bij in welke volgorde de knooppunten worden afgehandeld:  $R[v] = l$  d.e.s.d. als knooppunt  $v$  als  $l$ -de knooppunt is afgehandeld. Het algoritme is nu als volgt.

*Algoritme 2.2: Voorwaarts zoeken in een gerichte graaf*

1.  $N[v] = PRED[v] = R[v] = 0$ ,  $v = 1, 2, \dots, n$ ;  $F = I = D = C = \emptyset$ ;  $j = 0$ ;  $k = l = 1$ .
2. Kies een knooppunt  $v$  met  $N[v] = 0$ .
3.  $j := j + 1$ ;  $N[v] = j$ .
4. Doe voor iedere  $w \in L[v]$  het volgende:
  - a. Als  $N[w] = 0$ :  $F := F \cup (v, w)$ ;  $PRED[w] = v$  en voer de stappen 3 en 4 uit met  $v = w$ .
  - b. Als  $N[w] \neq 0$ : als  $N[w] > N[v]$ :  $I := I \cup (v, w)$ ;  
als  $N[w] \leq N[v]$ : als  $R[w] = 0$ :  $D := D \cup (v, w)$ ;  
als  $R[w] \neq 0$ :  $C := C \cup (v, w)$ .
5.  $R[v] := l$ ;  $l := l + 1$ .
6. Als er nog knooppunt  $v$  is met  $N[v] = 0$ :  $k := k + 1$  en ga naar stap 2.

### Voorbeeld 2.2

Beschouw nevenstaande graaf. Het algoritme verloopt als volgt.

$N[v] = PRED[v] = R[v] = 0$ ,  $v = 1, 2, \dots, 6$ ;  $F = I = D = C = \emptyset$ .

$j = 0$ ;  $k = l = 1$ .

$v = 1$ ;  $j = 1$ ;  $N[1] = 1$ ;  $w = 3$ :  $(1, 3) \in F$ ;  $PRED[3] = 1$ .

$v = 3$ ;  $j = 2$ ;  $N[3] = 2$ ;  $w = 4$ :  $(3, 4) \in F$ ;  $PRED[4] = 3$ .

$v = 4$ ;  $j = 3$ ;  $N[4] = 3$ ;  $w = 2$ :  $(4, 2) \in F$ ;  $PRED[2] = 4$ .

$v = 2$ ;  $j = 4$ ;  $N[2] = 4$ ;  $w = 3$ :  $(2, 3) \in D$ .

$v = 2$  is nu afgehandeld:  $R[2] = 1$ ;  $l = 2$ ; ga verder met  $v = 4$ .

$v = 4$  is nu afgehandeld:  $R[4] = 2$ ;  $l = 3$ ; ga verder met  $v = 3$ .

$v = 3$  is nu afgehandeld:  $R[3] = 3$ ;  $l = 4$ ; ga verder met  $v = 1$ .

$v = 1$ ;  $w = 4$ :  $(1, 4) \in I$ ;  $v = 1$  is nu afgehandeld:  $R[1] = 4$ ;  $l = 5$ ;  $k = 2$ .

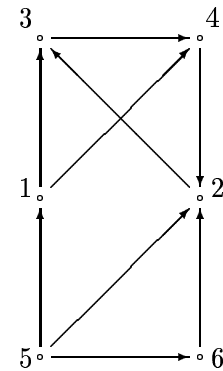
$v = 5$ ;  $j = 5$ ;  $N[5] = 5$ ;  $w = 1$ :  $(5, 1) \in C$ ;  $w = 2$ :  $(5, 2) \in C$ ;  $w = 6$ :  $(5, 6) \in F$ ;  $PRED[6] = 5$ .

$v = 6$ ;  $j = 6$ ;  $N[6] = 6$ ;  $w = 2$ :  $(6, 2) \in C$ ;  $v = 6$  is nu afgehandeld:  $R[6] = 5$ ;  $l = 6$ ; ga verder met  $v = 5$ .  $v = 5$  is nu afgehandeld:  $R[5] = 6$ ;  $l = 7$ .

We eindigen dus met twee componenten ( $k = 2$ ) en met:

$N[1] = 1$ ;  $N[2] = 4$ ;  $N[3] = 2$ ;  $N[4] = 3$ ;  $N[5] = 5$ ;  $N[6] = 6$ .

$F = \{(1, 3), (3, 4), (4, 2), (5, 6)\}$ ,  $I = \{(1, 4)\}$ ,  $D = \{(2, 3)\}$  en  $C = \{(5, 1), (5, 2), (6, 2)\}$ .





**Stelling 2.3** (i) de complexiteit van algoritme 2.2 is  $\mathcal{O}(p)$ , waarbij  $p = \max(n, m)$ ; (ii) de pijlen van  $F$  vormen een opspannend gericht bos met als wortels de knooppunten  $v$  met  $PRED[v] = 0$ ;

### Bewijs

Het bewijs gaat analoog aan het bewijs van Stelling 2.1 en Stelling 2.3 (i). ■

**Lemma 2.1** (i) een pijl  $(v, w) \in I$  d.e.s.d. als er in  $F$  geen pijl, maar wel een pad is van  $v$  naar  $w$ ; (ii) een pijl  $(v, w) \in D$  d.e.s.d. als er in  $F$  een pad is van  $w$  naar  $v$ ; (iii) een pijl  $(v, w) \in C$  d.e.s.d. als er in  $F$  geen pad is van  $v$  naar  $w$  en ook geen pad van  $w$  naar  $v$ .

### Bewijs

(i) Als  $(v, w) \in I$ , dan is  $v$  vóór  $w$  bezocht en is  $w$  afgehandeld terwijl we nu vanuit  $v$  bezig zijn met  $w \in L[v]$ . In  $F$  is er dan een pad van  $v$  naar  $w$ . Het omgekeerde gaat analoog.

(ii) Als  $(v, w) \in D$ , dan is  $w$  vóór  $v$  bezocht en is  $w$  is nog niet afgehandeld terwijl we nu vanuit  $v$  bezig zijn met  $w \in L[v]$ . In  $F$  is er dan een pad van  $w$  naar  $v$ . Het omgekeerde gaat analoog.

(iii) Dit onderdeel volgt uit (i) en (ii). ■

**Stelling 2.4** Laten de pijlen van  $F$  het opspannende gerichte bos  $T$  in  $G = (V, A)$  vormen. Zij  $G_1 = (V_1, A_1)$  een streng samenhangende component van  $G$ . Dan is de deelgraaf van  $T$  bestaande uit de knooppunten  $V_1$  en de daartussen lopende pijlen van  $F$  een gerichte opspannende boom van  $G_1$ .

### Bewijs

Kies twee knooppunten,  $v_1, v_2 \in V_1$  en neem aan dat  $N[v_1] < N[v_2]$ . Omdat  $G_1$  streng samenhangend is, is er in  $G_1$  een pad  $P$  van  $v_1$  naar  $v_2$ . Laat  $z_1$  het knooppunt op  $P$  zijn met de kleinste  $N$ -waarde (eventueel  $z_1 = v_1$ ) en laat  $T_1$  de deelboom van  $T$  zijn met wortel  $z_1$ .

Beschouw  $(v, w) \in A$  met  $v \in T_1$  en  $w \notin T_1$ . Omdat  $(v, w) \notin T_1$  is  $w$  eerder bezocht dan alle knooppunten van  $T_1$ :  $N[w] < N[z_1]$ . Hieruit volgt dat alle knooppunten van  $P[z_1, \dots, v_2]$  tot  $T_1$  behoren.

Omdat  $N[z_1] \leq N[v_1] < N[v_2]$ , en omdat  $z_1$  en  $v_2$  tot  $T_1$  behoren, behoren de tussenliggende waarden ook tot  $T_1$ :  $v_1 \in T_1$ . Er zijn dus in  $T$  paden van  $z_1$  naar  $v_1$  en van  $z_1$  naar  $v_2$ .

Kies vervolgens een  $v_3 \in V_1$  met  $v_3 \neq v_1, v_2, z_1$ . Herhaal het bovenstaande met voor  $v_1$  nu  $z_1$  en voor  $v_2$  de zojuist gekozen  $v_3$ . Dit geeft een  $z_2$  zdd. er in  $T$  paden zijn van van  $z_2$  naar  $z_1$  en van  $z_2$  naar  $v_3$ , dus ook van  $z_2$  naar  $v_1, v_2$  en  $v_3$ . Op deze wijze voortgaande zien we in dat er een  $z \in V_1$  is zdd. er in  $T$  paden zijn van  $z$  naar  $v$  voor iedere  $v \in V_1$ .

Het is nu voldoende om te laten zien dat deze deelgraaf geen andere knooppunten dan die van  $V_1$  bevat. Neem een willekeurige  $v$  uit deze deelgraaf. Er is een pad van  $z$  naar  $v$  en ga in de deelboom naar een eindpunt, zeg  $w$ , die tot  $V_1$  behoren. Omdat  $V_1$  streng samenhangend is, is er ook een pad van  $w$  naar  $z$ . De knooppunten  $v$  en  $z$  behoren dus tot dezelfde streng samenhangende component:  $v \in V_1$ . ■

### Gevolg

Als  $G$  streng samenhangend is, dan vormen de pijlen van  $F$  een gerichte opspannende boom.

## 2.2 Zijwaarts zoeken

Deze methode (in het Engels: breadth-first search) verschilt van het voorwaarts zoeken doordat, als een knooppunt wordt bezocht, eerst alle met dat knooppunt incidente takken (we bespreken alleen de versie voor een niet-gerichte graaf) worden bekeken voordat we verder gaan. De nog niet bezochte eindpunten van deze takken worden op een wachtlijst  $W$  geplaatst. Als  $v$  de wortel is, dan worden achtereenvolgens de knooppunten bezocht die via één tak, daarna via twee takken etc. bereikbaar zijn. Er wordt weer een opspannend gericht bos gecreëerd met pijlenverz.  $F$ .

*Algoritme 2.3: Zijwaarts zoeken in een niet-gerichte graaf*

1.  $N[v] = PRED[v] = 0$ ,  $v = 1, 2, \dots, n$ ;  $F = W = \emptyset$ ;  $j = 0$ ;  $k = 1$ .
2. Kies een knooppunt  $v$  met  $N[v] = 0$ ;  $j := j + 1$ ;  $N[v] = j$ ; voeg  $v$  achteraan  $W$  toe.
3. Als  $W = \emptyset$ : ga naar stap 5.
4. a. Neem het eerste knooppunt van  $W$ , zeg  $v$ , en verwijder dat uit  $W$ ;  
 b. Doe voor iedere  $w \in L[v]$  met  $N[w] = 0$  het volgende:  
 $j := j + 1$ ;  $N[w] = j$ ; voeg  $w$  achteraan  $W$  toe;  $F := F \cup (v, w)$ ;  $PRED[w] = v$ ;  
 c. Ga naar stap 3.
5. Als er nog knooppunt  $v$  is met  $N[v] = 0$ :  $k := k + 1$  en ga naar stap 2.

Voorbeeld 2.1 (vervolg)

$N[v] = PRED[v] = 0$ ,  $v = 1, 2, \dots, 6$ ;  $F = W = \emptyset$ ;  $j = 0$ ;  $k = 1$ .

$v = 1$ ;  $j = 1$ ;  $N[1] = 1$ ;  $W = \{1\}$ .

$v = 1$ ;  $W = \emptyset$ .

$w = 2$ ;  $j = 2$ ;  $N[2] = 2$ ;  $W = \{2\}$ ;  $(1, 2) \in F$ ;  $PRED[2] = 1$ .

$w = 3$ ;  $j = 3$ ;  $N[3] = 3$ ;  $W = \{2, 3\}$ ;  $(1, 3) \in F$ ;  $PRED[3] = 1$ .

$w = 4$ ;  $j = 4$ ;  $N[4] = 4$ ;  $W = \{2, 3, 4\}$ ;  $(1, 4) \in F$ ;  $PRED[4] = 1$ .

$v = 2$ ;  $W = \{3, 4\}$ .

$w = 5$ ;  $j = 5$ ;  $N[5] = 5$ ;  $W = \{3, 4, 5\}$ ;  $(2, 5) \in F$ ;  $PRED[5] = 2$ .

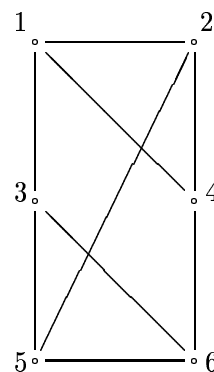
$v = 3$ ;  $W = \{4, 5\}$ .

$w = 6$ ;  $j = 6$ ;  $N[6] = 6$ ;  $W = \{4, 5, 6\}$ ;  $(3, 6) \in F$ ;  $PRED[6] = 3$ .

$v = 4$ ;  $W = \{5, 6\}$ .  $v = 5$ ;  $W = \{6\}$ .  $v = 6$ ;  $W = \emptyset$ .

We eindigen dus met één component ( $k = 1$ ) en met:

$N[1] = 1$ ;  $N[2] = 2$ ;  $N[3] = 3$ ;  $N[4] = 4$ ;  $N[5] = 5$ ;  $N[6] = 6$ .  $F = \{(1, 2), (1, 3), (1, 4), (2, 5), (3, 6)\}$ .



**Stelling 2.5** *De procedure van het zijwaarts zoeken heeft de volgende eigenschappen: (i) de complexiteit is  $\mathcal{O}(p)$ , waarbij  $p = \max(n, m)$ ; (ii) de pijlen van  $F$  vormen een opspannend gericht bos  $T$  met als wortels de knooppunten met  $PRED[v] = 0$ ; (iii) het aantal pijlen in  $T$  van een wortel  $v$  naar een knooppunt  $w$  is het minimum aantal takken om in de graaf van  $v$  naar  $w$  te gaan.*

### Bewijs

(i) Het bewijs gaat analoog aan het bewijs van Stelling 2.1.

- (ii) Het bewijs gaat analoog aan het bewijs van Stelling 2.3 (i).
- (iii) Het bewijs volgt uit de opmerking direct voorafgaand aan Algoritme 2.3 ■

## 2.3 Streng samenhangende componenten

In deze paragraaf gaan we uit van een gerichte graaf  $G = (V, A)$  en willen we de streng samenhangende componenten (afgekort: s.s.c. en in het Engels strongly connected components) van  $G$  bepalen. We gebruiken daartoe een variant van het voorwaarts zoeken. Voorwaarts zoeken verdeelt de graaf in samenhangende componenten, maar deze hoeven niet per se streng samenhangend te zijn. Wel is het zo (Stelling 2.4) dat de knooppunten van een s.s.c. in dezelfde samenhangende component komen: de met voorwaarts zoeken geconstrueerde componenten bevatten elk een aantal s.s.c., maar we weten nog niet hoe de onderverdeling er uitziet. Zo bevat in voorbeeld 2.2 de eerste component de knooppunten 1,2,3, en 4, terwijl  $\{1\}$  en  $\{2, 3, 4\}$  de s.s.c. zijn; de tweede component bestaat uit de knooppunten 5 en 6, die ieder zelf een s.s.c. zijn.

Bij het voorwaarts zoeken worden de pijlen van de graaf verdeeld in  $F, I, D$  en  $C$ . Het basis idee om een s.s.c. op te sporen is het ontdekken van rondes, omdat knooppunten die tot een ronde behoren in dezelfde s.s.c. zitten. Als we uitgaan van  $F$ , dan is volgens Lemma 2.1 de verz.  $I$  niet van belang (in  $F$  was al een pad, dus een pijl van  $I$  voegt niets toe voor de strenge samenhang).

Zij  $V_1$  de knooppunten van een s.s.c. met bijbehorende boom  $T_1$  van  $F$ . Dan is de wortel  $z$  van  $T_1$  het knooppunt van  $T_1$  met de kleinste  $N$ -waarde. De vraag is nu hoe we zo'n wortel  $z$  kunnen bepalen zonder  $T_1$  zelf te kennen. Het idee is om dit te doen door de laagste  $N$ -waarde van knooppunten op een ronde te zoeken.

We voeren hiervoor de volgende begrippen in:

$r(v)$  is de wortel van de deelboom van de s.s.c. waar  $v$  tot behoort.

$CURRENT(v) = \{w \in V \mid R[v] \leq R[r(w)]\}$ : dit is de verz. van knooppunten  $w$  zdd.  $v$  niet later is afgehandeld dan de wortel van de deelboom van de s.s.c. waar  $w$  toe behoort.

$Z[v] = \min\{N[v], \min\{N[w] \mid (v, w) \in D \cup C; w \in CURRENT(v)\}, \min\{Z[w] \mid (v, w) \in F\}\}$ .

De waarde  $Z[v]$  is dus het minimum van een drietal getallen, waaronder ook  $Z$ -getallen: dit is dus een recursieve definitie. We merken tevens op dat  $(v, w) \in D \cup C$  d.e.s.d. als  $N[w] \leq N[v]$ , zodat  $Z[v]$  gelijk is aan:

$\min\{N[v], \min\{N[w] \mid (v, w) \in A; N[w] \leq N[v]; w \in CURRENT(v)\}, \min\{Z[w] \mid (v, w) \in F\}\}$ .

We zullen bewijzen dat  $z$  een wortel van een s.s.c. is d.e.s.d. als  $Z[z] = N[z]$  en dat verder tot deze s.s.c. de knooppunten behoren die eerder dan  $z$  zijn afgehandeld en nog niet zijn ingedeeld in een s.s.c. We zullen dit eerst in een voorbeeld laten zien.

### Voorbeeld 2.2 (vervolg)

We hebben bij het voorwaarts zoeken gezien dat  $R[1] = 4$ ,  $R[2] = 1$ ,  $R[3] = 3$ ,  $R[4] = 2$ ,  $R[5] = 6$  en  $R[6] = 5$ . We weten, als we gebruik maken van de wetenschap dat  $\{1\}$ ,  $\{2, 3, 4\}$ ,  $\{5\}$  en  $\{6\}$  de

s.s.c. zijn, dat het volgende geldt:  $r(1) = 1$ ,  $r(2) = 3$ ,  $r(3) = 3$ ,  $r(4) = 3$ ,  $r(5) = 5$  en  $r(6) = 6$ .

Hiermee liggen de verz.  $\text{CURRENT}(v)$  vast:

$\text{CURRENT}(1) = \{1, 5, 6\}$ ,  $\text{CURRENT}(2) = \{1, 2, 3, 4, 5, 6\}$ ,  $\text{CURRENT}(3) = \{1, 2, 3, 4, 5, 6\}$ ,

$\text{CURRENT}(4) = \{1, 2, 3, 4, 5, 6\}$ ,  $\text{CURRENT}(5) = \{5\}$  en  $\text{CURRENT}(6) = \{5, 6\}$ .

De  $Z$ -getallen kunnen worden berekend in de volgorde waarin de knooppunten worden afgehandeld, d.w.z. in de volgorde van de  $R$ -getallen:

$Z[2] = \min\{N[2] = 4, \min\{N[3]\} = 2, \min\{\emptyset\} = \infty\} = 2 \neq N[2]$ : 2 is geen wortel van een s.s.c.

$Z[4] = \min\{N[4] = 3, \min\{\emptyset\} = \infty, \min\{Z[2] = 2\}\} = 2 \neq N[4]$ : 4 is geen wortel van een s.s.c.

$Z[3] = \min\{N[3] = 2, \min\{\emptyset\} = \infty, \min\{Z[4] = 2\}\} = 2 = N[3]$ : 3 is de wortel van een s.s.c. die bestaat uit  $\{3, 4, 2\}$ .

$Z[1] = \min\{N[1] = 1, \min\{\emptyset\} = \infty, \min\{Z[3] = 2\}\} = 1 = N[1]$ : 1 is de wortel van een s.s.c. die alleen uit  $\{1\}$  bestaat.

$Z[6] = \min\{N[6] = 6, \min\{\emptyset\} = \infty, \min\{\emptyset\} = \infty\} = 6 = N[6]$ : 6 is de wortel van een s.s.c. die alleen uit  $\{6\}$  bestaat.

$Z[5] = \min\{N[5] = 5, \min\{\emptyset\} = \infty, \min\{Z[6] = 6\}\} = 5 = N[5]$ : 5 is de wortel van een s.s.c. die alleen uit  $\{5\}$  bestaat.

We zullen nu de beweringen gaan bewijzen.

**Lemma 2.2** *Voor ieder knooppunt  $v$  geldt:  $Z[v] \geq N[r(v)]$ .*

### Bewijs

We beschouwen de volgende drie mogelijkheden voor  $Z[v]$ :

a.  $Z[v] = N[v]$ : dan kunnen we schrijven:  $N[r(v)] \leq N[(v)] = Z[v]$ .

b.  $Z[v] = N[w] < N[v]$  voor zekere  $w$  en  $v$  is niet later afgehandeld dan  $r(w)$ . Nu geldt:

$N[r(w)] \leq N[w] < N[v]$ : omdat  $v$  is niet later is afgehandeld dan  $r(w)$ , geldt  $r(v) = r(w)$ .

Maar dan kunnen we schrijven:  $Z[v] = N[w] \geq N[r(w)] = N[r(v)]$ .

c. Het laatste geval is:  $Z[v] = Z[w] < N[v]$  met  $(v, w) \in F$ .

Om dit geval te bewijzen passen we inductie naar de waarde  $R[v]$  toe.

Als  $R[v] = 1$ , dan is er geen enkele  $w$  met  $(v, w) \in F$ , zodat geval a of b van toepassing is, waarvoor reeds is bewezen dat  $Z[v] \geq N[r(v)]$ . Vervolgens beschouwen we de inductie stap.

Omdat  $(v, w) \in F$  geldt  $R[w] < R[v]$ . Dus voor  $w$  geldt het lemma volgens de inductieveronderstelling, zodat  $Z[w] \geq N[r(w)]$ . Nu zijn er twee mogelijkheden:

ðfwel  $r(w) = r(v)$ , en dan geldt:  $Z[v] = Z[w] \geq N[r(w)] = N[r(v)]$ .

ðfwel  $r(w) = w$ , en er geldt:  $Z[v] = Z[w] \geq N[r(w)] = N[w] \geq N[v] \geq N[r(v)]$ . ■

### Gevolg

Als  $z$  de wortel is van een s.s.c., dan geldt:  $Z[z] = N[z]$ .

### Bewijs

Omdat  $z$  de wortel is van een s.s.c., is  $r(z) = z$ , zodat we kunnen schrijven:

$Z[z] \geq N[r(z)] = N[z] \geq Z[z]$ . Dus  $Z[z] = N[z]$ . ■

**Stelling 2.6**  $z$  is de wortel van een streng samenhangende component d.e.s.d.  $Z[z] = N[z]$ .

### Bewijs

Op grond van bovenstaand gevolg is het voldoende te bewijzen dat  $Z[z] = N[z]$  impliceert dat  $z$  de wortel is van een s.s.c. Veronderstel dat  $Z[z] = N[z]$ , dat  $z$  ligt in de s.s.c.  $G_1 = (V_1, A_1)$  met bijbehorende boom  $T_1$  van  $F$ , en dat  $r \neq z$  de wortel is van  $T_1$ . In  $T_1$  is er dus een pad  $P_1$  van  $r$  naar  $z$  met pijlen van  $F$ . In verband met de strenge samenhang is er in  $G_1$  een pad  $P_2$  van  $z$  naar  $r$ . Omdat  $F$  geen kringen bevat behoren niet alle pijlen van  $P_2$  tot  $F$ . Beschouw nu onder alle mogelijke paden in  $G_1$  van  $z$  naar  $r$  een pad dat zo lang mogelijk pijlen van  $F$  kiest en noem dat weer  $P_2$ :  $P_2 = [z = v_0, v_1, \dots, v_{i-1}, v_i, \dots, v_k = r]$  met  $(v_{i-1}, v_i)$  de eerste pijl van  $P_2$  die niet tot  $F$  behoort. Dan behoort  $(v_{i-1}, v_i)$  tot  $D \cup C$  (als  $(v_{i-1}, v_i) \in I$ , dan is er ook een pad in  $F$  van  $z$  naar  $v_i$  en nemen we dit pad omdat het langer de pijlen van  $F$  volgt). Er geldt nu:

$Z[z] = Z[v_0] \leq Z[v_1] \cdots \leq Z[v_{i-1}]$  en  $N[v_i] < N[z]$  (want  $v_i$  moet eerder bezocht zijn dan  $z = v_0, v_1, \dots, v_{i-1}$ ). Omdat  $v_i \in G_1$  is  $r(v_i) = r$ , dus  $R[r(v_i)] = R[r] \geq R[v_{i-1}]$ , zodat  $v_i \in \text{CURRENT}(v_{i-1})$ . Hieruit volgt nu:  $Z[v_{i-1}] \leq N[v_i] < N[z] = Z[z] \leq Z[v_{i-1}]$ : tegenspraak. Onze veronderstelling dat  $r \neq z$  is dus onjuist, waarmee het bewijs geleverd is. ■

Om daadwerkelijk  $Z[v]$  te kunnen berekenen moeten we kunnen nagaan of  $w \in \text{CURRENT}(v)$ , d.w.z. of  $v$  eerder is afgehandeld dan  $r(w)$ . Zet daartoe de knooppunten zodra ze bezocht worden op een *stapel*  $S$ . De knooppunten blijven hierop staan totdat de wortel van hun "deelboom" wordt afgehandeld, d.w.z. totdat voor een  $z$  wordt ontdekt dat  $Z[z] = N[z]$ . Op dat moment worden de bovenste knooppunten van  $S$  t/m knooppunt  $z$  in een s.s.c. gestopt. De test of een knooppunt tot  $S$  behoort kan in  $\mathcal{O}(1)$  worden uitgevoerd door tevens een array  $S^*$  met componenten 0 of 1 bij te houden, waarbij  $S^*[v] = 1$  d.e.s.d. als  $v \in S$ . Hiermee is het niet meer nodig de  $R$ -waarden expliciet bij te houden.

De berekening van  $Z[v]$  verloopt nu als volgt:

1. Als  $v$  voor het eerst wordt bezocht:  $Z[v] := N[v]$ .
2. Als we  $(v, w)$  bekijken en  $N[w] \leq N[v]$ : als  $S^*[w] = 1$  en  $N[w] < Z[v]$ :  $Z[v] := N[w]$ .
3. Als  $w$  wordt afgehandeld en  $v = \text{PRED}[w]$ : als  $Z[w] < Z[v]$ :  $Z[v] := Z[w]$ .

*Algoritme 2.4: Bepaling streng samenhangende componenten in een gerichte graaf*

1.  $N[v] = \text{PRED}[v] = S^*[v] = 0$ ,  $v = 1, 2, \dots, n$ ;  $S = \emptyset$ ;  $j = 0$ ;  $k = 1$ .
2. Kies een knooppunt  $v$  met  $N[v] = 0$ .
3.  $j := j + 1$ ;  $N[v] = Z[v] = j$ ; voeg  $v$  aan de stapel  $S$  toe;  $S^*[v] = 1$ .
4. Doe voor iedere  $w \in L[v]$  het volgende:
  - a. Als  $N[w] = 0$ :  $\text{PRED}[w] = v$  en voer de stappen 3 en 4 uit met  $v = w$ .
  - b. Als  $N[w] \neq 0$ : als  $N[w] \leq N[v]$  en  $S^*[w] = 1$  en  $N[w] < Z[v]$ :  $Z[v] = N[w]$ .
5. Als  $\text{PRED}[v] \neq 0$  en  $Z[v] < Z[\text{PRED}[v]]$ :  $Z[\text{PRED}[v]] = Z[v]$ .
6. Als  $Z[v] = N[v]$ :
  - a. Doe voor alle  $x$  op de stapel  $S$  van bovenaf tot en met  $v$  het volgende: haal  $x$  van  $S$  af;  $S^*[x] = 0$ ;  $\text{COMP}[x] = k$ .

b.  $k := k + 1$ .

7. Als er nog knooppunt  $v$  is met  $N[v] = 0$ : ga naar stap 2.

Aan het einde van het algoritme heeft een knooppunt  $v$  voor  $\text{COMP}[v]$  de waarde  $k$  d.e.s.d. als  $v$  in de  $k$ -de streng samenhangende component zit.

### Voorbeeld 2.2 (vervolg)

$N[1] = 0$ ;  $N[2] = 0$ ;  $N[3] = 0$ ;  $N[4] = 0$ ;  $N[5] = 0$ ;  $N[6] = 0$ .

$\text{PRED}[1] = 0$ ;  $\text{PRED}[2] = 0$ ;  $\text{PRED}[3] = 0$ ;  $\text{PRED}[4] = 0$ ;  $\text{PRED}[5] = 0$ ;  $\text{PRED}[6] = 0$ .

$S^*[1] = 0$ ;  $S^*[2] = 0$ ;  $S^*[3] = 0$ ;  $S^*[4] = 0$ ;  $S^*[5] = 0$ ;  $S^*[6] = 0$ .  $S = \emptyset$ ;  $j = 0$ ;  $k = 1$ .

$v = 1$ ;  $j = 1$ ;  $N[1] = Z[1] = 1$ ;  $S = \{1\}$ ;  $S^*[1] = 1$ ;  $w = 3$ :  $\text{PRED}[3] = 1$ .

$v = 3$ ;  $j = 2$ ;  $N[3] = Z[3] = 2$ ;  $S = \{3, 1\}$ ;  $S^*[3] = 1$ ;  $w = 4$ :  $\text{PRED}[4] = 3$ .

$v = 4$ ;  $j = 3$ ;  $N[4] = Z[4] = 3$ ;  $S = \{4, 3, 1\}$ ;  $S^*[4] = 1$ ;  $w = 2$ :  $\text{PRED}[2] = 4$ .

$v = 2$ ;  $j = 4$ ;  $N[2] = Z[2] = 4$ ;  $S = \{2, 4, 3, 1\}$ ;  $S^*[2] = 1$ ;  $w = 3$ :  $Z[2] = 2$ .

$v = 2$  is afgehandeld; stap 5:  $2 = Z[2] < Z[4] = 3$ :  $Z[4] = 2$ ; stap 6 geeft geen verandering; ga verder met  $v = 4$ .

$v = 4$  is afgehandeld: de stappen 5 en 6 geven geen veranderingen; ga verder met  $v = 3$ .

$v = 3$  is afgehandeld: stap 5 levert niets op; stap 6:  $Z[3] = N[3] = 2$ :  $S = \{1\}$ ;  $S^*[2] = S^*[4] = S^*[3] = 0$ ;  $\text{COMP}[2] = \text{COMP}[4] = \text{COMP}[3] = 1$ ;  $k = 2$ ; ga verder met  $v = 1$ .

$v = 1$ ;  $w = 4$ : dit geeft verder geen veranderingen;  $v = 1$  is afgehandeld: stap 5 levert niets op; stap 6:  $Z[1] = N[1] = 1$ :  $S = \emptyset$ ;  $S^*[1] = 0$ ;  $\text{COMP}[1] = 2$ ;  $k = 3$ .

$v = 5$ ;  $j = 5$ ;  $N[5] = Z[5] = 5$ ;  $S = \{5\}$ ;  $S^*[5] = 1$ ; voor  $w = 1$  en  $w = 2$  zijn er geen veranderingen;  $w = 6$ :  $\text{PRED}[6] = 5$ .

$v = 6$ ;  $j = 6$ ;  $N[6] = Z[6] = 6$ ;  $S = \{6, 5\}$ ;  $S^*[6] = 1$ ;  $w = 2$ : geen veranderingen;

$v = 6$  is afgehandeld; stap 5 levert niets op; stap 6:  $Z[6] = N[6] = 6$ :  $S = \{5\}$ ;  $S^*[6] = 0$ ;  $\text{COMP}[6] = 3$ ;  $k = 4$ ; ga verder met  $v = 5$ .

$v = 5$  is afgehandeld; stap 5 levert niets op; stap 6:  $Z[5] = N[5] = 5$ :  $S = \emptyset$ ;  $S^*[5] = 0$ ;  $\text{COMP}[5] = 4$ ;  $k = 5$ .

De streng samenhangende componenten zijn:  $C_1 = \{2, 4, 3\}$ ;  $C_2 = \{1\}$ ;  $C_3 = \{6\}$  en  $C_4 = \{5\}$ .

**Stelling 2.7** *Algoritme 2.4 heeft complexiteit  $\mathcal{O}(p)$ , waarbij  $p = \max(n, m)$ .*

### Bewijs

Uit Stelling 2.3 volgt dat, zonder de bewerkingen aan  $Z$ ,  $S$  en  $S^*$  mee te tellen, de complexiteit  $\mathcal{O}(p)$  is. De bewerkingen aan  $Z$  zijn:  $\mathcal{O}(n)$  bij de eerste toekenning in stap 3; stap 4 geeft voor het totale algoritme  $\mathcal{O}(m)$  en stap 5  $\mathcal{O}(n)$ . Totaal geeft dit  $\mathcal{O}(p)$  voor het werk aan  $Z$ .

Voor het werk aan  $S$  en  $S^*$  geldt het volgende. Ieder knooppunt wordt éénmaal op  $S$  gezet en er ook éénmaal afgehaald; ieder knooppunt krijgt bij initialisatie de  $S^*$ -waarde 0, dan zodra het bereikt wordt de waarde 1 en tenslotte, nadat het van  $S$  is afgehaald weer de waarde 0. Verder wordt voor iedere pijl  $(v, w)$  in stap 4b getest of  $S^*[w] = 1$ . De totale hoeveelheid werk aan  $S$  en  $S^*$  is dus  $\mathcal{O}(n)$ . ■

## 2.4 Opgaven

### Opgave 2.1

Pas het voorwaarts zoeken toe op een niet-gerichte graaf met 5 knooppunten en met  $L[1] = \{2, 3, 5\}$ ,  $L[2] = \{1, 3, 4\}$ ,  $L[3] = \{1, 2, 3, 4, 5\}$ ,  $L[4] = \{2, 3\}$  en  $L[5] = \{1, 3\}$ .

### Opgave 2.2

Zij  $G$  een niet-gerichte samenhangende graaf en  $T$  de opspannende gerichte boom bestaande uit de pijlen van  $F$ , verkregen met het voorwaarts zoeken. Laat  $G_1$  een deelgraaf van  $G$  zijn die volledig is.

- Toon aan dat er in  $T$  een pad is waar alle knooppunten van  $G_1$  op liggen.
- Ga na of op dit pad ook andere knooppunten kunnen liggen.

### Opgave 2.3

Zij  $G$  een niet-gerichte graaf en  $T$  het opspannende gerichte bos bestaande uit de pijlen van  $F$ , verkregen met het voorwaarts zoeken. Zij  $T[v]$  het aantal knooppunten dat in  $T$  vanuit  $v$  kan worden bereikt, inclusief  $v$  zelf.

- Toon aan dat  $w$  in  $T$  vanuit  $v$  bereikbaar is d.e.s.d. als  $N[v] \leq N[w] < N[v] + T[v]$ .
- Pas algoritme 2.1 aan zdd. daarmee ook de getallen  $T[v]$ ,  $v \in V$ , worden berekend.

### Opgave 2.4

Stel een  $\mathcal{O}(m)$  algoritme op om na te gaan of een niet-gerichte samenhangende graaf bipartiet is.

### Opgave 2.5

Stel een algoritme op, gebaseerd op het voorwaarts zoeken, om een kringbasis in een niet-gerichte graaf te bepalen. Bepaal tevens de complexiteit van het voorgestelde algoritme.

### Opgave 2.6

Pas het voorwaarts zoeken toe op een gerichte graaf met 9 knooppunten en met  $L[1] = \{2\}$ ,  $L[2] = \{3, 4\}$ ,  $L[3] = \{1\}$ ,  $L[4] = \{3, 6\}$ ,  $L[5] = \{2, 6, 9\}$ ,  $L[6] = \emptyset$ ,  $L[7] = \{6, 8\}$ ,  $L[8] = \{4\}$  en  $L[9] = \{7, 8\}$ .

### Opgave 2.7

Beschouw een gerichte graaf  $G = (V, A)$ .

Stel een  $\mathcal{O}(p)$  algoritme op, waarbij  $p = \max(n, m)$ , om de knooppunten zó van 1 tot en met  $n$  te nummeren dat als  $(v, w) \in A$ , dan is  $N[v] < N[w]$ , indien dit mogelijk is; indien dit niet mogelijk is, toon dan aan dat  $G$  een ronde heeft en implementeer deze eigenschap in het algoritme door een variabele "ronde" dan de waarde 1 te geven.

### Opgave 2.8

Pas het zijwaarts zoeken toe op een niet-gerichte graaf met 5 knooppunten en met  $L[1] = \{2, 3, 5\}$ ,  $L[2] = \{1, 3, 4\}$ ,  $L[3] = \{1, 2, 3, 4, 5\}$ ,  $L[4] = \{2, 3\}$  en  $L[5] = \{1, 3\}$ .

### Opgave 2.9

Zij  $G = (V, E)$  een niet-gerichte samenhangende graaf. Kies een knooppunt  $v$ .

- Stel een  $\mathcal{O}(m)$  algoritme op om voor ieder knooppunt  $w$  het minimum aantal takken op een keten tussen  $v$  en  $w$  te bepalen.
- Pas dit algoritme toe (met  $v = 1$ ) op de graaf met de volgende structuurlijsten:  
 $L[1] = \{2, 3, 4\}$ ,  $L[2] = \{1, 4, 6\}$ ,  $L[3] = \{1, 4, 5\}$ ,  $L[4] = \{1, 2, 3, 5\}$ ,  $L[5] = \{3, 4\}$ ,  
 $L[6] = \{2, 7\}$  en  $L[7] = \{6\}$ .

### Opgave 2.10

Pas algoritme 2.4 toe op de gerichte graaf, gerepresenteerd door de volgende lijsten:

$L[1] = \{2\}$ ;  $L[2] = \{4, 8\}$ ;  $L[3] = \{9\}$ ;  $L[4] = \{6, 7\}$ ;  $L[5] = \{3, 4\}$ ;  $L[6] = \{4\}$ ;  $L[7] = \{6\}$ ;  $L[8] = \{1, 5\}$ ;  $L[9] = \{10\}$ ;  $L[10] = \{3, 11\}$ ;  $L[11] = \{9\}$ .

### Opgave 2.11 (onderdeel c heeft een \*-moeilijkheid)

In deze opgave bespreken we een ander algoritme om de streng samenhangende componenten van een gerichte graaf  $G = (V, A)$  te bepalen. Dit algoritme gaat als volgt.

- Voer het voorwaarts zoeken uit en nummer de knooppunten in de volgorde waarin ze worden afgehandeld (het eerst afgehandelde knooppunt krijgt nummer 1 etc).
- Construeer een andere gerichte graaf  $G_1 = (V, A_1)$  met  $(v, w) \in A_1$  d.e.s.d. als  $(w, v) \in A$ .
- Voer het voorwaarts zoeken uit in  $G_1$ , beginnend met het hoogst genummerde knooppunt (volgens de in stap 1 aangebrachte nummering). Indien met een nieuwe boom moet worden gestart, dan beginnen we weer met het nog niet bezochte knooppunt met het hoogste nummer.
- Neem als s.s.c. de samenhangende componenten die in stap 3 worden gevonden.

Beantwoord de volgende vragen over dit algoritme:

- Pas het algoritme toe op voorbeeld 2.2.
- Bepaal de complexiteit van de methode.
- Toon de correctheid van het algoritme aan.



# Hoofdstuk 3

## Kortste paden

### 3.1 Inleiding

Zij  $N = (V, A)$  een netwerk met *lengtefunctie*  $l$ . De lengte van een pad of ronde is de som van de lengtes van de pijlen op het pad of ronde. In dit hoofdstuk bespreken we de volgende problemen:

- bepaal het kortste pad van een bepaald knooppunt naar alle andere knooppunten;
- bepaal de kortste paden tussen ieder tweetal knooppunten.

**Aanname 3.1** *De lengte van iedere ronde is positief.*

Zonder deze aanname is het begrip kortste pad niet goed gedefinieerd (waarom?). Definieer:

$$\begin{cases} u_1^* = 0 \\ u_i^* = \text{de lengte van het kortste pad van } v_1 \text{ naar } v_i, i = 1, 2, \dots, n \end{cases}$$

In het dictaat Inleiding Besliskunde is het onderwerp kortste paden ook behandeld. Daar is de methode van Dijkstra besproken voor het kortste pad van een bepaald knooppunt naar alle andere knooppunten in het geval alle lengtes niet-negatief zijn. Verder is de simplex methode besproken. Deze methode is ook geldig voor negatieve lengtes. Beide methoden zijn nauw verwant met de volgende stelling<sup>1</sup>.

**Stelling 3.1 (Bellman-vergelijkingen)** (i) *De getallen  $u_i^*$ ,  $i = 1, 2, \dots, n$  voldoen aan de zo-*

*genaamde Bellman-vergelijkingen:* 
$$\begin{cases} u_1 = 0 \\ u_i = \min_{k \neq i} [u_k + l_{ki}], i = 2, 3, \dots, n \end{cases}$$

(ii) *Met iedere oplossing  $u$  van deze vergelijkingen correspondeert een opspannende boom met wortel  $v_1$  zdd.  $u_i$  de boomafstand is van  $v_1$  naar  $v_i$ ,  $i = 1, 2, \dots, n$ ; (iii) *Er is een opspannende boom van kortste paden; (iv)  $u_i^*$ ,  $i = 1, 2, \dots, n$  is de enige oplossing van de Bellman-vergelijkingen.**

In de volgende twee paragrafen zullen we nog een methode bespreken voor het kortste pad van een bepaald knooppunt naar alle andere knooppunten (voor het algemene geval met ook negatieve lengtes) en een methode om de kortste paden tussen ieder tweetal knooppunten te bepalen.

---

<sup>1</sup>Voor een bewijs van de stelling zie het dictaat Inleiding Besliskunde

## 3.2 Methode van Bellman en Ford

In deze methode worden de Bellman-vergelijkingen opgelost door de volgende iteratieve methode, ook wel *successieve approximatie* genoemd. Zonder beperking van de algemeenheid nemen we aan dat er tussen ieder tweetal knooppunten een pijl is.

We beginnen met als eerste benadering van  $u^*$ :  $u_1^1 = 0$ ;  $u_j^1 = l_{1j}$ ,  $2 \leq j \leq n$ .

In het algemeen geldt voor de  $k$ -de benadering  $u^k$ :  $u_j^k$  is de lengte van het kortste pad van  $v_1$  naar  $v_j$  met hoogstens  $k$  pijlen,  $k = 1, 2, \dots$ . De benadering  $u^{k+1}$  kan dan als volgt uit  $u^k$  worden bepaald:

$$u_j^{k+1} = \min \{u_j^k, \min_i [u_i^k + l_{ij}]\}, \quad j = 2, 3, \dots, n. \quad (3.1)$$

Omdat de lengte van een kortste pad hoogstens  $n - 1$  pijlen heeft, is  $u^{n-1} = u^*$ . Het algoritme van deze methode is dus als volgt, waarbij  $k(j)$  de voorganger van knooppunt  $j$  is op het kortste pad van  $v_1$  naar  $v_j$ ,  $j = 2, 3, \dots, n$ .

*Algoritme 3.1: Kortste paden van  $v_1$  naar  $v_i$ ,  $i = 2, 3, \dots, n$  volgens Bellman en Ford (versie 1)*

1.  $u_1 = 0$ ;  $w_j = u_j = l_{1j}$ ,  $2 \leq j \leq n$ ;  $k(j) = 1$ ,  $2 \leq j \leq n$ ;  $k = 1$ .

2. Voor  $j = 2, 3, \dots, n$  doe:

Voor  $i = 2, 3, \dots, n$  doe: als  $u_i + l_{ij} < w_j$ :  $w_j = u_i + l_{ij}$  en  $k(j) = i$ .

3. a. Voor  $j = 2, 3, \dots, n$  doe:  $u_j = w_j$ ;

b.  $k := k + 1$ ;

c. Als  $k = n - 1$ : stop.

Anders: ga naar stap 2.

Bovenstaand algoritme kan op de volgende wijze worden versneld:

1. Zodra  $u^{k+1} = u^k$ , zal er nooit meer iets veranderen. Dan is dus  $u^* = u^k$  en kunnen we stoppen.

2. Zij  $w_j^{k+1} = \min \{w_j^k, \min_{i < j} [w_i^{k+1} + l_{ij}], \min_{i > j} [w_i^k + l_{ij}]\}$ ,  $j = 2, 3, \dots, n$  (start met  $w^1 = u^1$ ).

Dan geldt:

(1) voor alle  $k$  en  $j$  is  $w_j^k \leq u_j^k$  en is er een pad van  $v_1$  naar  $v_j$  met lengte  $w_j^k$ ;

(2) omdat  $u_j^* \leq w_j^{n-1} \leq u_j^{n-1} = u_j^*$ ,  $2 \leq j \leq n$ , geeft  $w^{n-1}$  ook de lengte van kortste paden;

(3) meestal convergeert  $w^k$  veel sneller dan  $u^k$  naar  $u^*$ .

Bovenstaande overwegingen leiden tot het volgende algoritme.

*Algoritme 3.2: Kortste paden van  $v_1$  naar  $v_i$ ,  $i = 2, 3, \dots, n$  volgens Bellman en Ford (versie 2)*

1.  $u_1 = 0$ ;  $u_j = l_{1j}$ ,  $2 \leq j \leq n$ ;  $k(j) = 1$ ,  $2 \leq j \leq n$ ;  $k = 1$ ; klaar = 1.

2. Voor  $j = 2, 3, \dots, n$  doe:

Voor  $i = 2, 3, \dots, n$  doe:

Als  $u_i + l_{ij} < u_j$ :  $u_j = u_i + l_{ij}$ ,  $k(j) = i$  en klaar = 0.

3. a.  $k := k + 1$ ;

b. Als  $k = n - 1$  of klaar = 1: stop.

Anders: klaar = 1 en ga naar stap 2.

### Voorbeeld 3.1

Laat de lengtes van de pijlen van een graaf met 7 knooppunten gegeven zijn door nevenstaande lengtematrix  $L$ . Voor het getal  $\infty$  nemen we in de berekeningen 100. Tijdens de berekeningen geven we alleen aan wat verandert.

$$L = \begin{pmatrix} 0 & -3 & \infty & \infty & 2 & 3 & \infty \\ \infty & 0 & -5 & 15 & 12 & \infty & \infty \\ \infty & \infty & 0 & \infty & \infty & \infty & \infty \\ 6 & \infty & 8 & 0 & \infty & 4 & 11 \\ \infty & \infty & \infty & -7 & 0 & \infty & \infty \\ \infty & \infty & -10 & \infty & \infty & 0 & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty & 0 \end{pmatrix}$$

*Iteratie 1:*

$$u = (0, -3, 100, 100, 2, 3, 100).$$

$$k(2) = k(3) = k(4) = k(5) = k(6) = k(7) = 1. \quad k = 1; \text{ klaar} = 1.$$

*Iteratie 2:*

$$u_3 = -8; \quad k(3) = 2; \text{ klaar} = 0; \quad u_4 = 12; \quad k(4) = 2; \quad u_4 = -5; \quad k(4) = 5; \quad u_6 = -1; \quad k(6) = 4;$$

$$u(7) = 6; \quad k(7) = 4. \quad k = 2; \text{ klaar} = 1.$$

*Iteratie 3:*

$$u_3 = -11; \quad k(3) = 6; \text{ klaar} = 0. \quad k = 3; \text{ klaar} = 1.$$

*Iteratie 4:*

Geen veranderingen, dus we stoppen met  $u^* = (0, -3, -11, -5, 2, -1, 6)$  en met  $k = (-, 1, 6, 5, 1, 4, 4)$ .

De boom van kortste paden wordt gegeven door de pijlen  $\{(k(i), i), i = 2, 3, \dots, 7\}$ , dus door de pijlen  $\{(1, 2), (6, 3), (5, 4), (1, 5), (4, 6), (4, 7)\}$ .

**Stelling 3.2** *Algoritme 3.2 is correct en heeft complexiteit  $\mathcal{O}(n^3)$*

### Bewijs

De correctheid is hierboven reeds aangetoond. De eerste iteratie is van  $\mathcal{O}(n)$ . Daarna komen er nog hoogstens  $n - 2$  iteraties, die ieder complexiteit  $\mathcal{O}(n^2)$  hebben. ■

## 3.3 Methode van Floyd en Warshall

In plaats van de kortste paden vanuit één punt, beschouwen we in deze paragraaf de kortste paden tussen ieder tweetal punten. Laat  $u_{ij}^*$  de lengte van het kortste pad zijn van  $v_i$  naar  $v_j$  en  $k(i, j)$  het knooppunt dat direct voor  $j$  komt op het kortste pad van  $v_i$  naar  $v_j$ , voor alle  $i$  en  $j$ . We nemen weer aan dat het netwerk geen rondes met een negatieve lengte heeft.

Met achtereenvolgens ieder knooppunt als wortel zouden we bijvoorbeeld  $n$  keer het algoritme van Bellman en Ford kunnen toepassen. Dit geeft dan een methode met complexiteit  $\mathcal{O}(n^4)$ . Het kan echter ook met een methode van  $\mathcal{O}(n^3)$ , zoals Floyd en Warshall hebben voorgesteld.

In deze methode gebruiken we  $u_{ij}^k$  voor de lengte van het kortste pad van  $v_i$  naar  $v_j$  als we voor de tussenpunten slechts  $v_1, v_2, \dots, v_{k-1}$  mogen gebruiken. Hierbij is  $u_{ij}^1 = l_{ij}$  en  $k(i, j) = i$  voor alle  $i$  en  $j$ . Omdat het kortste pad van  $v_i$  naar  $v_j$  met de tussenpunten uit  $\{v_1, v_2, \dots, v_k\}$  knooppunt  $v_k$  niet of wel bevat, geldt:  $u_{ij}^{k+1} = \min[u_{ij}^k, u_{ik}^k + u_{kj}^k]$  voor alle  $i$  en  $j$ . Tenslotte merken we op dat  $u_{ij}^* = u_{ij}^{n+1}$  voor alle  $i$  en  $j$ . Het algoritme is dus als volgt.

*Algoritme 3.3: Kortste paden tussen ieder tweetal knooppunten volgens Floyd en Warshall*

1. Voor  $i = 1, 2, \dots, n$  doe:

Voor  $j = 1, 2, \dots, n$  doe:  $u_{ij} = l_{ij}$  en  $k(i, j) = i$ .

2. Voor  $k = 1, 2, \dots, n$  doe:

Voor  $i = 1, 2, \dots, n$  doe:

Voor  $j = 1, 2, \dots, n$  doe:

Als  $u_{ik} + u_{kj} < u_{ij}$ , dan:  $u_{ij} = u_{ik} + u_{kj}$ ;  $k(i, j) = k(k, j)$ .

**Stelling 3.3** *De complexiteit van de methode van Floyd en Warshall is  $\mathcal{O}(n^3)$*

### Bewijs

Het is direct duidelijk dat stap 1 complexiteit  $\mathcal{O}(n^2)$  heeft en stap 2  $\mathcal{O}(n^3)$ . ■

### Voorbeeld 3.2

Laat de lengtes van de pijlen van een graaf met zes knooppunten gegeven zijn door nevenstaande lengtematrix  $L$ . Tijdens de berekeningen geven we alleen aan wat verandert.

$$L = \begin{pmatrix} 0 & 3 & \infty & 2 & \infty & \infty \\ \infty & 0 & 10 & 6 & \infty & \infty \\ \infty & 1 & 0 & 9 & \infty & 3 \\ -1 & 8 & \infty & 0 & 4 & \infty \\ \infty & \infty & 2 & 3 & 0 & \infty \\ \infty & \infty & 1 & \infty & 4 & 0 \end{pmatrix}$$

*Iteratie 1:* De eerste  $U$ -matrix is gelijk aan  $L$  en  $k(i, j) = i$  voor  $i = 1, 2, \dots, 6$ .

*Iteratie 2:*  $u_{42} = 2$ ;  $k(4, 2) = 1$ .

*Iteratie 3:*  $u_{13} = 13$ ;  $k(1, 3) = 2$ ;  $u_{34} = 7$ ;  $k(3, 4) = 2$ ;  $u_{43} = 12$ ;  $k(4, 3) = 2$ .

*Iteratie 4:*  $u_{16} = 16$ ;  $k(1, 6) = 3$ ;  $u_{26} = 13$ ;  $k(2, 6) = 3$ ;  $u_{46} = 15$ ;  $k(4, 6) = 3$ ;  $u_{52} = 3$ ;  $k(5, 2) = 3$ ;  $u_{56} = 5$ ;  $k(5, 6) = 3$ ;  $u_{62} = 2$ ;  $k(6, 2) = 3$ ;  $u_{64} = 8$ ;  $k(6, 4) = 2$ .

*Iteratie 5:*  $u_{13} = 8$ ;  $k(1, 3) = 5$ ;  $u_{15} = 6$ ;  $k(1, 5) = 4$ ;  $u_{21} = 5$ ;  $k(2, 1) = 4$ ;  $u_{25} = 10$ ;  $k(2, 5) = 4$ ;  $u_{31} = 6$ ;  $k(3, 1) = 4$ ;  $u_{35} = 11$ ;  $k(3, 5) = 4$ ;  $u_{51} = 2$ ;  $k(5, 1) = 4$ ;  $u_{61} = 7$ ;  $k(6, 1) = 4$ ;  $u_{64} = 7$ ;  $k(6, 4) = 5$ .

*Iteratie 6:*  $u_{16} = 11$ ;  $k(1, 6) = 3$ ;  $u_{43} = 6$ ;  $k(4, 3) = 5$ ;  $u_{46} = 9$ ;  $k(4, 6) = 3$ ;  $u_{61} = 6$ ;  $k(6, 1) = 4$ ;  $u_{64} = 7$ ;  $k(6, 4) = 5$ .

*Iteratie 7:*  $u_{35} = 7$ ;  $k(3, 5) = 6$ .

Hieronder staan de verkregen matrices  $U^*$  en  $K$ .

$$U^* = \begin{pmatrix} 0 & 3 & 8 & 2 & 6 & 11 \\ 5 & 0 & 10 & 6 & 10 & 13 \\ 6 & 1 & 0 & 7 & 7 & 3 \\ -1 & 2 & 6 & 0 & 4 & 9 \\ 2 & 3 & 2 & 3 & 0 & 5 \\ 6 & 2 & 1 & 7 & 4 & 0 \end{pmatrix} \quad K = \begin{pmatrix} 1 & 1 & 5 & 1 & 4 & 3 \\ 4 & 2 & 2 & 2 & 4 & 3 \\ 4 & 3 & 3 & 2 & 6 & 3 \\ 4 & 1 & 5 & 4 & 4 & 3 \\ 4 & 3 & 5 & 5 & 5 & 3 \\ 4 & 3 & 6 & 5 & 6 & 6 \end{pmatrix}$$

Hiermee zijn alle kortste paden bepaald. Willen we bijvoorbeeld het kortste pad van  $v_6$  naar  $v_1$  hebben, dan lezen we af dat  $u_{61}^* = 6$ ; de route krijgen we door in de  $K$ -matrix vanaf  $k(6, 1)$  terug te lopen totdat we in knooppunt 6 zijn:  $k(6, 1) = 4$ , dus de laatste pijl is  $(4, 1)$ ;  $k(6, 4) = 5$ , dus de pijl daarvoor is  $(5, 4)$ ;  $k(6, 5) = 6$ , dus de eerste pijl is  $(6, 5)$ . Het pad is dus  $[6, 5, 4, 1]$ .

## 3.4 Enkele toepassingen

### *Het knapzakprobleem*

Iemand gaat op reis en moet een keuze maken uit  $n$  voorwerpen die hij wil meenemen. Voorwerp  $i$  heeft voor hem een waarde  $w_i$  en een volume  $a_i$ ,  $i = 1, 2, \dots, n$ . Het totale volume dat kan worden meegenomen is  $b$ . Welke voorwerpen neemt hij mee als hij de meegenomen waarde wil maximaliseren? Dit probleem kan als combinatorisch optimaliseringsprobleem worden geformuleerd met variabelen  $x_i \in \{0, 1\}$ ,  $i = 1, 2, \dots, n$ , waarbij  $x_i = 1$  d.e.s.d. als het  $i$ -de voorwerp wordt meegenomen. De formulering wordt dan:

$$\max \{ \sum_{i=1}^n w_i x_i \mid \sum_{i=1}^n a_i x_i \leq b; x_i \in \{0, 1\}, i = 1, 2, \dots, n \},$$

ofwel

$$-\min \{ \sum_{i=1}^n -w_i x_i \mid \sum_{i=1}^n a_i x_i \leq b; x_i \in \{0, 1\}, i = 1, 2, \dots, n \}.$$

We kunnen dit probleem ook modelleren als kortste pad probleem. We veronderstellen daarbij dat de volumes geheeltallig zijn. Neem knooppunten  $(i, k)$  voor  $0 \leq i \leq n + 1$  en  $0 \leq k \leq b$ . De  $i$  heeft betrekking op het  $i$ -de voorwerp en de  $k$  op het (opgebouwde) volume. Er zijn alleen pijlen van  $(i - 1, k)$  naar  $(i, k)$  en van  $(i - 1, k)$  naar  $(i, k + a_i)$ ,  $i = 1, 2, \dots, n$ ; de eerste pijl heeft als interpretatie dat het  $i$ -de voorwerp niet wordt gekozen en de tweede dat deze wel wordt gekozen; de eerste pijl heeft lengte 0 en de tweede lengte  $-w_i$ . Tenslotte zijn er pijlen van  $(n, k)$  naar  $(n + 1, b)$  met lengte 0. Het is nu direct in te zien dat het kortste pad van  $(0, 0)$  naar  $(n + 1, b)$  correspondeert met de optimale oplossing van het knapzakprobleem.

### *Vrachtschipprobleem*

Een vrachtschip maakt een rondreis langs een aantal havens, zeg langs de havens  $1, 2, \dots, n$ . Als de boot van haven  $i$  naar haven  $j$  vaart dan duurt dat  $t_{ij}$  uren en levert dat een winst van  $p_{ij}$  op. Voor een rondreis  $C$  langs een aantal havens noteren we met  $\mu(C)$  de winst per uur. Er geldt:  $\mu(C) = \frac{\sum_{(i,j) \in C} p_{ij}}{\sum_{(i,j) \in C} t_{ij}}$ . De directie vraagt zich af of er een rondreis gemaakt kan worden met een winst van meer dan  $\mu_0$  per uur. Deze vraagstelling kan ook als volgt worden geformuleerd: is er een rondreis  $C$  met  $\sum_{(i,j) \in C} [\mu_0 t_{ij} - p_{ij}] < 0$ ?

Beschouw een netwerk met als knooppunten de  $n$  havens en met  $l_{ij} = \mu_0 t_{ij} - p_{ij}$  voor alle  $i$  en  $j$ . Dan is het vrachtschipprobleem herschreven tot de vraag of er in dit netwerk een ronde met negatieve lengte is. Opgave 4 geeft antwoord op de vraag hoe dit kan worden nagegaan.

### *Schedulingprobleem*

Een call center moet telefonistes schedulen in een continu dienstrooster. Zij  $b_i$  het minimum aantal telefonistes dat nodig is gedurende uur  $i$ ,  $i = 1, 2, \dots, 24$ . Iedere telefoniste werkt in een shift van 8 opeenvolgende uren en zo'n shift kan op ieder uur beginnen. Het call center wil een cyclische schedule bepalen dat zich dagelijks herhaalt. Gegeven de getallen  $b_i$ ,  $i = 1, 2, \dots, 24$ ,

wordt gevraagd of er een cyclisch schedule mogelijk is met in totaal een gegeven aantal van  $N$  telefonistes.

Laat  $y_i$  het aantal telefonistes zijn dat hun dienst begint in uur  $i$ ,  $i = 1, 2, \dots, 24$ . We zoeken dus een toelaatbare oplossing  $y$ , met  $y_i \in \mathbb{N}_0$  ( $i = 1, 2, \dots, 24$ ) van het volgende stelsel:

$$\begin{cases} y_i + y_{i-1} + \dots + y_{i-7} & \geq b_i, \quad i = 8, 9, \dots, 24 \\ y_i + y_{i-1} + \dots + y_1 + y_{24} + y_{23} + \dots + y_{17+i} & \geq b_i, \quad i = 1, 2, \dots, 7 \\ y_1 + y_2 + \dots + y_{24} & = N \end{cases} \quad (3.2)$$

Definieer nieuwe variabelen  $u_i$ ,  $i = 1, 2, \dots, 24$ , waarbij  $u_i$  het aantal telefonistes is dat nieuw wordt ingezet gedurende de eerste  $i$  uren. Er geldt:  $u_1 = y_1$ ;  $u_2 = y_1 + y_2$ ;  $\dots$ ;  $u_{24} = y_1 + y_2 + \dots + y_{24}$ . Het stelsel (3.2) kan dus ook worden geschreven met variabelen  $u_i \in \mathbb{N}_0$ ,  $i = 1, 2, \dots, 24$ :

$$\begin{cases} u_i - u_{i-8} & \geq b_i, \quad i = 9, 10, \dots, 24 \\ u_i + u_{24} - u_{16+i} & \geq b_i, \quad i = 1, 2, \dots, 8 \\ u_i - u_{i-1} & \geq 0, \quad i = 2, 3, \dots, 24 \\ u_{24} & = N \end{cases} \quad (3.3)$$

Dit laatste stelsel is weer equivalent met het volgende stelsel waarin  $u_1 \geq 0$  en  $u_i \in \mathbb{N}$  ( $2 \leq i \leq 24$ ):

$$\begin{cases} u_{i-8} - u_i & \leq -b_i, \quad i = 9, 10, \dots, 24 \\ u_{16+i} - u_i & \leq N - b_i, \quad i = 1, 2, \dots, 8 \\ u_i - u_{i-1} & \leq 0, \quad i = 2, 3, \dots, 24 \end{cases} \quad (3.4)$$

Merk op dat als  $u$  een oplossing is van (3.4),  $u+c$  ook een oplossing is voor iedere constante vector  $c$ . We mogen dus  $u_1 = 0$  nemen. Merk bovendien op dat iedere rij van (3.4) precies één  $+1$  en één  $-1$  bevat, zodat de gespiegelde matrix van het stelsel (3.4) overeenkomt met de incidentiematrix van een gerichte graaf. Volgens Stelling 1.5 is deze matrix totaal unimodulair, zodat een eventuele oplossing van stelsel (3.4) in  $\mathbb{Z}$  is te vinden. Stelsel (3.2) heeft dus een toelaatbare oplossing  $y$  met  $y_i \in \mathbb{N}_0$  ( $i = 1, 2, \dots, 24$ ) d.e.s.d. als stelsel (3.4) met daaraan toegevoegd  $u_1 = 0$  een toelaatbare oplossing heeft.

Beschouw de gerichte graaf  $G = (V, A)$  met  $V = \{1, 2, \dots, 24\}$  en  $A = \{(i, i-8), 9 \leq i \leq 24\} \cup \{(i, 16+i), 1 \leq i \leq 8\} \cup \{(i, i-1), 2 \leq i \leq 24\}$ . Stelsel (3.4) met  $u_1 = 0$  is het stelsel dat behoort bij de simplex methode voor het kortste pad probleem op  $G$  t.a.v. de lengtefunctie  $l_{i,i-8} = -b_i$  ( $9 \leq i \leq 24$ ),  $l_{i,16+i} = N - b_i$  ( $1 \leq i \leq 8$ ) en  $l_{i,i-1} = 0$  ( $2 \leq i \leq 24$ )<sup>2</sup>.

**Lemma 3.1** *Stelsel (3.4) heeft een toelaatbare oplossing d.e.s.d. als  $l(C) \geq 0$  voor iedere ronde  $C$  in de graaf  $G = (V, A)$  met lengtefunctie  $l$ .*

### Bewijs

$\Rightarrow$  Beschouw een ronde  $C = \{i_1, i_2, \dots, i_k, i_{k+1} = i_1\}$ . Uit  $u_{i_{j+1}} - u_{i_j} \leq l_{i_j, i_{j+1}}$ ,  $1 \leq j \leq k$ , volgt:

<sup>2</sup>Zie de desbetreffende paragraaf in het dictaat Inleiding Besliskunde

$$0 = u_{i_{k+1}} - u_{i_1} = \sum_{j=1}^k (u_{i_{j+1}} - u_{i_j}) \leq \sum_{j=1}^k l_{i_j, i_{j+1}} = l(C).$$

⇐ Als er geen rondes met negatieve lengte zijn, dan is het begrip 'lengte van het kortste pad van knooppunt 1 naar knooppunt  $i$ ' goed gedefinieerd en noem deze lengte  $u_i$  voor  $i = 1, 2, \dots, 24$ . Deze  $u$  voldoet aan stelsel (3.4) met  $u_1 = 0$ . ■

Om het schedulingprobleem op te lossen moeten we dus nagaan of er al of niet een ronde met negatieve lengte is. Dit kan bijvoorbeeld worden gedaan met de methode van Bellman en Ford (zie Opgave 4). Het kan ook worden gedaan met de simplex methode voor het kortste pad probleem<sup>3</sup>.

### *Productieplanning*

Een productielijn bestaat uit  $n$  stadia en ieder stadium bestaat uit een bepaalde fabricage operatie gevolgd door een eventuele inspectie. Het product komt bij het begin van de productielijn binnen in batches van  $B$  exemplaren. Tijdens het fabricageproces kunnen in ieder stadium defecten ontstaan; de kans dat in stadium  $j$  in een exemplaar een defect ontstaat is een gegeven getal  $\alpha_j$ ,  $j = 1, 2, \dots, n$ . Een defect is niet repareerbaar en betekent productieverlies.

Aan het einde van ieder stadium kunnen we ofwel alle ofwel geen enkel exemplaar van een batch inspecteren. We nemen daarbij aan dat bij inspectie defecten ook werkelijk worden ontdekt en dat deze uit de batch worden verwijderd. Aan het einde van het laatste stadium vindt altijd inspectie plaats, zodat alleen goede eindproducten worden afgeleverd.

We gaan uit van de volgende bekende data:

$p_j$ : de productiekosten per exemplaar in stadium  $j$  ( $1 \leq j \leq n$ ).

$f_{ij}$ : de inspectiekosten per batch als aan het einde van stadium  $j$  een inspectie wordt gedaan, terwijl de vorige inspectie aan het einde van stadium  $i$  heeft plaats gevonden ( $0 \leq i < j \leq n$ );  $f_{0j}$  zijn de inspectiekosten als in stadium  $j$  voor het eerst inspectie plaats vindt).

$g_{ij}$ : de inspectiekosten per exemplaar als aan het einde van stadium  $j$  een inspectie wordt gedaan, terwijl de vorige inspectie aan het einde van stadium  $i$  heeft plaats gevonden ( $0 \leq i < j \leq n$ ).

Er wordt een optimaal inspectieplan gevraagd, d.w.z. minimaliseer de totale inspectie- en productiekosten voor een batch die aankomt met  $B$  exemplaren.

Dit probleem kan als volgt worden opgelost. Neem een gerichte graaf  $G = (V, A)$  met  $V = \{0, 1, \dots, n\}$  en  $A = \{(i, j) \mid i < j\}$ . Een pad is deze graaf van 0 naar  $n$  correspondeert dan éénduidig met een inspectieplan.

Laat  $B(i) = B \cdot \prod_{k=1}^i (1 - \alpha_k)$  = het verwachte aantal niet-defecte exemplaren aan het einde van stadium  $i$ ,  $1 \leq i \leq n$ . Als er dus aan het einde van stadium  $i$  inspectie plaats vindt, dan houden we  $B(i)$  goede exemplaren over. Neem  $l_{ij} = f_{ij} + B(i)g_{ij} + B(i) \sum_{k=i+1}^j p_k$  voor alle pijlen  $(i, j)$ . Het is nu direct duidelijk dat het kortste pad van 0 naar  $n$  correspondeert met het goedkoopste inspectieplan.

---

<sup>3</sup>Zie de opgaven over de simplex methode voor het kortste pad probleem in het dictaat Inleiding Besliskunde

## 3.5 Opgaven

### Opgave 3.1

Veronderstel dat  $u_j^k$  de lengte is van de kortste pijlreeks van  $v_1$  naar  $v_j$  met precies  $k$  pijlen,  $j = 2, 3, \dots, n$ . Geef een recurrente formule waaraan  $u^{k+1}$  voldoet.

### Opgave 3.2

Veronderstel dat  $G = (V, A)$  een netwerk is, waarbij  $(v_i, v_j) \in A$  impliceert dat  $i < j$ . Na hoeveel iteraties stopt algoritme 3.2?

### Opgave 3.3

Beschouw het netwerk met nevenstaande lengtematrix  $L$ . Pas algoritme 3.2 toe om de lengtes van de kortste paden en de boom van kortste paden te bepalen. Neem voor  $\infty$  de waarde 100.

$$L = \begin{pmatrix} 0 & \infty & 4 & 10 & 3 & \infty & \infty \\ \infty & 0 & -1 & -3 & 2 & 11 & 0 \\ \infty & 9 & 0 & 8 & 3 & 2 & 1 \\ \infty & 4 & 0 & 0 & 8 & 6 & 3 \\ \infty & 0 & 1 & 2 & 0 & 3 & -1 \\ \infty & 1 & -1 & 3 & 2 & 0 & 0 \\ \infty & 4 & 3 & \infty & \infty & 2 & 0 \end{pmatrix}$$

### Opgave 3.4

Beschouw een netwerk  $G = (V, A)$ , waarvan we niet weten of aan Aannname 3.1 is voldaan. Veronderstel dat er in  $v_1$  geen binnenkomende pijlen zijn en dat er voor iedere  $v_j \neq v_1$  een pad is van  $v_1$  naar  $v_j$ .

- Toon aan dat  $G$  een ronde met een negatieve lengte heeft d.e.s.d. als  $u_j^n < u_j^{n-1}$  voor zekere  $j$ , waarbij  $u_j^k$  wordt bepaald volgens formule (3.1).
- Gebruik onderdeel a om na te gaan of het netwerk uit Opgave 3.3, met daarin  $l_{62} = -1$  en de andere lengtes ongewijzigd, een ronde met negatieve lengte heeft.

### Opgave 3.5\*

Veronderstel dat  $G = (V, A)$  een netwerk is zonder rondes van negatieve lengtes, gegeven door de structuurlijsten  $L[v]$ ,  $v \in V$ . Beschouw het volgende algoritme.

- $T[1] = 1$ ;  $u_1 = 0$ ;  $Q = \{1\}$ .
- Voor  $j = 2, 3, \dots, n$  doe:  $u_j = \infty$  en  $T[j] = 0$ .
- Zolang  $Q \neq \emptyset$  doe:
  - Neem het eerste element van  $Q$ , zeg  $i$ , en verwijder  $i$  uit  $Q$ ;  $T[i] = 0$ .
  - Voor alle  $j \in L[i]$  met  $u_i + l_{ij} < u_j$  doe:
    - $u_j = u_i + l_{ij}$ ;  $k(j) = i$ ;
    - als  $T[j] = 0$ , dan: voeg  $j$  achteraan aan  $Q$  toe en  $T[j] = 1$ .

Beantwoord de volgende vragen over dit algoritme.

- Pas het algoritme toe op Voorbeeld 3.1.



- b. Toon aan dat voor iedere  $i \notin Q$  geldt:  $u_i + l_{ij} \geq u_j$  voor alle  $j$ .
- c. Toon aan dat  $Q$  in iedere iteratie een  $j$  bevat waarvoor geldt:  $u_j = u_j^*$ .
- d. Toon aan dat ieder knooppunt hoogstens  $n - 1$  keer in  $Q$  kan worden geplaatst.
- e. Toon aan dat het algoritme eindigt met de lengtes en de boom van kortste paden.
- f. Toon aan dat de complexiteit  $\mathcal{O}(n \cdot m)$  is.

### Opgave 3.6

Pas het algoritme van Floyd en Warshall toe op het de  $L$ -matrix uit Opgave 3.3.

### Opgave 3.7

Veronderstel dat het kortste pad van  $v_i$  naar  $v_j$  niet uniek is. Bijvoorbeeld, veronderstel dat er twee kortste paden zijn, zeg  $[v_i, v_2, v_4, v_j]$  en  $[v_i, v_{10}, v_1, v_j]$ . Welk pad wordt dan gekozen met het algoritme van Floyd en Warshall? Stel een regel op die deze vraag in het algemeen beantwoordt.

### Opgave 3.8

Beschrijf een methode met complexiteit  $\mathcal{O}(n^3)$  om de kortste ronde in een netwerk te bepalen (neem weer aan dat er geen ronde met negatieve lengte is).

### Opgave 3.9\*

Beschouw het volgende, van Dantzig afkomstige algoritme.

1. Voor  $i = 1, 2, \dots, n$  doe:  $u_{ii} = 0$ .
2. Voor  $k = 2, 3, \dots, n$  doe:
  - a. Voor  $l = 1, 2, \dots, k - 1$  doe:  $u_{kl} = \min_{1 \leq j \leq k-1} [l_{kj} + u_{jl}]$ ;  $u_{lk} = \min_{1 \leq j \leq k-1} [l_{jk} + u_{lj}]$ .
  - b. Voor  $i = 1, 2, \dots, k - 1$  doe:
 

Voor  $j = 1, 2, \dots, k - 1$  doe:  $u_{ij} = \min [u_{ij}, u_{ik} + u_{kj}]$ .

Beantwoord de volgende vragen over dit algoritme.

- a. Pas het algoritme toe op Voorbeeld 3.2.
- b. Toon aan dat dit algoritme de kortste paden tussen ieder tweetal knooppunten bepaalt.
- c. Toon aan dat de complexiteit  $\mathcal{O}(n^3)$  is.

# Hoofdstuk 4

## Netwerkstromen

### 4.1 Inleiding

Zij  $G = (V, A)$  een gerichte graaf en laten  $s$  en  $t$  twee knooppunten van deze graaf zijn. Een pad heet een  $(s, t)$ -pad als het begint in  $s$  en eindigt in  $t$ . We bestuderen het probleem om het maximum aantal knooppunt-disjuncte  $(s, t)$ -paden, d.w.z. paden die - op  $s$  en  $t$  na - geen knooppunten gemeen hebben, te bepalen. Op analoge wijze kunnen we het probleem bestuderen om een maximum aantal pijl-disjuncte  $(s, t)$ -paden te bepalen.

Laat  $W \subseteq V$  zdd.  $s \in W$  en  $t \notin W$ . De verzameling van de pijlen met beginpunt in  $W$  en eindpunt in  $V \setminus W$  heet een  $(s, t)$ -sneede. Het aantal pijlen in z'n sneede heet de *cardinaliteit* van de sneede.

**Stelling 4.1** (Menger 1927) *Het maximum aantal pijl-disjuncte  $(s, t)$ -paden is gelijk aan de minimale cardinaliteit van een  $(s, t)$ -sneede.*

#### Bewijs

Beschouw een sneede met minimale cardinaliteit. Ieder  $(s, t)$ -pad bevat een pijl van een  $(s, t)$ -sneede en bij pijl-disjuncte paden zijn deze pijlen verschillend: de minimale cardinaliteit van een  $(s, t)$ -sneede is minstens het maximum aantal pijl-disjuncte  $(s, t)$ -paden.

Zij  $P_1, P_2, \dots, P_k$  een maximum aantal pijl-disjuncte  $(s, t)$ -paden. We mogen wel aannemen dat deze paden enkelvoudig zijn. We moeten nu aantonen dat er een  $(s, t)$ -sneede is met hoogstens  $k$  elementen. Zij  $A'$  de pijlen van de paden  $P_1, P_2, \dots, P_k$  en laat  $G' = (V, A')$ . In  $G'$  geldt voor ieder knooppunt  $v \neq s, t$  dat  $\delta^+(v) = \delta^-(v)$ . Verder geldt:  $\delta^+(s) = k$ ,  $\delta^-(s) = 0$ ;  $\delta^+(t) = 0$ ,  $\delta^-(t) = k$ . Zij  $G^* = (V, A^*)$  met  $A^* = (A \setminus A') \cup (A')^{-1}$ , d.w.z. dat  $G^*$  uit  $G$  ontstaat door de pijlen van  $A'$  om te draaien.

Bewering:  $G^*$  heeft geen  $(s, t)$ -pad.

Bewijs:

Stel er is wel een  $(s, t)$ -pad, zeg  $P$ . De eerste en de laatste pijl van  $P$  behoren niet tot  $(A')^{-1}$ .

Zij  $P = A_1 \cup A_2$  met  $A_1 \subseteq [A \setminus (A')^{-1}]$  en  $A_2 \subseteq (A')^{-1}$ .

Beschouw  $G'' = (V, A'')$  met  $A'' = (A' \cup A_1) \setminus A_2^{-1}$ . Dit is de graaf  $G'$  met het pad  $P$ , maar zonder de pijlen van  $A_2^{-1}$  (deze laatste pijlen zowel uit  $G'$  als uit  $P$  halen). Voor de graden in deze graaf

geldt:  $\delta^+(v) = \delta^-(v)$ ,  $v \neq s, t$ ;  $\delta^+(s) = k + 1$ ,  $\delta^-(s) = 0$ ;  $\delta^+(t) = 0$ ,  $\delta^-(t) = k + 1$ .

In  $G'' \subseteq G$  zijn dus  $k + 1$  pijl-disjuncte  $(s, t)$ -paden: tegenspraak.

Laat  $W = \{v \mid v \text{ in } G^* \text{ bereikbaar vanuit } s\} : s \in W, t \notin W$ .  $W$  genereert dus een  $(s, t)$ -snede. Beschouw de pijlen van deze snede. Deze behoren niet tot  $(A')^{-1}$ . Dus geen enkele pijl van  $A'$  komt  $W$  binnen, d.w.z. iedere  $P_i$  verlaat  $W$  precies één keer:  $\delta_{G'}^+(W) = k$ . Anderzijds behoren de pijlen van deze snede ook niet tot  $A \setminus A'$ , zodat ook  $\delta_G^+(W) = k$ . De cardinaliteit van deze snede is dus  $k$ . ■

Een verz. knooppunten  $W$  zdd. ieder  $(s, t)$ -pad minstens één knooppunt van  $W$  bevat heet een  $(s, t)$ -scheidende knooppuntenverz. .

**Stelling 4.2** *Veronderstel dat  $(s, t) \notin A$ . Dan geldt: het maximum aantal knooppunt-disjuncte  $(s, t)$ -paden is gelijk aan de minimale cardinaliteit van een  $(s, t)$ -scheidende knooppuntenverz. in  $V \setminus \{s, t\}$ .*

### Bewijs

Construeer de volgende hulpgraaf  $G^*$ : vervang iedere  $v \neq s, t$  door twee knooppunten  $v^1$  en  $v^2$  en trek een pijl van  $v^1$  naar  $v^2$ ; iedere  $(v, w) \in A$  wordt vervangen door de pijl  $(v^2, w^1)$ . Merk op dat er een één-éénduidig verband is tussen de  $(s, t)$ -paden van  $G$  en  $G^*$  en dat knooppunt-disjuncte  $(s, t)$ -paden van  $G$  overeenkomen met pijl-disjuncte  $(s, t)$ -paden van  $G^*$ .

Beschouw in  $G^*$  een  $(s, t)$ -snede met minimale capaciteit. We mogen wel aannemen dat deze snede alleen pijlen van het type  $(v^1, v^2)$  bevat, want de  $\delta^+(v)$  pijlen  $(v^2, w_j^1)$  kunnen beter worden vervangen door de éne pijl  $(v^1, v^2)$  (knooppunten  $v \neq t$  zonder uitgaande pijlen doen niet ter zake). De pijlen van deze snede corresponderen aldus met een  $W \subseteq V \setminus \{s, t\}$ .  $W$  is  $(s, t)$ -scheidend omdat ieder  $(s, t)$ -pad in  $G^*$  een pijl van deze  $(s, t)$ -snede bevat.

We kunnen nu schrijven:

$$\begin{aligned} & \text{maximum aantal knooppunt-disjuncte } (s, t)\text{-paden in } G = \\ & \text{maximum aantal pijl-disjuncte } (s, t)\text{-paden in } G^* = \\ & \text{minimale cardinaliteit } (s, t)\text{-snede in } G^* = \\ & \text{cardinaliteit van een } (s, t)\text{-scheidende knooppuntenverz. in } G \geq \\ & \text{minimale cardinaliteit van een } (s, t)\text{-scheidende knooppuntenverz. in } G. \end{aligned}$$

Anderzijds geldt dat ieder  $(s, t)$ -pad uit een verz. knooppunt-disjuncte  $(s, t)$ -paden een element van een  $(s, t)$ -scheidende knooppuntenverz. moet bevatten en deze knooppunten zijn verschillend: minimale cardinaliteit van een  $(s, t)$ -scheidende knooppuntenverz. in  $G \geq$  maximum aantal knooppunt-disjuncte  $(s, t)$ -paden in  $G$ . ■

Zij  $G = (V, A)$  een gerichte graaf met *capaciteiten*  $b_{ij} \in \mathbb{Z}_+$  voor alle  $(i, j) \in A$ . Laten  $s$  en  $t$  twee knooppunten van  $V$  zijn. Een *stroom* van  $s$  naar  $t$  is een geheeltallige functie op  $A$  zdd.  $\sum_j x_{ij} - \sum_j x_{ji} = 0$  voor alle  $i \neq s, t$ . De *waarde*  $v(x)$  van stroom  $x$  is gedefinieerd door:  $v(x) = \sum_j x_{sj} - \sum_j x_{js}$ . Merk op dat ook geldt:  $v(x) = \sum_j x_{jt} - \sum_j x_{tj}$ .

Een stroom  $x$  heet *toelaatbaar* als  $0 \leq x_{ij} \leq b_{ij}$  voor alle  $(i, j) \in A$ .

Zij  $W \subseteq V$  zdd.  $s \in W$  en  $t \notin W$ , dan is de *capaciteit*  $c(W)$  van de bij  $W$  behorende  $(s, t)$ -snede gedefinieerd door:  $c(W) = \sum_{i \in W, j \notin W} b_{ij}$ .

We zullen nu nader ingaan op de volgende twee (duale) problemen:

- bepaal een maximale toelaatbare stroom, d.w.z. een stroom met maximale waarde;
- bepaal een  $(s, t)$ -snede met minimale capaciteit.

**Stelling 4.3** *Voor iedere toelaatbare stroom  $x$  en iedere  $(s, t)$ -snede  $W$  geldt:  $v(x) \leq c(W)$ .*

### Bewijs

Zij  $x$  een toelaatbare stroom en  $W$  een  $(s, t)$ -snede. Dan kunnen we schrijven:

$$\begin{aligned} v(x) &= \sum_j x_{sj} - \sum_j x_{js} = \sum_{i \in W} [\sum_j x_{ij} - \sum_j x_{ji}] \\ &= \sum_{i \in W} \sum_{j \in W} x_{ij} + \sum_{i \in W} \sum_{j \notin W} x_{ij} - \sum_{i \in W} \sum_{j \in W} x_{ji} - \sum_{i \in W} \sum_{j \notin W} x_{ji} \\ &= \sum_{i \in W} \sum_{j \notin W} x_{ij} - \sum_{i \in W} \sum_{j \notin W} x_{ji} \leq \sum_{i \in W} \sum_{j \notin W} b_{ij} = c(W). \end{aligned}$$

■

### Opmerking

Uit het bewijs van bovenstaande stelling volgt ook:

$v(x) = c(W)$  d.e.s.d. als  $x_{ij} = b_{ij}$  voor alle  $(i, j)$  met  $i \in W, j \notin W$  en tevens  $x_{ij} = 0$  voor alle  $(i, j)$  met  $i \notin W, j \in W$ . Als  $v(x) = c(W)$ , dan is  $x$  een maximale stroom en  $W$  een  $(1, n)$ -snede met minimale capaciteit.

**Stelling 4.4** *Er bestaat een toelaatbare stroom  $x$  en een  $(s, t)$ -snede  $W$  zdd.  $v(x) = c(W)$ .*

### Bewijs

Vervang iedere pijl  $(i, j)$  door  $b_{ij}$  pijlen van  $i$  naar  $j$  en pas de Stelling van Menger toe. Het resultaat volgt dan direct door te beseffen dat pijl-disjuncte  $(s, t)$ -paden overeenkomen met stromen met waarde 1.

■

### Opmerking

Stelling 4.4 staat in de literatuur bekend als *max flow - min cut theorem*.

## 4.2 Methode van Ford en Fulkerson

We zullen in deze paragraaf een methode beschrijven om een maximale stroom tussen twee knooppunten te bepalen, zeg van knooppunt 1 naar knooppunt  $n$ . Deze methode vindt ook een  $(1, n)$ -snede met minimale capaciteit.

Ga uit van een toelaatbare stroom  $x$  (bijv.  $x = 0$ ). Bij  $x$  behoort een gerichte graaf  $N_x = (V, A_x)$  met  $A_x = \{(i, j) \mid (i, j) \in A \text{ en } x_{ij} < b_{ij}\} \cup \{(j, i) \mid (i, j) \in A \text{ en } x_{ij} > 0\}$ . De pijlen uit de eerste verz. (de 'goede' richting) heten *voorwaarts* en de pijlen uit de tweede verz. (de 'verkeerde' richting) heten *achterwaarts*.

Zij  $P$  in  $N_x$  een pad van 1 naar  $n$ ; zo'n pad  $P$  heet een *groeiketen*. Indien een groeiketen bestaat, dan kan de waarde van de stroom worden vergroot door over de pijlen van  $P$  een hoeveelheid  $\Delta$  te vervoeren, waarbij  $\Delta = \min(\Delta_1, \Delta_2)$  met  $\Delta_1 = \min\{b_{ij} - x_{ij} \mid (i, j) \text{ voorwaartse pijl van } P\}$  en  $\Delta_2 = \min\{x_{ij} \mid (j, i) \text{ achterwaartse pijl van } P\}$ .

Voor de nieuwe stroom, noem deze weer  $x$ , geldt dus: 
$$x_{ij} := \begin{cases} x_{ij} + \Delta & (i, j) \text{ voorwaartse pijl van } P \\ x_{ij} - \Delta & (j, i) \text{ achterwaartse pijl van } P \\ x_{ij} & \text{als noch } (i, j) \text{ noch } (j, i) \text{ in } P \end{cases}$$

#### Voorbeeld 4.1

Beschouw nevenstaande graaf. De getallen bij de pijlen zijn de capaciteiten. Start met  $x = 0$ .

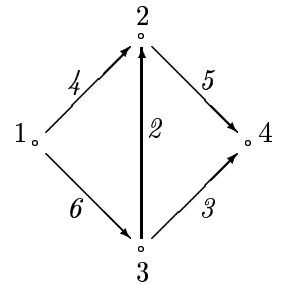
Neem  $P_1 = [1, 3, 2, 4]$ ;  $\Delta_1 = 2$ ;  $\Delta_2 = \infty$ ;  $\Delta = 2$ .  $x_{12} = 0$ ;  $x_{13} = 2$ ;  $x_{32} = 2$ ;  $x_{24} = 2$ ;  $x_{34} = 0$ .

Neem  $P_2 = [1, 2, 4]$ ;  $\Delta_1 = 3$ ;  $\Delta_2 = \infty$ ;  $\Delta = 3$ .  $x_{12} = 3$ ;  $x_{13} = 2$ ;  $x_{32} = 2$ ;  $x_{24} = 5$ ;  $x_{34} = 0$ .

Neem  $P_3 = [1, 2, 3, 4]$ ;  $\Delta_1 = 1$ ;  $\Delta_2 = 2$ ;  $\Delta = 1$ .  $x_{12} = 4$ ;  $x_{13} = 2$ ;  $x_{32} = 1$ ;  $x_{24} = 5$ ;  $x_{34} = 1$ .

Neem  $P_4 = [1, 3, 4]$ ;  $\Delta_1 = 2$ ;  $\Delta_2 = \infty$ ;  $\Delta = 2$ .  $x_{12} = 4$ ;  $x_{13} = 4$ ;  $x_{32} = 1$ ;  $x_{24} = 5$ ;  $x_{34} = 3$ .

Nu is er geen groepad meer. De waarde van de verkregen stroom is  $2 + 3 + 1 + 2 = 8$ . Neem  $W = \{1, 2, 3\}$ , dan is  $c(W) = 8$ .



**Stelling 4.5** Een toelaatbare stroom  $x$  is maximaal d.e.s.d. als er in  $N_x$  geen groeiketen is.

#### Bewijs

$\Rightarrow$  Stel er is wel een groeiketen. Hierboven hebben we gezien dat er dan een stroom is met een hogere waarde: tegenspraak.

$\Leftarrow$  Laat  $W = \{v \mid v \text{ is vanuit } v_1 \text{ in } N_x \text{ bereikbaar via een groeiketen}\}$ , dan is  $W$  een  $(1, n)$ -snede (anders is er een groeiketen van 1 naar  $n$ ). Verder geldt:

Als  $i \in W, j \notin W$ , dan is  $x_{ij} = b_{ij}$ ; als  $i \notin W, j \in W$ , dan is  $x_{ij} = 0$ . Volgens de opmerking na Stelling 4.3 is  $x$  een maximale stroom en  $W$  een  $(1, n)$ -snede met minimale capaciteit. ■

*Algoritme 4.1: Bepaling maximale stroom en snede met minimale capaciteit*

1. Start met een toelaatbare stroom (bijv.  $x = 0$ ).

2. a. Kies in  $N_x$  een groeiketen  $P$  van 1 naar  $n$  (als zo'n keten niet bestaat: ga naar stap 3).

b. Bepaal  $\Delta = \min(\Delta_1, \Delta_2)$  met  $\Delta_1 = \min\{b_{ij} - x_{ij} \mid (i, j) \text{ voorwaartse pijl van } P\}$  en  $\Delta_2 = \min\{x_{ij} \mid (j, i) \text{ achterwaartse pijl van } P\}$ .

c. 
$$x_{ij} := \begin{cases} x_{ij} + \Delta & (i, j) \text{ voorwaartse pijl van } P \\ x_{ij} - \Delta & (j, i) \text{ achterwaartse pijl van } P \\ x_{ij} & \text{als noch } (i, j) \text{ noch } (j, i) \text{ in } P \end{cases}$$

d. Ga naar stap 2a.

3.  $x$  is een maximale stroom en  $W = \{v \mid v \text{ is vanuit } v_1 \text{ in } N_x \text{ bereikbaar via een groeiketen}\}$  geeft een  $(1, n)$ -snede met minimale capaciteit.

**Stelling 4.6** *De methode van Ford en Fulkerson heeft complexiteit  $\mathcal{O}(m \cdot w)$ , met  $w$  de waarde van de maximale stroom.*

### Bewijs

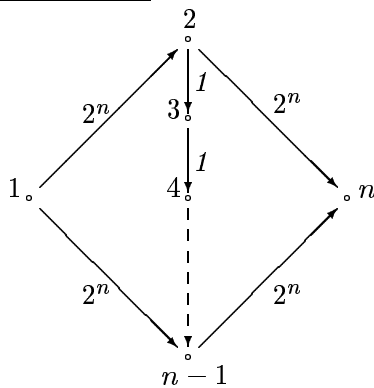
In iedere iteratie doen we het volgende:

- constructie  $N_x$ : complexiteit  $\mathcal{O}(m)$ ;
- bepaling groeiketen  $P$  (bijv. met voorwaarts zoeken): complexiteit  $\mathcal{O}(m)$ ;
- bepaling  $\Delta$ : complexiteit  $\mathcal{O}(n)$ ;
- aanpassen  $x$ : complexiteit  $\mathcal{O}(n)$ .

Per iteratie is de complexiteit dus  $\mathcal{O}(m)$ . Omdat er  $\mathcal{O}(w)$  iteraties zijn, is de totale complexiteit  $\mathcal{O}(m \cdot w)$ . ■

Bovenstaand resultaat betekent niet dat de methode polynomiaal is. De input-parameters zijn namelijk  $n$  en  $m$ ;  $w$  kan exponentieel zijn in deze parameters, zoals het volgende voorbeeld laat zien.

### Voorbeeld 4.2



Beschouw nevenstaand netwerk. De getallen bij de pijlen zijn de capaciteiten.

Als we starten met  $x = 0$ , dan kunnen afwisselend de ketens  $[1, 2, 3, \dots, n-1, n]$  en  $[1, n-1, n-2, \dots, 2, n]$  als groeiketen worden gekozen. In iedere iteratie is  $\Delta = 1$ , zodat er  $2^{n+1}$  iteraties zijn:  $w = 2^{n+1}$ .

Edmonds en Karp hebben in 1972 aangetoond<sup>1</sup> dat als  $P$  met de techniek van het zijwaarts zoeken wordt bepaald een polynomiaal algoritme ontstaat met complexiteit  $\mathcal{O}(m^2n)$ .

Bij het zijwaarts zoeken geldt voor iedere  $v_i$  die bezocht wordt dat dit gebeurt met een minimum aantal pijlen om dit knooppunt vanuit  $v_1$  te bereiken.

Laat  $\sigma_i^k$  het minimum aantal pijlen zijn om in de  $k$ -de iteratie via een groeiketen van  $v_1$  naar  $v_i$  te gaan en  $\tau_i^k$  het minimum aantal pijlen om in de  $k$ -de iteratie via een groeiketen van  $v_i$  naar  $v_n$  te gaan.

**Lemma 4.1** *Als in de  $k$ -de iteratie bij stroom  $x$  in  $N_x$  een groeiketen met zo min mogelijk pijlen wordt gekozen, dan geldt voor iedere  $1 \leq i \leq n$  dat  $\sigma_i^{k+1} \geq \sigma_i^k$  en  $\tau_i^{k+1} \geq \tau_i^k$ .*

<sup>1</sup>Edmonds, J. and R.M. Karp: *Theoretical improvements in algorithmic efficiency for network flow problems*, Journal of the ACM 19 (1972) 248-264.

## Bewijs

Veronderstel dat  $\sigma_i^{k+1} < \sigma_i^k$  voor zekere  $i$ . Omdat  $\sigma_1^{k+1} = \sigma_1^k = 0$ , is  $2 \leq i \leq n$ . Laat  $v_j$  een knooppunt zijn dat in  $N_x$  zo dicht mogelijk (d.w.z. met zo min mogelijk pijlen bereikbaar) bij  $v_1$  ligt en waarvoor geldt dat  $\sigma_j^{k+1} < \sigma_j^k$ . Er is dus een  $q$  zdd.  $\sigma_j^{k+1} = \sigma_q^{k+1} + 1 \geq \sigma_q^k + 1$ . Er geldt:  $\sigma_j^k \geq \sigma_j^{k+1} + 1 \geq \sigma_q^k + 2$ , d.w.z.  $(q, j)$  is geen pijl in het netwerk van iteratie  $k$ , maar dan zit de pijl  $(j, q)$  in de groeiketen van iteratie  $k$ :  $\sigma_q^k = \sigma_j^k + 1$ : tegenspraak.

De ongelijkheid  $\tau_i^{k+1} \geq \tau_i^k$  is analoog te bewijzen. ■

**Stelling 4.7** (i) Als in iedere iteratie een groeiketen met zo min mogelijk pijlen wordt gekozen, dan zijn er hoogstens  $\frac{1}{2}n \cdot m$  iteraties; (ii) de implementatie van Edmonds en Karp heeft de complexiteit  $\mathcal{O}(m^2n)$ .

## Bewijs

(i) Iedere groeiketen heeft een zogenaamde *blokkerende* pijl, dat is de pijl die  $\Delta$  bepaalt. Veronderstel dat  $(i, j)$  een blokkerende pijl is in iteratie  $k$ . De pijl  $(i, j)$  zit dan niet in  $A_x$  in iteratie  $k+1$ , maar pas nadat de pijl  $(j, i)$  is een groeiketen heeft gezeten. Stel  $(j, i)$  zit in iteratie  $p \geq k+1$  in een groeiketen. We kunnen dan schrijven:  $\sigma_i^p + \tau_i^p = [\sigma_j^p + 1] + \tau_i^p \geq [\sigma_j^k + 1] + \tau_i^k = [\sigma_i^k + 2] + \tau_i^k$ . De lengte van een groeiketen waar  $(j, i)$  voor het eerst na iteratie  $k$  in zit is dus minstens 2 meer dan de lengte van de groeiketen in iteratie  $k$  waar  $(i, j)$  blokkerend is. Het gevolg hiervan is dat iedere pijl maximaal  $\frac{1}{2}n$  keer blokkerend kan zijn. Omdat er  $m$  pijlen zijn, is het aantal iteraties hoogstens  $\frac{1}{2}n \cdot m$ .

(ii) Per iteratie is de complexiteit  $\mathcal{O}(m)$ , zodat de totale complexiteit van deze implementatie  $\mathcal{O}(m^2n)$  is. ■

### Voorbeeld 4.3

Beschouw nevenstaande netwerk. De getallen bij de pijlen zijn weer de capaciteiten. Start met  $x = 0$ .

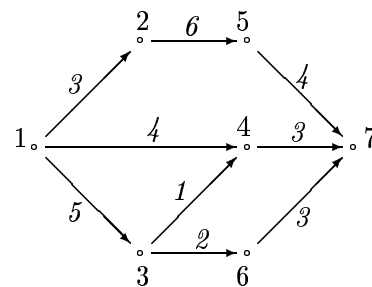
$P_1 = [1, 4, 7]$ ;  $\Delta = 3$ ;  $x_{14} = x_{47} = 3$ .

$P_2 = [1, 2, 5, 7]$ ;  $\Delta = 3$ ;  $x_{12} = x_{25} = x_{57} = 3$ .

$P_3 = [1, 3, 6, 7]$ ;  $\Delta = 2$ ;  $x_{13} = x_{36} = x_{67} = 2$ .

Er is geen groeipad meer. De waarde van de verkregen stroom is  $3 + 3 + 2 = 8$ . In  $N_x$  zijn de knooppunten 1, 4 en 3 bereikbaar:

$W = \{1, 3, 4\}$  geeft een snede met minimale capaciteit, nl. 8.



## 4.3 Methode van Dinic, Malhotra, Kumar en Maheshwari

Bij de methode van Ford en Fulkerson wordt de stroom in iedere iteratie verbeterd door een stroom over een groeiketen toe te voegen. De methode van Dinic<sup>2</sup>, Malhotra, Kumar en Maheshwari<sup>3</sup>

<sup>2</sup>Dinic, E.A.: *Algorithm for solution of a problem of maximum flow in a network with power estimation*, Soviet Mathematics Doklady 11 (1970) 1277 - 1280.

<sup>3</sup>Malhotra, V.M., M.P. Kumar and S.N. Maheshwari: *An  $\mathcal{O}(|V|^3)$  algorithm for finding maximum flows in networks*, Information Processing Letters 7 (1978) 277 - 278.

(DMKM) werkt anders. De extra stroom wordt niet noodzakelijkerwijs over een keten gestuurd, maar zo mogelijk over een aantal ketens tegelijk. Het zal blijken dat hiermee een  $\mathcal{O}(n^3)$  algoritme verkregen kan worden.

We gebruiken hiervoor een van het netwerk  $N$  en stroom  $x$  afgeleid *gelaagd netwerk*  $N_x$ . Een gelaagd netwerk is een netwerk waarin de knooppunten zijn verdeeld in  $V_1 \cup V_2 \cup \dots \cup V_p$  zdd.  $V_1 = \{v_1\}$ ,  $V_p = \{v_n\}$  en met alleen pijlen van  $V_i$  naar  $V_{i+1}$ ,  $i = 1, 2, \dots, p - 1$ .

De constructie van  $N_x$  als gelaagd netwerk kan worden bepaald met de techniek van het zijwaarts zoeken in de graaf met voorwaartse en achterwaartse pijlen. Voor ieder knooppunt  $v_k$  dat wordt bezocht houden we het volgende bij:

- in welke  $V_i$  het zit door  $LABEL[k] = i$  als  $v_k \in V_i$ ;
- welke pijlen er binnenkomen door  $PRED[k] = \{j \mid (j, k) \in N_x\}$ ;
- welke pijlen er uitgaan door  $SUCC[k] = \{j \mid (k, j) \in N_x\}$ .

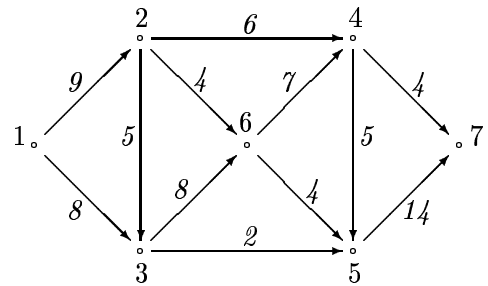
Knooppunten  $v_k \neq v_n$  die geen opvolgers hebben gekregen, d.w.z.  $SUCC[k] = \emptyset$ , worden weer verwijderd, evenals de in zo'n knooppunt binnenkomende pijlen. Verder houden we voor iedere voorwaartse en achterwaartse pijl  $(i, j)$  met  $CAP[i, j]$  bij wat er nog over vervoerd kan worden, de restcapaciteit. We stoppen zodra de laag met knooppunt  $v_n$  is bereikt.

Voorbeeld 4.4

Beschouw nevenstaand netwerk, waarin naast de pijlen de capaciteiten zijn aangegeven.

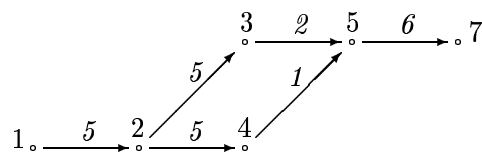
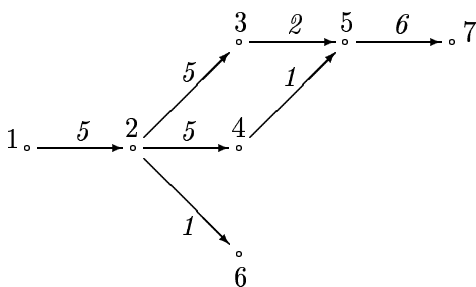
Veronderstel dat we reeds de volgende stroom met waarde 12 hebben:

$$x_{12} = 4, x_{13} = 8, x_{23} = 0, x_{24} = 1, x_{26} = 3, x_{35} = 0, \\ x_{36} = 8, x_{64} = 7, x_{65} = 4, x_{45} = 4, x_{47} = 4, x_{57} = 8.$$



Het gelaagde netwerk staat hieronder aan de linkerkant met bij de pijlen de restcapaciteit. We zien dat knooppunt 6 geen uitgaande pijlen heeft. We verwijderen dus knooppunt 6 met de pijl  $(2, 6)$ , waarna we het netwerk dat hieronder rechts staat overhouden. Hiervoor geldt:

$$V_1 = \{1\}; V_2 = \{2\}; V_3 = \{3, 4, 6\}; V_4 = \{5\}; V_6 = \{7\}.$$

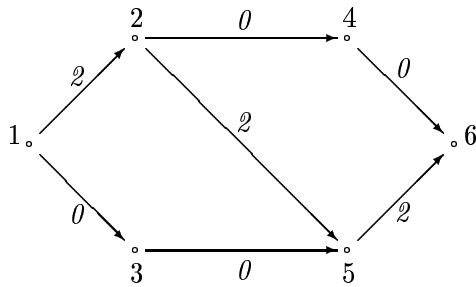


In het gelaagde netwerk  $N_x$  gaan we een zogenaamde *verzadigde stroom* bepalen. Een verzadigde stroom is een stroom zdd. op ieder pad in  $N_x$  van  $v_1$  naar  $v_n$  niets extra kan worden vervoerd. Een dergelijke stroom hoeft niet maximaal te zijn zoals onderstaand voorbeeld toont.



### Voorbeeld 4.5

In dit voorbeeld zijn alle capaciteiten 2 en staan in de figuur de  $x$ -waarden van een verzadigde stroom  $x$ . De waarde van de maximale stroom is 4 en is 2 voor de verzadigde stroom  $x$ .



Nadat het gelaagde netwerk  $N_x$  met het zijwaarts zoeken is bepaald, wordt als volgt een verzadigde stroom bepaald:

*Algoritme 4.2: Algoritme om een verzadigde stroom in een gelaagd netwerk  $N_x$  te bepalen*

1. Bepaal voor iedere  $v_j \neq v_1, v_n$ :  $d_j^+ = \sum_k CAP[j, k]$ ,  $d_j^- = \sum_k CAP[k, j]$  en  $d_j = \min(d_j^+, d_j^-)$ .
2. Laat  $d_r = \min_j d_j$  en stel  $v_r \in V_i$ .
3. a. Vanuit  $v_r$  wordt de hoeveelheid  $d_r$  'doorgeduwd' naar  $V_{i+1}, V_{i+2}, \dots, V_p = \{v_n\}$ .  
b. Vanuit  $v_r$  wordt de hoeveelheid  $d_r$  'teruggeduwd' naar  $V_{i-1}, V_{i-2}, \dots, V_1 = \{v_1\}$ .  
Tijdens het doorduwen en terugduwen worden de restcapaciteiten en de getallen  $d_j^+, d_j^-$  en  $d_j$  steeds aangepast. Zodra een knooppunt  $v_j$  een  $d_j$ -waarde 0 krijgt, wordt het knooppunt verwijderd en worden de capaciteiten van aangrenzende knooppunten zonodig aangepast. Zodra  $v_1$  of  $v_n$  wordt verwijderd stopt het algoritme.
4. Zolang  $v_1$  of  $v_n$  niet verwijderd zijn: ga naar stap 2.

Het DMKM-algoritme ziet er in totaliteit als volgt uit.

*Algoritme 4.3: Maximale stroom van  $v_1$  naar  $v_n$  volgens Dinic, Malhotra, Kumar en Maheshwari*

1. Start met  $x = 0$ .
2. a. Bepaal het gelaagde netwerk  $N_x$ .  
b. Als  $v_n$  niet bereikt kan worden:  $x$  is een maximale stroom en  $W$  geeft een snede met minimale capaciteit, waarbij  $W$  bestaat uit de knooppunten die vanuit  $v_1$  wel bereikt zijn.
3. a. Bepaal een verzadigde stroom in  $N_x$  met algoritme 4.2.  
b. Pas de stroom  $x$  aan en ga naar stap 2.

### Voorbeeld 4.4 (vervolg)

We gaan uit van het gelaagde netwerk  $N_x$  dat eerder hebben bepaald en dat hiernaast staat.

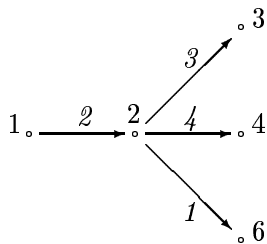
$d_1 = 5, d_2 = 5, d_3 = 2, d_4 = 1, d_5 = 3$  en  $d_7 = 6$ .  $r = 4$  en  $d_r = 1$ . Deze hoeveelheid 1 wordt vanuit  $v_4$  via  $v_5$  naar  $v_7$  doorgeduwd en via  $v_2$  naar  $v_1$  teruggeduwd. Knooppunt  $v_4$  wordt vervolgens uit het netwerk verwijderd en de capaciteiten worden aangepast (zie hierna).



In dit laatste gelaagde netwerk  $N_x$  hebben we:  $d_1 = 4, d_2 = 4, d_3 = 2, d_5 = 2$  en  $d_7 = 5$ .  $r = 3$  en  $d_r = 2$ . Deze hoeveelheid 2 wordt vanuit  $v_3$  via  $v_5$  naar  $v_7$  doorgeduwd en via  $v_2$  naar  $v_1$  teruggeduwd. Knooppunt  $v_3$  wordt vervolgens uit het netwerk verwijderd, waarna ook de andere knooppunten één voor één verdwijnen.

Deze verzadigde stroom wordt bij de oorspronkelijke stroom opgeteld. Dit geeft een nieuwe stroom  $x$  met waarde 15 en met  $x_{12} = 7, x_{13} = 8, x_{23} = 2, x_{24} = 2, x_{26} = 3, x_{35} = 2, x_{36} = 8, x_{64} = 7, x_{65} = 4, x_{45} = 5, x_{47} = 4$  en  $x_{57} = 11$ .

We gaan nu naar de volgende iteratie. Hieronder staat het nieuwe gelaagde netwerk  $N_x$ .



Omdat knooppunt  $v_7$  kan niet worden bereikt, is de stroom  $x$  maximaal; de waarde is 15. De verz.  $W = \{1, 2, 3, 4, 6\}$  geeft een snede met minimale capaciteit; deze capaciteit is ook 15.

**Stelling 4.8** *In opeenvolgende iteraties is het minimum aantal pijlen om in  $N_x$  van  $v_1$  naar  $v_n$  te komen strict stijgend.*

**Bewijs**

Volgens Lemma 4.1 is het minimum aantal pijlen om van  $v_1$  naar  $v_n$  te komen niet-dalend. In  $N_x$  heeft ieder pad van  $v_1$  naar  $v_n$  dezelfde lengte, zeg lengte  $p$ . Laat  $y$  een verzadigde stroom zijn in  $N_x$  en zij  $x' = x + y$ .

Veronderstel dat in  $N_{x'}$  een pad  $P$  is van  $v_1$  naar  $v_n$  (d.w.z. er kan over  $P$  nog wat worden vervoerd van  $v_1$  naar  $v_n$ ) met ook  $p$  pijlen, zeg  $P = [v_1, v_2, \dots, v_p, v_{p+1} = v_n]$ . Omdat  $y$  een verzadigde stroom is, bestaat dit pad  $P$  niet in  $N_x$  (anders kon er nog wel wat extra's worden vervoerd). Laat  $(v_i, v_{i+1})$  een pijl van  $P$  zijn die niet in  $N_x$  zit. Veronderstel dat dit een voorwaartse pijl is (voor een achterwaartse pijl gaat het bewijs analoog).

Laat  $\sigma_j$  en  $\sigma'_j$  het aantal pijlen in  $N_x$  resp.  $N_{x'}$  van  $v_1$  naar  $v_j$  zijn en  $\tau_j$  en  $\tau'_j$  het aantal pijlen in  $N_x$  resp.  $N_{x'}$  van  $v_j$  naar  $v_n$ . Voor de knooppunten op pad  $P$  geldt dat  $\sigma_i < \sigma_{i+1}$ , anders geldt immers (we gebruiken Lemma 4.1):  $p = \sigma_{i+1} + \tau_{i+1} \leq \sigma_i + \tau_{i+1} \leq \sigma'_i + \tau'_{i+1} = \sigma'_i + \tau'_i - 1 = p - 1$ : tegenspraak.

We onderscheiden de volgende twee gevallen:

- a. Als  $(v_{i+1}, v_i) \in N_x$ :  $p = \sigma'_i + 1 + \tau'_{i+1} \geq \sigma_i + 1 + \tau_{i+1} = \sigma_{i+1} + 1 + 1 + \tau_{i+1} = p + 2$ : tegenspraak.
- b. Als  $(v_{i+1}, v_i) \notin N_x$ : in  $N_x$  is de pijl  $(v_i, v_{i+1})$  in beide richtingen niet aanwezig:  $x'_{i,i+1} = x_{i,i+1}$ . Omdat  $x'_{i,i+1} = x_{i,i+1}$  en  $(v_i, v_{i+1}) \in N_{x'}$  als voorwaartse pijl, geldt dat  $x_{i,i+1} = x'_{i,i+1} < b_{i,i+1}$ . Uit  $\sigma_i < \sigma_{i+1}$ , volgt dan dat  $(v_i, v_{i+1}) \in N_x$ : tegenspraak. ■

**Stelling 4.9** *De DMKM methode is correct en heeft complexiteit  $\mathcal{O}(n^3)$ .*

### Bewijs

De methode eindigt als er geen groeiketen meer is van  $v_1$  naar  $v_n$ . De bijbehorende stroom is dus maximaal en het algoritme correct.

Volgens Stelling 4.8 zijn er maximaal  $n - 1$  iteraties. We moeten dus aantonen dat de hoeveelheid werk per iteratie  $\mathcal{O}(n^2)$  is. Per iteratie is het aantal keren dat we een verzadigde stroom kunnen vinden  $\mathcal{O}(n)$ , d.w.z. het algoritme om een verzadigde stroom te vinden heeft hoogstens  $n$  deeliteraties, want na iedere deeliteratie verdwijnt minstens één knooppunt. Verder wordt per iteratie het volgende gedaan:

- Opstellen van het netwerk  $N_x$  met bijgeborende restcapaciteit en  $d_j$ 's: de complexiteit hiervan is  $\mathcal{O}(m) \leq \mathcal{O}(n^2)$ .
- Bepalen  $v_r$ : per deeliteratie  $\mathcal{O}(n)$ , dus per iteratie  $\mathcal{O}(n^2)$ .
- Het 'doorduwen' en 'terugduwen' van  $d_r$  met de daarbijbehorende aanpassingen:

Per pijl maken we onderscheid tussen de volgende handelingen:

*Volmaken:* dan wordt de restcapaciteit 0. Dit kan per iteratie maar één keer gebeuren.

Het werk hieraan is per pijl  $\mathcal{O}(1)$ ; voor alle pijlen is het werk dus  $\mathcal{O}(m) \leq \mathcal{O}(n^2)$ .

*Bijwerken:* per deeliteratie wordt vanuit ieder knooppunt hoogstens één pijl bijgewerkt (omdat de restcapaciteit niet volledig benut wordt); het werk voor alle knooppunten is per deeliteratie dus  $\mathcal{O}(n)$ ; er zijn  $\mathcal{O}(n)$  deeliteraties, dus in totaal per iteratie is het werk  $\mathcal{O}(n^2)$ .

- Het verwijderen van knooppunten met bijbehorende aanpassingen: per iteratie gebeurt dit hoogstens  $n$  keer en per knooppunt  $v$  is het werk  $\mathcal{O}(\delta(v))$ ; in totaal is de complexiteit dus  $\mathcal{O}(\sum_v \delta(v)) = \mathcal{O}(m) \leq \mathcal{O}(n^2)$ .

Per iteratie is het werk dus inderdaad  $\mathcal{O}(n^2)$ , zodat de complexiteit van het totale algoritme  $\mathcal{O}(n^3)$  is. ■

## 4.4 Stromen met onder- en bovengrenzen

In deze paragraaf veronderstellen we dat aan iedere pijl  $(i, j)$  twee getallen  $a_{ij}, b_{ij} \in \mathbb{N}_0$  zijn toegevoegd met  $a_{ij} \leq b_{ij}$ . Een stroom  $x$  heet *toelaatbaar* als  $a_{ij} \leq x_{ij} \leq b_{ij}$  voor alle pijlen  $(i, j)$ . De getallen  $a_{ij}$  en  $b_{ij}$  heten de *onder-* resp. *bovengrenzen*.

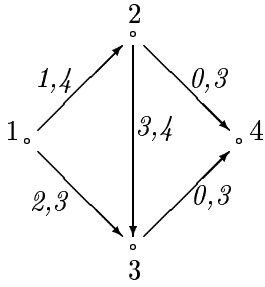
Om in dit model een maximale stroom te vinden, moeten we eerst een toelaatbare stroom bepalen ( $x = 0$  voldoet i.h.a. niet). Daartoe stellen we een *geassocieerd netwerk*  $\bar{N} = (\bar{V}, \bar{A})$  op met ondergrenzen 0 en capaciteiten  $\bar{b}_{ij}$ .

Het geassocieerd netwerk is als volgt gedefinieerd:

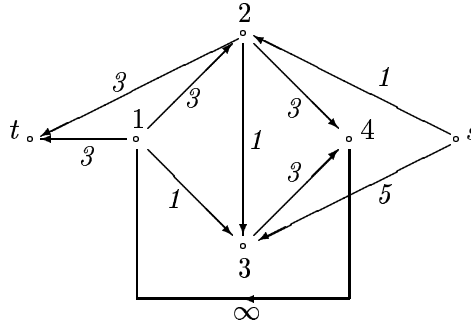
$\bar{V} = V \cup s \cup t$ ;  $\bar{A} = A \cup (n, 1) \cup A_s \cup A_t$ , waarbij  $A_s = \{(s, j) \mid j \in V \text{ en } \sum_k a_{kj} > 0\}$  en

$$A_t = \{(j, t) \mid j \in V \text{ en } \sum_k a_{jk} > 0\}; \bar{b}_{n1} = \infty; \bar{b}_{ij} = \begin{cases} b_{ij} - a_{ij} & (i, j) \in A \\ \sum_k a_{kj} & i = s, (s, j) \in A_s \\ \sum_k a_{ik} & j = t, (i, j) \in A_t \end{cases}$$

Voorbeeld 4.6



De getallen bij de pijlen  $(i, j)$  zijn  $a_{ij}, b_{ij}$ .



De getallen bij de pijlen zijn de capaciteiten  $\bar{b}_{ij}$ .

Een toelaatbare stroom  $\bar{x}$  van  $s$  naar  $t$  in  $\bar{N}$  heet een *volgemaakte stroom* als  $\bar{x}_{sj} = \bar{b}_{sj}$  voor alle  $j$  met  $(s, j) \in A_s$ , d.w.z. een stroom met waarde  $v(\bar{x}) = \sum_{i,j} a_{ij}$ . Merk op dat een volgemaakte stroom een maximale stroom is en dat ook geldt  $\bar{x}_{it} = \bar{b}_{it}$  voor alle  $i$  met  $(i, t) \in A_t$ .

**Stelling 4.10** Als  $\bar{x}$  een volgemaakte stroom is in  $\bar{N}$  dan is  $x$ , gedefinieerd door  $x_{ij} = \bar{x}_{ij} + a_{ij}$  voor alle  $(i, j) \in A$ , een toelaatbare stroom in  $N$  met  $v(x) = \bar{x}_{n1}$ .

**Bewijs**

Omdat  $0 \leq \bar{x} \leq \bar{b}$ , geldt:  $a_{ij} \leq x_{ij} \leq b_{ij}$  voor alle  $(i, j) \in A$ . Voor  $i \neq 1, n$  geldt:

$$\sum_{j \in V} x_{ij} - \sum_{j \in V} x_{ji} = \sum_{j \in V} [\bar{x}_{ij} + a_{ij}] - \sum_{j \in V} [\bar{x}_{ji} + a_{ij}] = [\sum_{j \in V} \bar{x}_{ij} + \bar{b}_{it}] - [\sum_{j \in V} \bar{x}_{ji} + \bar{b}_{si}] = [\sum_{j \in V} \bar{x}_{ij} + \bar{x}_{it}] - [\sum_{j \in V} \bar{x}_{ji} + \bar{x}_{si}] = \text{nettoproductie in knooppunt } v_i \text{ in netwerk } \bar{N} = 0.$$

Voor  $i = 1$  geldt:

$$\sum_{j \in V} x_{1j} - \sum_{j \in V} x_{j1} = \{[\sum_{j \in V} \bar{x}_{1j} + \bar{x}_{1t}] - [\sum_{j \in V} \bar{x}_{j1} + \bar{x}_{s1}] - \bar{x}_{n1}\} + \bar{x}_{n1} = \text{nettoproductie in knooppunt } v_1 \text{ in netwerk } \bar{N} + \bar{x}_{n1} = \bar{x}_{n1}.$$

Analoog geldt voor  $i = n$ :  $\sum_{j \in V} x_{nj} - \sum_{j \in V} x_{jn} = -\bar{x}_{n1}$ . ■

**Stelling 4.11** Als  $N$  een toelaatbare stroom  $x$  heeft, dan heeft  $\bar{N}$  een volgemaakte stroom  $\bar{x}$ .

**Bewijs**

Neem  $\bar{x}_{ij} = x_{ij} - a_{ij}$  voor alle  $(i, j) \in A$ ,  $\bar{x}_{n1} =$  waarde stroom  $x$ ,  $\bar{x}_{sj} = \sum_i a_{ij}$  voor alle  $(s, j) \in A_s$ ;  $\bar{x}_{it} = \sum_j a_{ij}$  voor alle  $(i, t) \in A_t$ . Analoog aan Stelling 4.10 is direct in te zien dat  $0 \leq \bar{x}_{ij} \leq \bar{b}_{ij}$  voor alle  $(i, j) \in \bar{A}$ . Verder geldt:

$$\text{Voor } 1 \leq i \leq n : \sum_{j \in \bar{V}} \bar{x}_{ij} - \sum_{j \in \bar{V}} \bar{x}_{ji} = 0; \text{ voor } i = s : \sum_{j \in \bar{V}} \bar{x}_{sj} = \sum_{(j,i)} a_{ji} \geq 0;$$

$$\text{voor } i = t : \sum_{i \in \bar{V}} \bar{x}_{it} = \sum_{(i,j)} a_{ij} \geq 0.$$

Hieruit volgt dat  $\bar{x}$  een volgemaakte stroom is. ■

Indien we in  $\bar{N}$  een volgemaakte stroom  $\bar{x}$  hebben, dan geeft dit dus een toelaatbare stroom  $x$  voor  $N$ . Met een kleine aanpassing kan dan met een van de eerder besproken methoden (Ford en

Fulkerson of DMKM) een maximale stroom worden gevonden. De aanpassing betreft het begrip achterwaartse pijl. In plaats van  $x_{ij} > 0$  moet nu voor een achterwaartse pijl gelden dat  $x_{ij} > a_{ij}$ . De restcapaciteit is dan  $x_{ij} - a_{ij}$ .

*Algoritme 4.4: Algoritme voor een maximale stroom in een netwerk met onder- en bovengrenzen*

1. Construeer het netwerk  $\bar{N}$  en bepaal daarin een maximale stroom  $\bar{x}$ .
2. Als  $\bar{x}$  geen volgemaakte stroom is:  $N$  heeft geen toelaatbare stroom en stop.  
Als  $\bar{x}$  wel een volgemaakte stroom is: ga naar stap 3.
3. Bepaal een maximale stroom in  $N$  uitgaande van de toelaatbare stroom  $x$ , gedefinieerd door  $x_{ij} = \bar{x}_{ij} + a_{ij}$  voor alle  $(i, j) \in A$ .

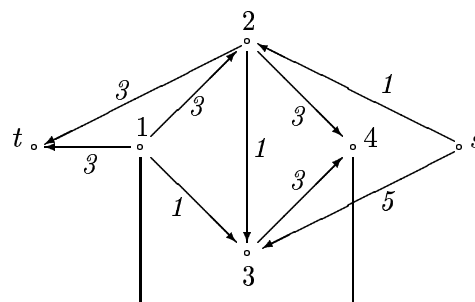
#### Voorbeeld 4.6 (vervolg)

Een maximale stroom van  $s$  naar  $t$  in het hier-naast staande netwerk  $\bar{N}$  krijgen we via de volgende groeiketens  $P$ :

$P = [s, 2, t]$  met hoeveelheid 1.

$P = [s, 3, 4, 1, t]$  met hoeveelheid 3.

De nu verkregen stroom is maximaal, maar is geen volgemaakte stroom. Het oorspronkelijke netwerk heeft dus geen toegelaten stroom.



## 4.5 Minimale kostenstromen

In dit model hebben we een netwerk met ondergrenzen 0, capaciteiten  $b_{ij}$  en kosten  $c_{ij}$  voor iedere pijl  $(i, j) \in A$ . De kosten worden per eenheid van vervoer berekend, dus een stroom  $x$  heeft kosten  $c(x) = \sum_{(i,j) \in A} c_{ij} x_{ij}$ . We veronderstellen in deze paragraaf dat de kosten over een ronde niet-negatief zijn.

We zullen een algoritme opstellen dat de volgende twee problemen oplost:

- a. Bepaal een stroom met een gegeven waarde  $w$  en minimale kosten.
- b. Bepaal een stroom met maximale waarde en minimale kosten.

De oplosmethode lijkt sterk op de methode van Ford en Fulkerson. In plaats van *een* groeiketen van  $v_1$  naar  $v_n$  moet nu de *kortste* groeiketen in  $N_x$  worden gevonden t.a.v. de volgende, van de stroom  $x$  afhankelijke, lengtefunctie:

$$l_{ij} = \begin{cases} c_{ij} & \text{als } (i, j) \text{ een voorwaartse pijl is} \\ -c_{ji} & \text{als } (i, j) \text{ een achterwaartse pijl is} \end{cases} \quad (4.1)$$

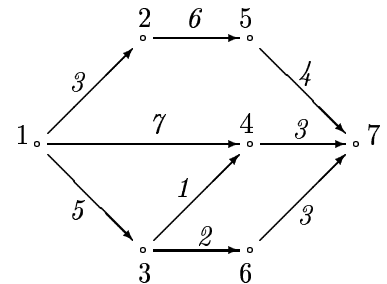
Het algoritme is als volgt ( $k$  geeft de waarde van de stroom en  $c$  de kosten; indien we een maximale stroom met minimale kosten wensen, dan moeten we  $w = \infty$  nemen).

*Algoritme 4.5: Algoritme voor een minimale kostenstroom*

1. Start met  $x = 0$ ,  $c = 0$  en  $k = 0$ .
2. a. Bepaal een kortste groeiketen  $P$  van  $v_1$  naar  $v_n$  t.a.v. de lengtefunctie (4.1), en laat  $l(P)$  de lengte van dit pad  $P$  zijn (als er geen groeiketen is van  $v_1$  naar  $v_n$  is, dan is  $x$  een maximale stroom met waarde  $k$  en minimale kosten  $c$ ; stop).
  - b. Bepaal een maximale hoeveelheid  $\Delta$  die over  $P$  kan worden vervoerd.
  - c. Als  $k + \Delta \geq w$ :  $\Delta := w - k$ .
3. a. Pas  $x$  aan door de stroom over  $P$  erbij op te tellen;  $c := c + \Delta \cdot l(P)$ ;  $k := k + \Delta$ .
  - b. Als  $k = w$ :  $x$  is een minimale kostenstroom met waarde  $w$  en stop.  
Anders: ga naar stap 2a.

Voorbeeld 4.7

Beschouw nevenstaande netwerk, waarin naast de pijlen de kosten  $c_{ij}$  zijn aangegeven. Alle bovengrenzen  $b_{ij}$  nemen we gelijk aan 2. Gevraagd wordt een minimale kostenstroom met waarde 5.



*Iteratie 1:*

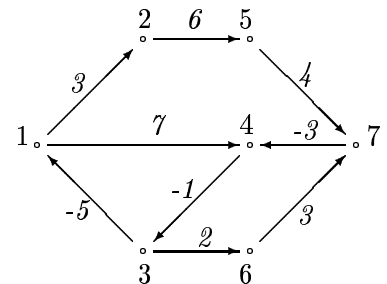
$x = 0$ ;  $c = 0$ ;  $k = 0$ . Bij de start is de lengtefunctie gelijk aan de kostenfunctie.

Het kortste pad  $P$  van 1 naar 7 is:

$P = [1, 3, 4, 7]$ ;  $l(P) = 9$ ;  $\Delta = 2$ ;

$c = 2 \cdot 9 = 18$ ;  $k = 2$ .

De nieuwe lengtefunctie staat in nevenstaand netwerk.



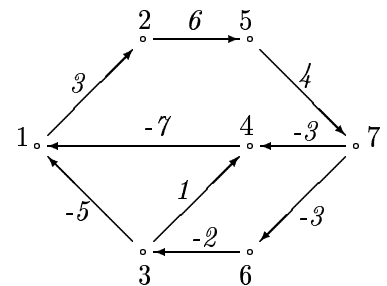
*Iteratie 2:*

Het kortste pad  $P$  van 1 naar 7 is:

$P = [1, 4, 3, 6, 7]$ ;  $l(P) = 11$ ;  $\Delta = 2$ .

$c = 18 + 2 \cdot 11 = 40$ ;  $k = 2 + 2 = 4$ .

De nieuwe lengtefunctie staat in nevenstaand netwerk.



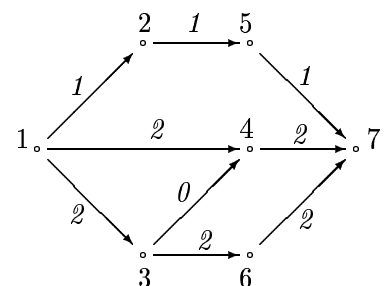
*Iteratie 3:*

Het kortste pad  $P$  van 1 naar 7 is:

$P = [1, 2, 5, 7]$ ;  $l(P) = 13$ ;  $\Delta = 2$ ;  $\Delta = 1$ .

$c = 40 + 1 \cdot 13 = 53$ ;  $k = 4 + 1 = 5$ .

Omdat  $k = w$ , is de stroom een minimale kostenstroom met waarde 5. De optimale stroom staat in nevenstaand netwerk.



Zij  $P$  een enkelvoudig pad (of ronde) in een netwerk met voorwaartse en achterwaartse pijlen. Hierbij behoort een *elementaire pad (of ronde) stroom*  $y^P$  met betrekking tot  $x$ , gedefinieerd door:

$$y_{ij}^P = \begin{cases} 1 & \text{als } (i, j) \in P \text{ en voorwaarts} \\ -1 & \text{als } (j, i) \in P \text{ en achterwaarts} \\ 0 & \text{als noch } (i, j) \text{ noch } (j, i) \text{ tot } P \text{ behoort} \end{cases}$$

Het is eenvoudig in te zien dat  $y^P$  inderdaad een stroom is, d.w.z. dat (vooreen pad behalve het begin- en eindpunt) in ieder punt  $i$  geldt:  $\sum_j y_{ij}^P = \sum_j y_{ji}^P$ .

We zullen nu ingaan op de correctheid van deze methode. Laat  $N$  een netwerk zijn met daarin een (niet noodzakelijkerwijs toegelaten) stroom  $x$ . De gerichte graaf  $N_x^* = (V^*, A^*)$  is als volgt gedefinieerd:

$$V^* = V; \quad A^* = \begin{cases} x_{ij} & \text{pijlen van } i \text{ naar } j \text{ als } x_{ij} \geq 0 \\ -x_{ij} & \text{pijlen van } j \text{ naar } i \text{ als } x_{ij} < 0 \end{cases}$$

**Stelling 4.12** *Zij  $x$  een stroom van  $v_1$  naar  $v_n$  met waarde  $k$ . Dan geldt:  $x = \sum_{i=1}^k y^i + \sum_{i=k+1}^q y^i$*

met  $y^i = \begin{cases} \text{elementaire pad-stroom van } v_1 \text{ naar } v_n, & 1 \leq i \leq k \\ \text{elementaire ronde-stroom,} & k+1 \leq i \leq q \end{cases}$

### Bewijs

In  $N_x^*$  geldt:  $\delta^+(v_i) - \delta^-(v_i) = 0$  voor alle  $i \neq 1, n$ ;  $\delta^+(v_1) - \delta^-(v_1) = k$ ;  $\delta^+(v_n) - \delta^-(v_n) = -k$ . Voeg  $k$  pijlen toe van  $v_n$  naar  $v_1$ . Dit geeft een Eulergraaf, dus alle pijlen vormen tezamen één ronde. Laat vervolgens de toegevoegde pijlen uit deze ronde weg: er ontstaan  $k$  enkelvoudige paden van  $v_1$  naar  $v_n$ , plus eventueel nog een aantal enkelvoudige ronden. Bij de enkelvoudige paden horen elementaire pad-stromen en bij de enkelvoudige ronden horen elementaire ronde-stromen. ■

**Stelling 4.13** *Zij  $x$  een toelaatbare stroom van  $v_1$  naar  $v_n$  met waarde  $k$  en  $l^x$  de lengtefunctie volgens (4.1) die bij stroom  $x$  behoort. Dan geldt:  $x$  is een minimale kostenstroom bij waarde  $k$  d.e.s.d. als  $l^x(C) \geq 0$  voor iedere enkelvoudige ronde  $C$  in  $N_x$ .*

### Bewijs

$\Rightarrow$  Stel  $C$  is een enkelvoudige ronde in  $N_x$  met  $l^x(C) < 0$ . Vervoer een hoeveelheid 1 over deze ronde en laat  $y$  de elementaire ronde-stroom bij  $C$  zijn. Nu geldt dat  $c(y) = l^x(C)$ , immers:

Als  $(i, j) \in C$  een voorwaartse pijl is, dan geldt:  $l_{ij} = c_{ij}$  en  $y_{ij} = 1$ ; dus  $l_{ij} = c_{ij}y_{ij}$ .

Als  $(i, j) \in C$  een achterwaartse pijl is, dan geldt:  $l_{ij} = -c_{ji}$  en  $y_{ji} = -1$ ; dus  $l_{ij} = c_{ji}y_{ji}$ .

Er geldt dus:

$$\begin{aligned} l^x(C) &= \sum_{(i,j) \in C} l_{ij} = \sum_{(i,j) \text{ voorwaarts in } C} l_{ij} + \sum_{(i,j) \text{ achterwaarts in } C} l_{ij} \\ &= \sum_{(i,j) \text{ voorwaarts in } C} c_{ij}y_{ij} + \sum_{(i,j) \text{ achterwaarts in } C} c_{ji}y_{ji} = c(y). \end{aligned}$$

Laat  $x' = x + y$ . Dan is  $x'$  een toelaatbare stroom met waarde  $k$  en  $c(x') = c(x) + c(y) = c(x) + l^x(C) < c(x)$ : tegenspraak.

$\Leftarrow$  Stel  $x'$  is ook een toelaatbare stroom met waarde  $k$ . Dan is  $x' - x$  een (niet noodzakelijkerwijs

toelaatbare) stroom met waarde 0. Volgens Stelling 4.12 geldt:  $x' - x = \sum_{p=1}^q y^p$  met  $y^p$  een elementaire ronde-stroom bij ronde  $C^p$  in  $N_{x'-x}^*$ ,  $1 \leq p \leq q$ . Laat:

$I_1^p = \{(i, j) \in C^p \text{ en } (x' - x)_{ij} > 0\}$ . Dan geldt:  $b_{ij} \geq x'_{ij} > x_{ij}$ , zodat  $(i, j)$  voorwaarts is in  $N_x$ . Dit betekent:  $l_{ij} = c_{ij}$  en  $y_{ij}^p = 1$ :  $l_{ij} = c_{ij}y_{ij}^p$ .

$I_2^p = \{(j, i) \in C^p \text{ en } (x' - x)_{ij} < 0\}$ . Dan geldt:  $x_{ij} \geq x'_{ij} > 0$ , zodat  $(i, j)$  achterwaarts is in  $N_x$ . Dit betekent:  $l_{ji} = -c_{ij}$  en  $y_{ij}^p = -1$ :  $l_{ji} = c_{ij}y_{ij}^p$ .

Er geldt dus  $l^x(C^p) = c(y^p)$ , zodat  $c(x') = c(x) + \sum_{p=1}^q c(y^p) = c(x) + \sum_{p=1}^q l^x(C^p) \geq c(x)$ :  $x$  is een minimale kostenstroom bij waarde  $k$ . ■

**Stelling 4.14** *Zij  $x$  een minimale kostenstroom van  $v_1$  naar  $v_n$  met waarde  $k$ ,  $P$  het kortste pad van  $v_1$  naar  $v_n$  t.o.v. de lengtefunctie  $l^x$ , gedefinieerd in (4.1), met bijbehorende elementaire pad-stroom  $y$ , en laat  $\Delta$  de maximale hoeveelheid zijn die over  $P$  nog vervoerd kan worden. Dan geldt voor  $x^* = x + y$  dat  $x^*$  een minimale kostenstroom is bij waarde  $k + \Delta$  en dat  $c(x^*) = c(x) + \Delta \cdot l^x(P)$ .*

### Bewijs

Het is duidelijk dat  $x^*$  een toelaatbare stroom is met waarde  $k + \Delta$  en met kosten  $c(x) + l^x(P)$ . Stel dat  $x'$  ook een toelaatbare stroom is met waarde  $k + \Delta$ . Dan is  $x' - x$  een stroom met waarde  $\Delta$ . Volgens Stelling 4.12 is  $x' - x = \sum_{i=1}^{\Delta} y^i + \sum_{i=\Delta+1}^q y^i$  met

$$y^i = \begin{cases} \text{elementaire pad-stroom } P_i \text{ van } v_1 \text{ naar } v_n \text{ in } N_{x'-x}^*, & 1 \leq i \leq \Delta \\ \text{elementaire ronde-stroom } C_i \text{ in } N_{x'-x}^*, & \Delta + 1 \leq i \leq q \end{cases}$$

Analoog aan Stelling 4.13 kan worden aangetoond dat  $l^x(P_i) = c(y^i)$ ,  $1 \leq i \leq \Delta$ , en  $l^x(C_i) = c(y^i)$ ,  $\Delta + 1 \leq i \leq q$ . We kunnen nu schrijven (we gebruiken daarbij tevens dat  $P$  het kortste pad is m.b.t.  $l^x$  en dat volgens Stelling 4.13  $l^x(C_i) \geq 0$ ,  $\Delta + 1 \leq i \leq q$ ):

$$c(x') = c(x) + \sum_{i=1}^{\Delta} l^x(P_i) + \sum_{i=\Delta+1}^q l^x(C_i) \geq c(x) + \sum_{i=1}^{\Delta} l^x(P) = c(x) + \Delta \cdot l^x(P) = c(x^*),$$

d.w.z.  $x^*$  is een minimale kostenstroom bij waarde  $k + \Delta$ . ■

### Opmerking

Omdat de bepaling van het kortste pad in een netwerk waarin ook negatieve lengtes voorkomen complexiteit  $\mathcal{O}(n^3)$  heeft, heeft algoritme 4.5 complexiteit  $\mathcal{O}(w \cdot n^3)$ , met  $w =$  de waarde van de stroom. Er bestaat een aangepaste versie van het algoritme die complexiteit  $\mathcal{O}(w \cdot n^2)$  heeft (zie Opgave 20).

## 4.6 Minimale circulatiestromen in netwerk met onder- en bovengrenzen

In deze paragraaf beschouwen we het probleem om een toelaatbare circulatiestroom te vinden met minimale kosten in een netwerk met onder- en bovengrenzen. Een *circulatiestroom* is een



stroom zdd. in ieder knooppunt evenveel binnenkomt als er uitgaat. Voor iedere pijl  $(i, j)$  in het netwerk hebben we kosten  $c_{ij}$ , ondergrens  $a_{ij}$  en bovengrens  $b_{ij}$ . Het probleem kan nu als volgt als LP-probleem worden geformuleerd:

$$\min \left\{ \sum_{(i,j)} c_{ij} x_{ij} \mid \begin{array}{l} \sum_j x_{ij} - \sum_j x_{ji} = 0, \quad i = 1, 2, \dots, n \\ a_{ij} \leq x_{ij} \leq b_{ij}, \quad (i, j) \in A \end{array} \right\} \quad (4.2)$$

Dit circulatieprobleem is een generalisatie van de volgende eerder besproken problemen:

1. Kortste pad probleem van  $v_1$  naar  $v_n$ : neem  $a_{n1} = b_{n1} = 1$ ;  $a_{ij} = 0, b_{ij} = 1$  voor alle  $(i, j) \neq (n, 1)$ ;  $c_{ij} = l_{ij}$  voor alle  $(i, j)$ .
2. Maximale stroom probleem van  $v_1$  naar  $v_n$ : neem  $c_{n1} = -1$ ;  $c_{ij} = 0$  voor alle  $(i, j) \neq (n, 1)$ .
3. Minimale kostenstroom probleem met waarde  $k$ : neem  $a_{n1} = b_{n1} = k$ .

Het duale probleem van (4.2) luidt:

$$\max \left\{ \sum_{(i,j)} a_{ij} v_{ij} - \sum_{(i,j)} b_{ij} w_{ij} \mid \begin{array}{l} u_i - u_j + v_{ij} - w_{ij} = c_{ij}, \quad (i, j) \in A \\ v_{ij}, w_{ij} \geq 0, \quad (i, j) \in A \end{array} \right\} \quad (4.3)$$

Laat  $z_{ij} = u_i - u_j - c_{ij}$  voor alle  $(i, j) \in A$ . Omdat  $a_{ij} \leq b_{ij}$  voor alle  $(i, j)$  is het direct in te zien dat, als  $u$  gegeven is (dan is ook  $z$  gegeven), de optimale oplossing van (4.3) is:

$$\begin{array}{l} \text{als } z_{ij} \geq 0 : v_{ij} = 0, w_{ij} = z_{ij}; \\ \text{als } z_{ij} \leq 0 : v_{ij} = -z_{ij}, w_{ij} = 0. \end{array}$$

Beide gevallen zijn samen te vatten in de oplossing  $v_{ij} = z_{ij}^-$ ,  $w_{ij} = z_{ij}^+$  voor alle  $(i, j)$ .

Zij  $x$  een stroom,  $u$  willekeurig en  $v_{ij} = z_{ij}^-$ ,  $w_{ij} = z_{ij}^+$  voor alle  $(i, j) \in A$ . Dan volgt uit de theorie van de lineaire programmering dat beide oplossingen optimaal zijn d.e.s.d. als

$$\text{voor alle } (i, j) \in A \text{ geldt: } \begin{cases} v_{ij} \cdot (x_{ij} - a_{ij}) = 0 \\ w_{ij} \cdot (b_{ij} - x_{ij}) = 0 \\ a_{ij} \leq x_{ij} \leq b_{ij} \end{cases}$$

Dit houdt in dat voor een stroom  $x$  en een gegeven  $u$  en voor  $z_{ij} = u_i - u_j - c_{ij}$  de stroom  $x$

$$\text{optimaal is d.e.s.d. als voor iedere } (i, j) \in A \text{ geldt: } \begin{cases} z_{ij} < 0 \text{ impliceert } x_{ij} = a_{ij} \\ z_{ij} > 0 \text{ impliceert } x_{ij} = b_{ij} \\ z_{ij} = 0 \text{ impliceert } a_{ij} \leq x_{ij} \leq b_{ij} \end{cases} \quad (4.4)$$

#### Voorbeeld 4.8

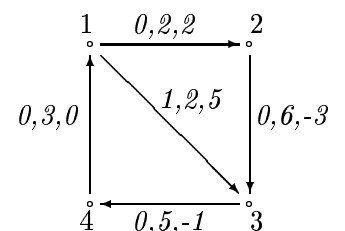
Beschouw nevenstaand circulatienetwerk.

Bij iedere pijl staat het drietal  $a_{ij}, b_{ij}, c_{ij}$ .

Neem  $x = u = 0$ .

Voor de pijl  $(1, 2)$  geldt  $z_{12} = -2$ , dus deze pijl voldoet aan (4.4).

Voor de pijl  $(3, 4)$  geldt  $z_{34} = 1$ , dus deze pijl voldoet niet aan (4.4).



Als een pijl voldoet aan (4.4) dan zeggen we dat deze *in orde* is; voldoet een pijl niet, dan is deze *niet in orde*. Een pijl  $(i, j)$  is dus in orde als geldt:

$$x_{ij} = a_{ij} \text{ indien } z_{ij} < 0; \quad a_{ij} \leq x_{ij} \leq b_{ij} \text{ indien } z_{ij} = 0; \quad x_{ij} = b_{ij} \text{ indien } z_{ij} > 0.$$

Als maat voor het niet in orde zijn, bij gegeven  $x$  en  $u$  (met de bij  $u$  behorende  $z$ , d.w.z.  $z_{ij} = u_i - u_j - c_{ij}$ ), nemen we:

$$k_{ij} = \begin{cases} |x_{ij} - a_{ij}| & \text{als } z_{ij} < 0 \text{ of als } z_{ij} = 0 \text{ en } x_{ij} < a_{ij} \\ |b_{ij} - x_{ij}| & \text{als } z_{ij} > 0 \text{ of als } z_{ij} = 0 \text{ en } x_{ij} > b_{ij} \\ 0 & \text{anders} \end{cases} \quad (4.5)$$

We zullen nu een iteratiestap gaan beschrijven, uitgaande van een stroom  $x$  en een gegeven  $u$  met bijbehorende  $z$ . Als  $k_{ij} = 0$  voor alle  $(i, j)$ , dan is de stroom  $x$  optimaal. Als dit niet het geval is, dan kiezen we een pijl  $(p, q)$  met  $k_{pq} > 0$ . We proberen nu  $x$  te veranderen over een ronde zdd. geen enkel  $k$ -getal groter wordt en  $k_{pq}$  kleiner wordt; lukt dit niet, dan gaan we  $u$  veranderen zdd. eveneens geen enkel  $k$ -getal groter wordt, maar zdd. we na een aantal van deze stappen in de eerder genoemde situatie komen. Op deze wijze ontstaat een eindig algoritme.

We construeren een hulpgraaf  $G^* = (V, A^*)$  zdd. een stroom in  $G^*$  geen enkel  $k$ -getal doet toenemen. Verder houden we met  $d$ -getallen bij hoeveel over de pijlen vervoerd kan worden. De pijlen van  $A^*$  ontstaan als volgt:

indien  $z_{ij} < 0$  : als  $x_{ij} < a_{ij}$  : neem  $(i, j) \in A^*$  en laat  $d_{ij} = k_{ij}$ ;

als  $x_{ij} > a_{ij}$  : neem  $(j, i) \in A^*$  en laat  $d_{ji} = k_{ij}$ .

indien  $z_{ij} = 0$  : als  $x_{ij} < b_{ij}$  : neem  $(i, j) \in A^*$  en laat  $d_{ij} = b_{ij} - x_{ij}$ ;

als  $x_{ij} > a_{ij}$  : neem  $(j, i) \in A^*$  en laat  $d_{ji} = x_{ij} - a_{ij}$ .

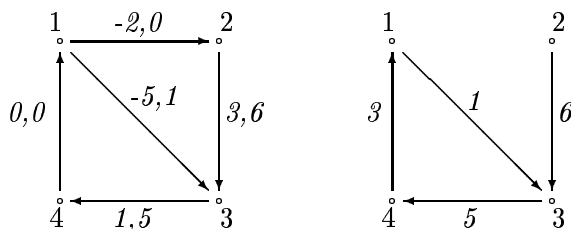
indien  $z_{ij} > 0$  : als  $x_{ij} < b_{ij}$  : neem  $(i, j) \in A^*$  en laat  $d_{ij} = k_{ij}$ ;

als  $x_{ij} > b_{ij}$  : neem  $(j, i) \in A^*$  en laat  $d_{ji} = k_{ij}$ .

Veronderstel dat  $k_{pq}$  daalt als we stroom over de pijl  $(p, q)$  sturen. We gaan nu in  $G^*$  vanuit knooppunt  $q$  zijwaarts zoeken totdat we knooppunt  $p$  hebben bereikt (als  $k_{pq}$  daalt als we stroom over de pijl in de richting  $(q, p)$  sturen, dan zoeken we vanuit  $p$  totdat we  $q$  bereiken). Indien  $p$  wordt bereikt, dan hebben we met de pijl  $(p, q)$  een ronde  $C$ . Vervolgens gaan we over  $C$  zoveel mogelijk vervoeren. Dit procédé herhalen we totdat ofwel  $k_{pq} = 0$ , ofwel  $p$  niet vanuit  $q$  bereikt kan worden.

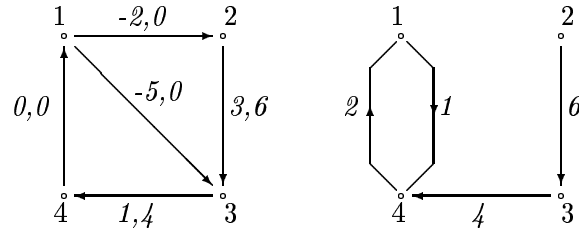
#### Voorbeeld 4.8 (vervolg)

Start met  $x = 0$  en  $u = 0$ , dus  $z = -c$ . Hiernaast staat het netwerk met bij iedere pijl  $(i, j)$  het getallenpaar  $z_{ij}, k_{ij}$ . Daarnaast staat de graaf  $G^*$  met bij de pijlen de  $d$ -getallen met de hoeveelheden die hierover kunnen worden vervoerd.



Kies voor  $(p, q)$  de pijl  $(3, 4)$ . Dit geeft ronde  $[4, 1, 3, 4]$  en hierover kan 1 worden vervoerd.

Hiernaast staat weer het netwerk met bij iedere pijl  $(i, j)$  het getallenpaar  $z_{ij}, k_{ij}$ . Daarnaast staat de graaf  $G^*$  met bij de pijlen de  $d$ -getallen. We kiezen weer  $(p, q) = (3, 4)$  en zien dat in  $G^*$  geen pad van 4 naar 3 is.



Als  $p$  niet meer vanuit  $q$  bereikt kan worden, dan gaan we als volgt  $u$  aanpassen.

Zij  $W = \{i \mid i \text{ is in } G^* \text{ bereikbaar vanuit } q\}$ . Dan geldt dus:  $q \in W$  en  $p \notin W$ . Als we voor een bepaalde  $(i, j)$  de waarde van  $u_i$  en  $u_j$  op dezelfde wijze veranderen, d.w.z.  $u_i := u_i + c$  en  $u_j := u_j + c$  voor zekere constante  $c$ , dan verandert  $z_{ij}$  niet en verandert er dus ook niets aan  $G^*$  en  $k_{ij}$  ( $x$  verandert immers ook niet). We gaan nu de  $u$ -waarde binnen  $W$  als volgt veranderen.

$$u_i := \begin{cases} u_i + c & \text{als } i \in W \\ u_i & \text{als } i \notin W. \end{cases} \quad \text{Dan geldt: } z_{ij} := \begin{cases} z_{ij} + c & \text{als } i \in W, j \notin W \\ z_{ij} - c & \text{als } i \notin W, j \in W \\ z_{ij} & \text{anders.} \end{cases} \quad (4.6)$$

We gaan nu  $c$  zó bepalen dat de  $k$ -getallen niet groter worden en  $c$  zo groot mogelijk wordt (daarmee bereiken we dat  $G^*$  meer pijlen krijgt, waardoor wellicht  $p$  wel vanuit  $q$  bereikt kan worden). Uit (4.5) en het feit dat de  $k$ -getallen niet mogen toenemen volgt:

Als  $i \in W, j \notin W$ : in  $G^*$  is geen pijl  $(i, j)$  en  $z_{ij}$  neemt met  $c$  toe.

Als  $z_{ij} < 0$  (dan is  $x_{ij} \geq a_{ij}$ ) geldt dat  $c$  willekeurig kan toenemen als  $x_{ij} \geq b_{ij}$ ; als  $x_{ij} < b_{ij}$ , dan kan  $c$  maximaal  $-z_{ij}$  zijn.

Als  $z_{ij} \geq 0$  (dan is  $x_{ij} \geq b_{ij}$ ), dan mag  $c$  willekeurig groot worden.

Dus  $c$  kan voor een pijl  $(i, j)$  met  $i \in W, j \notin W, x_{ij} < b_{ij}$  maximaal  $-z_{ij}$  zijn.

Laat  $I_1 = \{(i, j) \mid i \in W, j \notin W, z_{ij} < 0; x_{ij} \leq b_{ij}\}$  en  $c_1 = \min\{-z_{ij} \mid (i, j) \in I_1\}$

(we laten  $x_{ij} = b_{ij}$  toe i.v.m. het convergentiebewijs (zie Stelling 4.15)).

Als  $i \notin W, j \in W$ : in  $G^*$  is geen pijl  $(j, i)$  en  $z_{ij}$  neemt met  $c$  af.

Als  $z_{ij} \leq 0$  (dan is  $x_{ij} \leq a_{ij}$ ) geldt mag  $c$  willekeurig groot worden.

Als  $z_{ij} > 0$  (dan is  $x_{ij} \leq b_{ij}$ ) geldt dat als  $x_{ij} \leq a_{ij}$   $c$  willekeurig mag toenemen; als  $x_{ij} > a_{ij}$ , dan kan  $c$  maximaal  $z_{ij}$  zijn.

Dus  $c$  kan voor een pijl  $(i, j)$  met  $i \notin W, j \in W, x_{ij} > a_{ij}$  maximaal  $z_{ij}$  zijn.

Laat nu  $I_2 = \{(i, j) \mid i \notin W, j \in W, z_{ij} > 0; x_{ij} \geq a_{ij}\}$  en  $c_2 = \min\{z_{ij} \mid (i, j) \in I_2\}$ .

Als keuze voor  $c$  nemen we:  $c = \min\{c_1, c_2\}$ . Het is mogelijk dat zowel  $I_1$  als  $I_2$  leeg zijn. In dat geval is het probleem ontoelaatbaar, zoals de volgende stelling laat zien.

**Stelling 4.15** *Indien  $I_1 = \emptyset$  en  $I_2 = \emptyset$ , dan is probleem (4.2) ontoelaatbaar.*

### Bewijs

Laat  $x$  de circulatiestroom tijdens het algoritme zijn waarvoor (bij zekere  $u$ )  $I_1 = \emptyset$  en  $I_2 = \emptyset$ .

Omdat  $x$  een circulatoriestroom is, geldt:

$$\sum_{i \in W, j \notin W} x_{ij} = \sum_{i \notin W, j \in W} x_{ij} \quad (4.7)$$

Voor iedere pijl  $(i, j)$  met  $i \in W, j \notin W$  geldt: ofwel  $z_{ij} \geq 0$  en dan is  $x_{ij} \geq b_{ij}$ , ofwel  $z_{ij} < 0$  en dan is  $x_{ij} > b_{ij}$  (omdat  $I_1 = \emptyset$ ). Analoog geldt voor iedere pijl  $(i, j)$  met  $i \notin W, j \in W$ : ofwel  $z_{ij} \leq 0$  en dan is  $x_{ij} \leq a_{ij}$ , ofwel  $z_{ij} > 0$  en dan is  $x_{ij} < a_{ij}$ . Omdat  $p \notin W, q \in W$  is dus  $x_{pq} \leq a_{pq}$ . Verder weten we dat  $k_{pq} > 0$ . Stel  $x_{pq} = a_{pq}$ . Omdat  $k_{pq} > 0$  is  $z_{pq} > 0$ , d.w.z.  $(p, q) \in I_2$ : tegenspraak! Dus is  $x_{pq} < a_{pq}$ .

Stel dat (4.2) een toelaatbare oplossing  $x^*$  heeft. Dan volgt met behulp van (4.7),  $x_{ij} \geq b_{ij}$  voor  $i \in W, j \notin W$ ,  $x_{ij} \leq a_{ij}$  voor  $i \notin W, j \in W$ , en  $x_{pq} < a_{pq}$ :

$$0 = \sum_{i \in W, j \notin W} x_{ij}^* - \sum_{i \notin W, j \in W} x_{ij}^* \leq \sum_{i \in W, j \notin W} b_{ij} - \sum_{i \notin W, j \in W} a_{ij} < \sum_{i \in W, j \notin W} x_{ij} - \sum_{i \notin W, j \in W} x_{ij} = 0: \text{ tegenspraak.} \quad \blacksquare$$

*Algoritme 4.6: Algoritme voor een circulatoriestroom met minimale kosten*

1. a. Start met  $x = 0$  en een willekeurige  $u$ .  
b. Bepaal voor alle  $(i, j) \in A$ :  $z_{ij} = u_i - u_j - c_{ij}$  en  $k_{ij}$  volgens (4.5).
2. Als  $k_{ij} = 0$  voor alle  $(i, j) \in A$ :  $x$  is een optimale oplossing (stop).  
Anders: ga naar stap 3.
3. a. Kies  $(p, q)$  met  $k_{pq} > 0$  (zo mogelijk dezelfde keuze als in vorige iteratie).  
b. Construeer de graaf  $G^*$ .  
c. Als toename van de stroom over  $(p, q)$  het getal  $k_{pq}$  verlaagt:  
zoek een pad in  $G^*$  van  $q$  naar  $p$ ;  
anders: zoek in  $G^*$  een pad van  $p$  naar  $q$ .  
d. Als het is gelukt om in onderdeel c het gewenste pad te vinden: ga naar stap 4;  
anders: ga naar stap 5.
4. a. Breid het gevonden pad met  $(p, q)$  of  $(q, p)$  uit tot een ronde  $C$ .  
b. Vervoer zoveel mogelijk over  $C$  in de gewenste richting.  
c. Pas op de ronde  $C$  de stroom  $x$  en de getallen  $k_{ij}$  (zie (4.5)) aan.  
d. Ga naar stap 2.
5. a. Bepaal  $W = \{i \mid i \text{ is in } G^* \text{ bereikbaar vanuit het beoogde vertrekpunt}\}$ .  
b. Bepaal  $I_1 = \{(i, j) \mid i \in W, j \notin W; z_{ij} < 0; x_{ij} \leq b_{ij}\}$  en  
 $I_2 = \{(i, j) \mid i \notin W, j \in W; z_{ij} > 0; x_{ij} \geq a_{ij}\}$ .  
c. Als  $I_1 = I_2 = \emptyset$ : het probleem is ontoelaatbaar (stop);  
anders: ga naar stap 6.
6. a.  $c_1 = \min\{-z_{ij} \mid (i, j) \in I_1\}$ ;  $c_2 = \min\{z_{ij} \mid (i, j) \in I_2\}$  en  $c = \min\{c_1, c_2\}$ .  
b. Pas de getallen  $z$  en  $k$  aan volgens (4.6) resp. (4.5).  
c. Ga naar stap 2.

Voorbeeld 4.8 (vervolg)

We zijn gebleven in stap 5.  $W = \{4, 1\}$ .

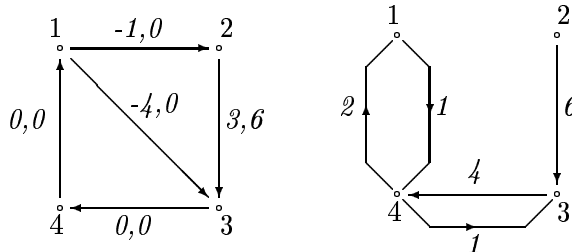
$I_1 = \{(1, 2), (1, 3)\}$ ;  $I_2 = \{(3, 4)\}$ .

$c_1 = 2$ ,  $c_2 = 1$ ;  $c = 1$ .

$z_{12} = -1$ ,  $z_{13} = -4$ ;  $z_{34} = 0$ .

We gaan nu naar een volgende iteratie.

Hiernaast staat  $G$  met  $z$ - en  $k$ -getallen en de graaf  $G^*$ .



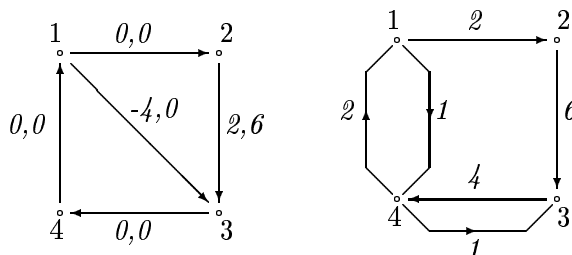
Neem  $(p, q) = (2, 3)$ .  $W = \{3, 4, 1\}$ .

$I_1 = \{(1, 2)\}$ ;  $I_2 = \{(2, 3)\}$ .

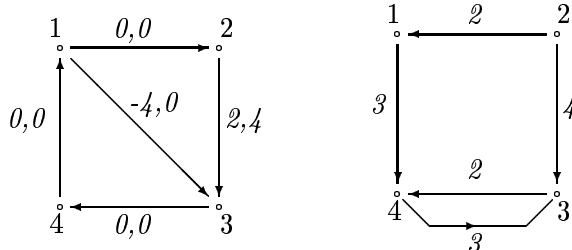
$c_1 = 1$ ,  $c_2 = 3$ ;  $c = 1$ .  $z_{12} = 0$ ,  $z_{23} = 2$ .

We gaan nu naar de derde iteratie.

Hiernaast staat  $G$  met  $z$ - en  $k$ -getallen en de graaf  $G^*$ .



Neem  $(p, q) = (2, 3)$ . We vinden nu de ronde  $[2, 3, 4, 1, 2]$ , waarover 2 vervoerd kan worden. Hiermee komen we bij iteratie 4. De grafen  $G$  (met de  $z$ - en  $k$ -getallen) en  $G^*$  staan weer hiernaast.



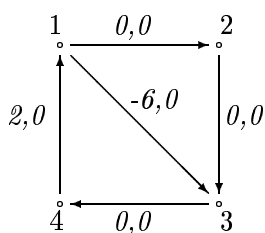
$(p, q) = (2, 3)$ .  $W = \{3, 4\}$ .

$I_1 = \emptyset$ ;  $I_2 = \{(2, 3)\}$ .

$c_1 = \infty$ ,  $c_2 = 2$ ;  $c = 2$ .

$z_{41} = 2$ ,  $z_{13} = -6$ ,  $z_{23} = 0$ .

Tenslotte komt de vijfde iteratie.



We constateren dat alle  $k$ -getallen 0 zijn geworden. We hebben dus een circulatiestroom met minimale kosten. Deze stroom is:  $x_{12} = 2$ ,  $x_{13} = 1$ ,  $x_{23} = 2$ ,  $x_{34} = 3$  en  $x_{41} = 3$ .

**Stelling 4.16** *Algoritme 4.6 is eindig en correct.*

**Bewijs**

Merk allereerst op dat als het algoritme stopt, òfwel een optimale oplossing is gevonden (als in stap 2 wordt gestopt) òfwel het probleem ontoelaatbaar is (als in stap 5c wordt gestopt).

Tijdens het algoritme wordt geen enkel  $k$ -getal ooit groter. Als er een ronde wordt gevonden, dan neemt ten minste één  $k$ -getal, namelijk  $k_{pq}$ , af. Veronderstel dat in een iteratie  $k_{pq}$  niet daalt.

Indien in de volgende iteratie geen ronde wordt gevonden, gebeurt het volgende:

- knooppunten die in  $W$  zaten blijven in  $W$ ;
- $z_{ij}$  wordt 0 voor de  $(i, j)$  die het minimum oplevert bij de bepaling van  $c$ ;
- $z_{ij}$  stijgt voor  $i \in W, j \notin W$ ;  $z_{ij}$  daalt voor  $i \notin W, j \in W$ .

Het aantal elementen van  $I_1 \cup I_2$  daalt dus in zo'n iteratie. Dit kan slechts eindig vaak gebeuren.

Hieruit volgt de eindigheid van het algoritme. ■

### Opmerking

Per iteratie is de hoeveelheid werk als volgt:

- constructie van de graaf  $G^*$ :  $\mathcal{O}(m)$ ;
- zoeken naar een pad en bepalen van het vervoer over de ronde:  $\mathcal{O}(m)$ ;
- aanpassen van de stroom  $x$ , de  $z$ -getallen en de  $k$ -getallen:  $\mathcal{O}(m)$ ;
- bepalen van  $I_1$ ,  $I_2$  en  $c$ :  $\mathcal{O}(m)$ .

Een iteratie heeft dus complexiteit  $\mathcal{O}(m)$ . We hebben in bovenstaand bewijs gezien dat als geen enkel  $k$ -getal afneemt,  $I_1 \cup I_2$  kleiner wordt. Dit kan hoogstens  $\mathcal{O}(m)$  keer gebeuren. Omdat de som van alle  $k$ -getallen bij de start hoogstens  $m \cdot U$  is, met  $U = \max_{(i,j)} b_{ij}$ , is de complexiteit van algoritme 4.6  $\mathcal{O}(m^3 \cdot U)$ .

## 4.7 Enkele toepassingen

### Afronden matrixelementen

Veronderstel dat we een  $n \times m$  matrix  $M$  met elementen  $m_{ij} \in \mathbb{R}_+$  hebben. De rij- en kolomsommen noteren we met  $r_i$ ,  $1 \leq i \leq n$ , resp.  $k_j$ ,  $1 \leq j \leq m$ . Er wordt nu gevraagd de getallen  $m_{ij}$ ,  $r_i$  en  $k_j$  naar boven of naar beneden af te ronden zdd. de rij- en kolomsommen kloppen. De vraag is: kan dit en, zo ja, hoe doe je dat?

We construeren de volgende gerichte graaf  $G = (V, A)$  met onder- en bovengrenzen  $a_{ij}$  resp.  $b_{ij}$ .

$V$ : we nemen een apart knooppunt  $s$  (beginpunt) en een apart knooppunt  $t$  (eindpunt); verder nemen we knooppunten voor iedere rij ( $v_i$ ,  $1 \leq i \leq n$ ) en voor iedere kolom ( $w_j$ ,  $1 \leq j \leq m$ ).

$A$ : er zijn pijlen van  $s$  naar  $v_i$  met ondergrenzen  $[r_i]$  en bovengrenzen  $\lceil r_i \rceil$ ,  $1 \leq i \leq n$ ;

analoog zijn er pijlen van  $w_j$  naar  $t$  met ondergrenzen  $\lfloor k_j \rfloor$  en bovengrenzen  $\lceil k_j \rceil$ ,  $1 \leq j \leq m$ ;

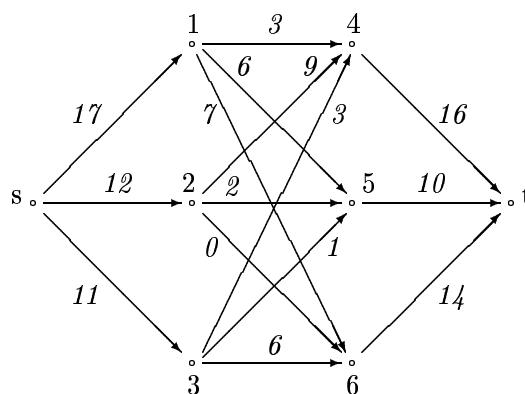
tenslotte zijn er pijlen van  $v_i$  naar  $w_j$  met ondergrenzen  $\lfloor m_{ij} \rfloor$  en bovengrenzen  $\lceil m_{ij} \rceil$ .

Het afrondingsprobleem is nu equivalent met het probleem: is er in  $G$  een toelaatbare stroom van  $s$  naar  $t$ ?

### Voorbeeld 4.9

Laat  $M = \begin{pmatrix} 3.1 & 6.8 & 7.3 \\ 9.6 & 2.4 & 0.7 \\ 3.6 & 1.2 & 6.5 \end{pmatrix}$  met rijsummen

17.2, 12.7 en 11.3, en met kolomsommen 16.3, 10.4 en 14.5. Dit geeft nevenstaande graaf  $G$ . De getallen bij de pijlen zijn de ondergrenzen; de bovengrenzen zijn 1 hoger. Een toelaatbare stroom  $x$  in  $G$  is:  $x_{s1} = 17, x_{s2} = 12, x_{s3} = 11, x_{14} = 3, x_{15} = 7, x_{16} = 7, x_{24} = 9, x_{25} = 2, x_{26} = 1, x_{34} = 4, x_{35} = 1, x_{36} = 6, x_{4t} = 16, x_{5t} = 10, x_{6t} = 14$ .



Bovenstaande oplossing geeft de afgeronde matrix  $\begin{pmatrix} 3 & 7 & 7 \\ 9 & 2 & 1 \\ 4 & 1 & 6 \end{pmatrix}$  met rijssommen 17, 12 en 11, en met kolomsommen 16, 10 en 14.

### *Een scheduling probleem*

Veronderstel dat  $n$  taken op  $m$  identieke machines moeten worden uitgevoerd. Taak  $j$  heeft bewerkingstijd  $p_j$ , deadline  $d_j$  en is beschikbaar voor bewerking op tijdstip  $r_j$ ,  $1 \leq j \leq n$ , waarbij we aannemen dat  $d_j \geq r_j + p_j$ ,  $1 \leq j \leq n$ .

Verder nemen we aan dat alle data geheel zijn en dat de bewerking van een taak mag worden afgebroken en later (ook op een andere machine) mag worden afgemaakt.

De vraagstelling luidt: is er een planning zdd. alle deadlines gehaald worden?

Dit probleem kan als volgt worden opgelost.

Rangschik alle getallen  $r_j$  en  $d_j$  in niet dalende volgorde. Dit geeft een aantal intervallen die we noteren als  $T_{kl}$  als het interval  $[k, l]$  betreft. In elke eenheid van het interval  $T_{kl}$  kan taak  $j$  (geheel of voor een deel) worden uitgevoerd d.e.s.d. als  $r_j \leq k$  en  $l \leq d_j$ .

We construeren de volgende gerichte graaf  $G = (V, A)$  met capaciteiten  $b_{ij}$ .

$V$ : we nemen een apart knooppunt  $s$  (beginpunt) en een apart knooppunt  $t$  (eindpunt); verder is er een knooppunt voor iedere taak  $j$  (knooppunt  $v_j$ ) en ieder interval  $T_{kl}$  (knooppunt  $w_{kl}$ ).  
 $A$ : er zijn pijlen van  $s$  naar  $v_j$  met capaciteit  $p_j$ , pijlen van  $v_j$  naar  $w_{kl}$  als  $r_j \leq k$  en  $l \leq d_j$  met capaciteit  $l - k$ , en tenslotte pijlen van  $w_{kl}$  naar  $t$  met capaciteit  $m \cdot (l - k)$  (dit is het beschikbare aantal machine-eenheden in interval  $T_{kl}$ ).

Het scheduling probleem is nu equivalent met het volgende probleem: is er in  $G$  een volgemaakte stroom van  $s$  naar  $t$  (d.w.z. een stroom met waarde  $\sum_{j=1}^n p_j$ )?

### *Werkzaamheden op 2 processoren*

Subroutines van een computerprogramma moeten ieder aan een van twee processoren worden toegewezen zdd. de totale tijd voor de berekeningen en de communicatie minimaal is.

Laat  $\alpha_i$  de kosten van subroutine  $i$  als deze op processor 1 wordt uitgevoerd,  $\beta_i$  de kosten van subroutine  $i$  als deze op processor 2 wordt uitgevoerd, en  $c_{ij}$  de kosten als de subroutines  $i$  en  $j$  op verschillende processoren worden uitgevoerd.

Dit probleem kan als volgt als een *minimale snede probleem* worden geformuleerd.

We construeren de volgende graaf  $G = (V, A)$ .

$V$ : we nemen een knooppunt  $s$  (voor processor 1) en een knooppunt  $t$  (voor processor 2) en verder een knooppunt  $v_j$  voor iedere subroutine  $j$ ,  $1 \leq j \leq n$ .

$A$ : er zijn pijlen van  $s$  naar  $v_j$  (met capaciteit  $\beta_j$ ) en van  $v_j$  naar  $t$  (met capaciteit  $\alpha_j$ ),  $1 \leq j \leq n$ ; verder is er een pijl van  $v_i$  en  $v_j$  en van  $v_j$  naar  $v_i$  als  $c_{ij} \neq 0$  met capaciteit  $c_{ij}$ .

We zullen laten zien dat er een één-éénduidig verband is tussen de  $(s, t)$ -snede en de planningen van de subroutines op de processoren zdd. de capaciteit van de snede overeenkomt met de totale tijdsduur voor de berekeningen en de communicatie.

Zij  $I_1$  en  $I_2$  de indexverz. van de subroutines die op processor 1 resp. processor 2 worden

uitgevoerd. De totale tijdsduur voor de berekeningen en de communicatie is dan:  $\sum_{i \in I_1} \alpha_i + \sum_{i \in I_2} \beta_i + \sum_{i \in I_1, j \in I_2} c_{ij}$ . Voor de bijbehorende  $(s, t)$ -snede nemen we  $W = \{s, I_1\}$ . Deze heeft de capaciteit: de capaciteit van de pijlen van  $s$  naar  $I_2$  + de capaciteit van de pijlen van  $I_1$  naar  $I_2$  + de capaciteit van de pijlen van  $I_1$  naar  $t$ :  $\sum_{i \in I_2} \beta_i + \sum_{i \in I_1, j \in I_2} c_{ij} + \sum_{i \in I_1} \alpha_i$ .

Omgekeerd geldt analoog dat als we uitgaan van een  $(s, t)$ -snede hierbij een planning hoort zdd. de capaciteit van de snede gelijk is aan de totale tijdsduur voor de berekeningen en de communicatie van deze planning.

### Scheduling van tankers

Een tanker-bedrijf is contractueel verplicht een aantal opdrachten, zeg  $n$  opdrachten, uit te voeren. Iedere opdracht is van het volgende type: vervoer een scheepslanding van een bepaald product, zeg product  $j$ , van plaats  $k$  naar plaats  $l$  zdd. het in plaats  $l$  op een bepaalde dag, zeg dag  $d_j$ , aankomt ( $1 \leq j \leq n$ ). In verband met bederf mag een transport ook niet eerder aankomen dan op de geplande dag. Het bedrijf wil het minimum aantal schepen bepalen om deze orders uit te voeren. De vervoerstijden tussen de verschillende plaatsen zijn bekend en afhankelijk van het feit of het een vol of een leeg schip betreft: van plaats  $k$  naar  $l$  duurt een beladen rit  $t_{kl}^1$  en een rit zonder vracht  $t_{kl}^0$ .

Een voorbeeld van een dergelijk probleem met 4 opdrachten en vervoer tussen 4 plaatsen staat hieronder (volle vrachten van A en B naar C en D).

Taak	Van	Naar	Dag
1	A	C	3
2	A	C	8
3	B	D	3
4	B	D	6

Vol	Naar C	Naar D
Van A	3	2
Van B	2	3

Leeg	Naar A	Naar B
Van C	2	1
Van D	1	2

Om dit probleem op te lossen construeren we de volgende gerichte graaf  $G = (V, A)$ .

$V$ : we nemen een apart knooppunt  $s$  (beginpunt) en een apart knooppunt  $t$  (eindpunt); verder zijn er twee knooppunten voor iedere opdracht  $j$  (knooppunt  $v_j$  met als interpretatie het begin van opdracht  $j$  en knooppunt  $w_j$  met als interpretatie het einde van opdracht  $j$ ).

$A$ : er zijn pijlen van  $s$  naar  $v_j$  met ondergrens 0 en bovengrens 1 (als we hier een hoeveelheid 1 over versturen betekent dit dat opdracht  $j$  als eerste door een tanker wordt uitgevoerd), pijlen van  $v_j$  naar  $w_j$  met onder- en bovengrens 1 (dit betekent dat opdracht  $j$  moet worden uitgevoerd), en pijlen van  $w_j$  naar  $t$  met ondergrens 0 en bovengrens 1 (als we hier een hoeveelheid 1 over versturen betekent dit dat opdracht  $j$  als laatste door een tanker wordt uitgevoerd); tenslotte zijn er pijlen van  $w_j$  naar  $v_i$  indien het volgende geldt: de starttijd van opdracht  $i$  (dit is de leveringsdag - de tijdsduur van de volle tanker van beginpunt opdracht  $i$  naar zijn bestemming) is niet eerder dan de leveringsdag van opdracht  $j$  + de tijdsduur van de lege tanker van bestemming opdracht  $j$  naar het beginpunt van opdracht  $i$ ; dit betekent dat opdracht  $i$  na opdracht  $j$  met dezelfde tanker kan worden uitgevoerd.

In bovenstaand voorbeeld geeft dit laatste alleen de pijlen van  $w_1$  naar  $v_2$  en van  $w_3$  naar  $v_2$  (ga dit zelf na).



Het resultaat van bovenstaand netwerk is dat ieder pad van  $s$  naar  $t$  correspondeert met een toelaatbare planning voor één tanker. Omdat de bovengrenzen van de pijlen 1 zijn, correspondeert een stroom van  $s$  naar  $t$  met waarde  $k$  met  $k$  disjuncte paden, dus met  $k$  verschillende tankers. Het oorspronkelijke probleem is dus equivalent met het vinden van een minimale stroom in bovenstaand netwerk. Hoe zo stroom gevonden kan worden wordt behandeld in Opgave 11.

## 4.8 Opgaven

### Opgave 4.1

Zij  $G = (V, A)$  een gerichte graaf en laten  $s, t_1, t_2, \dots, t_k$  verschillende knooppunten zijn.

Bewijs dat er  $k$  pijl-disjuncte paden  $P_1, P_2, \dots, P_k$  zijn zdd.  $P_i$  een pad van  $s$  naar  $t_i$  is ( $1 \leq i \leq k$ ) d.e.s.d. als  $\delta^+(W) \geq |\{i \mid t_i \notin W\}|$  voor iedere  $W \subseteq V$  met  $s \in W$ .

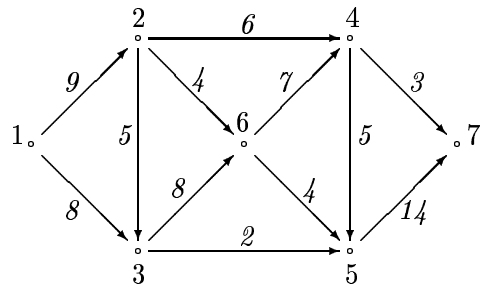
### Opgave 4.2

Beschouw het probleem om een maximale stroom te vinden van  $v_1$  naar  $v_n$  in een netwerk met capaciteiten. Neem aan dat  $v_1$  alleen uitgaande en  $v_n$  alleen binnenkomende pijlen heeft. Is de volgende uitspraak juist of onjuist?

De maximale stroom kan niet worden gevonden zonder iedere pijl van het netwerk minstens één keer te bekijken.

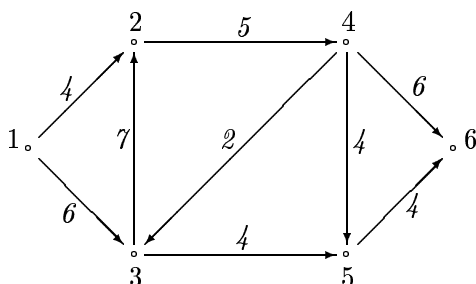
### Opgave 4.3

Bepaal een maximale stroom en een  $(1, 7)$ -snede met minimale capaciteit voor nevenstaand netwerk. De getallen bij de pijlen zijn weer de capaciteiten. Start met  $x = 0$ . Pas de implementatie van Edmonds en Karp toe.



### Opgave 4.4

- Beschrijf een alternatieve procedure, analoog aan die van Ford en Fulkerson, om een maximale stroom te bepalen die groeiketens vanuit  $v_n$  in plaats van vanuit  $v_1$  tracht op te sporen.
- Pas de procedure toe op onderstaand netwerk.

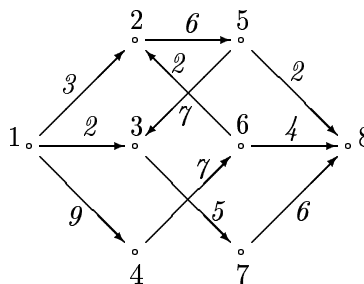


- Beschrijf een procedure om een pijl op te sporen die de eigenschap heeft dat het verhogen van

- de capaciteit van deze pijl de waarde van de maximale stroom doet toenemen.
- d. Pas de in onderdeel c beschreven methode toe op het netwerk uit onderdeel b.
- e. Bestaat er altijd zo'n pijl als in onderdeel c genoemd? Verklaar uw antwoord.

### Opgave 4.5

Bepaal een maximale stroom en een  $(1, 8)$ -sneede met minimale capaciteit voor nevenstaand netwerk. De getallen bij de pijlen zijn weer de capaciteiten. Start met  $x = 0$ . Pas de DMKM methode toe.



### Opgave 4.6\*

Beschouw een netwerk  $N$ , met knooppuntenverz.  $V$ , waarin alle capaciteiten gelijk aan 1 zijn. Toon, via de volgende tussenstappen, aan dat de DMKM methode complexiteit  $\mathcal{O}(m \cdot n^{\frac{2}{3}})$  heeft.

- a. Bewijs dat een iteratie van de DMKM methode in  $\mathcal{O}(m)$  kan worden uitgevoerd.
- Laat  $w$  de waarde maximale stroom in  $N$  zijn en laat  $p$  het minimum aantal pijlen zijn om in  $N$  vanuit knooppunt  $v_1$  knooppunt  $v_n$  te bereiken.
- Zij  $V_i = \{v \in V \mid v \text{ in } N \text{ via het zijwaarts zoeken bereikbaar vanuit } v_1 \text{ met } i \text{ pijlen}\}, 0 \leq i \leq p$ .
- b. Bewijs dat voor ieder tweetal  $V_i$  en  $V_{i+1}$  met  $0 \leq i \leq p - 1$  geldt dat het aantal knooppunten in minstens één van de twee verz. minstens  $\sqrt{w}$  is.
- c. Bewijs dat  $p \leq \frac{2n}{\sqrt{w}}$  is.
- d. Toon als volgt aan dat het aantal iteraties maximaal  $3n^{\frac{2}{3}} + 1$  is:
- Beschouw eerst het geval dat  $w \leq n^{\frac{2}{3}}$  is;
  - In het geval dat  $w > n^{\frac{2}{3}}$  is, beschouw dan de iteratie waarin de waarde van de stroom voor het eerst het getal  $w - n^{\frac{2}{3}}$  overschrijdt en toon de bewering aan door onderdeel c toe te passen.

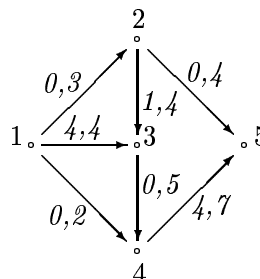
### Opgave 4.7\*

Beschouw een netwerk waarin alle capaciteiten gelijk aan 1 zijn en waarin bovendien geldt dat  $\min[\delta^+(v), \delta^-(v)] \leq 1$  voor alle knooppunten  $v$ .

Toon - op analoge wijze als in opgave 4.6, waarbij nu in onderdeel b moet worden bewezen dat  $\#V_i \geq w, 1 \leq i \leq p - 1$  - aan dat de DMKM methode in dit geval complexiteit  $\mathcal{O}(m \cdot \sqrt{n})$  heeft.

### Opgave 4.8

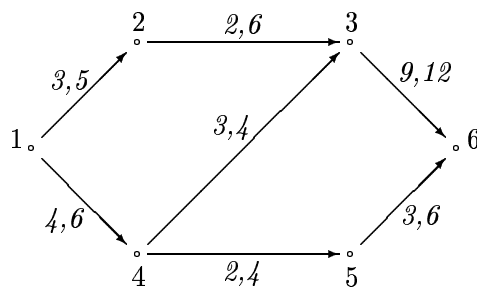
Ga met algoritme 4.4 na of nevenstaand netwerk, waarin bij iedere pijl de onder- en de bovengrens staat, een toelaatbare stroom heeft. Zo ja, bepaal een maximale stroom.



### Opgave 4.9

Beschouw het hiernaast getekende netwerk met onder- en bovengrenzen.

- Toon aan dat dit netwerk geen toelaatbare stroom heeft.
- Verander de ondergrens van de pijl  $(3, 6)$  in 5. Toon aan dat dit nieuwe netwerk een toelaatbare stroom heeft en bepaal een maximale stroom.



### Opgave 4.10

Beschouw een netwerk  $N$  met onder- en bovengrenzen  $a_{ij}$  resp.  $b_{ij}$ . Veronderstel dat  $N$  een toelaatbare stroom heeft. Voor een  $(1, n)$ -snede  $(W, V - W)$  definiëren we de capaciteit  $c(W)$  door:  $c(W) = \sum_{i \in W, j \notin W} b_{ij} - \sum_{i \notin W, j \in W} a_{ij}$ .

Toon het volgende aan: de waarde van een maximale stroom is gelijk aan de capaciteit van een  $(1, n)$ -snede met minimale capaciteit.

### Opgave 4.11

- Beschrijf een methode om een *minimale stroom* te bepalen in een netwerk met onder- en bovengrenzen.
- Pas de methode toe op het netwerk uit opgave 9b.
- Veronderstel dat er een toelaatbare stroom bestaat. Geeft de methode om een toelaatbare stroom te vinden dan altijd een minimale stroom?

### Opgave 4.12\*

Beschouw een netwerk met onder- en bovengrenzen.

- Formuleer het probleem om een maximale stroom te bepalen als een lineair programmeringsprobleem.
- Stel het duale probleem op en geef er een interpretatie aan.
- Bewijs met de dualiteitsstelling van de lineaire programmering dat de waarde van de maximale stroom gelijk is aan de capaciteit van de  $(1, n)$ -snede met de kleinste capaciteit, waarbij de capaciteit is gedefinieerd zoals in opgave 10.

### Opgave 4.13\*

Zij  $N$  een netwerk met ondergrenzen  $a_{ij}$  en bovengrenzen  $b_{ij}$  zdd. positieve ondergrenzen alleen voorkomen bij pijlen  $(1, j)$  en  $(i, n)$ . Knooppunt 1 heeft alleen uitgaande en knooppunt  $n$  alleen binnenkomende pijlen.

- Bewijs dat  $N$  een toelaatbare stroom heeft d.e.s.d. als:

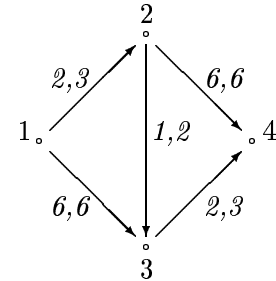
$$\sum_{i \in W, j \notin W} b_{ij} \geq \sum_{j \in W} a_{1j} \text{ voor iedere } W \text{ met } 1 \notin W \text{ en } n \notin W;$$

$$\sum_{i \in W, j \notin W} b_{ij} \geq \sum_{i \notin W} a_{in} \text{ voor iedere } W \text{ met } 1 \in W \text{ en } n \in W.$$

- b. Bewijs dat  $N$  een toelaatbare stroom heeft d.e.s.d. als er twee stromen  $x^1$  en  $x^2$  bestaan waarvoor geldt:
- $0 \leq x_{ij}^1 \leq b_{ij}$  en  $0 \leq x_{ij}^2 \leq b_{ij}$  voor alle  $(i, j) \in A$ ;  
 $a_{1j} \leq x_{1j}^1$  voor alle  $(1, j) \in A$ ;  $a_{in} \leq x_{in}^2$  voor alle  $(i, n) \in A$ .

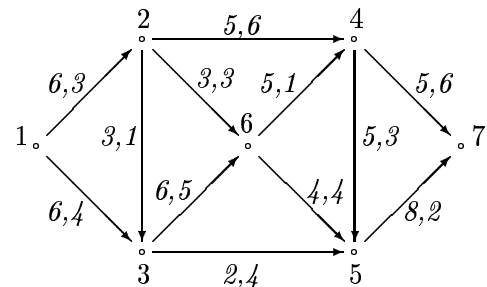
**Opgave 4.14**

Bepaal een maximale stroom met minimale kosten voor nevenstaand netwerk. De getal-paren bij de pijlen zijn de capaciteiten (het eerste getal) en de kosten (het tweede getal).



**Opgave 4.15**

Bepaal een maximale stroom met minimale kosten voor nevenstaand netwerk. De getal-paren bij de pijlen zijn de capaciteiten (het eerste getal) en de kosten (het tweede getal).



**Opgave 4.16**

Zij  $x$  een maximale stroom en  $(W, V - W)$  een minimale  $(1, n)$ -snede. Toon aan dat  $x$  een minimale kostenstroom is d.e.s.d. als iedere ronde die geheel in  $W$  of geheel in  $V - W$  ligt een niet-negatieve lengte m.b.t.  $l$ , gedefinieerd in (4.1), heeft.

**Opgave 4.17**

Beschouw het probleem om een maximale stroom met minimale kosten te bepalen in een netwerk met onder- en bovengrenzen. Beschrijf een methode om dit probleem op te lossen.

**Opgave 4.18\***

Beschouw het volgende transportprobleem. Er zijn  $m$  voorraadplaatsen, waarbij in plaats  $i$  een voorraad  $a_i$  is,  $1 \leq i \leq m$ . Daarnaast zijn er  $n$  bestemmingen en in bestemming  $j$  is een hoeveelheid  $b_j$  nodig,  $1 \leq j \leq n$ . Neem aan dat  $\sum_{i=1}^m a_i = \sum_{j=1}^n b_j$ .

Verder is gegeven dat er per eenheid van vervoer van voorraadplaats  $i$  naar bestemming  $j$  transportkosten  $c_{ij}$  zijn. Gevraagd wordt een transportschema op te stellen met minimale kosten.

- Formuleer dit transportprobleem als een minimale kostenstroom probleem.
- Omgekeerd, formuleer voor een minimale kostenstroom probleem met waarde  $w$  een transportprobleem zdd. een toelaatbare stroom voor het minimale kostenstroom probleem overeenkomt met een toelaatbaar transportschema met dezelfde kosten.

### Opgave 4.19\*

- Formuleer het probleem om een minimale kostenstroom probleem met waarde  $w$  te bepalen als een lineair programmeringsprobleem.
- Stel het duale probleem op en geef de orthogonaliteitsrelaties.
- Zij  $x$  een minimale kostenstroom met waarde  $w$ . Toon m.b.v. de orthogonaliteitsrelaties aan dat de lengte van iedere ronde niet-negatief is.
- Zij  $x$  een toelaatbare basisoplossing van het LP-probleem. Stel dat de lengte van iedere ronde m.b.t.  $l$ , gedefinieerd in (4.1), niet-negatief is. Bewijs (zonder Stelling 4.13 te gebruiken) dat  $x$  een minimale kostenstroom is.

### Opgave 4.20\*

Beschouw in het algoritme om een minimale kostenstroom te bepalen de volgende veranderingen in de lengte-functie  $l^k$  tijdens de  $k$ -de iteratie:

$$l_{ij}^k = \begin{cases} c_{ij} + \pi_i^k - \pi_j^k & \text{als } (i, j) \text{ een voorwaartse pijl is} \\ -c_{ji} + \pi_i^k - \pi_j^k & \text{als } (i, j) \text{ een achterwaartse pijl is,} \end{cases}$$

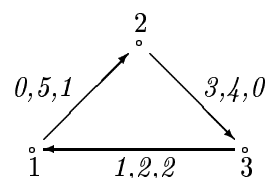
waarbij de getallen  $\pi_i^k$ ,  $1 \leq i \leq n$ , als volgt recursief worden bepaald:

$$\begin{cases} \pi_i^1 = 0 \\ \pi_i^k = \pi_i^{k-1} + u_i^{k-1} \text{ met } u_i^{k-1} \text{ de lengte van het kortste pad van } v_1 \text{ naar } v_n \text{ in iteratie } k-1 \end{cases}$$

- Pas deze gewijzigde methode toe op het probleem uit opgave 14.
- Toon aan dat een pad  $P$  van  $v_1$  naar  $v_n$  een kortste pad is t.o.v. de lengte-functie  $l$ , gedefinieerd in (4.1), d.e.s.d. als  $P$  een kortste pad is t.o.v. de in deze opgave gedefinieerde lengte-functie  $l^k$ .
- Toon aan dat  $l_{ij}^k \geq 0$  voor alle  $i, j$  en  $k$  (neem hierbij aan dat  $c_{ij} \geq 0$  voor alle  $(i, j)$ ).
- Beschrijf een methode met complexiteit  $\mathcal{O}(w \cdot n^2)$  om een minimale kostenstroom met waarde  $w$  te bepalen (neem hierbij aan dat  $c_{ij} \geq 0$  voor alle  $(i, j)$ ).

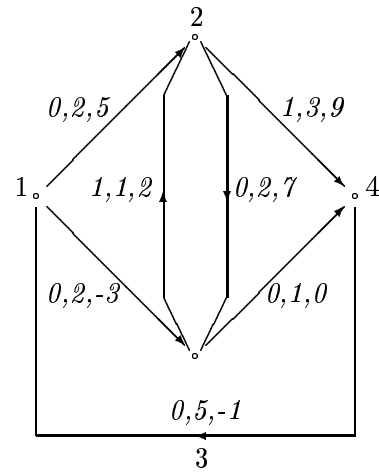
### Opgave 4.21

Los het probleem om een circulatiestroom met minimale kosten te vinden op nevenstaand netwerk. De getallen bij de pijlen zijn de ondergrenzen (het eerste getal), bovengrenzen (het tweede getal) en de kosten (het derde getal). Start met  $x = 0$  en  $u = 0$ .



**Opgave 4.22**

Los het probleem om een circulatiestroom met minimale kosten te vinden op nevenstaand netwerk. De getallen bij de pijlen zijn de ondergrenzen (het eerste getal), bovengrenzen (het tweede getal) en de kosten (het derde getal). Start met  $x = 0$  en  $u = 0$ .

**Opgave 4.23**

Beschouw probleem (4.2) om een circulatiestroom met minimale kosten te vinden. Toon aan dat er een toelaatbare oplossing van (4.2) bestaat d.e.s.d. als voor iedere  $W \subseteq V$  geldt dat  $\sum_{i \in W, j \notin W} b_{ij} \geq \sum_{i \notin W, j \in W} a_{ij}$ .

**Opgave 4.24**

Los het probleem uit paragraaf 4.7 om tankers optimaal te schedulen voor de gegeven data op.

# Hoofdstuk 5

## Netwerk simplex methode

### 5.1 Inleiding

Optimaliseringsproblemen in netwerken (gerichte graaf  $G = (V, A)$  met één of meer functies op de pijlen) zijn vaak als lineair programmeringsprobleem te formuleren. Dit soort problemen is in principe met de simplex methode op te lossen. We kunnen echter de structuur van het netwerk benutten om efficiëntere implementaties te verkrijgen. In feite komt zo'n speciale implementatie overeen met het alleen hoeven uitrekenen van de eerste kolom (het getransformeerde rechterlid) en de onderste rij (de getransformeerde doelfunctie). Deze kolom en rij horen bij een basisoplossing en de daarbij corresponderende duale oplossing. De elementen in het inwendige van het tableau hoeven dus niet te worden berekend. Tevens zal blijken dat de berekeningen om de basisoplossing en bijbehorende duale oplossing te bepalen eenvoudig in het netwerk zijn uit te voeren.

We beschouwen het minimale kostenprobleem met onder- en bovengrenzen en met in knooppunt  $i$  een gegeven *nettoproductie*  $s_i$ ,  $1 \leq i \leq n$  met  $\sum_{i=1}^n s_i = 0$  (anders is het probleem ontoelaatbaar).

$$\min \left\{ \sum_{(i,j) \in A} c_{ij} x_{ij} \mid \begin{array}{l} \sum_j x_{ij} - \sum_j x_{ji} = s_i, \quad 1 \leq i \leq n \\ a_{ij} \leq x_{ij} \leq b_{ij}, \quad (i, j) \in A \end{array} \right\} \quad (5.1)$$

In vectornotatie kan het probleem geschreven worden als:

$$\min \left\{ cx \mid \begin{array}{l} Nx = s \\ a \leq x \leq b \end{array} \right\} \quad (5.2)$$

#### Opmerkingen

1. De matrix  $N$  is de incidentiematrix van de gerichte graaf en dus totaal unimodulair (zie Stelling 1.5). Als de getallen  $a_{ij}$ ,  $b_{ij}$  en  $s_i$  geheeltallig zijn, wat we zullen aannemen, is iedere basisoplossing geheeltallig.
2. Als alle ondergrenzen  $a_{ij} = 0$  zijn, dan noemen we de getallen  $b_{ij}$  *capaciteiten* in plaats van bovengrenzen; als tevens de bovengrenzen  $\infty$  zijn, dan spreken we over een minimale stroom-

probleem zonder capaciteiten.

3. Een *stroom* is een functie  $x : A \rightarrow \mathbb{Z}$  zdd.  $Nx = s$ ; indien ook voldaan is aan  $a \leq x \leq b$ , dan heet de stroom *toelaatbaar*.

### Speciale gevallen

1. Het *kortste pad probleem* van  $v_1$  naar de andere knooppunten  $v_i$ ,  $2 \leq i \leq n$ , is een probleem van de vorm (5.1), maar zonder capaciteiten, met  $s_1 = n - 1$  en  $s_i = -1$ ,  $2 \leq i \leq n$ , en met  $c_{ij} = l_{ij}$  voor alle  $(i, j) \in A$ :

$$\min \left\{ \sum_{(i,j) \in A} l_{ij} x_{ij} \left| \begin{array}{l} \sum_j x_{1j} - \sum_j x_{j1} = n - 1 \\ \sum_j x_{ij} - \sum_j x_{ji} = -1, \quad 2 \leq i \leq n \\ x_{ij} \geq 0, \quad (i, j) \in A \end{array} \right. \right\} \quad (5.3)$$

In het dictaat Inleiding Besliskunde is de netwerk simplex methode behandeld voor het kortste pad probleem. Deze methode heeft complexiteit  $\mathcal{O}(n^3)$ .

2. Het *maximale stroom probleem* krijgen we door alle nettoproducties  $s_i$ , ondergrenzen  $a_{ij}$  en kosten  $c_{ij}$  gelijk aan 0 te nemen, de pijl  $(n, 1)$  toe te voegen met bovengrens  $\infty$  en kosten -1. Dit geeft de equivalente formulering:

$$\max \left\{ x_{n1} \left| \begin{array}{l} \sum_j x_{ij} - \sum_j x_{ji} = 0, \quad 1 \leq i \leq n \\ 0 \leq x_{ij} \leq b_{ij}, \quad (i, j) \in A \end{array} \right. \right\} \quad (5.4)$$

3. Het *transportprobleem* is ook een speciaal geval van (5.1). In een transportprobleem is de graaf bipartiet, d.w.z.  $V = V_1 \cup V_2$  en de pijlen lopen van  $V_1$  naar  $V_2$ . In  $v_i \in V_1$  dient een hoeveelheid  $a_i$  verzonden te worden en in  $w_j \in V_2$  wordt een hoeveelheid  $b_j$  ontvangen, waarbij  $\sum_{i \in V_1} a_i = \sum_{j \in V_2} b_j$ . Voor iedere eenheid die van  $v_i \in V_1$  naar  $w_j \in V_2$  wordt vervoerd worden vervoerskosten  $c_{ij}$  berekend. Welk vervoersschema voldoet aan de eisen en is zo goedkoop mogelijk? Dit probleem geeft de volgende LP-formulering:

$$\min \left\{ \sum_{(i,j) \in A} c_{ij} x_{ij} \left| \begin{array}{l} \sum_j x_{ij} = a_i, \quad i \in V_1 \\ \sum_i x_{ij} = b_j, \quad j \in V_2 \\ x_{ij} \geq 0, \quad (i, j) \in A \end{array} \right. \right\} \quad (5.5)$$

Als  $V_1 = V_2$  en  $a_i = b_j = 1$  voor alle  $(i, j)$ , dan krijgen we het *toewijzingsprobleem*.

Van dit probleem luidt de LP-formulering:

$$\min \left\{ \sum_{(i,j) \in A} c_{ij} x_{ij} \left| \begin{array}{l} \sum_j x_{ij} = 1, \quad i \in V_1 \\ \sum_i x_{ij} = 1, \quad j \in V_2 \\ x_{ij} \geq 0, \quad (i, j) \in A \end{array} \right. \right\} \quad (5.6)$$

4. Het *circulatiestroom probleem* (4.2) uit paragraaf 4.6 is het speciale geval waarin alle  $s_i = 0$  zijn:

$$\min \left\{ \sum_{(i,j) \in A} c_{ij} x_{ij} \left| \begin{array}{l} \sum_j x_{ij} - \sum_j x_{ji} = 0, \quad 1 \leq i \leq n \\ a_{ij} \leq x_{ij} \leq b_{ij}, \quad (i, j) \in A \end{array} \right. \right\} \quad (5.7)$$



## 5.2 Bases en opspannende bomen

We veronderstellen in deze paragraaf dat alle ondergrenzen 0 en alle bovengrenzen  $\infty$  zijn, d.w.z. dat we een probleem zonder capaciteiten hebben. Zij  $T$  een *opspannende boom* van de bij  $G$  behorende niet-gerichte graaf. Een *stroom*  $x$ , d.w.z. een geheeltallige functie op de pijlen die voldoet aan  $Nx = s$ , heet een *basisstroom* bij  $T$  indien  $x_{ij} = 0$  voor alle  $(i, j) \notin T$ . Het volgende lemma laat zien dat iedere opspannende boom op unieke wijze een basisstroom bepaalt.

**Lemma 5.1** *Zij  $T$  een opspannende boom. Dan is er een unieke stroom  $x$  met  $x_{ij} = 0$  voor alle  $(i, j) \notin T$ . Bovendien geldt voor iedere  $(i, j) \in T$ : als de pijl  $(i, j)$  de verbinding vormt van component  $T_i$  naar component  $T_j$  in  $T - \{(i, j)\}$ , dan is  $x_{ij} = \sum_{k \in T_i} s_k$ .*

### Bewijs

Zij  $B$  de  $n \times (n-1)$  matrix die uit de incidentiematrix  $N$  ontstaat door de  $n-1$  kolommen behorend bij boom  $T$  te nemen. Volgens Stelling 1.6 zijn de kolommen van  $B$  lineair onafhankelijk. Het stelsel  $Bx = s$  heeft dus een unieke oplossing die i.v.m. de totaal unimodulariteit ook geheeltallig is. Bij  $T$  hoort dus een unieke stroom  $x$ .

Zij  $(i, j) \in T$  en laat  $(i, j)$  de verbinding vormen tussen de componenten  $T_i$ , die  $i$  bevat, en  $T_j$ , waar  $j$  toe behoort.  $\sum_{k \in T_i} s_k$  is de nettoproductie van  $T_i$ . Omdat de pijl  $(i, j)$  de enige verbinding is van  $T_i$  met de rest van de knooppunten (d.w.z. met  $T_j$ ) is  $x_{ij} = \sum_{k \in T_i} s_k$ . ■

De verkregen oplossing hoeft niet toelaatbaar te zijn:  $x_{ij}$  kan negatief zijn voor sommige  $(i, j)$ 's. Als  $x_{ij} \geq 0$  voor alle  $(i, j)$ , dan is  $x$  een *toegelaten basisstroom*. Omdat de kolommen van  $N$  behorende bij positieve  $x$ -componenten lineair onafhankelijk zijn (ze behoren immers tot  $T$ ), is een toegelaten basisstroom  $x$  ook een toegelaten basisoplossing van het LP-probleem.

Het tweede deel van bovenstaand lemma geeft tevens een efficiënte methode om de bij een opspannende boom  $T$  behorende basisstroom  $x$  te bepalen. Merk op, omdat  $\sum_{k \in T} s_k = 0$ , dat voor  $(i, j) \in T$  geldt:  $x_{ij} = \sum_{k \in T_i} s_k = -\sum_{k \in T_j} s_k$ .

Met de methode van het *voorwaarts zoeken* krijgt iedere knooppunt  $v_i$  een nummer  $N(i)$  in de volgorde waarin dit knooppunt wordt bezocht en een unieke voorganger  $PRED[i]$  (behalve de wortel). Doorloop nu de knooppunten in de omgekeerde volgorde waarin ze zijn bezocht. Beschouw het eerste knooppunt (dus het knooppunt dat het laatst is bezocht), zeg knooppunt  $j$  met  $PRED[j] = i$ . Dit is een eindpunt van de boom, zodat  $T_j = \{j\}$ . Als de pijl tussen  $i$  en  $j$  van  $i$  naar  $j$  loopt geldt dus  $x_{ij} = -s_j$ ; anders geldt  $x_{ji} = s_j$ . Stel dat we vervolgens verder gaan in  $i$  ( $i$  is dus het op één na laatst bezochte knooppunt) en dat  $PRED[i] = l$ . Als  $(l, i) \in A$ , dan is  $x_{li} = -\sum_{k \in T_i} s_k = -(s_i + s_j)$ ; als  $(i, l) \in A$ , dan is  $x_{il} = \sum_{k \in T_i} s_k = (s_i + s_j)$ . Indien we dus aan het einde van de vorige stap de nettoproductie  $s_j$  van  $j$  hadden overgeheveld naar  $i$ , dan krijgen we dezelfde toewijzing aan de  $x$ -variabele. Dit herhaalt zich en geeft de volgende procedure, waarin de ge-update nettoproducties met  $t$ -getallen worden genoteerd. De complexiteit van dit

algoritme is hetzelfde als van het voorwaarts zoeken:  $\mathcal{O}(m)$  (de initialisatie  $x_{ij} = 0$  voor alle  $(i, j) \in A$  geeft  $\mathcal{O}(m)$ ; de rest van het werk is  $\mathcal{O}(n)$ ).

*Algoritme 5.1: Algoritme om een basisstroom te bepalen bij opspannende boom  $T$*

1. a. Kies een wortel, zeg knooppunt 1. Pas de methode van het voorwaarts zoeken toe om in  $T$  een nummering van de knooppunten met getallen  $N[\cdot]$  aan te brengen in de volgorde waarin de knooppunten zijn bezocht; verder krijgt ieder knooppunt  $j \neq 1$  een waarde  $PRED[j]$  van zijn voorganger.
  - b. Laat  $t_i = s_i$ ,  $1 \leq i \leq n$ ,  $x_{ij} = 0$  voor alle  $(i, j) \in A$  en  $k = n$ .
2. a. Neem  $j$  zdd.  $N[j] = k$  en laat  $i = PRED[j]$ .
  - b. Als  $(i, j) \in A$ :  $x_{ij} = -t_j$ ;  
anders:  $x_{ji} = t_j$ .
  - c.  $t_i = t_i + t_j$ .
3. a.  $k := k - 1$ .
  - b. Als  $k \neq 1$ : ga naar stap 2;  
anders: stop.

### Voorbeeld 5.1

Hiernaast staat een netwerk. De nettoproducties zijn:

$$s_1 = 2, s_2 = 0, s_3 = -2, s_4 = 2, s_5 = 1, s_6 = -2,$$

$$s_7 = 0 \text{ en } s_8 = -1. \text{ Neem als opspannende boom}$$

$$T = \{(1, 2), (2, 3), (4, 3), (3, 8), (8, 6), (5, 6), (6, 7)\}.$$

Voorwaarts zoeken geeft de volgende nummering en

$$\text{voorgangers: } N[1] = 1; N[2] = 2, PRED[2] = 1;$$

$$N[3] = 3, PRED[3] = 2; N[4] = 4, PRED[4] = 3;$$

$$N[5] = 8, PRED[8] = 3; N[6] = 6, PRED[6] = 8;$$

$$N[7] = 5, PRED[7] = 6; N[8] = 7, PRED[8] = 6.$$

$$t_1 = 2, t_2 = 0, t_3 = -2, t_4 = 2, t_5 = 1, t_6 = -2, t_7 = 0 \text{ en } t_8 = -1. x_{12} = x_{14} = x_{23} = x_{24} = x_{38} = x_{43} = x_{53} = x_{54} = x_{56} = x_{57} = x_{67} = x_{86} = 0.$$

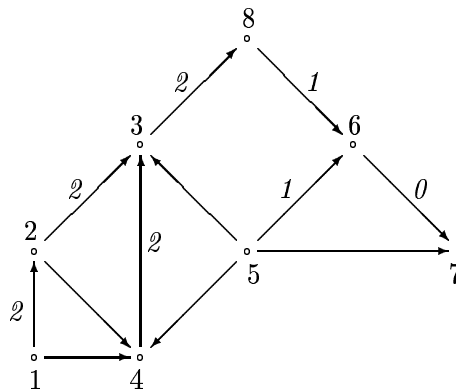
$$j = 7; i = 6; x_{67} = 0; t_6 = -2; k = 7. \quad j = 5; i = 6; x_{56} = 1; t_6 = -1; k = 6.$$

$$j = 6; i = 8; x_{86} = 1; t_8 = -2; k = 5. \quad j = 8; i = 3; x_{38} = 2; t_3 = -4; k = 4.$$

$$j = 4; i = 3; x_{43} = 2; t_3 = -2; k = 3. \quad j = 3; i = 2; x_{23} = 2; t_2 = -2; k = 2.$$

$$j = 2; i = 1; x_{12} = 2; t_1 = 0; k = 1.$$

De basisstroom, die in dit geval toelaatbaar is, staat in bovenstaande figuur.



### Opmerking

In het algemeen kunnen *gedegeneerde* oplossingen, d.w.z. oplossingen waarin ten minste één basisvariabele de waarde 0 heeft, voorkomen. In bovenstaand voorbeeld hebben we een gedegeneerde oplossing, want de basisvariabele  $x_{67} = 0$ . Het is ook mogelijk dat een bepaalde oplossing  $x$  basisoplossing is bij meer dan één opspannende boom. In voorbeeld 5.1 is de verkregen oplossing  $x$  ook de unieke basisstroom bij de boom die verkregen wordt door de pijl  $(6, 7)$  te vervangen door de pijl  $(5, 7)$  (ga dit zelf na).

### 5.3 Algoritme voor problemen zonder capaciteiten

Bij de simplex methode gaan we van een toegelaten basisoplossing naar een (betere) toegelaten basisoplossing. We kiezen eerst een pivotkolom; dit is een kolom behorende bij een niet-basisvariabele, waarvoor de *gereduceerde kosten* negatief zijn. Het duale probleem van (5.1), met  $a_{ij} = 0$  en  $b_{ij} = \infty$  voor alle  $(i, j) \in A$ , luidt:

$$\max \left\{ \sum_{i=1}^n s_i u_i \mid u_i - u_j \leq c_{ij}, (i, j) \in A \right\} \quad (5.8)$$

Zij  $T$  een opspannende boom met bijbehorende basisoplossing  $x$ . Veronderstel dat deze toelaatbaar is. Op grond van de orthogonaliteit geldt voor de bijbehorende duale oplossing  $u$  dat  $u_i - u_j = c_{ij}$  voor alle  $(i, j) \in T$ . De gereduceerde kosten  $\bar{c}_{ij}$  zijn de verschilvariabelen van het duale probleem, d.w.z.

$$\bar{c}_{ij} = c_{ij} - u_i + u_j, (i, j) \in A. \quad (5.9)$$

Omdat  $u$  op een constante na bepaald is, kunnen we  $u_1 = 0$  nemen. Analoog aan algoritme 5.1 om de basisoplossing  $x$  uit de boom  $T$  te berekenen, kunnen de getallen  $u_i$ ,  $2 \leq i \leq n$ , ook in boom  $T$  efficiënt worden berekend door de knooppunten af te lopen in de volgorde waarin ze zijn bezocht met het voorwaarts zoeken. Dit geeft het volgende algoritme, dat complexiteit  $\mathcal{O}(n)$  heeft.

*Algoritme 5.2: Algoritme om de duale oplossing  $u$  te bepalen voor een basisoplossing  $x$  die behoort bij de opspannende boom  $T$ .*

1. a. Kies een wortel, zeg knooppunt 1. Pas de methode van het voorwaarts zoeken toe om in  $T$  een nummering van de knooppunten met getallen  $N[\cdot]$  aan te brengen in de volgorde waarin de knooppunten zijn bezocht; verder krijgt ieder knooppunt  $j \neq 1$  een waarde  $PRED[j]$  van zijn voorganger.
  - b. Laat  $u_1 = 0$  en  $k = 2$ .
2. a. Neem  $j$  zdd.  $N[j] = k$  en laat  $i = PRED[j]$ .
  - b. Als  $(i, j) \in A$ :  $u_j = u_i - c_{ij}$ ;  
anders:  $u_j = u_i + c_{ji}$ .
3. a.  $k := k + 1$ .
  - b. Als  $k \leq n$ : ga naar stap 2;  
anders: stop.

#### Voorbeeld 5.1 (vervolg)

Laat  $c_{12} = 2$ ,  $c_{14} = -1$ ,  $c_{23} = 1$ ,  $c_{24} = 3$ ,  $c_{38} = 2$ ,  $c_{43} = 2$ ,  $c_{53} = 4$ ,  $c_{54} = 2$ ,  $c_{56} = 3$ ,  $c_{57} = 2$ ,  $c_{67} = 0$ ,  $c_{86} = -1$ . De berekening van de bij  $x$  behorende duale variabelen  $u$  gaat als volgt.  
 $k = 2$ ;  $j = 2$ ;  $i = 1$ ;  $u_2 = -2$ .  $k = 3$ ;  $j = 3$ ;  $i = 2$ ;  $u_3 = -3$ .  $k = 4$ ;  $j = 4$ ;  $i = 3$ ;  $u_4 = -1$ .

$k = 5; j = 8; i = 3; u_8 = -5. \quad k = 6; j = 6; i = 8; u_6 = -4. \quad k = 7; j = 5; i = 6; u_5 = -1.$   
 $k = 8; j = 7; i = 6; u_7 = -4.$

Veronderstel dat  $x_{kl}$  de variabele is die behoort bij de pivotkolom, d.w.z.  $(k, l) \notin T$  en  $\bar{c}_{kl} < 0$ . De variabele  $x_{kl}$  komt nu in de basis en een andere variabele gaat de basis verlaten. Door  $(k, l)$  aan  $T$  toe te voegen ontstaat precies één kring, zeg  $C$ , en  $(k, l)$  behoort tot  $C$ . Doorloop  $C$  in de richting van  $(k, l)$ . Aan een pijl die we in de goede richting doorlopen kan een willekeurige niet-negatieve waarde van de stroom worden toegevoegd; een pijl die we in tegengestelde richting doorlopen, zeg we lopen van  $j$  naar  $i$  terwijl de pijl van  $i$  naar  $j$  loopt, kan maximaal  $x_{ij}$  omlaag.

Laat  $I_1 = \{(i, j) \mid (i, j) \text{ wordt in } C \text{ in de goede richting doorlopen}\}$ ,  $I_2 = \{(i, j) \mid (i, j) \text{ wordt in } C \text{ in de tegengestelde richting doorlopen}\}$  en  $\Delta = \min\{x_{ij} \mid (i, j) \in I_2\}$ . Dan gaat de pijl die het minimum van  $\Delta$  bepaalt, zeg dit is pijl  $(p, q)$ , uit de boom (deze pijl is in het algemeen niet uniek). Als  $I_2 = \emptyset$ , dan heeft het probleem een *oneindige oplossing*, d.w.z. dat de kosten naar  $-\infty$  kunnen gaan. De stroom  $x$  verandert in stroom  $x'$  op de volgende wijze:

$$x'_{ij} = \begin{cases} x_{ij} + \Delta, & (i, j) \in I_1 \\ x_{ij} - \Delta, & (i, j) \in I_2 \\ x_{ij}, & (i, j) \notin I_1 \cup I_2 \end{cases}$$

Het kan voorkomen dat  $\Delta = 0$ . In dat geval blijft de stroom onveranderd en verandert alleen de boom  $T$ . We spreken dan over een *gedegeneerde oplossing*.

Voor alle  $(i, j) \in T$  geldt  $\bar{c}_{ij} = 0$ , d.w.z.  $u_j = u_i - c_{ij}$ . Omdat  $u_1 = 0$  volgt dat  $u_j$  de lengte is in de boom  $T$  om van  $j$  naar de wortel 1 te gaan, waarbij de lengte gelijk is aan de kosten als de pijl in de goede richting wordt doorlopen en tegengesteld aan de kosten als de pijl in de tegengestelde richting wordt doorlopen (het bewijs gaat met inductie naar het aantal pijlen op het pad van  $j$  naar 1).

Door  $(p, q)$  uit  $T$  te verwijderen valt de boom uiteen in twee delen:  $T_1$  (met knooppunt 1) en  $T_2$ . De pijl  $(p, q)$  verbindt  $T_1$  met  $T_2$ . Omdat  $T - (p, q) + (k, l)$  weer een opspannende boom is, verbindt  $(k, l)$  ook  $T_1$  met  $T_2$ . Dit betekent dat in de nieuwe duale oplossing, zeg  $u'$ , alle componenten uit  $T_1$  onveranderd zijn en dat de componenten van  $T_2$  met een constante zijn veranderd.

Als  $k \in T_1$  en  $l \in T_2$ , dan kunnen we schrijven:

$$u'_l = u'_k - c_{kl} = u_k - c_{kl} = u_l - \bar{c}_{kl}, \text{ dus alle } u\text{-getallen in } T_2 \text{ gaan omhoog met } -\bar{c}_{kl}.$$

Als  $l \in T_1$  en  $k \in T_2$ , dan kunnen we schrijven:

$$u'_k = u'_l + c_{kl} = u_l + c_{kl} = u_k + \bar{c}_{kl}, \text{ dus alle } u\text{-getallen in } T_2 \text{ gaan omlaag met } -\bar{c}_{kl}.$$

Voor de nieuwe  $\bar{c}$ -getallen, noteer deze met  $\bar{c}'$ , geldt op grond van het bovenstaande het volgende:

$$\bar{c}'_{ij} = \begin{cases} \bar{c}_{ij} - \bar{c}_{kl} & \text{als } i \in T_1, j \in T_2, k \in T_1, l \in T_2 \text{ of als } i \in T_2, j \in T_1, k \in T_2, l \in T_1 \\ \bar{c}_{ij} + \bar{c}_{kl} & \text{als } i \in T_1, j \in T_2, k \in T_2, l \in T_1 \text{ of als } i \in T_2, j \in T_1, k \in T_1, l \in T_2 \\ \bar{c}_{ij} & \text{in alle andere gevallen} \end{cases}$$

*Algoritme 5.3: Algoritme voor een minimale kosten stroom in een netwerk zonder capaciteiten.*

1. a. Start met een toelaatbare basisstroom  $x$  bij opspannende boom  $T$ .  
 b. Bepaal de bij  $x$  behorende duale oplossing  $u$  m.b.v. Algoritme 5.2 en bereken de bijbehorende gereduceerde kosten  $\bar{c}_{ij}$ ,  $(i, j) \notin T$ , via (5.9).
2. Als  $\bar{c}_{ij} \geq 0$  voor alle  $(i, j) \notin T$ :  $x$  is optimaal voor (5.1) en  $u$  optimaal voor (5.8) en stop.  
 anders: ga naar stap 3.
3. a. Kies een  $(k, l)$  met  $\bar{c}_{kl} < 0$ .  
 b. Bepaal de kring  $C$  die ontstaat door  $(k, l)$  aan  $T$  toe te voegen en doorloop  $C$  in de richting van  $(k, l)$ .  
 c. Laat  $I_1 = \{(i, j) \mid (i, j) \text{ wordt in } C \text{ in de goede richting doorlopen}\}$  en  $I_2 = \{(i, j) \mid (i, j) \text{ wordt in } C \text{ in de tegengestelde richting doorlopen}\}$ .  
 d. Als  $I_2 = \emptyset$ , dan heeft het probleem een oneindige oplossing en stop;  
 anders:  $\Delta = \min\{x_{ij} \mid (i, j) \in I_2\}$  en neem  $(p, q) \in I_2$  zdd.  $x_{pq} = \Delta$ .
4. a.  $x_{ij} := \begin{cases} x_{ij} + \Delta, & (i, j) \in I_1 \\ x_{ij} - \Delta, & (i, j) \in I_2 \\ x_{ij}, & (i, j) \notin I_1 \cup I_2 \end{cases}$   
 b. Bepaal in  $T - (p, q)$  de deelbomen  $T_1$  (met knooppunt 1) en  $T_2$ .  
 c. Als  $k \in T_1$  en  $l \in T_2$ :  
 (i)  $u_i := u_i - \bar{c}_{kl}$ ,  $i \in T_2$ ;  
 (ii)  $\bar{c}_{ij} := \bar{c}_{ij} - \bar{c}_{kl}$ ,  $i \in T_1, j \in T_2$ ;  
 (iii)  $\bar{c}_{ij} := \bar{c}_{ij} + \bar{c}_{kl}$ ,  $i \in T_2, j \in T_1$ .  
 Als  $k \in T_2$  en  $l \in T_1$ :  
 (i)  $u_i := u_i + \bar{c}_{kl}$ ,  $i \in T_2$ ;  
 (ii)  $\bar{c}_{ij} := \bar{c}_{ij} + \bar{c}_{kl}$ ,  $i \in T_1, j \in T_2$ ;  
 (iii)  $\bar{c}_{ij} := \bar{c}_{ij} - \bar{c}_{kl}$ ,  $i \in T_2, j \in T_1$ .  
 d.  $T := T - (p, q) + (k, l)$ .  
 e. Ga naar stap 2.

### Opmerkingen

1. Wat te doen als er niet direct een toelaatbare basisstroom ter beschikking is? In dat geval beschouwen we een modificatie van het probleem. Voeg een extra knooppunt 0 toe met  $s_0 = 0$  en met pijlen  $(i, 0)$  als  $s_i \geq 0$  en pijlen  $(0, i)$  als  $s_i < 0$ .  
 Neem nu als eerste opspannende boom  $T = \{(i, 0) \mid s_i \geq 0\} \cup \{(0, i) \mid s_i < 0\}$ .  
 Voor de bijbehorende basisstroom  $x$  geldt:  $x_{i0} = s_i$  ( $i$  zdd.  $s_i \geq 0$ ),  $x_{0i} = -s_i$  ( $i$  zdd.  $s_i < 0$ ) en  $x_{ij} = 0$  voor alle  $(i, j)$  die niet incident met 0 zijn. Dit is dus een toelaatbare basisstroom. Door alle pijlen die incident met 0 zijn de kosten  $M$  te geven, met  $M$  een zeer groot getal, wordt bereikt dat geen van deze pijlen in een optimale oplossing voorkomt (is dit toch het geval, dan is het probleem ontoelaatbaar). In Opgave 5.6 wordt ingegaan op de vraag welke waarde voor  $M$  moet worden gekozen.
2. In stap 3b moet de kring  $C$  worden gevonden die ontstaat door  $(k, l)$  aan  $T$  toe te voegen.

Dit kan m.b.v. de informatie die we met het voorwaarts zoeken verkrijgen. Loop in  $T$  zowel vanuit  $k$  als  $l$  terug naar de wortel 1 via de voorgangers. In omgekeerde volgorde geeft dit twee ketens:  $P_k$  van 1 naar  $k$  en  $P_l$  van 1 naar  $l$ . Deze ketens hebben een gemeenschappelijk beginstuk (in ieder geval punt 1), zeg  $P_{kl}$ . Nu geldt:  $C = P_k \cup P_l - P_{kl} + (k, l)$ .

Aldus kan de bepaling van  $C$  in  $\mathcal{O}(n)$  worden uitgevoerd. Eén stap van het algoritme is dus in  $\mathcal{O}(m)$  uit te voeren.

3. Als iedere iteratie niet-gedegeneerd is, dan is hiermee ook de eindigheid gegarandeerd.

In het algemeen is degeneratie echter niet te voorkomen. Gelukkig is voor het degeneratie probleem een oplossing door te werken met zogenaamde *streng toelaatbare bomen*. We komen hier later in deze paragraaf, na het volgende voorbeeld, op terug.

### Voorbeeld 5.1 (vervolg)

Hieronder geven we alleen waarden aan die veranderen.

*Iteratie 1:*

$\bar{c}_{14} = -2$ ;  $\bar{c}_{24} = 4$ ;  $\bar{c}_{53} = 2$ ;  $\bar{c}_{54} = 2$ ;  $\bar{c}_{57} = -1$ . Neem  $(k, l) = (1, 4)$ . Dan is  $C = [1, 4, 3, 2, 1]$ .

$I_1 = \{(1, 4), (4, 3)\}$ ;  $I_2 = \{(2, 3), (1, 2)\}$ ;  $\Delta = 2$ ;  $(p, q) = (2, 3)$ .

$x_{14} = 2$ ;  $x_{43} = 4$ ;  $x_{23} = 0$ ;  $x_{12} = 0$ .

$T_1 = \{1, 2\}$ ;  $T_2 = \{3, 4, 5, 6, 7, 8\}$ ;  $u_3 = -1$ ;  $u_4 = 1$ ;  $u_5 = 1$ ;  $u_6 = -2$ ;  $u_7 = -2$ ;  $u_8 = -3$ ;

$\bar{c}_{14} = 0$ ;  $\bar{c}_{23} = 2$ ;  $\bar{c}_{24} = 6$ ;  $T = \{(1, 2), (1, 4), (4, 3), (3, 8), (8, 6), (5, 6), (6, 7)\}$ .

*Iteratie 2:*

$(k, l) = (5, 7)$ . Dan is  $C = [5, 7, 6, 5]$ ;  $I_1 = \{(5, 7)\}$ ;  $I_2 = \{(6, 7), (5, 6)\}$ .

$\Delta = 0$ , dus  $x$  verandert niet.  $(p, q) = (6, 7)$ ;  $T_1 = \{1, 2, 3, 4, 5, 6, 8\}$ ;  $T_2 = \{7\}$ .

$u_7 = -1$ ;  $\bar{c}_{67} = 1$ ;  $\bar{c}_{57} = 0$ .  $T = \{(1, 2), (1, 4), (4, 3), (3, 8), (8, 6), (5, 6), (5, 7)\}$ .

*Iteratie 3:*

Alle  $\bar{c}$ -getallen zijn niet-negatief. De optimale oplossing van (5.1) is  $x_{12} = 0$ ;  $x_{14} = 2$ ;  $x_{23} = 0$ ;  $x_{24} = 0$ ;  $x_{38} = 2$ ;  $x_{43} = 4$ ;  $x_{53} = 0$ ;  $x_{54} = 0$ ;  $x_{56} = 1$ ;  $x_{57} = 0$ ;  $x_{67} = 0$ ;  $x_{86} = 1$  en de optimale oplossing van het duale probleem (5.8) is  $u_1 = 0$ ;  $u_2 = -2$ ;  $u_3 = -1$ ;  $u_4 = 1$ ;  $u_5 = 1$ ;  $u_6 = -2$ ;  $u_7 = -1$ ;  $u_8 = -3$ .

Zij  $T$  een opspannende boom met wortel 1 en met bijbehorende toelaatbare basisstroom  $x$ . Een pijl  $(i, j) \in T$  heet *naar de wortel toe gericht* als het unieke pad  $P_i$  in  $T$  van  $i$  naar 1 begint met de pijl  $(i, j)$  in de goede richting te doorlopen.  $T$  heet *streng toelaatbaar* als voor iedere  $(i, j) \in T$  met stroom 0 dit een naar de wortel toe gerichte pijl is. In een streng toelaatbare opspannende boom kan dus vanuit ieder knooppunt een positieve stroom naar de wortel worden gestuurd zonder de niet-negativiteit aan te tasten.

### Opmerkingen

1. De hierboven in een eerdere opmerking geconstrueerde eerste toelaatbare basisstroom hoort bij een opspannende boom die streng toelaatbaar is (neem dan als wortel knooppunt 0).
2. Een niet-gedegeneerde toelaatbare basisstroom hoort bij een opspannende boom die streng toelaatbaar is.

3. Een niet-gedegeneerde opspannende toelaatbare boom levert in de volgende iteratie een gedegeneerde basisstroom op d.e.s.d. de uittredende pijl  $(p, q)$  niet uniek is.

We zullen nu gaan aantonen dat voor een speciale keuze van de uittredende pijl  $(p, q)$  alle opspannende bomen van het algoritme streng toelaatbaar en dat daaruit volgt dat het algoritme eindig is omdat alle opspannende bomen verschillend zijn.

Als  $(k, l)$  de intredende pijl is en de uittredende pijl  $(p, q)$  niet uniek is, dan wordt deze als volgt bepaald:

Beschouw de kring  $C$  die in  $T$  ontstaat door  $(k, l)$  aan  $T$  toe te voegen. We hebben eerder gezien dat  $C = P_k \cup P_l - P_{kl} + (k, l)$  met  $P_k$  en  $P_l$  de ketens van 1 naar  $k$  resp.  $l$  in  $T$  en  $P_{kl}$  het gemeenschappelijk beginstuk van beide ketens. Laat  $w$  het laatste knooppunt van  $P_{kl}$  zijn. We noemen  $w$  de *top* van  $C$ . We kiezen nu als uittredende pijl  $(p, q)$  de laatste blokkerende pijl van  $I_2$  (d.w.z. een pijl met stroom gelijk aan  $\Delta$ ), als we  $C$  vanuit  $w$  rondlopen in de richting van  $(k, l)$ .

**Stelling 5.1** *Met bovenstaande keuze voor de uittredende pijl  $(p, q)$  zijn de opspannende bomen die in het algoritme voorkomen steeds streng toelaatbaar.*

### Bewijs

We passen inductie toe. We hebben gezien dat de eerste opspannende boom streng toelaatbaar kan worden gekozen. Het is dus voldoende om te bewijzen dat in een nieuwe boom nog steeds vanuit ieder knooppunt positieve stroom naar de wortel kan worden gestuurd. Dit is het geval als in de nieuwe boom van ieder knooppunt van de kring  $C$  positieve stroom naar de top  $w$  kan worden gestuurd (de rest van het netwerk blijft immers ongewijzigd).

Zij  $C = \{w, \dots, q, p, \dots, w\}$ . Omdat  $(p, q)$  de laatste blokkerende pijl is, kan er van  $\{p, \dots, w\}$  in de nieuwe boom een positieve stroom naar  $w$  worden gestuurd. We moeten dus nog aantonen dat er van  $\{q, \dots, w\}$  in de nieuwe boom ook positieve stroom naar  $w$  kan worden gestuurd. Als  $\Delta > 0$ , dan kan over  $C$  een positieve stroom in de richting van  $(k, l)$  rond worden gestuurd, dus ook terug. Resteert nog het geval dat  $\Delta = 0$ , d.w.z.  $x_{pq} = 0$  en in dat geval verandert de stroom niet.

Omdat van  $q$  geen positieve stroom naar  $p$  kan worden gestuurd, moet in de oude boom een positieve stroom vanuit  $q$  niet via  $\{q, p, \dots, w\}$ , maar de andere kant op, via  $\{q, \dots, w\}$  gaan. Op dit stuk van  $C$  kan  $(k, l)$  dus niet liggen (want dit stuk moet tot de oude boom behoren), d.w.z.  $(k, l)$  ligt op  $\{q, p, \dots, w\}$ . Het stuk  $\{q, \dots, w\}$  was ook in de oude boom aanwezig en er kan dus een positieve stroom vanuit deze knooppunten naar de wortel worden gestuurd. ■

**Stelling 5.2** *Met bovenstaande keuze voor de uittredende pijl zijn alle opspannende bomen verschillend en dus is het algoritme eindig.*

### Bewijs

Zij  $T$  een opspannende boom die streng toelaatbaar is en veronderstel dat  $T$  in het algoritme

wordt opgevolgd door  $T'$  met  $T' = T + (k, l) - (p, q)$ . Associeer met een opspannende boom twee getallen:  $c(T) = \sum_{(i,j) \in A} c_{ij} x_{ij}$  en  $u(T) = \sum_{i \in V} u_i$ . We zullen nu aantonen dat geldt: ofwel  $c(T') < c(T)$  ofwel  $c(T') = c(T)$  en  $u(T') < u(T)$ . Een opspannende boom genereert éénduidig een  $x$  en een  $u$  (met als interpretatie de lengtefunctie), zodat als het bovenstaande is bewezen inderdaad alle bomen verschillend zijn.

Als  $\Delta \neq 0$ , dan volgt uit de theorie van de lineaire programmering dat de waarde van de doel-functie omlaag gaat, d.w.z.  $c(T) < c(T')$ .

Beschouw tenslotte het geval dat  $\Delta = 0$ . Dan verandert  $x$  niet, zodat  $c(T') = c(T)$  en  $x_{kl} = 0$ . Omdat  $T'$  weer streng toelaatbaar is, gaat in  $T'$  de keten van  $l$  naar de wortel niet via  $(k, l)$ . Dit houdt in dat  $l \in T_1$  en  $k \in T_2$ , zodat  $u(T') = u(T) + \bar{c}_{kl} \cdot |T_2| < u(T)$ , omdat  $\bar{c}_{kl} < 0$ . ■

## 5.4 Problemen met onder- en bovengrenzen

Beschouw het minimale kostenprobleem in de gedaante (5.1) en veronderstel dat  $a_{ij} < b_{ij}$  voor alle  $(i, j) \in A$  (als  $a_{ij} = b_{ij}$ , dan kan de pijl  $(i, j)$  worden weggelaten en de waarden van  $s_i$  en  $s_j$  met  $a_{ij}$  verlaagd resp. verhoogd). Het duale probleem van (5.1) luidt:

$$\max \left\{ \sum_{i=1}^n s_i u_i + \sum_{(i,j)} a_{ij} v_{ij} - \sum_{(i,j)} b_{ij} w_{ij} \mid \begin{array}{l} u_i - u_j + v_{ij} - w_{ij} = c_{ij}, \quad (i, j) \in A \\ v_{ij}, w_{ij} \geq 0, \quad (i, j) \in A \end{array} \right\} \quad (5.10)$$

We werken ook in deze paragraaf met opspannende bomen  $T$  en daarbij behorende (toelaatbare) stromen  $x$ . Voor deze stromen  $x$  geldt dat als  $a_{ij} < x_{ij} < b_{ij}$ , dan  $(i, j) \in T$ . De pijlen van  $A$  worden verdeeld in  $(T, L, U)$  met  $T$  de opspannende boom,  $L$  en  $U$  de pijlen die niet tot de boom behoren en op hun ondergrens resp. bovengrens staan. Een dergelijk drietal  $(T, L, U)$  heet een *boomstructuur*.

Bij een boomstructuur berekenen we de bijbehorende stroom als volgt:

1.  $x_{ij} = a_{ij}$ ,  $(i, j) \in L$ ;  $x_{ij} = b_{ij}$ ,  $(i, j) \in U$ .
2.  $s'_i = s_i - \sum_{(i,j) \in L} a_{ij} - \sum_{(i,j) \in U} b_{ij} + \sum_{(j,i) \in L} a_{ji} + \sum_{(j,i) \in U} b_{ji}$ ,  $i \in V$ .
3. Los op:  $Bx_T = s'$  met  $B$  de matrix van de kolommen van  $N$  behorende bij opspannende boom  $T$  en  $x_T$  het bijbehorende deel van de  $x$ -vector.

Met dezelfde argumentatie als in Lemma 5.1 kan worden aangetoond dat  $x_T$  uniek is. Als  $a_{ij} \leq x_{ij} \leq b_{ij}$  voor alle  $(i, j) \in T$ , dan heten de boomstructuur en de stroom  $x$  *toelaatbaar*. Een toelaatbare boomstructuur en stroom heten *streng toelaatbaar* als voor iedere  $(i, j) \in T$  met  $x_{ij} = a_{ij}$  dit een naar de wortel toe gerichte pijl is op het unieke pad  $P_i$  (dit pad verbindt in  $T$  de wortel met  $i$ ), en voor iedere  $(i, j) \in T$  met  $x_{ij} = b_{ij}$  dit een van de wortel af gerichte pijl is op het pad  $P_j$ . Ook nu kan in een streng toelaatbare opspannende boom vanuit ieder knooppunt stroom naar de wortel worden gestuurd zonder de toelaatbaarheid aan te tasten.



### Voorbeeld 5.2

Beschouw nevenstaand netwerk. De nettoproducties zijn:

$$s_1 = 9, s_2 = s_3 = s_4 = s_5 = 0, s_6 = -9.$$

Alle ondergrenzen zijn 0 en voor de bovengrenzen geldt:

$$b_{12} = 8, b_{13} = 3, b_{23} = 3, b_{24} = 7, b_{25} = 2, b_{35} = 3,$$

$$b_{54} = 4, b_{46} = 5 \text{ en } b_{56} = 6.$$

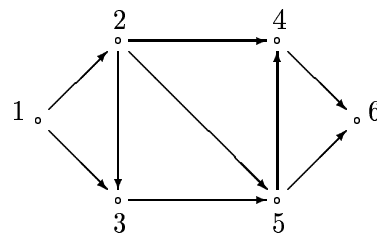
De kosten zijn:  $c_{12} = 3, c_{13} = 2, c_{23} = 2, c_{24} = 5, c_{25} = 2, c_{35} = 4, c_{54} = 5, c_{46} = 3$  en  $c_{56} = 4$ .

Neem  $T = \{(1, 2), (1, 3), (2, 4), (2, 5), (5, 6)\}$ ,  $L = \{(2, 3), (5, 4)\}$  en  $U = \{(3, 5), (4, 6)\}$ .

$$s'_1 = 9, s'_2 = 0, s'_3 = -3, s'_4 = -5, s'_5 = 3, s'_6 = -4.$$

Voor de boomoplossing  $x_T$  geldt:  $x_{12} = 6, x_{13} = 3, x_{24} = 5, x_{25} = 1$  en  $x_{56} = 4$ .

Merk op dat deze oplossing streng toelaatbaar is.



Ook in dit geval kunnen we weer starten met een boomstructuur en stroom die streng toelaatbaar zijn. We voegen daartoe weer een extra knooppunt 0 toe en verbinden dit met alle andere knooppunten (in welke richting: zie verder) met ondergrenzen 0, bovengrenzen  $\infty$  en kosten  $M$  (voldoende groot getal). We nemen  $L = A$  en  $U = \emptyset$ . Voor de aangepaste  $s$ -getallen geldt:  $s'_0 = 0$  en  $s'_i = s_i - \sum_{(i,j)} a_{ij} + \sum_{(j,i)} a_{ji}$ ,  $1 \leq i \leq n$ . Richt de pijlen van  $i$  naar 0 als  $s'_i \geq 0$  en van 0 naar  $i$  als  $s'_i < 0$ . Voor de bijbehorende stroom geldt:  $x_{0i} = -s'_i$  en  $x_{i0} = s'_i$ . Deze stroom is streng toelaatbaar.

Uitgaande van de boomstructuur  $(T, L, U)$  met bijbehorende stroom  $x$  bepalen we de bijbehorende duale oplossing. Voor  $(i, j) \in T$  geldt  $v_{ij} = w_{ij} = 0$ , zodat  $u$  bepaald kan worden met Algoritme 5.2. Uit de orthogonaliteitsrelaties  $v_{ij} \cdot (x_{ij} - a_{ij}) = w_{ij} \cdot (b_{ij} - x_{ij}) = 0$  voor alle  $(i, j) \in A$ , volgt: als  $(i, j) \in L$ , dan is  $w_{ij} = 0$ , zodat voor de optimaliteit moet gelden  $v_{ij} \geq 0$ , d.w.z.  $\bar{c}_{ij} = c_{ij} - u_i + u_j \geq 0$ ; als  $(i, j) \in U$ , dan is  $v_{ij} = 0$ , zodat voor de optimaliteit moet gelden  $w_{ij} \geq 0$ , d.w.z.  $\bar{c}_{ij} = c_{ij} - u_i + u_j \leq 0$ .

Een intredende pijl  $(k, l)$  kan dus worden gekozen uit  $L^-$  of  $U^+$ , waarbij  $L^- = \{(i, j) \in L \mid \bar{c}_{ij} < 0\}$  en  $U^+ = \{(i, j) \in U \mid \bar{c}_{ij} > 0\}$  (als  $L^- \cup U^+ = \emptyset$ , dan is  $x$  optimaal voor (5.1) en  $u$  (met bijbehorende  $v$  en  $w$ ) optimaal voor (5.10).

Als  $(k, l)$  als intredende pijl is gekozen, dan bepalen we weer de unieke kring  $C$  met top  $w$ . De kring  $C$  wordt in de richting  $(k, l)$  doorlopen als  $(k, l) \in L$  en in de richting  $(l, k)$  als  $(k, l) \in U$ . Laat  $\Delta = \min(\Delta_1, \Delta_2)$  met  $\Delta_1 = \min\{x_{ij} - a_{ij} \mid (i, j) \text{ wordt in } C \text{ in de verkeerde richting doorlopen}\}$  en  $\Delta_2 = \min\{b_{ij} - x_{ij} \mid (i, j) \text{ wordt in } C \text{ in de goede richting doorlopen}\}$ . Als  $\Delta = \infty$  (wat het geval is als alle pijlen van  $C$  in de goede richting worden doorlopen en de bovengrens van al deze pijlen  $\infty$  is), dan is er een *oneindige oplossing*.

Als uittredende pijl  $(p, q)$  nemen we weer de laatst blokkerende pijl als we  $C$  vanaf de top  $w$  doorlopen. Vervolgens wordt de stroom aangepast over  $C$  en worden de  $u$ - en  $\bar{c}$ -getallen aangepast. De eindigheid volgt weer uit dezelfde overwegingen als in de vorige paragraaf het geval was. Het is mogelijk dat  $(k, l) = (p, q)$ , namelijk als de stroom over deze pijl van de ondergrens naar de bovengrens gaat (of omgekeerd).

Het algoritme voor een minimale kostenstroom in een netwerk met onder- en bovengrenzen is als volgt.

*Algoritme 5.4: Algoritme voor een minimale kostenstroom in een netwerk met onder- en bovengrenzen.*

1. a. Start met een streng toelaatbare basisstroom  $x$  bij opspannende boom  $T$ .  
 Laat  $L = \{(i, j) \notin T \mid x_{ij} = a_{ij}\}$  en  $U = \{(i, j) \notin T \mid x_{ij} = b_{ij}\}$ .  
 b. Bepaal de bij  $x$  behorende duale oplossing  $u$  m.b.v. Algoritme 5.2 en bereken de bijbehorende gereduceerde kosten  $\bar{c}_{ij}$ ,  $(i, j) \notin T$ , via (5.9).
2. Als  $\bar{c}_{ij} \geq 0$  voor alle  $(i, j) \in L$  en  $\bar{c}_{ij} \leq 0$  voor alle  $(i, j) \in U$ :  $x$  is optimaal voor (5.1) en  $u$  optimaal voor (5.10) en stop.  
 anders: ga naar stap 3.
3. a. Kies een  $(k, l) \in L$  met  $\bar{c}_{kl} < 0$  of  $(k, l) \in U$  met  $\bar{c}_{kl} > 0$ .  
 b. Bepaal de kring  $C$  die ontstaat door  $(k, l)$  aan  $T$  toe te voegen, bepaal de top  $w$  en doorloop  $C$  in de richting  $(k, l)$  als  $(k, l) \in L$  en in de richting  $(l, k)$  als  $(k, l) \in U$ .  
 c. Laat  $\Delta_1 = \min\{x_{ij} - a_{ij} \mid (i, j) \text{ wordt in } C \text{ in de verkeerde richting doorlopen}\}$  en  $\Delta_2 = \min\{b_{ij} - x_{ij} \mid (i, j) \text{ wordt in } C \text{ in de goede richting doorlopen}\}$ .  
 Neem  $\Delta = \min(\Delta_1, \Delta_2)$ .  
 d. Als  $\Delta = \infty$ , dan heeft het probleem een oneindige oplossing en stop;  
 anders: neem  $(p, q)$  als laatste blokkerende van  $C$ , als we  $C$  vanuit de top  $w$  doorlopen.
4. a.  $x_{ij} := \begin{cases} x_{ij} + \Delta, & (i, j) \in C \text{ en wordt in de goede richting doorlopen} \\ x_{ij} - \Delta, & (i, j) \in C \text{ en wordt in de verkeerde richting doorlopen} \\ x_{ij}, & (i, j) \notin C \end{cases}$   
 b.  $T := T + (k, l) - (p, q)$ ; voeg  $(p, q)$  aan  $L$  toe (als  $(p, q)$  in  $C$  in de verkeerde richting wordt doorlopen) of aan  $U$  (als  $(p, q)$  in  $C$  in de goede richting wordt doorlopen); laat  $(k, l)$  uit  $L$  weg (als  $(k, l)$  tot  $L$  behoorde) of uit  $U$  (als  $(k, l)$  tot  $U$  behoorde).  
 c. Bepaal  $u$  m.b.v. Algoritme 5.2.  
 d. Bepaal  $\bar{c}_{ij}$ ,  $(i, j) \notin T$ .  
 e. Ga naar stap 2.

### Opmerkingen

1. Net als in stap 4c van Algoritme 5.3 kan ook hier de berekening van nieuwe  $u$  en  $\bar{c}$  worden beperkt tot  $T_2$  resp. de pijlen die tussen  $T_1$  en  $T_2$  lopen (zie ook opgave 10).
2. Het is lang onbekend geweest of er een polynomiaal algoritme bestaat voor het minimale kostenstroomprobleem via de netwerk simplex methode. In 1997 is aangetoond<sup>1</sup> dat hiervoor een  $\mathcal{O}(m^2 n^2 \log n)$  implementatie geconstrueerd kan worden.

<sup>1</sup>Orlin, J.B.: *A polynomial time primal network simplex algorithm for minimum cost flows*, Mathematical Programming **78** (1997) 109-129.

### Voorbeeld 5.2 (vervolg)

Hieronder geven we alleen waarden aan die veranderen.

*Iteratie 1:*

$$u_1 = 0, u_2 = -3; u_3 = -2, u_4 = -8, u_5 = -5, u_6 = -9. \bar{c}_{23} = 3; \bar{c}_{35} = 1; \bar{c}_{54} = 2; \bar{c}_{46} = 2.$$

Neem  $(k, l) = (3, 5)$ . Dan is  $C = [5, 3, 1, 2, 5]$ .  $\Delta_1 = 3; \Delta_2 = 1, \Delta = 1. (p, q) = (2, 5)$ .

$x_{35} = 2; x_{13} = 2; x_{12} = 7; x_{25} = 2. T = \{(1, 2), (1, 3), (2, 4), (5, 6), (3, 5)\}$ .  $(2, 5)$  komt in  $U$ .

$$u_1 = 0, u_2 = -3; u_3 = -2, u_4 = -8, u_5 = -6, u_6 = -10. \bar{c}_{23} = 3; \bar{c}_{25} = -1; \bar{c}_{54} = 3; \bar{c}_{46} = 1.$$

*Iteratie 2:*

$(k, l) = (4, 6)$ . Dan is  $C = [6, 4, 2, 1, 3, 5, 6]$ .  $\Delta_1 = 5; \Delta_2 = 1, \Delta = 1. (p, q) = (3, 5)$ .

$x_{46} = 4; x_{24} = 4; x_{12} = 6; x_{13} = 3; x_{35} = 3; x_{56} = 5. T = \{(1, 2), (1, 3), (2, 4), (5, 6), (4, 6)\}$ .  $(3, 5)$  komt in  $U$ .

$$u_1 = 0, u_2 = -3; u_3 = -2, u_4 = -8, u_5 = -7, u_6 = -11. \bar{c}_{23} = 3; \bar{c}_{25} = -2; \bar{c}_{35} = -1; \bar{c}_{54} = 4.$$

*Iteratie 3:*

Alle  $\bar{c}$ -getallen bij pijlen van  $L$  en  $U$  zijn niet-negatief resp. niet-positief. De optimale oplossing van (5.1) is  $x_{12} = 6; x_{13} = 3; x_{23} = 0; x_{24} = 4; x_{25} = 2; x_{35} = 3; x_{46} = 4; x_{54} = 0; x_{56} = 5$  en de optimale oplossing van het duale probleem (5.10) is  $u_1 = 0; u_2 = -3; u_3 = -2; u_4 = -8; u_5 = -7; u_6 = -11$ . Hierbij horen de volgende  $v$ - en  $w$ -waarden:  $v_{12} = x_{12} = 0; v_{13} = w_{13} = 0; v_{23} = 3; w_{23} = 0; v_{24} = w_{24} = 0; v_{25} = 0; w_{25} = 2; v_{35} = 0; w_{35} = 1; v_{46} = w_{46} = 0; v_{54} = 4; w_{54} = 0; v_{56} = w_{56} = 0$ .

## 5.5 Maximale stroom probleem

In paragraaf 5.1 hebben we gezien dat we het maximale stroom probleem krijgen als speciaal geval van het minimale kostenstroom probleem door alle nettoproducties  $s_i$ , ondergrenzen  $a_{ij}$  en kosten  $c_{ij}$  gelijk aan 0 te nemen, de pijl  $(n, 1)$  toe te voegen met ondergrens  $-\infty$ , bovengrens  $\infty$  en kosten -1.

Dit geeft de equivalente formulering:

$$\max \left\{ x_{n1} \mid \begin{array}{l} \sum_j x_{ij} - \sum_j x_{ji} = 0, \quad 1 \leq i \leq n \\ 0 \leq x_{ij} \leq b_{ij}, \quad (i, j) \in A \end{array} \right\} \quad (5.11)$$

met als duaal probleem

$$\min \left\{ \sum_{(i,j)} b_{ij} w_{ij} \mid \begin{array}{l} u_i - u_j + w_{ij} \geq 0, \quad (i, j) \in A \\ -u_1 + u_n = 1 \\ w_{ij} \geq 0, \quad (i, j) \in A \end{array} \right\} \quad (5.12)$$

De beperking  $-u_1 + u_n = 1$  hoort bij de extra pijl  $(n, 1)$ . Met behulp van verschilvariabele  $v_{ij}$  is de ongelijkheid  $u_i - u_j + w_{ij} \geq 0$  equivalent met  $u_i - u_j + w_{ij} - v_{ij} = 0; v_{ij} \geq 0$ . Dit houdt in dat (5.12) vervangen kan worden door:

$$\min \left\{ \sum_{(i,j)} b_{ij} w_{ij} \mid \begin{array}{l} -u_i + u_j + v_{ij} - w_{ij} = 0, \quad (i, j) \in A \\ -u_n + u_1 = -1 \\ v_{ij}, w_{ij} \geq 0, \quad (i, j) \in A \end{array} \right\} \quad (5.13)$$

Vergelijken we (5.13) met (5.10), dan zien we dat de  $u$  in (5.13) overeenkomt met  $-u$  in (5.10). Dit heeft tot gevolg dat voor de  $u$  in (5.13) de interpretatie is dat  $u_i$  de lengte is van het boompad van 1 naar  $i$  in plaats van  $i$  naar 1.

We starten met een toelaatbare opspannende boom  $T$  die de extra pijl  $(n, 1)$  bevat. We zullen laten zien dat  $(n, 1)$  ook in alle volgende bomen voorkomt (iedere toelaatbare stroom met positieve waarde moet de extra pijl  $(n, 1)$  bevatten). Bij een boom  $T$  zullen we met  $T_1$  en  $T_2$  de deelbomen noteren die samen  $T - (n, 1)$  vormen, met knooppunt 1 in  $T_1$  en knooppunt  $n$  in  $T_2$ . Omdat  $u_1 = 0, l_{1n} = -c_{n1} = 1$  en alle andere lengtes 0 zijn, geldt:  $u_i = \begin{cases} 0 & \text{als } i \in T_1 \\ 1 & \text{als } i \in T_2. \end{cases}$

Zij  $(T, L, U)$  een toelaatbare boomstructuur die  $(n, 1)$  bevat met bijbehorende toelaatbare stroom  $x$  en duale oplossing  $(u, v, w)$ . In de vorige paragraaf hebben we gezien dat de oplossingen optimaal zijn als  $L^- \cup U^+ = \emptyset$ , waarbij  $L^- = \{(i, j) \in L \mid u_i - u_j < 0\}$  en  $U^+ = \{(i, j) \in U \mid u_i - u_j > 0\}$ . Uit de bovenstaande waarden voor  $u_i, i \in T_1 \cup T_2$ , volgt dat  $u_i - u_j < 0$  alleen het geval is als  $i \in T_1$  en  $j \in T_2$ , en  $u_i - u_j > 0$  impliceert  $i \in T_2$  en  $j \in T_1$ . We kunnen dus schrijven:  $L^- = \{(i, j) \in L \mid i \in T_1, j \in T_2\}$  en  $U^+ = \{(i, j) \in U \mid i \in T_2, j \in T_1\}$ .

**Lemma 5.2** *Als  $L^- \cup U^+ = \emptyset$ , dan is  $x$  een maximale stroom en  $(T_1, T_2)$  een  $(1, n)$ -snede met minimale capaciteit.*

### Bewijs

Uit bovenstaande afleiding weten we dat als  $L^- \cup U^+ = \emptyset$  de oplossingen  $x$  en  $(u, v, w)$  optimaal zijn voor (5.11) resp. (5.13). Dus  $x$  is een maximale stroom,  $v, w \geq 0$ ,  $x_{n1} = \sum_{(i,j)} b_{ij}w_{ij}$ , en  $v_{ij}(x_{ij} - a_{ij}) = w_{ij}(b_{ij} - x_{ij}) = 0$  voor alle  $(i, j) \in A$ .

Als  $w_{ij} \neq 0$ , d.w.z.  $w_{ij} > 0$ , dan geldt:  $x_{ij} = b_{ij} > a_{ij}$ , dus  $v_{ij} = 0$  en  $0 < w_{ij} = -u_i + u_j$ , dus  $u_i = 0, u_j = 1$  en  $w_{ij} = 1$ . Hieruit volgt: waarde stroom  $= x_{n1} = \sum_{i \in T_1, j \in T_2} b_{ij} = \text{capaciteit snede } (T_1, T_2)$ . Omdat de capaciteit van iedere  $(n, 1)$ -snede minstens de waarde van de maximale stroom is, is  $(T_1, T_2)$  een  $(n, 1)$ -snede met minimale capaciteit. ■

Als  $(k, l) \in L^- \cup U^+$ , dan kan  $(k, l)$  als intredende pijl worden gekozen.  $T \cup (k, l)$  bevat een kring  $C$  die zowel  $T_1$  als  $T_2$  bezoekt en dus de pijl  $(n, 1)$  bevat. We doorlopen  $C$  in de richting van de pijl  $(n, 1)$  (ga dit zelf na). De pijl  $(n, 1)$  heeft bovengrens  $\infty$  en kan dus nooit blokkerend zijn. Alle volgende bomen bevatten dus ook  $(n, 1)$ .

Omdat  $C$  de pijl  $(n, 1)$  bevat, ontdekken we in feite een pad van 1 naar  $n$  en doorover vervoeren we  $\Delta$ . De methode lijkt dus erg op die van Ford en Fulkerson. Alleen is daar altijd  $\Delta > 0$ , terwijl bij de netwerk simplex methode  $\Delta = 0$  kan zijn, waarmee degeneratie en gevaar voor cycling ontstaat. Dit is weer op te lossen met *streng toelaatbare bomen*. Zoals de netwerk simplex methode in de context van het maximale stroom probleem eenvoudiger is dan in het algemene geval (in feite moeten we alleen kijken hoe de splitsing in  $T_1$  en  $T_2$  is), zo is ook het concept van streng toelaatbare bomen eenvoudiger.

We beschouwen hiertoe knooppunt  $n$  als wortel en noemen een toelaatbare boom  $T$  streng toelaatbaar als vanuit iedere  $i \in T_2$  (dus niet voor alle knooppunten) in  $T$  een positieve stroom naar  $n$  kan worden gezonden. Dus iedere pijl  $(i, j) \in T_2$  met  $x_{ij} = 0$  is naar  $n$  toe gericht en iedere pijl  $(i, j) \in T_2$  met  $x_{ij} = b_{ij}$  is van  $n$  af gericht. Merk op dat we altijd met een streng toelaatbare boom kunnen starten door een opspannende boom voor de knooppunten  $\{1, 2, \dots, n-1\}$  te nemen en daaraan de pijl  $(n, 1)$  toe te voegen. Voor deze boom is  $T_2 = \{n\}$ , zodat de voorwaarde geldt. Als uittredende pijl nemen we de laatste blokkerende pijl als we  $C$  beginnend met de pijl  $(n, 1)$  doorlopen. Ga zelf na (zie Opgave 11) dat in opeenvolgende iteraties de eigenschap van strenge toelaatbaarheid behouden blijft.

Veronderstel nu dat we degeneratie hebben, d.w.z.  $\Delta = 0$ . Omdat we in een streng toelaatbare boom altijd een positieve stroom vanuit  $T_2$  naar  $n$  kunnen sturen, heeft de blokkerende pijl beide uiteinden in  $T_1$ , zodat  $T_2$  in de volgende iteratie echt groter wordt (zowel  $k$  als  $l$  behoren er o.a. toe). Het aantal opeenvolgende gedegenerende iteraties is dus eindig, zelfs hoogstens  $n$  (voor het algemene minimale kostenstroom probleem is het aantal opeenvolgende gedegenerende iteraties ook eindig, maar kan wel exponentieel groot zijn). Er kan worden bewezen<sup>2</sup> dat als voor de intredende pijl  $(k, l)$  de pijl van  $L^- \cup U^+$  wordt gekozen die het dichtst bij knooppunt 1 ligt (deze is met zijwaarts zoeken eenvoudig te vinden) dat het aantal iteraties hoogstens  $nm$  is en dat de totale complexiteit  $\mathcal{O}(n^2m)$  is.

*Algoritme 5.5: Algoritme voor een maximale stroom en snede met minimale capaciteit in een netwerk met capaciteiten.*

1. Start met een streng toelaatbare boom  $T$  die  $(n, 1)$  bevat (neem bijv. een opspannende boom in  $\{1, 2, \dots, n-1\}$  plus de pijl  $(n, 1)$ ), neem  $U = \emptyset$ ,  $L = \{(i, j) \mid (i, j) \notin T\}$ , en  $x = 0$ .
2. a. Bepaal de knooppunten van de deelbomen  $T_1$  (met  $1 \in T_1$ ) en  $T_2$  die uit  $T$  ontstaan door  $(n, 1)$  weg te laten.  
b. Bepaal  $L^- = \{(i, j) \in L \mid i \in T_1, j \in T_2\}$  en  $U^+ = \{(i, j) \in U \mid i \in T_2, j \in T_1\}$ .  
c. Als  $L^- \cup U^+ = \emptyset$ :  $x$  is een maximale stroom en  $(T_1, T_2)$  een  $(n, 1)$ -snede met minimale capaciteit (stop).
3. a. Kies een  $(k, l) \in L^- \cup U^+$  die het dichtst bij knooppunt 1 ligt (via zijwaarts zoeken).  
b. Bepaal de kring  $C$  die ontstaat door  $(k, l)$  aan  $T$  toe te voegen en doorloop  $C$  vanuit de pijl  $(n, 1)$ .  
c. Laat  $\Delta_1 = \min\{x_{ij} \mid (i, j) \text{ wordt in } C \text{ in de verkeerde richting doorlopen}\}$  en  $\Delta_2 = \min\{b_{ij} - x_{ij} \mid (i, j) \text{ wordt in } C \text{ in de goede richting doorlopen}\}$ .  
Neem  $\Delta = \min(\Delta_1, \Delta_2)$  en neem  $(p, q)$  als laatste blokkerende van  $C$ .
4. a.  $x_{ij} := \begin{cases} x_{ij} + \Delta, & (i, j) \in C \text{ en wordt in de goede richting doorlopen} \\ x_{ij} - \Delta, & (i, j) \in C \text{ en wordt in de verkeerde richting doorlopen} \\ x_{ij}, & (i, j) \notin C \end{cases}$

---

<sup>2</sup>Goldfarb, D. and J. Hao: *A primal simplex algorithm that solves the maximum flow problem in at most  $nm$  pivots and  $\mathcal{O}(n^2m)$  time*, Mathematical Programming 47 (1990) 353-366.

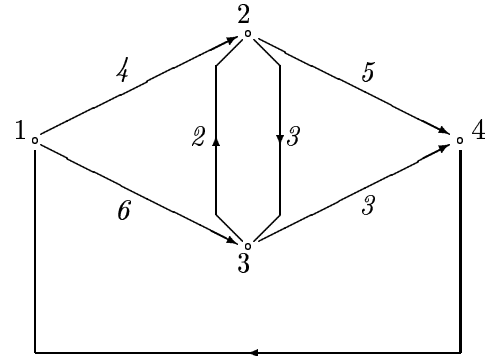
- b.  $T := T + (k, l) - (p, q)$ ; voeg  $(p, q)$  aan  $L$  toe (als  $(p, q)$  in  $C$  in de verkeerde richting wordt doorlopen) of aan  $U$  (als  $(p, q)$  in  $C$  in de goede richting wordt doorlopen); laat  $(k, l)$  uit  $L$  weg (als  $(k, l)$  tot  $L$  behoorde) of uit  $U$  (als  $(k, l)$  tot  $U$  behoorde).
- c. Ga naar stap 2.

### Voorbeeld 5.3

Hiernaast staat het netwerk met bij iedere pijl  $(i, j)$  de capaciteit.

#### Iteratie 1

1. Start met  $T = \{(1, 2), (1, 3), (4, 1)\}$ ;  $U = \emptyset$ ;  
 $L = \{(2, 3), (3, 2), (2, 4), (3, 4)\}$ ;  $x = 0$ .
2.  $T_1 = \{1, 2, 3\}$ ;  $T_2 = \{4\}$ ;  $L^- = \{(2, 4), (3, 4)\}$ ;  $U^+ = \emptyset$ .
3.  $(k, l) = (2, 4)$ ;  $C = [4, 1, 2, 4]$ ;  $\Delta_1 = \infty$ ;  $\Delta_2 = 4$ ;  $\Delta = 4$ .  
 $(p, q) = (1, 2)$ .
4.  $x_{41} = x_{12} = x_{24} = 4$ ;  $T = \{(1, 3), (2, 4), (4, 1)\}$ ;  
 $U = \{(1, 2)\}$ ;  $L = \{(2, 3), (3, 2), (3, 4)\}$ .



#### Iteratie 2

2.  $T_1 = \{1, 3\}$ ;  $T_2 = \{2, 4\}$ ;  $L^- = \{(3, 2), (3, 4)\}$ ;  $U^+ = \emptyset$ .
3.  $(k, l) = (3, 2)$ ;  $C = [4, 1, 3, 2, 4]$ ;  $\Delta_1 = \infty$ ;  $\Delta_2 = 1$ ;  $\Delta = 1$ ;  $(p, q) = (2, 4)$ .
4.  $x_{41} = 5$ ,  $x_{13} = 1$ ,  $x_{32} = 1$ ,  $x_{24} = 5$ ;  $T = \{(1, 3), (3, 2), (4, 1)\}$ ;  $U = \{(1, 2), (2, 4)\}$ ;  
 $L = \{(2, 3), (3, 4)\}$ .

#### Iteratie 3

2.  $T_1 = \{1, 2, 3\}$ ;  $T_2 = \{4\}$ ;  $L^- = \{(3, 4)\}$ ;  $U^+ = \emptyset$ .
3.  $(k, l) = (3, 4)$ ;  $C = [4, 1, 3, 4]$ ;  $\Delta_1 = \infty$ ;  $\Delta_2 = 3$ ;  $\Delta = 3$ ;  $(p, q) = (3, 4)$ .
4.  $x_{41} = 8$ ,  $x_{13} = 4$ ,  $x_{34} = 3$ ;  $T = \{(1, 3), (3, 2), (4, 1)\}$ ;  $U = \{(1, 2), (2, 4), (3, 4)\}$ ;  $L = \{(2, 3)\}$ .

#### Iteratie 4

2.  $T_1 = \{1, 2, 3\}$ ;  $T_2 = \{4\}$ ;  $L^- = U^+ = \emptyset$ . De stroom  $x$ , waarbij  $x_{12} = 4$ ,  $x_{13} = 4$ ,  $x_{24} = 5$ ,  $x_{32} = 1$ ,  $x_{34} = 3$ , is optimaal met waarde 8. De snede  $(T_1 = \{1, 2, 3\}, T_2 = \{4\})$  is een snede met minimale capaciteit; deze capaciteit is ook 8.

## 5.6 Opgaven

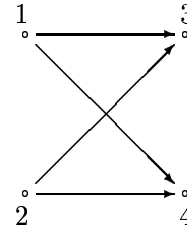
### Opgave 5.1

De knooppunten van de verz.  $V^+ = \{i \in V \mid s_i > 0\}$  en  $V^- = \{i \in V \mid s_i < 0\}$  heten de *bronnen* resp. de *putten*.

- a. Toon aan dat ieder probleem van de vorm (5.1) met meer dan één bron en/of meer dan één put kan worden getransformeerd tot een equivalent probleem met slechts één bron en één put.
- b. Toon aan dat ieder probleem van de vorm (5.1) kan worden getransformeerd tot een equivalent probleem met  $s_i = 0$  voor alle  $i$ .

**Opgave 5.2**

Hiernaast staat een netwerk. De nettoproducties zijn:  
 $s_1 = 2$ ,  $s_2 = 3$ ,  $s_3 = -3$  en  $s_4 = -2$ . Bepaal voor alle  
 (vier) opspannende bomen van deze graaf de bijbe-  
 horende basisstroom.



**Opgave 5.3**

Zij  $T$  een opspannende boom en  $B$  de  $(n - 1) \times (n - 1)$  matrix die uit de incidentiematrix  $N$  ontstaat door de  $n - 1$  kolommen behorend bij boom  $T$  te nemen en de eerste rij weg te laten. Beschouw het stelsel  $Bx = s'$ , met  $x$  de  $(n - 1)$ -dimensionale vector die behoort bij  $T$  en waarbij  $s'$  uit  $s$  wordt verkregen door de eerste rij weg te laten.

- Toon aan dat, na henummering van de rijen en kolommen van  $B$ ,  $B$  kan worden getransformeed tot een bovendriehoeksmatrix, zeg  $B^*$ , waardoor het stelsel - na dezelfde henummering van  $x$  en  $s'$  - overgaat in het stelsel  $B^*x^* = s^*$  met  $B^*$  een bovendriehoeksmatrix.
- Laat zien dat deze henummering en het oplossen van  $B^*x^* = s^*$  met een  $\mathcal{O}(n)$  algoritme kan worden uitgevoerd.
- Pas bovenstaande procedure toe op voorbeeld 5.1 met de daarin aangegeven boom  $T$ .

**Opgave 5.4**

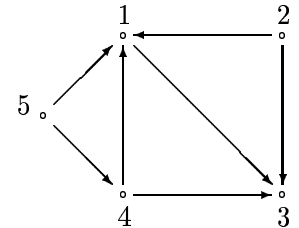
Veronderstel dat  $x$  voldoet aan  $Nx = s$ ;  $x \geq 0$ . Beschrijf hoe  $x$  gewijzigd kan worden in een basisoplossing  $x'$ , terwijl geldt dat  $x'_{ij} = 0$  voor alle  $(i, j)$  waarvoor ook  $x_{ij} = 0$ .

**Opgave 5.5**

Beschouw het minimale kostenstroomprobleem zonder capaciteiten in nevenstaand netwerk. De nettoproducties zijn:

$s_1 = 0$ ,  $s_2 = 3$ ,  $s_3 = -7$ ,  $s_4 = 1$  en  $s_5 = 3$ . Neem alle kosten  $c_{ij} = 1$ .

- Formuleer het bijbehorende LP-probleem en het duale probleem.
- Pas de netwerk simplex methode toe om de oplossing van beide problemen te bepalen. Begin met  $T = \{(2, 1), (2, 3), (4, 3), (5, 4)\}$ .



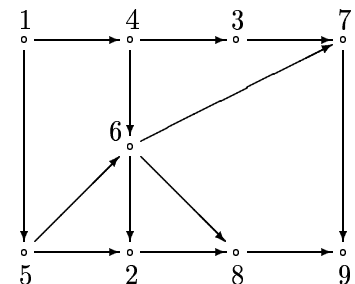
**Opgave 5.6**

Toon aan dat  $M > \frac{1}{2}(n - 1)^2 \cdot \max_i |s_i| \cdot \max_{(i,j)} |c_{ij}|$  voldoet aan de eis dat een pijl  $(0, i)$  of  $(i, 0)$  uit opmerking 1 van paragraaf 5.3 stroom 0 heeft in een optimale oplossing, tenzij het probleem ontoelaatbaar is.

**Opgave 5.7**

Beschouw nevenstaand netwerk. De nettoproducties zijn:  
 $s_1 = 3$ ,  $s_2 = -1$ ,  $s_3 = 0$ ,  $s_4 = 0$ ,  $s_5 = 0$ ,  $s_6 = 0$ ,  $s_7 = 0$ ,  $s_8 = 0$ ,  
 en  $s_9 = -2$ . Voor de kosten geldt:  $c_{14} = 1$ ,  $c_{15} = 2$ ,  $c_{28} = 1$ ,  
 $c_{37} = 1$ ,  $c_{43} = 3$ ,  $c_{46} = 4$ ,  $c_{52} = 4$ ,  $c_{56} = 1$ ,  $c_{62} = 1$ ,  $c_{67} = 2$ ,  
 $c_{68} = 2$ ,  $c_{79} = 3$  en  $c_{89} = 2$ .

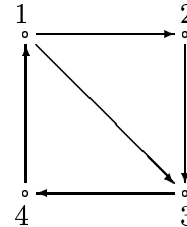
- Laat  $T = \{(1, 5), (5, 6), (6, 2), (6, 8), (8, 9), (7, 9), (3, 7), (4, 3)\}$ .  
 Bepaal de basisstroom die bij  $T$  behoort.



- b. Bepaal voor iedere pijl die niet tot  $T$  behoort de gereduceerde kosten  $\bar{c}_{ij}$ .
- c. Is deze oplossing optimaal?
- d. Met hoeveel moet  $c_{56}$  ten minste worden verhoogd om een andere optimale oplossing te krijgen?
- e. Als we  $s_1$  vervangen door  $3 + \delta$  en  $s_9$  door  $-2 - \delta$  met hoeveel verandert dan de optimale waarde van de doelfunctie?

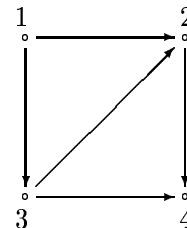
### Opgave 5.8

Hiernaast staat een netwerk. Neem aan dat alle ondergrenzen 0 zijn en alle bovengrenzen  $\infty$ . De kosten zijn:  $c_{12} = 4$ ,  $c_{13} = 1$ ,  $c_{23} = -2$ ,  $c_{34} = 4$ ;  $c_{41} = 0$ . De nettoproducties zijn:  $s_1 = 2$ ,  $s_2 = 3$ ,  $s_3 = -3$  en  $s_4 = -2$ . Los voor deze data het minimale kostenstroom op. Neem als eerste opspannende boom  $T = \{(1, 2), (2, 3), (3, 4)\}$ .



### Opgave 5.9

Hiernaast staat een netwerk. Voor de onder-, bovengrenzen en kosten geldt:  $a_{12} = 0$ ,  $b_{12} = 6$ ,  $c_{12} = 1$ ;  $a_{13} = 0$ ,  $b_{13} = 3$ ,  $c_{13} = 4$ ;  $a_{24} = 0$ ,  $b_{24} = 4$ ,  $c_{24} = 2$ ;  $a_{32} = 1$ ,  $b_{32} = 8$ ,  $c_{32} = -1$ ;  $a_{34} = 1$ ,  $b_{34} = 5$ ,  $c_{34} = 6$ . De nettoproducties zijn:  $s_1 = 5$ ,  $s_2 = 0$ ,  $s_3 = 3$  en  $s_4 = -8$ . Los voor deze data het minimale kostenstroom op. Neem de eerste streng toelaatbare opspannende boom volgens de constructie met het extra knooppunt 0 en neem  $M = 100$ .



### Opgave 5.10

Beschouw de lengtefunctie  $l : A \rightarrow \mathbb{Z}$  met  $l_{ij} = c_{ij}$  als  $(i, j)$  in de goede richting wordt doorlopen en  $l_{ji} = -c_{ij}$  als  $(i, j)$  in de verkeerde richting wordt doorlopen. Toon het volgende aan voor  $(k, l)$ ,  $(p, q)$  en  $C$  als in algoritme 5.4:

a. 
$$l(C) = \begin{cases} \bar{c}_{kl} & \text{als } (k, l) \in L \\ -\bar{c}_{kl} & \text{als } (k, l) \in U \end{cases}$$

b. Zij  $T - (p, q) = T_1 \cup T_2$  met de wortel 1 in  $T_1$  en laat  $T' = T + (k, l) - (p, q)$ .

Laat  $u'$  de boomafstand in  $T'$  zijn. Dan geldt: 
$$u'_i = \begin{cases} u_i, & i \in T_1 \\ u_i + l(C), & i \in T_2, k \in T_2 \text{ en } l \in T_1 \\ u_i - l(C), & i \in T_2, l \in T_2 \text{ en } k \in T_1 \end{cases}$$

### Opgave 5.11

Toon aan dat de in Algoritme 5.5 gebruikte opspannende bomen alle streng toelaatbaar zijn.



# Hoofdstuk 6

## Koppelingen

### 6.1 Inleiding

Zij  $(G, E)$  een niet-gerichte, normale graaf.  $M \subseteq E$  heet een *koppeling* als er geen twee takken van  $M$  aan elkaar grenzen. Een koppeling  $M$  heeft *maximale cardinaliteit* als er geen koppeling  $M'$  bestaat met  $\#M' > \#M$ .  $\mu(G) = \max\{\#M \mid M \text{ is een koppeling}\}$  heet het *koppelingstgetal* van de graaf  $G$ .

Omdat iedere tak van een koppeling twee knooppunten bindt en ieder knooppunt incident is met hoogstens één tak van de koppeling, kan een koppeling nooit meer dan  $\frac{1}{2}n$  elementen bevatten. Een koppeling die  $\frac{1}{2}n$  elementen bevat heet een *volmaakte koppeling*. Als  $(i, j)$  een tak van de koppeling is, dan heten de knooppunten  $i$  en  $j$  *gebonden* en zijn ze elkaars *partner*; knooppunten die niet gebonden zijn, zijn *vrij*. Het aantal vrije knooppunten is dus  $n - 2 \cdot \#M$ .

Een keten  $C$  heet een *wisselketen* m.b.t. een koppeling  $M$  als de takken van  $C$  afwisselend wel en niet tot de koppeling behoren. Als de keten gesloten is, dan spreken we van een *wisselkring*; een wisselkring bevat een even aantal takken en de helft ervan behoort tot  $M$ .

Een wisselketen heet een *groeiketen* als beide uiteinden vrij zijn. Een groeiketen heeft een oneven aantal takken, zeg  $2p + 1$ , waarvan er  $p$  tot  $M$  behoren. Bij een koppeling  $M$  en groeiketen  $C$  kunnen we  $M \oplus C = M \cup C - M \cap C$  beschouwen. Dit zijn de takken die op  $C$  niet en buiten  $C$  wel tot  $M$  behoren.  $M \oplus C$  is weer een koppeling, en heeft één tak meer dan  $M$ .

#### **Stelling 6.1** (*Stelling van Berge*)

*Een koppeling  $M$  heeft maximale cardinaliteit d.e.s.d. als er geen groeiketen m.b.t.  $M$  is.*

#### **Bewijs**

$\Rightarrow$  Als er een groeiketen  $C$  is, dan kan de koppeling worden uitgebreid tot  $M \oplus C$ .

$\Leftarrow$  Het is voldoende om aan te tonen dat als  $M$  geen maximale cardinaliteit heeft, er een groeiketen bestaat. Als  $M$  geen maximale cardinaliteit heeft, dan is er een koppeling  $M'$  met  $\#M' > \#M$ . Beschouw de graaf  $G' = (V, M \oplus M')$ . Ieder knooppunt is incident met hoogstens één tak van  $M$  en hoogstens één tak van  $M'$ : ieder knooppunt heeft in  $G'$  hoogstens graad 2. De componenten

van  $G'$  zijn dus: geïsoleerde knooppunten, wisselketens waarvan de takken afwisselend tot  $M$  en  $M'$  behoren en wisselketens met evenveel takken van  $M$  als van  $M'$ . Omdat  $\#M' > \#M$ , is er een wisselketen die begint en eindigt met een tak van  $M'$ . Dit is een groeiketens t.o.v.  $M$ . ■

### **Gevolg**

Zij  $M$  een koppeling. Dan bestaat er een koppeling  $M^*$  met maximale cardinaliteit die alle knooppunten bindt die ook gebonden zijn door  $M$ .

### **Bewijs**

Als  $M$  geen maximale cardinaliteit heeft, dan is er een groeiketens m.b.t.  $M$ . Dit geeft een koppeling  $M' = M \oplus C$  met één tak meer dan  $M$  en waarin gebonden knooppunten gebonden blijven. Op deze wijze kunnen we doorgaan totdat een koppeling  $M^*$  met maximale cardinaliteit is verkregen. ■

Een deelverz.  $W \subseteq V$  heet *onafhankelijk* als geen enkel tweetal knooppunten van  $W$  door een tak verbonden is.  $\alpha(G) = \max\{\#W \mid W \subseteq V \text{ onafhankelijk}\}$  heet het *onafhankelijkheidsgetal* van de graaf  $G$ .

Een deelverz.  $W \subseteq V$  heet een *knooppuntenbedekking* als iedere tak minstens één eindpunt in  $W$  heeft.  $\tau(G) = \min\{\#W \mid W \subseteq V \text{ knooppuntenbedekking}\}$  heet het *knooppuntenbedekkingsgetal* van de graaf  $G$ .

Een deelverz.  $F \subseteq E$  heet een *takkenbedekking* als ieder knooppunt incident is met minstens één tak van  $F$ .  $\rho(G) = \min\{\#F \mid F \subseteq E \text{ takkenbedekking}\}$  heet het *takkenbedekkingsgetal* van de graaf  $G$ .

**Lemma 6.1** *Voor iedere graaf  $G = (V, E)$  geldt:  $\alpha(G) \leq \rho(G)$  en  $\mu(G) \leq \tau(G)$ .*

### **Bewijs**

Omdat verschillende knooppunten van een onafhankelijke verz. bedekt moeten worden door verschillende takken geldt:  $\alpha(G) \leq \rho(G)$ . Evenzo moeten verschillende takken van een koppeling bedekt worden door verschillende knooppunten:  $\mu(G) \leq \tau(G)$ . ■

### **Stelling 6.2** (*Stelling van Gallai*)

*Neem aan dat  $G$  geen geïsoleerde knooppunten heeft. Dan geldt:  $\alpha(G) + \tau(G) = \rho(G) + \mu(G) = n$ .*

### **Bewijs**

Merk allereerst op dat  $W \subseteq V$  onafhankelijk is d.e.s.d. als  $V - W$  een knooppuntenbedekking is. Dus  $W$  is een onafhankelijke verz. met maximale cardinaliteit d.e.s.d. als  $V - W$  een knooppuntenbedekking is met minimale cardinaliteit. Als  $W$  een onafhankelijke verz. is met maximale cardinaliteit, dan geldt:  $n = \#W + \#(V - W) = \alpha(G) + \tau(G)$ .

Zij  $M \subseteq E$  en  $F \subseteq E$  een koppeling resp. takkenbedekking zdd.  $\#M = \mu(G)$  en  $\#F = \rho(G)$ . Voeg voor ieder vrij knooppunt aan  $M$  een tak toe die incident is met het vrije knooppunt.

Dit geeft een takkenbedekking  $F'$  met  $\#F' = \#M + (n - 2 \cdot \#M) = n - \#M$ . Hieruit volgt:  $\#F' + \#M = n$ .

Anderzijds, verwijder uit  $F$  takken totdat voor de eerste keer ieder knooppunt incident is met hoogstens één tak. Dit geeft een koppeling  $M'$ . Daar  $F$  een takkenbedekking is met minimale cardinaliteit, bevat  $F$  geen ketens met drie (of meer) takken, zodat door het verwijderen van een tak precies één vrij knooppunt ontstaat. Hieruit volgt:  $\#F - \#M' =$  aantal vrije knooppunten t.o.v.  $M' = n - 2 \cdot \#M'$ , wat oplevert:  $\#F + \#M' = n$ . Door bovenstaande resultaten te combineren krijgen we:  $n = \#F' + \#M \geq \#F + \#M \geq \#F + \#M' = n$ , d.w.z.  $\rho(G) + \mu(G) = \#F + \#M = n$ . ■

Voor iedere  $W \subseteq V$  definiëren we:  $o(W) =$  het aantal componenten in de deelgraaf  $G_W$  (de deelgraaf met de knooppunten  $W$  en met de takken van  $E$  die beide uiteinden in  $W$  hebben) dat een oneven aantal knooppunten heeft.

**Lemma 6.2** *Voor iedere koppeling  $M$  en iedere  $W \subseteq V$  geldt:  $\#M \leq n - \frac{1}{2}[\#W + o(W)]$ .*

### Bewijs

Zij  $M$  een koppeling en laat  $W \subseteq V$ . Iedere component met een oneven aantal knooppunten heeft minstens één vrij knooppunt. In  $G_W$  zijn dus minstens  $o(W)$  vrije knooppunten, d.w.z. hoogstens  $\#W - o(W)$  gebonden knooppunten. Binnen  $G_W$  is het aantal takken van  $M$  dus hoogstens  $\frac{1}{2}[\#W - o(W)]$ . Het aantal takken van  $M$  dat incident is met  $V - W$  is hoogstens  $\#(V - W)$ . Hieruit volgt:  $\#M \leq \frac{1}{2}[\#W - o(W)] + \#(V - W) = n + \frac{1}{2}[\#W - o(W) - 2 \cdot \#W] = n - \frac{1}{2}[\#W + o(W)]$ . ■

### Opmerking

$\#W + o(W)$  is even, dus  $n - \frac{1}{2}[\#W + o(W)]$  is een geheel getal.

### Stelling 6.3 (Stelling van Tutte)

*Zij  $G = (V, E)$  een samenhangende graaf. Dan geldt:  $G$  heeft een volmaakte koppeling d.e.s.d. als  $o(W) \leq \#(V - W)$  voor alle  $W \subseteq V$ .*

### Bewijs

$\Rightarrow$  Als  $G$  een volmaakte koppeling  $M$  heeft, dan geldt volgens Lemma 6.2:  $\frac{1}{2}n \leq n - \frac{1}{2}[\#W + o(W)]$ , d.w.z.  $o(W) \leq \#(V - W)$  voor alle  $W \subseteq V$ .

$\Leftarrow$  Stel  $o(W) \leq \#(V - W)$  voor alle  $W \subseteq V$ . Neem  $V = W$ , dan volgt  $o(V) = 0$ , d.w.z.  $n$  is even. Veronderstel dat  $G$  geen volmaakte koppeling heeft. Door aan  $G$  een tak toe te voegen kan  $o(W)$  alleen kleiner worden, terwijl  $\#(V - W)$  hetzelfde blijft: aan de voorwaarde van de Stelling blijft dus voldaan. Voeg nu steeds takken toe zolang  $G$  geen volmaakte koppeling heeft. Aldus ontstaat een graaf  $G' = (V, E')$  die aan de voorwaarde van de Stelling voldoet, geen volmaakte koppeling heeft, maar het toevoegen van iedere tak geeft wel een volmaakte koppeling (deze bestaat inderdaad omdat  $n$  even is).

Laat  $W_0 = \{w \in V \mid w \text{ is in } G' \text{ met niet alle andere knooppunten verbonden}\}$ .

*Bewering: De componenten van  $G_{W_0}$  zijn volledig.*

*Bewijs:*

Het is voldoende te bewijzen dat als  $v, w, x \in W_0$  en  $(v, w), (w, x) \in E'$ , dan ook  $(v, x) \in E'$ . Stel  $(v, x) \notin E'$ . Door het toevoegen van  $(v, x)$  ontstaat een volmaakte koppeling  $M_1$  in  $G'$  die  $(v, x)$  bevat. Omdat  $w \in W_0$  is er een  $y \in V$  met  $(w, y) \notin E'$ . Door  $(w, y)$  toe te voegen ontstaat een volmaakte koppeling  $M_2$  die  $(w, y)$  bevat. Beschouw het symmetrische verschil  $M_1 \oplus M_2$ . Omdat  $M_1$  en  $M_2$  beide volmaakte koppelingen zijn, kunnen we in ieder knooppunt waar we met een wisselketen aankomen ook weer weg:  $M_1 \oplus M_2$  bestaat uit een aantal wisselkringen, zeg  $C_1, C_2, \dots, C_p$ . De takken  $(v, x)$  en  $(w, y)$  behoren tot deze kringen, zeg  $(v, x) \in C_1$  en  $(w, y) \in C_2$ .

Als  $C_1 \neq C_2$ :

Dan is  $C_1 - (v, w)$  in  $G'$  een groeiketen m.b.t.  $M_1$ , zodat  $M_1$  in  $G'$  is uit te breiden tot een volmaakte koppeling: tegenspraak.

Als  $C_1 = C_2$ :

Doorloop deze kring vanuit  $w$  in de richting van  $y$  totdat we een van de uiteinden van  $(v, x)$  tegenkomen, zeg  $v$ . Laat  $P$  het deel van  $C_1$  zijn dat in  $y$  begint en eindigt in  $v$  en verleng deze keten met  $(v, w)$ . Dit is in  $G'$  een groeiketen m.b.t.  $M_2$ , wat ook een tegenspraak geeft.

We tonen nu aan dat  $o(W_0) \leq \#(V - W_0)$  tot een tegenspraak leidt.

In  $G'$  kan als volgt een volmaakte koppeling worden geconstrueerd. Beschouw eerst de deelgraf(en) voortgebracht door  $W_0$ . We hebben gezien dat deze volledig zijn. In de componenten met een even aantal knooppunten kan dus een volmaakte koppeling worden gekozen en in de andere componenten een koppeling met per component één vrij knooppunt. Het aantal vrije knooppunten is dus  $o(W)$ . Er is gegeven dat  $o(W) \leq \#(V - W_0)$ . Bind nu in de koppeling ieder van de vrije knooppunten van  $W_0$  met een ander punt van  $V - W_0$  (dit kan want de knooppunten van  $V - W_0$  zijn met alle andere knooppunten verbonden). De resterende knooppunten van  $V - W_0$ , dit zijn er  $n - \#(W_0) - o(W_0)$  en dat aantal is even omdat zowel  $n$  als  $\#(W_0) + o(W_0)$  even zijn, worden alle gebonden (ook dit kan omdat de knooppunten van  $V - W_0$  met alle andere knooppunten zijn verbonden). Aldus is er een volmaakte koppeling geconstrueerd, wat een tegenspraak oplevert. ■

**Stelling 6.4** (*Stelling van Tutte-Berge*)

*Zij  $G$  een samenhangende graaf. Dan geldt:  $\mu(G) = n - \frac{1}{2} \max_{W \subseteq V} [\#W + o(W)]$ .*

**Bewijs**

Noteer het gehele getal  $n - \frac{1}{2} \max_{W \subseteq V} [\#W + o(W)]$  met  $p$ . Uit Lemma 6.2 volgt dat  $\mu(G) \leq \min_{W \subseteq V} \{n - \frac{1}{2} [\#W + o(W)]\} = n - \frac{1}{2} \max_{W \subseteq V} [\#W + o(W)] = p$ .

Laat  $G' = (V', E')$  met  $V' = V \cup X$ , waarbij  $X = \{x_1, x_2, \dots, x_{n-2p}\}$ , terwijl  $E'$  uit  $E$  wordt verkregen door alle toegevoegde knooppunten  $x_i$  met alle knooppunten van  $V$  te verbinden.

*Geval 1:  $G'$  heeft een volmaakte koppeling, zeg  $M'$ .*

Neem  $M$  door in  $M'$  de takken incident met  $X$  weg te laten. Dan is  $M$  een koppeling in  $G$  en deze koppeling bindt minstens  $n - (n - 2p) = 2p$  knooppunten van  $G$ :  $\mu(G) \geq p$ , waarmee het bewijs geleverd is.

*Geval 2:  $G'$  heeft geen volmaakte koppeling.*

Volgens Stelling 6.3 is er een  $W' \subseteq V'$  met  $o(W') > \#(V' - W')$ .

a.  $W' \cap X \neq \emptyset$  :

Omdat de knooppunten van  $X$  met alle andere verbonden zijn, heeft  $W'$  slechts één component, d.w.z.  $1 \geq o(W') > \#(V' - W')$ . Dus  $\#(V' - W') = 0$ , d.w.z.  $V' = W'$ . Omdat  $G'$  een even aantal knooppunten heeft, geldt:  $0 = o(V') = o(W')$ , wat in tegenspraak is met  $o(W') > \#(V' - W')$ .

b.  $W' \cap X = \emptyset$  :

Nu is  $W' \subseteq V$ , zodat  $p = n - \frac{1}{2} \max_{W \subseteq V} [\#W + o(W)] \leq n - \frac{1}{2} [\#W' + o(W')]$ . Hieruit volgt:  $o(W') \leq 2n - 2p - \#W' = \#(V' - W')$  : tegenspraak. ■

### Gevolg

Uit de Stellingen 6.2 en 6.4 volgt:  $\rho(G) = \frac{1}{2} \max_{W \subseteq V} [\#W + o(W)]$ .

Veronderstel dat de graaf *bipartiet* is, d.w.z.  $G = (V_1 \cup V_2, E)$  en iedere tak heeft één eindpunt in  $V_1$  en één eindpunt in  $V_2$ . Laat  $V_1 = \{v_1, v_2, \dots, v_p\}$  en  $V_2 = \{w_1, w_2, \dots, w_q\}$ . Het probleem om een koppeling met maximale cardinaliteit te vinden kan worden geformuleerd als het volgende LP-probleem:

$$\max \left\{ \sum_{(i,j)} x_{ij} \left| \begin{array}{l} \sum_j x_{ij} \leq 1, \quad i = 1, 2, \dots, p \\ \sum_i x_{ij} \leq 1, \quad j = 1, 2, \dots, q \\ x_{ij} \geq 0, \quad (i, j) \in E \end{array} \right. \right\} \quad (6.1)$$

De coëfficiëntenmatrix van (6.1) is de incidentiematrix van de graaf  $G$ . Omdat deze totaal unimodulair is (Stelling 1.3) zijn de hoekpunten van (6.1) geheeltallig. Iedere basisoplossing is dus een  $(0, 1)$ -oplossing.

**Stelling 6.5** *Er is een één-éénduidig verband tussen de hoekpunten van (6.1) en de koppelingen in  $G$  zdd. de waarde van de doelfunctie de cardinaliteit van de koppeling is.*

### Bewijs

Zij  $x$  een hoekpunt van (6.1), dan is  $x$  een  $(0, 1)$ -vector. Uit de beperkingen van (6.1) volgt dat voor iedere  $i$  hoogstens één  $x_{ij}$  positief is, dus gelijk aan 1; evenzo geldt dat voor iedere  $j$  hoogstens één  $x_{ij}$  positief is, dus gelijk aan 1. Laat  $M = \{(i, j) \mid x_{ij} = 1\}$ , dan is  $M$  een koppeling en  $\#M = \sum_{(i,j)} x_{ij}$ .

Omgekeerd, zij  $M$  een koppeling. Neem  $x_{ij} = 1$  als  $(i, j) \in M$  en  $x_{ij} = 0$  als  $(i, j) \notin M$ . Stel  $x$  is geen hoekpunt, d.w.z.  $x = \lambda x^1 + (1 - \lambda)x^2$  met  $x^1$  en  $x^2$  toelaatbaar voor (6.1) en  $0 < \lambda < 1$ .

Dan geldt:

$$\text{als } (i, j) \in M : 1 = x_{ij} = \lambda x_{ij}^1 + (1 - \lambda)x_{ij}^2 \leq 1 \rightarrow x_{ij} = x_{ij}^1 = x_{ij}^2 = 1;$$

$$\text{als } (i, j) \notin M : 0 = x_{ij} = \lambda x_{ij}^1 + (1 - \lambda)x_{ij}^2 \geq 0 \rightarrow x_{ij} = x_{ij}^1 = x_{ij}^2 = 0.$$

Dus  $x = x^1 = x^2$ , d.w.z.  $x$  is wel een hoekpunt. ■

## Toepassing

Beschouw een  $n \times n$  matrix  $A = (a_{ij})$  met niet-negatieve coëfficiënten en zdd. iedere rij- en kolomsom 1 is; een dergelijke matrix wordt *dubbelstochastisch* genoemd. Een dubbelstochastische matrix heet een *permutatiematrix* als alle positieve elementen gelijk aan 1 zijn, d.w.z. dat iedere rij en iedere kolom precies één 1 bevat. Een dubbelstochastische matrix voldoet aan het stelstel

$$\begin{cases} \sum_j x_{ij} = 1, & i = 1, 2, \dots, n \\ \sum_i x_{ij} = 1, & j = 1, 2, \dots, n \\ x_{ij} \geq 0, & \text{voor alle } i \text{ en } j \end{cases}$$

Omdat de matrix van dit stelstel totaal unimodulair is (het is de incidentiematrix van de volledige bipartiete graaf  $K_{n,n}$ ), zijn de hoekpunten geheeltallig, d.w.z. dat zij corresponderen met permutatiematrices. Hiermee is het volgende bewezen.

### **Stelling 6.6** (*Stelling van Birkhoff-Von Neumann*)

*Iedere dubbelstochastische matrix is een convexe combinatie van permutatie matrices.*

Beschouw het duale probleem van (6.1):

$$\min \left\{ \sum_{i=1}^p s_i + \sum_{j=1}^q t_j \mid \begin{array}{l} s_i + t_j \geq 1, \quad (i, j) \in E \\ s_i, t_j \geq 0, \quad 1 \leq i \leq p, 1 \leq j \leq q \end{array} \right\} \quad (6.2)$$

**Stelling 6.7** *Er is een één-éénduidig verband tussen de  $(0, 1)$ -oplossingen van (6.2) en de knooppuntenbedekkingen van  $G$  zdd. de waarde van de doelfunctie het aantal elementen in de bedekking is.*

### **Bewijs**

Zij  $(s, t)$  een  $(0, 1)$ -oplossing van (6.2). Neem  $W_1 = \{i \in V_1 \mid s_i = 1\}$  en  $W_2 = \{j \in V_2 \mid t_j = 1\}$ . Omdat  $s_i + t_j \geq 1$  voor iedere  $(i, j) \in E$  is dit een knooppuntenbedekking.

Omgekeerd, zij  $W_1 \cup W_2$  een knooppuntenbedekking.

Neem  $s_i = \begin{cases} 1 & \text{als } i \in W_1 \\ 0 & \text{als } i \notin W_1 \end{cases}$  en  $t_j = \begin{cases} 1 & \text{als } j \in W_2 \\ 0 & \text{als } j \notin W_2. \end{cases}$

Dan is  $\sum_{i=1}^p s_i + \sum_{j=1}^q t_j = \#(W_1 \cup W_2)$  en  $(s, t)$  is een toelaatbare  $(0, 1)$ -oplossing van (6.2). ■

### **Stelling 6.8** (*Stelling van König*)

*In een bipartiete graaf  $G$  geldt:  $\mu(G) = \tau(G)$ .*

### **Bewijs**

Zij  $(s, t)$  is een optimaal hoekpunt van (6.2). Dan is  $(s, t)$  geheeltallig, dus een  $(0, 1)$ -oplossing. Deze correspondeert met een minimale knooppuntenbedekking  $W = W_1 \cup W_2$  en de waarde van de doelfunctie is  $\tau(G)$ . Omdat volgens de dualiteitsstelling van de lineaire programmering de optimale waarden van (6.1) en (6.2) aan elkaar gelijk zijn, geldt:  $\mu(G) = \tau(G)$ . ■

### **Gevolg**

$\alpha(G) = \rho(G) = n - \mu(G) = n - \tau(G)$ .

## 6.2 Equivalente combinatorische resultaten

We zullen in deze paragraaf de onderlinge relaties van een achttal combinatorische resultaten (stellingen) bepreken. De eerste vier resultaten (A t/m D) blijken equivalent te zijn, evenals de volgende drie (E, F en G). Het laatste resultaat (H) is het sterkst: hieruit volgt E (dus ook F en G) en uit F (dus ook uit E en G) volgt A (dus ook B, C, en D). We zullen nu eerst de acht resultaten formuleren.

*Resultaat A (Stelling van König)*

In een bipartiete graaf is het minimum aantal elementen in een knooppuntenbedekking gelijk aan het aantal elementen in een maximale koppeling.

*Resultaat B (Stelling van König-Egervary)*

Zij  $A$  een  $m \times n$  matrix met  $(0,1)$ -elementen. Het minimum aantal rijen en kolommen dat alle 1'en bevat is gelijk aan het maximum aantal onafhankelijke 1'en (dat zijn 1'en die in verschillende rijen en kolommen staan).

*Resultaat C (Stelling van Hall)*

Zij  $E$  een eindige niet-lege verz. en  $\mathcal{S} = (S_1, S_2, \dots, S_n)$  een collectie van deelverz. van  $E$ .  $\mathcal{S}$  is een transversaal d.e.s.d. als voor iedere  $k = 1, 2, \dots, n$  de vereniging van ieder  $k$ -tal  $S_i$ 's minstens  $k$  elementen bevat.

Het volgende resultaat betreft partieel geordende verzamelingen. Het is op het eerste gezicht wat verrassend dat een dergelijk probleem ook equivalent met de vorige is.

Zij  $X$  een eindige verz. met daarop een binaire relatie  $\leq$ . Het tweetal  $(X, \leq)$  heet een *partieel geordende verz.* als de volgende eigenschappen gelden:

- (i)  $x \leq x$  voor alle  $x \in X$  (reflexiviteit);
- (ii)  $x \leq y$  en  $y \leq x$  impliceert  $x = y$  voor alle  $x, y \in X$  (anti-symmetrie);
- (iii)  $x \leq y$  en  $y \leq z$  impliceert  $x \leq z$  voor alle  $x, y, z \in X$  (transitiviteit).

Twee elementen  $x$  en  $y$  met  $x \leq y$  en/of  $y \leq x$  heten *vergelijkbaar*; is dit niet het geval dan heten ze *onvergelijkbaar*. Als  $x \leq y$  en  $x \neq y$ , dan is  $x < y$ . Een verz. elementen  $Y \subseteq X$  zdd. ieder tweetal uit  $Y$  vergelijkbaar is, heet een *keten* en  $\#Y$  is de *lengte* van de keten. Een verz. ketens *bedekt*  $X$  als ieder element van  $X$  in minstens één keten zit. Een verz. elementen  $A \subseteq X$  waarvan geen enkel tweetal vergelijkbaar is, noemen we een *anti-keten* met  $\#A$  de lengte ervan.

Merk op dat iedere keten hoogstens één element gemeen heeft met iedere anti-keten. Dus de maximale lengte van een anti-keten is hoogstens het minimum aantal ketens dat  $X$  bedekt. Het volgende resultaat zegt dat er gelijkheid is.

*Resultaat D (Stelling van Dilworth)*

Als  $(X, \leq)$  een eindige partieel geordende verz. is, dan is het maximum aantal elementen in een anti-keten gelijk aan het minimum aantal ketens dat  $X$  kan bedekken.

*Resultaat E (Stelling van Menger)*

Het maximum aantal pijl-disjuncte paden van  $s$  naar  $t$  is gelijk aan het aantal elementen in een minimale  $(s, t)$ -snede.

*Resultaat F (Stelling van Ford en Fulkerson)*

Zij  $N = (V, A)$  een netwerk met geheeltallige capaciteiten  $b$ . Dan is de maximale stroom geheeltalig en heeft een waarde die gelijk is aan de capaciteit van een snede met een minimale capaciteit.

*Resultaat G (Stelling van Hoffman)*

Zij  $N = (V, A)$  een netwerk met onder- en bovengrenzen  $a$  resp.  $b$ .  $N$  heeft een toelaatbare circulatiestroom d.e.s.d. als  $\sum_{i \in W, j \notin W} b_{ij} \geq \sum_{i \notin W, j \in W} a_{ij}$  voor alle  $W \subseteq V$ .

*Resultaat H (Dualiteitsstelling lineaire programmering)*

Beschouw een eindig LP-probleem en het duale probleem. Dan zijn de optima (maximum resp. minimum) van beide problemen aan elkaar gelijk.

De acht resultaten zijn alle waar. Een aantal is reeds bewezen: A, E, F en G in dit dictaat; B en C in Discrete Wiskunde en H in Inleiding Besliskunde. Alleen resultaat D is dus nieuw.

**Lemma 6.3** *Uit resultaat A volgt resultaat B.*

**Bewijs**

Zij  $X$  een  $m \times n$  (0,1)-matrix. Construeer de volgende bipartiete graaf  $G = (V_1 \cup V_2, E)$  :

$V_1 = \{1, 2, \dots, m\}$ ,  $V_2 = \{1, 2, \dots, n\}$  en  $E = \{(i, j) \mid x_{ij} = 1\}$ .

Onafhankelijke 1'en van  $X$  corresponderen met een koppeling; rijen en kolommen die tezamen alle 1'en bedekken corresponderen met een knooppuntenbedekking. Uit resultaat A (het minimum aantal elementen in een knooppuntenbedekking gelijk aan het aantal elementen in een maximale koppeling) volgt dat het minimum aantal rijen en kolommen dat alle 1'en bevat gelijk is aan het maximum aantal onafhankelijke 1'en. ■

**Lemma 6.4** *Uit resultaat B volgt resultaat D.*

**Bewijs**

Zij  $(X \leq)$  een eindig partieel geordende verz. met  $X = \{1, 2, \dots, n\}$ . Beschouw een willekeurige ketenbedekking  $C_1, C_2, \dots, C_p$  en een willekeurige anti-keten  $C^*$  met  $q$  elementen. Dan is  $q \leq p$ , omdat ieder element van  $C^*$  uit een andere  $C_i$  moet komen, want de elementen van  $C^*$  zijn niet vergelijkbaar. We moeten dus bewijzen dat er een ketenbedekking  $C_1, C_2, \dots, C_p$  is en een anti-keten  $C^*$  met  $\#C^* = p$ , want dan is de bedekking minimaal en de anti-keten maximaal.

Construeer de volgende  $n \times n$  (0,1)-matrix  $A$  :  $a_{ij} = 1$  d.e.s.d. als  $i < j$ . Beschouw een keten  $C = \{i_1, i_2, \dots, i_k\}$  met  $i_1 < i_2 < \dots < i_k$ . Dan zijn de 1'en in de posities  $(i_1, i_2), (i_2, i_3), \dots, (i_{k-1}, i_k)$  onafhankelijk, want ze staan in verschillende rijen en kolommen. Op grond van hetzelfde argument geldt dat als  $C_1, C_2, \dots, C_p$  disjuncte ketens zijn die tezamen  $X$  bedekken, dan zijn de daarbij



behorende 1'en onafhankelijk (we mogen ons beperken tot disjuncte ketens, want laat anders "dubbele elementen" weg).

Omgekeerd, beschouw een verz. onafhankelijke 1'en van de bij  $(X, \leq)$  behorende matrix  $A$ , zeg  $(i_1, j_1), (i_2, j_2), \dots, (i_k, j_k)$ . Deze corresponderen met een disjuncte ketenbedekking, immers: neem iedere  $i$  die niet in een rij of kolom van de onafhankelijke 1'en voorkomt als aparte keten  $\{i\}$ ; verder beginnen we met de keten  $\{i_1, j_1\}$ ; we nemen daarna  $(i_2, j_2)$ : als  $i_2 = j_1$ , dan verlengen we  $\{i_1, j_1\}$  tot  $\{i_1, j_1, j_2\}$ , anders nemen we een nieuwe keten  $\{i_2, j_2\}$  als nieuwe keten, etc.

Iedere onafhankelijke 1 geeft dus aanleiding tot één keten minder dan het maximale aantal van  $n$  ketens. Dus als de onafhankelijke verz.  $n-p$  1'en bevat, dan heeft de bedekking  $p$  ketens. Hieruit volgt: als er maximaal  $n-p$  onafhankelijke 1'en zijn, dan heeft een minimale ketenbedekking  $p$  ketens. Als er maximaal  $n-p$  onafhankelijke 1'en zijn, dan is volgens resultaat B het minimum aantal rijen en kolommen dat alle 1'en bevat eveneens  $n-p$ . Beschouw nu een minimum aantal van  $n-p$  rijen en kolommen die tezamen alle 1'en bevatten. Laat  $J$  de verz. van deze kolommen zijn en  $I$  de verz. van de rijen waarin 1'en voorkomen die niet in de kolommen van  $J$  zitten:  $i \in I$  d.e.s.d. als  $i < j$  voor zekere  $j \notin J$ . Nu geldt:  $I \cap J = \emptyset$  (immers: stel  $k \in I \cap J$ , dan is er een  $j \notin J$  met  $k < j$  en een  $i \notin I$  met  $i < k$ ; dus  $i < j$ , d.w.z.  $a_{ij} = 1$  en noch rij  $i$  noch kolom  $j$  is doorgestreept).

Nu is  $\{1, 2, \dots, n\} \setminus (I \cup J)$  een anti-keten  $C^*$ , immers: stel niet, dan is  $i < j$  voor zekere  $i, j \notin I \cup J$ :  $a_{ij} = 1$  en dit element is niet doorgestreept: tegenspraak.

Tevens geldt:  $\#C^* = n - \#(I \cup J) = n - (\#I + \#J) = n - (n-p) = p$ , d.w.z.  $p$  is het maximum aantal elementen in de anti-keten  $C^*$  en is ook gelijk aan het minimum aantal ketens dat  $X$  kan bedekken. ■

**Lemma 6.5** *Uit resultaat D volgt resultaat C.*

### Bewijs

Zij  $E = \{1, 2, \dots, m\}$  een eindige niet-lege verz. en  $\mathcal{S} = (S_1, S_2, \dots, S_n)$  een collectie van deelverz. van  $E$  zdd. voor iedere  $k = 1, 2, \dots, n$  de vereniging van ieder  $k$ -tal  $S_i$ 's minstens  $k$  elementen bevat (dus  $m \geq n$ ).

Neem  $X = X_1 \cup X_2$ , met  $X_1 = \{1, 2, \dots, m\}$  en  $X_2 = \{n+1, n+2, \dots, n+m\}$ , en daarin de volgende partiële ordening:  $i \leq j$  als  $i \leq m$ ,  $j > m$  en  $i \in S_{j-m}$ . Dus een keten bestaat uit één of twee elementen.

Zij  $C^*$  een anti-keten met  $p$  elementen van  $X_1$  en  $q$  van  $X_2$ . De  $q$  elementen van  $X_2$  sluiten minstens  $q$  elementen van  $X_1$  uit:  $\#C^* = p + q \leq (m - q) + q = m$ . Omdat  $X_1$  een anti-keten is met  $m$  elementen, is  $X_1$  een anti-keten met maximum aantal elementen. Uit resultaat D volgt dat het minimum aantal ketens om  $X$  te bedekken  $m$  is. Omdat iedere keten één of twee elementen bevat, is een minimale ketenbedekking van de vorm:  $C_1 = \{i_1, m+j_1\}, C_2 = \{i_2, m+j_2\}, \dots, C_n = \{i_n, m+j_n\}, C_{n+1} = \{i_{n+1}\}, C_{n+2} = \{i_{n+2}\}, \dots, C_m = \{i_m\}$ . Dus  $\{i_1, i_2, \dots, i_n\}$  is een stelsel representanten, d.w.z.  $\mathcal{S} = (S_1, S_2, \dots, S_n)$  is een transversaal.

De andere kant van resultaat C (als  $\mathcal{S} = (S_1, S_2, \dots, S_n)$  een transversaal is, geldt dat voor iedere  $k = 1, 2, \dots, n$  de vereniging van ieder  $k$ -tal  $S_i$ 's minstens  $k$  elementen bevat) is triviaal. ■

**Lemma 6.6** *Uit resultaat C volgt resultaat A.*

**Bewijs**

Zij  $G = (V_1 \cup V_2, E)$  een bipartiete graaf,  $W_1 \cup W_2$  een knooppuntenbedekking en  $M$  een koppeling. Iedere tak van  $M$  heeft minstens één eindpunt in  $W_1 \cup W_2$ , en verschillende takken van  $M$  corresponderen met verschillende eindpunten:  $\#(W_1 \cup W_2) \geq \#M$ . We moeten dus nog bewijzen dat er een knooppuntenbedekking  $W_1 \cup W_2$  en een koppeling  $M$  is met  $\#(W_1 \cup W_2) = \#M$ .

Neem voor  $W_1 \cup W_2$  een minimale knooppuntenbedekking en stel dat  $W_1 = \{1, 2, \dots, p\}$  en  $W_2 = \{p+1, p+2, \dots, p+q\}$ . Laat  $S_i = \begin{cases} \{j \in V_2 - W_2 \mid (i, j) \in E\}, & i = 1, 2, \dots, p \\ \{j \in V_1 - W_1 \mid (j, i) \in E\}, & i = p+1, p+2, \dots, p+q. \end{cases}$

Nu geldt dat de vereniging van een  $k$ -tal  $S_i$ 's minstens  $k$  elementen bevat, immers:

Stel dat het niet zo is: dan is er dus een  $k$ -tal  $S_i$ 's dat minder dan  $k$  elementen bevat. De takken van de  $\{S_i \mid 1 \leq i \leq p\}$  en  $\{S_i \mid p+1 \leq i \leq p+q\}$  hebben geen gemeenschappelijke eindpunten: er is dus binnen de eerste of de tweede groep een  $j$ -tal  $S_i$ 's waarvan de vereniging minder dan  $j$  elementen bevat. Maar dan is er een kleinere bedekking te vinden door de hoogstens  $j-1$  eindpunten te nemen i.p.v. de  $j$  eindpunten van  $W_1$  of  $W_2$ .

Volgens resultaat C is  $(S_1, S_2, \dots, S_{p+q})$  een transversaal met corresponderende representanten, zeg  $w_1, w_2, \dots, w_{p+q}$ . Nu vormen de takken  $(i, w_i)$ ,  $1 \leq i \leq p$  en  $(w_i, i)$ ,  $p+1 \leq i \leq p+q$  een koppeling  $M$  en  $\#M = \#(W_1 \cup W_2)$ . ■

**Gevolg**

De resultaten A, B, C en D zijn equivalent.

**Lemma 6.7** *Uit resultaat E volgt resultaat F.*

**Bewijs**

Dit bewijs is te vinden in Stelling 4.4. ■

**Lemma 6.8** *Uit resultaat F volgt resultaat G.*

**Bewijs**

Zij  $x$  een toelaatbare circulatiestroom en kies een  $W \subseteq V$ . Dan kunnen we schrijven:

$$0 = \text{stroom uit } W - \text{stroom in } W = \sum_{i \in W, j \notin W} x_{ij} - \sum_{i \notin W, j \in W} x_{ij} \leq \sum_{i \in W, j \notin W} b_{ij} - \sum_{i \notin W, j \in W} a_{ij}.$$

Omgekeerd, stel  $\sum_{i \in W, j \notin W} b_{ij} \geq \sum_{i \notin W, j \in W} a_{ij}$ . In paragraaf 4.4 hebben we gezien dat er een toelaatbare stroom is d.e.s.d. als het hulpnetwerk  $\bar{N}$  een stroom heeft met waarde  $\sum_{i,j} a_{ij}$ .

Laat  $(\bar{W}, V - \bar{W})$  een  $(s, t)$ -snede in  $\bar{N}$  zijn en zij  $W = \bar{W} - \{s\}$ . Nu geldt:

$$\begin{aligned} c(\bar{W}) &= c(s \rightarrow V - W) + c(W \rightarrow V - W) + c(W \rightarrow t) \\ &= \sum_{i \in V, j \notin W} a_{ij} + \sum_{i \in W, j \notin W} (b_{ij} - a_{ij}) + \sum_{i \in W, j \in V} a_{ij} \\ &\geq \sum_{i \in V, j \notin W} a_{ij} + \sum_{i \notin W, j \in W} a_{ij} - \sum_{i \in W, j \notin W} a_{ij} + \sum_{i \in W, j \in V} a_{ij} = \sum_{i,j} a_{ij}. \end{aligned}$$

Dus de capaciteit van iedere snede is minstens  $\sum_{i,j} a_{ij}$ . Volgens resultaat F heeft  $\bar{N}$  een maximale stroom met waarde minstens  $\sum_{i,j} a_{ij}$ . Omdat de waarde ook niet meer kan zijn, is de waarde van de maximale stroom in  $\bar{N}$  gelijk aan  $\sum_{i,j} a_{ij}$ . ■

**Lemma 6.9** *Uit resultaat G volgt resultaat E.*

**Bewijs**

Zij  $G = (V, A)$  een gerichte graaf met  $s, t \in V$ . Altijd geldt:

het maximum aantal pijl-disjuncte  $(s, t)$ -paden  $\leq$  de minimale cardinaliteit van de  $(s, t)$ -smeden. Geef nu iedere pijl ondergrens 0 en bovengrens 1. Voeg de pijl  $(t, s)$  toe en geef deze pijl zowel de onder- als de bovengrens de waarde van de minimale cardinaliteit van de  $(s, t)$ -smeden. Nu geldt in deze nieuwe graaf voor iedere  $W \subseteq V$ :  $\sum_{i \in W, j \notin W} b_{ij} \geq \sum_{i \notin W, j \in W} a_{ij}$  (we moeten dus alleen het geval nagaan waarin  $t \notin W$  en  $s \in W$ , want alleen dan kan  $\sum_{i \notin W, j \in W} a_{ij}$  positief zijn; voor dit geval geldt:  $\sum_{i \in W, j \notin W} b_{ij} = c(W) \geq a_{ts} = \sum_{i \notin W, j \in W} a_{ij}$ ).

Volgens het resultaat van G is er dus een circulatorstroom, waaruit volgt dat de waarde van de maximale stroom gelijk is aan de minimale capaciteit van de  $(s, t)$ -smeden. Hiermee is aangetoond dat het maximum aantal pijl-disjuncte  $(s, t)$ -paden gelijk is aan de minimale cardinaliteit van de  $(s, t)$ -smeden. ■

**Gevolg**

De resultaten E, F en G zijn equivalent.

**Lemma 6.10** *Uit resultaat H volgt resultaat F.*

**Bewijs**

De waarde van de maximale stroom is gelijk aan het optimum van LP-probleem (5.11). Volgens resultaat H is de waarde ook gelijk aan het optimum van het duale probleem (5.12). Het is dus voldoende om aan te tonen dat het optimum van dit duale probleem de minimale capaciteit van de  $(1, n)$ -smeden geeft.

Voor een  $(1, n)$ -smede  $(W, V - W)$  definiëren we:  $u_i = \begin{cases} 0 & i \in W \\ 1 & i \notin W \end{cases}$  en  $w_{ij} = \begin{cases} 1 & i \in W, j \notin W \\ 0 & \text{anders.} \end{cases}$

Het is direct in te zien dat  $(u, w)$  toelaatbaar is voor (5.12) en dat de waarde van de doelfunctie gelijk is aan de capaciteit van de snede.

Omgekeerd, zij  $(u, w)$  een hoekpunt van (5.12). Omdat de matrix die bij het stelsel van (5.12) hoort totaal unimodulair is, volgt uit de regel van Cramer en het feit dat het rechterlid een eenheidsvector is dat  $(u, w)$  een  $(0,1)$ -oplossing is. Hieruit volgt onder andere:  $u_1 = 0$ ,  $u_n = 1$  en  $w_{ij} = 1$  als  $u_i = 0$  en  $u_j = 1$ . Neem  $W = \{i \mid u_i = 0\}$ , dan is  $(W, V - W)$  een  $(1, n)$ -smede en geldt voor de waarde van de doelfunctie:  $\sum_{i,j} b_{ij}w_{ij} \geq \sum_{i \in W, j \notin W} b_{ij}w_{ij} = c(W)$ . Hiermee is aangetoond dat het optimum van (5.12) de minimale capaciteit van de  $(1, n)$ -smeden geeft. ■

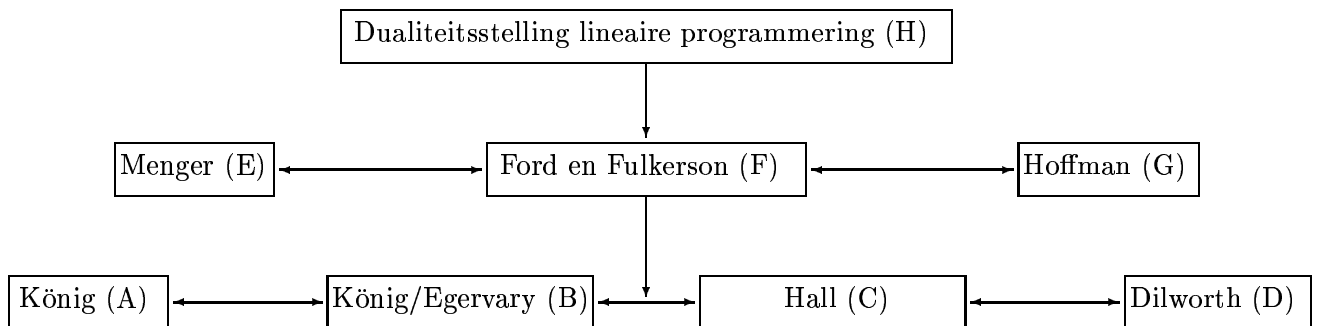
**Lemma 6.11** *Uit resultaat F volgt resultaat A.*

**Bewijs**

Zij  $G = (V_1 \cup V_2, E)$  een bipartiete graaf. Construeer als volgt netwerk  $N = (V, A)$ . Neem  $V = V_1 \cup V_2 \cup s \cup t$  en  $A = \{(s, v^1) \mid v^1 \in V_1\} \cup \{(v^1, v^2) \mid v^1 \in V_1, v^2 \in V_2\} \cup \{(v^2, t) \mid v^2 \in V_2\}$ . De pijlen van de drie deelverz. van  $A$  krijgen als capaciteit resp. 1,  $\infty$  en 1.

Merk op dat een stroom in  $N$  van  $s$  naar  $t$  correspondeert met een koppeling  $M$  in  $G$  zdd. de waarde van de stroom overeenkomt met het aantal elementen van de koppeling, en omgekeerd. Door een maximale stroom te nemen en resultaat F toe te passen krijgen we:  $\mu(G) =$  de minimale capaciteit van de  $(s, t)$ -seden in  $N$ , zeg de snede  $(W, V - W)$ . Omdat deze minimale capaciteit eindig is, bevat deze geen pijl van het type  $(v^1, v^2)$  met  $v^1 \in V_1$  en  $v^2 \in V_2$ . De verz.  $W$  is dus van het type  $s \cup W_1 \cup W_2$  met  $W_1 \subseteq V_1, W_2 \subseteq V_2$  en geen pijlen van  $W_1$  naar  $V_2 - W_2$ . De capaciteit van deze snede is  $\#(V_1 - W_1) + \#W_2$ . Merk op dat  $(V_1 - W_1) \cup W_2$  een knooppuntenbedekking is in  $G$  (anders is er immers een tak is tussen  $W_1$  en  $V_2 - W_2$ ). Hiermee is dus bewezen dat  $\mu(G) \geq \tau(G)$ . De ongelijkheid de andere kant op is triviaal (iedere tak van de koppeling heeft (minstens) één ander knooppunt nodig om de tak te bedekken). ■

De in deze paragraaf afgeleide resultaten geven we hieronder schematisch weer.



### 6.3 Koppeling met maximale cardinaliteit in bipartiete graaf

Laat  $G = (V_1 \cup V_2, E)$  een bipartiete graaf zijn. In de vorige paragraaf hebben we gezien hoe uit de Stelling van Ford en Fulkerson (resultaat F) de Stelling van König (resultaat A) volgt. Dit geeft in feite een methode om een koppeling met maximale cardinaliteit in een bipartiete graaf te vinden (zie Lemma 6.11).

Construeer het netwerk  $N = (V, A)$ , met twee extra knooppunten  $s$  en  $t$ , zoals in Lemma 6.11 is gedaan. Bepaal in  $N$  een maximale stroom  $x$ . Omdat iedere stroom uit disjuncte ketens bestaat, correspondeert de maximale stroom  $x$  met een maximale koppeling  $M$ . Aangezien aan de voorwaarden van Opgave 4.7 is voldaan, kan dit worden uitgevoerd m.b.v. de DMKM methode met complexiteit  $\mathcal{O}(m \cdot \sqrt{n})$ .

We zullen nu in enkele lemma's en een stelling bepaalde eigenschappen van koppelingen in bipartiete grafen afleiden en daarna expliciet het  $\mathcal{O}(m \cdot \sqrt{n})$  algoritme opstellen.

**Lemma 6.12** *Zij  $M_1$  en  $M_2$  twee koppelingen met  $m_1 = \#M_1 > M_2 = m_2$ . Dan geldt: (i) in  $M_1 \oplus M_2$  zijn minstens  $m_1 - m_2$  groeiketens m.b.t.  $M_2$  met disjuncte knooppunten; (ii) er is een groeiketen m.b.t.  $M_2$  met hoogstens  $\frac{m_1 + m_2}{m_1 - m_2}$  takken.*

### Bewijs

(i) De takken van  $M_1 \oplus M_2$  bestaan uit ketens en kringen met disjuncte knooppunten. Omdat de kringen evenveel takken van  $M_1$  als  $M_2$  hebben, zijn er minstens  $m_1 - m_2$  ketens met begin- en eindpunt in  $M_1$ , en dit zijn groeiketens m.b.t.  $M_2$ .

(ii) De  $m_1 - m_2$  ketens uit onderdeel (i) hebben tezamen hoogstens  $m_1 + m_2$  takken; dus er is een keten met hoogstens  $\frac{m_1+m_2}{m_1-m_2}$  takken. ■

**Lemma 6.13** *Zij  $M$  een koppeling en  $C$  een kortste groeiketen m.b.t.  $M$  en laat  $C'$  een groeiketen zijn m.b.t.  $M' = M \oplus C$ . Dan geldt:  $\#C' \geq \#C + 2 \cdot \#(C \cap C')$ .*

### Bewijs

Beschouw  $M'' = M' \oplus C'$ . Dit is een koppeling en  $\#M'' = \#M + 2$ . Volgens Lemma 6.12 zijn er in  $M'' \oplus M$  twee disjuncte groeiketens  $C_1, C_2$  m.b.t.  $M$ , zodat  $\#(M'' \oplus M) \geq \#C_1 + \#C_2 \geq 2 \cdot \#C$ .

Verder geldt:  $M'' \oplus M = [(M \oplus C) \oplus C'] \oplus M = C \oplus C'$ , zodat we kunnen schrijven:

$$2 \cdot \#C \leq \#C_1 + \#C_2 \leq \#(M'' \oplus M) = \#(C \oplus C') = \#C + \#C' - 2 \cdot \#(C \cap C'), \text{ dus:}$$

$$\#C' \geq \#C + 2 \cdot \#(C \cap C'). \quad \blacksquare$$

**Lemma 6.14** *Start met  $M_1 = \emptyset$  en neem  $M_{i+1} = M_i \oplus C_i$ , waarbij  $C_i$  de kortste groeiketen is m.b.t.  $M_i$ ,  $i = 1, 2, \dots$ . Dan geldt: (i)  $\#C_{i+1} \geq \#C_i$ ,  $i = 1, 2, \dots$ ; (ii) als  $\#C_i = \#C_j$ , dan hebben  $C_i$  en  $C_j$  disjuncte knooppunten.*

### Bewijs

(i) Uit Lemma 6.13 volgt:  $\#C_{i+1} \geq \#C_i + 2 \cdot \#(C_i \cap C_{i+1}) \geq \#C_i$ .

(ii) Stel dat  $\#C_i = \#C_j$ , met  $i < j$ , terwijl  $C_i$  en  $C_j$  niet knooppunt-disjunct zijn. Dan bestaan er indices  $k$  en  $l$ , met  $i \leq k < l \leq j$ , zdd.  $C_k$  en  $C_l$  niet knooppunt-disjunct zijn, terwijl  $C_{k+1}, C_{k+2}, \dots, C_{l-1}$  knooppunt-disjunct zijn met zowel  $C_k$  als  $C_l$ . Op grond van deze knooppunt-disjunctheid is  $C_l$  een groeiketen m.b.t.  $M_k \oplus C_k$ . Dus volgens Lemma 6.13 geldt:  $\#C_l \geq \#C_k + 2 \cdot \#(C_k \cap C_l)$ . Omdat  $\#C_k = \#C_l$  is  $C_k \cap C_l = \emptyset$ , d.w.z.  $C_k$  en  $C_l$  hebben geen takken gemeen. Maar als  $C_k$  en  $C_l$  een gemeenschappelijk knooppunt  $v$  hebben en  $C_l$  is een groeiketen m.b.t.  $M_k \oplus C_k$ , dan hebben zij de tak van  $M_k \oplus C_k$  gemeen die incident is met  $v$ . Dus  $C_k$  en  $C_l$  zijn niet tak-disjunct: tegenspraak. ■

We zullen nu ingaan op het algoritme. In het algoritme zullen we in een iteratie alle kortste groeiketens, die dus volgens Lemma 6.14 disjuncte knooppunten hebben, opnemen in een nieuwe koppeling. De volgende stelling toont aan dat er op deze wijze hoogstens  $2\sqrt{p} + 1$  iteraties zijn met  $p = \mu(G)$ .

**Stelling 6.9** *Start met  $M_1 = \emptyset$  en neem  $M_{i+1} = M_i \oplus C_i$ , waarbij  $C_i$  een kortste groeiketen is m.b.t.  $M_i$ ,  $i = 1, 2, \dots$ . Als in iedere iteratie disjuncte kortste groeiketens worden toegevoegd totdat dit niet meer mogelijk is, dan is het aantal iteraties hoogstens  $2\sqrt{p} + 1$  met  $p = \mu(G)$ .*

## Bewijs

Omdat  $p = \mu(G)$ , is het voldoende om aan te tonen dat  $\{\#C_1, \#C_2, \dots, \#C_p\}$  hoogstens  $2\sqrt{p}+1$  verschillende getallen bevat. Omdat  $C_i$  een groeiketen is, is  $\#C_i$  oneven voor iedere  $i$ . Verder is  $\#M_i = i$ ,  $i = 1, 2, \dots, p$ . Laat  $r = \lceil \sqrt{p} \rceil$  en  $q = p - r$ . Pas Lemma 6.12 toe voor de koppelingen  $M_p$  en  $M_q$ :  $\#C_q \leq \frac{p+q}{p-q} = \frac{2p-r}{r} = \frac{2p}{r} - 1$ .

Dus voor iedere  $0 \leq i \leq q$  is  $\#C_q$  één van de getallen uit  $\{1, 3, \dots, \frac{2p}{r} - 1\}$  en dit zijn hoogstens  $\frac{p}{r} \leq \sqrt{p}$  getallen. In de overige  $p - q$  getallen  $\#C_{q+1}, \#C_{q+2}, \dots, \#C_p$  komen hoogstens  $p - q = r \leq \sqrt{p} + 1$  verschillende oneven getallen voor. In totaal zijn er dus hoogstens  $2\sqrt{p} + 1$  verschillende getallen. ■

Op grond van Lemma 6.14 en Stelling 6.9 gaan we in fasen de koppeling uitbreiden, waarbij we in één fase disjuncte kortste groeiketens toevoegen totdat dit niet meer kan. Volgens Lemma 6.14 zijn deze kortste groeiketens disjunct. Het zijn dus ook groeiketens t.o.v. de koppeling waarmee de fase begint. Dit geeft aanleiding tot het volgende algoritme.

*Algoritme 6.1: Algoritme om een koppeling met maximale cardinaliteit te bepalen in een bipartiete graaf*

1. Start met  $M = \emptyset$ .
2. Bepaal, met zijwaarts zoeken vanuit de vrije knooppunten van  $V_1$ , naar ieder knooppunt dat met een wisselketen bereikbaar is een kortste groeiketen m.b.t.  $M$  totdat we de laag hebben gehad waarin we ten minste één vrij knooppunt van  $V_2$  hebben aangetroffen. De bijbehorende gerichte deelgraaf noteren we met  $G^*$  en de verz. van vrije knooppunten in deze laatste laag noteren we met  $X$ .  
(deze stap wordt hieronder nader uitgewerkt en daarin wordt o.a. met deelverz.  $S_k$  gewerkt, waarbij  $S_k \subseteq V_1$  voor  $k$  oneven en  $S_k \subseteq V_2$  voor  $k$  even).  
Als er geen groeiketen is:  $M$  is een koppeling met maximale cardinaliteit en  $W_1 \cup W_2$  is een knooppuntenbedekking met minimale cardinaliteit, waarbij  $W_1 = \{i \in V_1 \mid i \notin S_1 \cup S_3 \cup \dots\}$  en  $W_2 = \{j \in V_2 \mid j \in S_2 \cup S_4 \cup \dots\}$  (stop).
3. Bepaal in  $G^*$ , door vanuit een gevonden vrij knooppunt van  $X$  terug te lopen totdat we in een vrij knooppunt van  $S_1$  komen, een (met vorige groeiketens van deze iteratie disjuncte) groeiketen  $C$  (ook deze stap wordt hieronder verder uitgewerkt).  
Laat  $M := M \oplus C$  en herhaal stap 3 totdat het niet meer mogelijk is een nieuwe disjuncte groeiketen te vinden.
4. Ga naar stap 2.

*Uitwerking stap 2 van Algoritme 6.1*

1.  $S_1 = \{\text{vrije knooppunten van } V_1\}$ ;  $k = 1$ ;  $X = \emptyset$ .
2. a.  $S_{k+1} = \emptyset$ .  
b. Voor iedere  $v_i \in S_k$  doe:  
voor iedere  $v_j \in V_2$  met  $(v_i, v_j) \in E \setminus M$  doe:  
als  $v_j \notin \bigcup_{j=1}^k S_j$ : voeg  $v_j$  aan  $S_{k+1}$  toe en neem  $(v_i, v_j)$  op in  $G^*$ .

- c.  $k := k + 1$  en ga naar stap 3.
3. a. Als  $S_k = \emptyset$ : er is geen groeiketen (stop);  
 Anders: (i)  $S_{k+1} = \emptyset$ ;  
 (ii) voor iedere  $v_i \in S_k$  doe:  
 als  $v_i$  vrij is ( $v_i$  is eindpunt van een kortste groeiketen): voeg  $v_i$  aan  $X$  toe;  
 anders: laat  $v_j \in V_1$  zdd.  $(v_i, v_j) \in M$ : voeg  $v_j$  aan  $S_{k+1}$  toe;  
 neem  $(v_i, v_j)$  op in  $G^*$ .
- b. Als  $X = \emptyset$ :  $k := k + 1$  en ga naar stap 2;  
 Anders: einde stap 2 van algoritme 6.1.

*Uitwerking stap 3 van Algoritme 6.1* (laat  $A$  de pijlenverz. van  $G^*$  zijn).

1.  $l = 0$ ; kies een  $w \in X$  die nog niet gelabeld is met 'onderzocht' (als zo'n  $w$  niet bestaat: stop).
2. Label  $w$  met 'onderzocht';  $l := l + 1$ ;  $C[l] = w$ .
3. Kies een nog niet met 'onderzocht' gelabelde pijl naar  $w$ , zeg  $(v, w) \in A$  en ga naar stap 4.  
 Als zo'n pijl niet bestaat: als  $l = 1$ : ga naar stap 1;  
 anders:  $l := l - 2$ , laat  $w = C[l]$  en ga naar stap 3.
4. Label  $(v, w)$  met 'onderzocht'.  
 Als  $v$  reeds met 'onderzocht' was gelabeld: ga naar stap 3;  
 Anders: label  $v$  met 'onderzocht';  $l := l + 1$ ;  $C[l] = v$ .
5. Als  $v$  vrij is: ga naar stap 6;  
 Anders: laat  $w$  zdd.  $(w, v) \in M$  en ga naar stap 2.
6.  $C = \{C[1], C[2], \dots, C[l]\}$ ;  $M := M \oplus C$ ; ga naar stap 1.

### Voorbeeld 6.1

Beschouw nevenstaande bipartiete graaf.

*Iteratie 1:*

Stap 2:

$S_1 = \{1, 2, 3, 4, 5\}$ ;  $k = 1$ ;  $X = \emptyset$ ;  $A = \emptyset$ ;  $S_2 = \emptyset$ .

$v_1$ :  $S_2 = \{6, 9\}$ ;  $A := \{(1, 6), (1, 9)\}$ .

$v_2$ :  $S_2 = \{6, 9, 7, 10\}$ ;  $A := A \cup \{(2, 7), (2, 10)\}$ .

$v_3$ :  $S_2 = \{6, 9, 7, 10, 8\}$ ;  $A := A \cup \{(3, 8), (3, 10)\}$ .

$v_4$ :  $A := A \cup \{(4, 6), (4, 7)\}$ .

$v_5$ :  $A := A \cup \{(5, 8), (5, 9)\}$ .

$k = 2$ ;  $S_3 = \emptyset$ ;  $X = \{6, 9, 7, 10, 8\}$ .

Stap 3:

$l = 0$ ;  $w = 6$ ;  $v_6$  onderzocht;  $l = 1$ ;  $C[1] = 6$ ;  $v = 1$ ;  $(1, 6)$  onderzocht;  $v_1$  onderzocht;

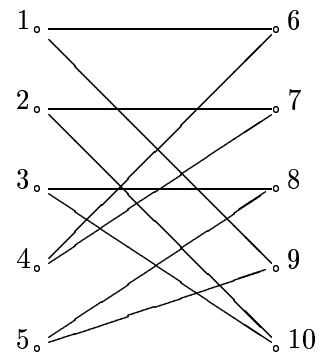
$l = 2$ ;  $C[2] = 1$ ;  $C = \{(6, 1)\}$ ;  $M = \{(1, 6)\}$ .

$l = 0$ ;  $w = 9$ ;  $v_9$  onderzocht;  $l = 1$ ;  $C[1] = 9$ ;  $v = 1$ ;  $(1, 9)$  onderzocht;  $v = 5$ ;  $(5, 9)$  onderzocht;

$v_5$  onderzocht;  $l = 2$ ;  $C[2] = 5$ ;  $C = \{(9, 5)\}$ ;  $M = \{(1, 6), (5, 9)\}$ .

$l = 0$ ;  $w = 7$ ;  $v_7$  onderzocht;  $l = 1$ ;  $C[1] = 7$ ;  $v = 2$ ;  $(2, 7)$  onderzocht;  $v_2$  onderzocht;

$l = 2$ ;  $C[2] = 2$ ;  $C = \{(7, 2)\}$ ;  $M = \{(1, 6), (5, 9), (2, 7)\}$ .



$l = 0$ ;  $w = 10$ ;  $v_{10}$  onderzocht;  $l = 1$ ;  $C[1] = 10$ ;  $v = 2$ ;  $(2, 10)$  onderzocht;  $v = 3$ ;  
 $(3, 10)$  onderzocht;  $v_3$  onderzocht;  $l = 2$ ;  $C[2] = 3$ ;  $C = \{(10, 3)\}$ ;  $M = \{(1, 6), (5, 9), (2, 7), (3, 10)\}$ .  
 $l = 0$ ;  $w = 8$ ;  $v_8$  onderzocht;  $l = 1$ ;  $C[1] = 8$ ;  $v = 3$ ;  $(3, 8)$  onderzocht;  $v = 5$ ;  $(5, 9)$  onderzocht;  
 $l = 0$ . Einde iteratie

*Iteratie 2:*

Stap 2:

$S_1 = \{4\}$ ;  $k = 1$ ;  $X = \emptyset$ ;  $S_2 = \emptyset$ ;  $v_4 : S_2 = \{6, 7\}$ ;  $A = \{(4, 6), (4, 7)\}$ ;  $k = 2$ ;  $S_3 = \emptyset$ .  
 $v_6 : S_3 = \{1\}$ ;  $A := A \cup (6, 1)$ ;  $v_1 : S_4 = \{9\}$ ;  $A := A \cup (1, 9)$ ;  $v_2 : S_4 = \{9, 10\}$ ;  $A := A \cup (2, 10)$ .  
 $k = 4$ ;  $S_5 = \emptyset$ .  
 $v_9 : S_5 = \{5\}$ ;  $A := A \cup (9, 5)$ ;  $v_{10} : S_5 = \{5, 3\}$ ;  $A := A \cup (10, 3)$ .  
 $k = 5$ ;  $S_6 = \emptyset$ .  
 $v_5 : S_6 = \{8\}$ ;  $A := A \cup (5, 8)$ ;  $v_3 : S_6 = \{8\}$ ;  $A := A \cup (3, 8)$ .  
 $k = 6$ ;  $S_7 = \emptyset$ .  
 $v_8 : X = \{8\}$ .

Stap 3:

$l = 0$ ;  $w = 8$ ;  $v_8$  onderzocht;  $l = 1$ ;  $C[1] = 8$ ;  $v = 5$ ;  $(5, 8)$  onderzocht;  $v_5$  onderzocht;  
 $l = 2$ ;  $C[2] = 5$ ;  $w = 9$ ;  $v_9$  onderzocht;  $l = 3$ ;  $C[3] = 9$ ;  $v = 1$ ;  $(1, 9)$  onderzocht;  
 $v_1$  onderzocht;  $l = 4$ ;  $C[4] = 1$ ;  $w = 6$ ;  $v_6$  onderzocht;  $l = 5$ ;  $C[5] = 6$ ;  $v = 4$ ;  $(4, 6)$  onderzocht;  
 $v_4$  onderzocht;  $l = 6$ ;  $C[6] = 4$ ;  $C = \{8, 5, 9, 1, 6, 4\}$ ;  $M = \{(4, 6), (1, 9), (5, 8), (2, 7), (3, 6)\}$ .  
 $l = 0$ . Einde iteratie

*Iteratie 3:*

$S_1 = \emptyset$ ;  $k = 1$ ;  $X = \emptyset$ ;  $S_2 = \emptyset$ . Stop.

$M$  is een koppeling met maximale cardinaliteit.  $W = V_1$  is een knooppuntenbedekking met minimale cardinaliteit.

**Stelling 6.10** *Het algoritme is correct en heeft complexiteit  $\mathcal{O}(m \cdot \sqrt{n})$ .*

**Bewijs**

Het algoritme stopt als er geen groeiketen meer is: de koppeling is dus maximaal.

De verkregen verz.  $W_1 \cup W_2$  is een knooppuntenbedekking, immers:

Stel niet, d.w.z. er is een  $(v_i, v_j) \in E$  met  $v_i \notin W_1$  en  $v_j \notin W_2$ . Knooppunt  $v_i$  is dus via een wisselketen bereikbaar vanuit een vrij knooppunt van  $V_1$  (eindigend met een tak van  $M$ ) en  $v_j$  kan niet worden bereikt: tegenspraak.

$|W_1 \cup W_2| = |M|$ , d.w.z.  $W_1 \cup W_2$  is een knooppuntenbedekking met minimale cardinaliteit, nl:

Stel  $V_1$  heeft  $r$  vrije knooppunten en het  $i$ -de vrije knooppunt genereert een wisselboom (laat in de graaf  $G^*$  takken weg die een kring genereren en niet tot de koppeling behoren weg (dit heeft geen invloed op de bereikbaarheid) met  $t_i$  takken. Deze wisselboom heeft  $t_i + 1$

knooppunten, waarvan er  $\frac{1}{2}t_i + 1$  tot  $V_1$  en  $\frac{1}{2}t_i$  tot  $V_2$  behoren. Er geldt nu:

$$\#M = \#V_1 - r = \#V_1 - \sum_{i=1}^r [\frac{1}{2}t_i + 1] + \sum_{i=1}^r [\frac{1}{2}t_i] = \#W_1 + \#W_2.$$

Het aantal iteraties is hoogstens  $2\sqrt{p} + 1$  met  $p = \#M$ , dus het aantal iteraties is  $\mathcal{O}(\sqrt{n})$ . Het werk per iteratie is  $\mathcal{O}(m)$  (ga dit zelf na), dus in totaal is de complexiteit  $\mathcal{O}(m \cdot \sqrt{n})$ . ■



## 6.4 Koppeling met maximale gewicht in bipartiete graaf

In deze paragraaf is aan iedere tak  $(v_i, v_j)$  van de bipartiete graaf een gewicht  $w_{ij} \geq 0$  toegekend. We zijn geïnteresseerd in een koppeling met maximaal gewicht. Als  $w_{ij} = 1$  voor iedere  $(v_i, v_j)$ , dan hebben we het model van de vorige paragraaf waarin een koppeling met maximale cardinaliteit werd gezocht.

De LP-formulering van dit probleem is:

$$\max \left\{ \sum_{(i,j) \in E} w_{ij} x_{ij} \left| \begin{array}{l} \sum_j x_{ij} \leq 1, \quad i \in V_1 \\ \sum_i x_{ij} \leq 1, \quad j \in V_2 \\ x_{ij} \geq 0, \quad (i,j) \in E \end{array} \right. \right\} \quad (6.3)$$

De coëfficiëntenmatrix van (6.3) is de incidentiematrix van de graaf  $G$ . Omdat deze totaal unimodulair is (Stelling 1.3) zijn de hoekpunten van (6.1) geheeltallig. Iedere basisoplossing is dus een  $(0, 1)$ -oplossing en deze correspondeert met een koppeling.

Het duale probleem van (6.3) is:

$$\min \left\{ \sum_{i \in V_1} s_i + \sum_{j \in V_2} t_j \left| \begin{array}{l} s_i + t_j \geq w_{ij}, \quad (i,j) \in E \\ s_i, t_j \geq 0, \quad i \in V_1, j \in V_2 \end{array} \right. \right\} \quad (6.4)$$

De bij dit paar LP-problemen behorende *orthogonaliteitsrelaties* zijn:

$$x_{ij} \cdot [s_i + t_j - w_{ij}] = 0 \text{ voor alle } i, j \in E \quad (6.5)$$

$$s_i \cdot [1 - \sum_j x_{ij}] = 0 \text{ voor alle } i \in V_1 \quad (6.6)$$

$$t_j \cdot [1 - \sum_i x_{ij}] = 0 \text{ voor alle } j \in V_2 \quad (6.7)$$

We zullen het probleem om een koppeling met maximaal gewicht te vinden oplossen met de zogenaamde *Hongaarse methode*. Hierbij hebben we in iedere iteratie een toelaatbare oplossing  $x$  van (6.3), een toelaatbare oplossing  $(s, t)$  van (6.4) zdd. (6.5) en (6.7) gelden. Uit de theorie van de lineaire programmering volgt dat, indien ook aan (6.6) is voldaan, de oplossingen  $x$  en  $(s, t)$  optimaal zijn voor resp. (6.3) en (6.4).

We starten met  $x = 0$ ,  $t = 0$  en  $s_i = \max_{(i,j)} w_{ij}$ ,  $i \in V_1$ . Het is direct duidelijk dat deze startoplossingen voldoen aan de gestelde eisen. De algemene iteratiestap ziet er als volgt uit.

Beschouw de deelgraaf met als takkenverz.  $E' = \{(i, j) \mid s_i + t_j = w_{ij}\}$ . Voor deze takken mag  $x_{ij} = 1$  zijn (vgl. (6.5)), d.w.z. dat ze in de koppeling kunnen komen. In deze deelgraaf gaan we disjuncte kortste groeiketens zoeken met de techniek uit de vorige paragraaf. Met deze groeiketens kan de koppeling worden uitgebreid. Hierdoor verandert  $x$  op een aantal takken van  $E'$  en  $(s, t)$  blijft onveranderd. Merk op dat daarmee de vereiste eigenschappen blijven gelden. Als  $V_1$  geen vrije knooppunten meer heeft of als  $s_i = 0$  voor alle vrije knooppunten van  $V_1$ , dan is ook aan (6.6) voldaan, zodat de oplossingen weer optimaal zijn. Indien  $V_1$  wel een vrij knooppunt  $v_i$  heeft met  $s_i > 0$  en er in de deelgraaf met takken  $E'$  geen groeiketen meer is, dan gaan we

$(s, t)$  aanpassen zdd. nog steeds aan de vereiste voorwaarden wordt voldaan en  $E'$  groter wordt (dit kan slechts eindig vaak gebeuren).

Aan het einde van de procedure om een groeiketen te vinden hebben we  $W_1$  en  $W_2$ , de via wisselketens bereikbare knooppunten van  $V_1$  resp.  $V_2$ . Bepaal nu de volgende getallen.

$$\pi_j = \min_{i \in W_1} [s_i + t_j - w_{ij}], \quad j \notin W_2; \quad \delta_1 = \min_{j \notin W_2} \pi_j; \quad \delta_2 = \min_{i \in W_1} s_i; \quad \delta = \min(\delta_1, \delta_2).$$

Als  $i \in W_1$  en  $j \notin W_2$ , dan is de tak  $(i, j)$  geen tak van de koppeling (anders was  $i$  via  $j$  bereikt, zodat  $j \in W_2$ ) en behoort deze ook niet tot  $E' \setminus M$  (anders was  $j$  uit  $i$  bereikbaar, zodat eveneens  $j \in W_2$ ). Dus  $(i, j) \notin E'$  en hieruit volgt dat  $\pi_j > 0$ ,  $j \notin W_2$ , zodat  $\delta_1 > 0$ .

Uit de beginwaarde van  $s$  en de aanpassing die hieronder wordt gegeven volgt dat vrije knooppunten een minimale  $s$ -waarde hebben; deze is positief omdat als  $s_i = 0$  voor alle vrije knooppunten we optimale oplossingen hebben. Dus ook  $\delta_2 > 0$ , waarmee  $\delta > 0$  is.

We passen nu als volgt de oplossing  $(s, t)$  aan:  $s_i := s_i - \delta$ ,  $i \in W_1$ ;  $t_j := t_j + \delta$ ,  $j \in W_2$ . Deze nieuwe oplossing voldoet weer aan de gestelde eisen (ga dit zelf na). Als  $\delta = \delta_2$ , dan wordt  $s_i = 0$  voor alle vrije knooppunten en zijn de oplossingen optimaal.

Als  $\delta = \delta_1 < \delta_2$ , dan is er een  $i \in W_1$  en een  $j \notin W_2$  waarvoor  $s_i + t_j$  gelijk aan  $w_{ij}$  wordt;  $E'$  wordt dus uitgebreid en  $j$  wordt hiermee bereikbaar. In  $E'$  zijn meer knooppunten bereikbaar (als  $i \notin W_1$  en  $j \in W_2$ , dan wordt  $s_i + t_j$  groter, zodat  $(i, j)$  uit  $E'$  kan verdwijnen; echter: omdat  $i \notin W_1$  en  $j \in W_2$ , is  $i$  niet vrij en zit  $(i, j)$  niet in de koppeling:  $(i, j)$  komt niet voor in een groeiketen m.b.t.  $M$ , dus deze tak is niet van belang). Vervolgens wordt het zoeken naar een groeiketen voortgezet en wordt bovenstaand procédé herhaald. Na een eindig aantal iteraties wordt ofwel een groeiketen gevonden ofwel wordt ontdekt dat de koppeling optimaal is. Samengevat is het algoritme als volgt.

*Algoritme 6.2: Algoritme om een koppeling met maximale gewicht te bepalen in bipartiete graaf*

1.  $M = \emptyset$ ;  $t_j = 0, j \in V_2$ ;  $s_i = \max_{(k,l)} w_{kl}, i \in V_1$ ;  $E' = \{(i, j) \mid s_i + t_j = w_{ij}\}$ ;  $G' = (V_1 \cup V_2, E')$ .
2. a. Bepaal in  $G'$  met zijwaarts zoeken disjuncte kortste groeiketens m.b.t.  $M$ , zeg  $C_1, C_2, \dots, C_k$ .  
b. Als er geen groeiketen is: ga naar stap 3.  
Anders:  $M := M \oplus C_1 \oplus C_2 \cdots \oplus C_k$  en ga naar stap 2.
3. Als  $V_1$  geen vrije knooppunten heeft: stop ( $M$  is een koppeling met maximaal gewicht).  
Anders: ga naar stap 4.
4. Laten  $W_1$  en  $W_2$  de in  $G'$  bereikbare knooppunten van  $V_1$  resp.  $V_2$  zijn.
  - a.  $\pi_j = \min_{i \in W_1} [s_i + t_j - w_{ij}], j \notin W_2$ ;  $\delta_1 = \min_{j \notin W_2} \pi_j$ ;  $\delta_2 = \min_{i \in W_1} s_i$ ;  $\delta = \min(\delta_1, \delta_2)$ ;
  - b.  $s_i := s_i - \delta, i \in W_1$ ;  $t_j := t_j + \delta, j \in W_2$ .
  - c. Als  $\delta = \delta_2$ : stop ( $M$  is een koppeling met maximaal gewicht);  
Anders:  $F' = \{(i, j) \notin E' \mid s_i + t_j = w_{ij} \text{ met } i \in W_1, j \notin W_2\}$ ;  
 $F'' = \{(i, j) \in E' \mid s_i + t_j > w_{ij} \text{ met } i \notin W_1, j \in W_2\}$ ;  
 $E' := E' + F' - F''$ ;
  - d. Ga verder in stap 2 met het zoeken van groeiketens door  $F'$  aan de wisselboom toe te voegen.

### Voorbeeld 6.2

We gaan een koppeling met maximaal gewicht bepalen in de volledige bipartiete graaf  $K_{5,5}$ . De gewichten staan in nevenstaande  $5 \times 5$  matrix  $W$ . De knooppunten van  $V_1$  nummeren we met 1 t/m 5 en die van  $V_2$  met 6 t/m 10.

$$\begin{pmatrix} 3 & 8 & 9 & 1 & 6 \\ 1 & 4 & 1 & 5 & 5 \\ 7 & 2 & 7 & 9 & 2 \\ 3 & 1 & 6 & 8 & 8 \\ 2 & 6 & 3 & 6 & 2 \end{pmatrix}$$

*Iteratie 1:*

$M = \emptyset$ ;  $s_i = 9$ ,  $1 \leq i \leq 5$ ;  $t_j = 0$ ,  $6 \leq j \leq 10$ ;  $E' = \{(1, 8), (3, 9)\}$ .

$C_1 = [1, 8]$ ;  $C_2 = [3, 9]$ ;  $M = \{(1, 8), (3, 9)\}$ .

*Iteratie 2:*

Er worden geen groeiketens gevonden.  $W_1 = \{2, 4, 5\}$ ,  $W_2 = \emptyset$ .

$\pi_6 = 6$ ,  $\pi_7 = 3$ ,  $\pi_8 = 3$ ,  $\pi_9 = 1$ ,  $\pi_{10} = 1$ ;  $\delta_1 = 1$ ;  $\delta_2 = 9$ ;  $\delta = 1$ ;  $s_2 = 8$ ,  $s_4 = 8$ ,  $s_5 = 8$ .

$F' = \{(4, 9), (4, 10)\}$ ;  $F'' = \emptyset$ ;  $E' = \{(1, 8), (3, 9), (4, 9), (4, 10)\}$ .

$C_1 = [4, 10]$ ;  $M = \{(1, 8), (3, 9), (4, 10)\}$ .

*Iteratie 3:*

Er worden geen groeiketens gevonden.  $W_1 = \{2, 5\}$ ,  $W_2 = \emptyset$ .

$\pi_6 = 6$ ,  $\pi_7 = 2$ ,  $\pi_8 = 5$ ,  $\pi_9 = 2$ ,  $\pi_{10} = 3$ ;  $\delta_1 = 2$ ;  $\delta_2 = 8$ ;  $\delta = 2$ ;  $s_2 = 6$ ,  $s_5 = 6$ .

$F' = \{(5, 7), (5, 9)\}$ ;  $F'' = \emptyset$ ;  $E' = \{(1, 8), (3, 9), (4, 9), (4, 10), (5, 7), (5, 9)\}$ .

$C_1 = [5, 7]$ ;  $M = \{(1, 8), (3, 9), (4, 10), (5, 7)\}$ .

*Iteratie 4:*

Er worden geen groeiketens gevonden.  $W_1 = \{2\}$ ,  $W_2 = \emptyset$ .

$\pi_6 = 5$ ,  $\pi_7 = 2$ ,  $\pi_8 = 5$ ,  $\pi_9 = 1$ ,  $\pi_{10} = 1$ ;  $\delta_1 = 1$ ;  $\delta_2 = 6$ ;  $\delta = 1$ ;  $s_2 = 5$ .

$F' = \{(2, 9), (2, 10)\}$ ;  $F'' = \emptyset$ ;  $E' = \{(1, 8), (3, 9), (4, 9), (4, 10), (5, 7), (5, 9), (2, 9), (2, 10)\}$ .

Er worden geen groeiketens gevonden.  $W_1 = \{2, 3, 4\}$ ,  $W_2 = \{9, 10\}$ .

$\pi_6 = 2$ ,  $\pi_7 = 1$ ,  $\pi_8 = 2$ ;  $\delta_1 = 1$ ;  $\delta_2 = 5$ ;  $\delta = 1$ ;  $s_2 = 4$ ,  $s_3 = 8$ ,  $s_4 = 7$ ;  $t_9 = 1$ ,  $t_{10} = 1$ .

$F' = \{(2, 7)\}$ ;  $F'' = \{(5, 9)\}$ ;  $E' = \{(1, 8), (3, 9), (4, 9), (4, 10), (5, 7), (2, 9), (2, 10), (2, 7)\}$ .

Er worden geen groeiketens gevonden.  $W_1 = \{2, 3, 4, 5\}$ ,  $W_2 = \{7, 9, 10\}$ .

$\pi_6 = 1$ ,  $\pi_8 = 1$ ;  $\delta_1 = 1$ ;  $\delta_2 = 4$ ;  $\delta = 1$ ;  $s_2 = 3$ ,  $s_3 = 7$ ,  $s_4 = 6$ ,  $s_5 = 5$ ;  $t_7 = 1$ ,  $t_9 = 2$ ,  $t_{10} = 2$ .

$F' = \{(3, 6), (3, 8), (4, 8)\}$ ;  $F'' = \emptyset$ ;

$E' = \{(1, 8), (3, 9), (4, 9), (4, 10), (5, 7), (2, 9), (2, 10), (2, 7), (3, 6), (3, 8), (4, 8)\}$ .

$C_1 = [2, 9, 3, 6]$ ;  $M = \{(1, 8), (2, 9), (4, 10), (5, 7), (3, 6)\}$ .

*Iteratie 5:*

Er worden geen groeiketens gevonden.  $V_1$  heeft geen vrije knooppunten:  $M$  is een koppeling met maximaal gewicht en  $(s, t)$  een optimale oplossing van het duale LP-probleem; het optimale gewicht is 35 en dat is gelijk aan  $\sum_i s_i + \sum_j t_j$ .

**Stelling 6.11** *Het algoritme van de Hongaarse methode heeft complexiteit  $\mathcal{O}(n^3)$ .*

### Bewijs

In iedere iteratie wordt de koppeling minstens één element groter: het aantal iteraties is  $\mathcal{O}(n)$ .

Het werk per iteratie splitsen we op in:

- werk om de wisselketens en groeiketens te construeren:  $\mathcal{O}(m) = \mathcal{O}(n^2)$ ;
  - werk aan  $E'$ : omdat per iteratie iedere tak ofwel aan  $E'$  wordt toegevoegd, ofwel uit  $E'$  wordt verwijderd (ofwel er gebeurt niets mee), heeft dit complexiteit  $\mathcal{O}(m) = \mathcal{O}(n^2)$ ;
  - aanpassing van de  $s_i$ 's en  $t_j$ 's: per deeliteratie is de complexiteit  $\mathcal{O}(n)$  en het aantal deeliteraties is  $\mathcal{O}(n)$  (want per deeliteratie wordt  $W_1 \cup W_2$  echt groter);
  - bepaling  $\delta_1$ : bepaal  $\pi_j = \min_{i \in W_1} \{s_i + t_j - w_{ij}\}, j \notin W_2$  als volgt:
    - \* bij de eerste deeliteratie berekenen we dit 'gewoon': dit geeft complexiteit  $\mathcal{O}(n^2)$ ;
    - \* bij de volgende deeliteraties wordt  $\pi_j$  met  $\delta$  verkleind (dus voor alle  $\pi_j$ 's complexiteit  $\mathcal{O}(n)$  per deeliteratie), ofwel als  $i \in V_1$  aan  $W_1$  wordt toegevoegd, dan berekenen wij:  $\pi_j := \min\{\pi_j, s_i + t_j - w_{ij}\}$ , wat  $\mathcal{O}(1)$  is; in totaal is dit werk per iteratie dus ook  $\mathcal{O}(n^2)$ .
  - bepaling  $\delta_2$ : per deeliteratie heeft het werk complexiteit  $\mathcal{O}(n)$ , dus per iteratie  $\mathcal{O}(n^2)$ ;
- De totale complexiteit van de methode is dus  $\mathcal{O}(n^3)$ . ■

## 6.5 Gilmore-Gomory en Gale-Shapley koppelingen

Bschouw de volledige bipartiete graaf  $K_{n,n}$  met knooppuntenverz.  $V_1 \cup V_2$  en met gewichten  $w_{ij}$ . Laat  $V_1 = \{v_1, v_2, \dots, v_n\}$  en  $V_2 = \{w_1, w_2, \dots, w_n\}$ .

Veronderstel dat bij iedere  $v_i \in V_1$  en iedere  $w_j \in V_2$  reële getallen  $\alpha_i$  resp.  $\beta_j$  behoren en dat de knooppunten zó genummerd zijn dat  $\alpha_1 \geq \alpha_2 \geq \dots \geq \alpha_n$  en  $\beta_1 \geq \beta_2 \geq \dots \geq \beta_n$ .

We spreken van een *Gilmore-Gomory graaf* als de gewichten  $w_{ij}$  voldoen aan:

$$w_{ij} = \begin{cases} \int_{\alpha_i}^{\beta_j} f(y) dy & \text{als } \beta_j \geq \alpha_i \\ \int_{\beta_j}^{\alpha_i} g(y) dy & \text{als } \beta_j < \alpha_i, \end{cases}$$

waarbij  $f(y)$  en  $g(y)$  gegeven functies zijn met  $f(y) + g(y) \geq 0$  voor alle  $y$ .

**Stelling 6.12** (i) De koppeling  $M_1 = \{(v_i, w_i), 1 \leq i \leq n\}$  is een volmaakte koppeling met minimaal gewicht; (ii) de koppeling  $M_2 = \{(v_i, w_{n+1-i}), 1 \leq i \leq n\}$  is een volmaakte koppeling met maximaal gewicht.

### Bewijs

(i) Zij  $M \neq M_1$  een willekeurige volmaakte koppeling. Dan zijn er in  $M$  takken  $(v_i, w_j)$  en  $(v_k, w_l)$  met  $i < k$  en  $j > l$ . Beschouw  $M' = M \cup (v_k, w_j) \cup (v_i, w_l) - (v_i, w_j) - (v_k, w_l)$ .

Laat  $\Delta w = \text{gewicht}(M') - \text{gewicht}(M)$ , dan zullen we aantonen dat  $\Delta w \leq 0$ .

Als  $\alpha_k \geq \beta_l$ :

$$\Delta w = \int_{\beta_j}^{\alpha_k} g(y) dy + \int_{\beta_l}^{\alpha_i} g(y) dy - \int_{\beta_j}^{\alpha_i} g(y) dy - \int_{\beta_l}^{\alpha_k} g(y) dy = 0.$$

Als  $\beta_j \leq \alpha_k \leq \beta_l \leq \alpha_i$ :

$$\Delta w = \int_{\beta_j}^{\alpha_k} g(y) dy + \int_{\beta_l}^{\alpha_i} g(y) dy - \int_{\beta_j}^{\alpha_i} g(y) dy - \int_{\alpha_k}^{\beta_l} f(y) dy = - \int_{\alpha_k}^{\beta_l} [g(y) + f(y)] dy \leq 0.$$

Als  $\alpha_k \leq \beta_j \leq \beta_l \leq \alpha_i$ :

$$\Delta w = \int_{\alpha_k}^{\beta_j} f(y)dy + \int_{\beta_l}^{\alpha_i} g(y)dy - \int_{\beta_j}^{\alpha_i} g(y)dy - \int_{\alpha_k}^{\beta_l} f(y)dy = - \int_{\beta_j}^{\beta_l} [g(y) + f(y)]dy \leq 0.$$

Als  $\beta_j \leq \alpha_k \leq \alpha_i \leq \beta_l$ :

$$\Delta w = \int_{\beta_j}^{\alpha_k} g(y)dy + \int_{\alpha_i}^{\beta_l} f(y)dy - \int_{\beta_j}^{\alpha_i} g(y)dy - \int_{\alpha_k}^{\beta_l} f(y)dy = - \int_{\alpha_k}^{\alpha_i} [g(y) + f(y)]dy \leq 0.$$

Als  $\alpha_k \leq \beta_j \leq \alpha_i \leq \beta_l$ :

$$\Delta w = \int_{\alpha_k}^{\beta_j} f(y)dy + \int_{\alpha_i}^{\beta_l} f(y)dy - \int_{\beta_j}^{\alpha_i} g(y)dy - \int_{\alpha_k}^{\beta_l} f(y)dy = - \int_{\beta_j}^{\alpha_i} [g(y) + f(y)]dy \leq 0.$$

Als  $\alpha_k \leq \alpha_i \leq \beta_j \leq \beta_l$ :

$$\Delta w = \int_{\alpha_k}^{\beta_j} f(y)dy + \int_{\alpha_i}^{\beta_l} f(y)dy - \int_{\alpha_i}^{\beta_j} f(y)dy - \int_{\alpha_k}^{\beta_l} f(y)dy = 0.$$

Aldus zien we dat een koppeling zonder gekruiste takken, d.w.z. de koppeling  $M_1$ , minimaal gewicht heeft.

(ii) Dit deel gaat analoog: niet gekruiste takken vervangen we door gekruiste takken, waardoor het gewicht niet kleiner wordt. De koppeling  $M_2$  heeft dus maximaal gewicht. ■

Beschouw een groep personen bestaande uit  $n$  mannen en  $n$  vrouwen. Iedere man rangschikt de vrouwen naar zijn voorkeur, en evenzo rangschikt iedere vrouw demannen naar haar voorkeur. Een volmaakte koppeling heet *instabiel* als er een man en een vrouw zijn die niet aan elkaar gekoppeld zijn, maar die beiden elkaar verkiezen boven hun huidige partners. Een stabiele koppeling, d.w.z. een koppeling die niet instabiel is, heet een *Gale-Shapley koppeling* als iedere man in iedere andere stabiele koppeling niet beter af is, d.w.z. niet gekoppeld is aan een vrouw die de voorkeur heeft boven zijn huidige partner.

Een Gale-Shapley koppeling kan als volgt worden gevonden (het correctheidsbewijs volgt later).

*Algoritme 6.3: Algoritme om een Gale-Shapley koppeling te bepalen*

Voor  $i = 1, 2, \dots, n$ :

Koppel de  $i$ -de man als volgt:

- a. Geef man  $i$  de vrouw van zijn volgende voorkeur (begin met de hoogste voorkeur), zeg dit is vrouw  $j$ .
- b. Als vrouw  $j$  man  $i$  accepteert, d.w.z. ze was òfwel nog niet gekoppeld òfwel ze verkiest man  $i$  boven haar huidige partner:
  - \* koppel man  $i$  en vrouw  $j$ ;
  - \* als vrouw  $j$  haar huidige partner, zeg man  $k$ , verlaat:
    - probeer man  $k$  te koppelen volgens dezelfde procedure.

Als vrouw  $j$  man  $i$  niet accepteert, d.w.z. als ze aan een man is gekoppeld die haar voorkeur heeft boven man  $i$ : ga naar a.

### Voorbeeld 6.3

Beschouw onderstaande voorkeursmatrices:  $M$  voor de mannen en  $V$  voor de vrouwen:

$$M = \begin{pmatrix} 2 & 5 & 1 & 3 & 4 \\ 1 & 2 & 3 & 4 & 5 \\ 2 & 3 & 5 & 4 & 1 \\ 1 & 3 & 2 & 4 & 5 \\ 5 & 3 & 2 & 1 & 4 \end{pmatrix} \text{ en } V = \begin{pmatrix} 2 & 4 & 5 & 3 & 1 \\ 4 & 3 & 5 & 1 & 2 \\ 1 & 3 & 4 & 2 & 5 \\ 4 & 2 & 1 & 3 & 5 \\ 5 & 2 & 3 & 1 & 4 \end{pmatrix}.$$

De betekenis hiervan is als volgt:  $m_{ij} = k$  betekent dat vrouw  $k$  bij man  $i$  op de  $j$ -de plaats komt; man 1 heeft dus als voorkeursvolgorde: vrouw 2, vrouw 5, vrouw 1, vrouw 3 en als laatste vrouw 4.

Voor de vrouwen geldt:  $v_{ij} = k$  betekent dat vrouw  $i$  man  $j$  voorkeur  $k$  geeft; vrouw 1 heeft dus als voorkeursvolgorde: man 5, man 1, man 4, man 2 en als laatste man 3.

De betekenissen zijn dus niet symmetrisch!

Schematisch gaat het algoritme als volgt (noem de mannen A t/m E en de vrouwen 1 t/m 5); het getal boven de lijnen is het nummer van de toewijzing.

$$\begin{array}{l} \text{A} \text{ --- } \overset{1}{\text{---}} \text{---} 2 \text{ --- } \overset{7}{\text{---}} \text{---} 5 \text{ --- } \overset{9}{\text{---}} \text{---} 1 \\ \text{B} \text{ --- } \overset{2}{\text{---}} \text{---} 1 \text{ --- } \overset{6}{\text{---}} \text{---} 2 \text{ --- } \overset{14}{\text{---}} \text{---} 3 \text{ --- } \overset{15}{\text{---}} \text{---} 4 \\ \text{C} \text{ --- } \overset{3}{\text{---}} \text{---} 2 \text{ --- } \overset{4}{\text{---}} \text{---} 3 \text{ --- } \overset{11}{\text{---}} \text{---} 5 \\ \text{D} \text{ --- } \overset{5}{\text{---}} \text{---} 1 \text{ --- } \overset{10}{\text{---}} \text{---} 3 \\ \text{E} \text{ --- } \overset{8}{\text{---}} \text{---} 5 \text{ --- } \overset{12}{\text{---}} \text{---} 3 \text{ --- } \overset{13}{\text{---}} \text{---} 2 \end{array}$$

De Gale-Shapley koppeling is: man A met vrouw 1; man B met vrouw 4; man C met vrouw 5; man D met vrouw 3 en man E met vrouw 2.

**Stelling 6.13** *Algoritme 6.3 is correct en heeft complexiteit  $\mathcal{O}(n^2)$ .*

### Bewijs

De mannen krijgen de vrouwen in afnemende voorkeur en de vrouwen krijgen de mannen in toenemende voorkeur toegewezen, dus een verbroken koppeling wordt nooit meer hersteld: het algoritme is dus eindig.

Het algoritme eindigt op het moment dat de laatste vrouw een partner krijgt toegewezen (dan hebben ook alle mannen een partner). Dit is op het laatst het geval als een man gekoppeld wordt aan de vrouw van zijn laagste voorkeur en alle andere mannen gekoppeld zijn aan hun op één na laagste voorkeur: het algoritme heeft dus hoogstens  $n \times (n - 1) + 1 = n^2 - n + 1$  stappen.

De verkregen koppeling is stabiel, immers:

Stel man  $i$  en vrouw  $j$  zijn niet aan elkaar gekoppeld, terwijl ze liever elkaar hebben dan hun huidige partners. Man  $i$  was op zeker moment aan vrouw  $j$  gekoppeld, wat in strijd is met de eigenschap dat de vrouwen er "steeds beter op worden".

Een vrouw heet een *mogelijke kandidaat* voor een man als er een stabiele koppeling is waarin ze gekoppeld zijn. Merk op dat het voldoende is om te bewijzen dat een mogelijke kandidaat nooit meer wordt verlaten, want dan krijgt iedere man de "best mogelijke kandidaat".

Veronderstel dat tijdens het algoritme man  $i$  aan vrouw  $j$  is gekoppeld, waarbij vrouw  $j$  een mogelijke kandidaat is voor man  $i$ , maar dat dit koppel wordt verbroken en dat het voor de eerste keer is dat zo'n mogelijke combinatie wordt verbroken. We moeten nu laten zien dat dit tot een tegenspraak leidt.

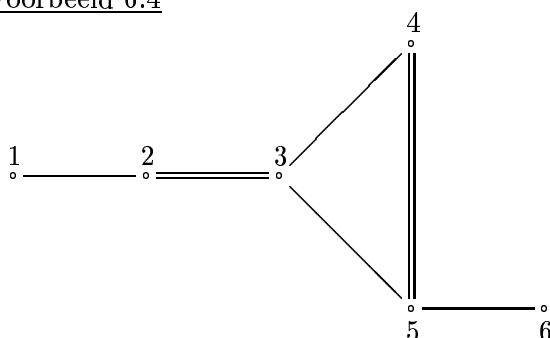
Stel dat het koppel  $(i, j)$  wordt verbroken omdat vrouw  $j$  een aanzoek krijgt van man  $k$ , die ze dus verkiest boven man  $i$ . Omdat vrouw  $j$  een mogelijke kandidaat is voor man  $i$ , is er een stabiele koppeling  $M$  waarin  $(i, j)$  een koppel is en stel dat man  $k$  daarin aan vrouw  $l$  is gekoppeld, d.w.z. vrouw  $l$  is een mogelijke kandidaat voor man  $k$ . Omdat tijdens het algoritme  $(i, j)$  het eerst mogelijke koppel is dat wordt verbroken, is op het moment dat man  $k$  een aanzoek doet aan vrouw  $j$  het koppel  $(k, l)$  nog niet gevormd: man  $k$  verkiest vrouw  $j$  boven vrouw  $l$ . We hebben reeds gezien dat vrouw  $j$  ook man  $k$  verkiest boven man  $i$ : koppeling  $M$  is dus niet stabiel, wat de gewenste tegenspraak oplevert.

Wat de complexiteit betreft geldt het volgende: het aantal iteraties is  $\mathcal{O}(n^2)$  en het werk per iteratie is  $\mathcal{O}(1)$  (het vergelijken van twee elementen). De totale complexiteit is dus  $\mathcal{O}(n^2)$ . ■

## 6.6 Koppeling met maximale cardinaliteit in een willekeurige graaf

In een willekeurige graaf  $G = (V, E)$  is het vinden van een groeiketen lastiger dan in een bipartiete graaf. In een bipartiete graaf komen we afwisselend in  $V_1$  en  $V_2$ , wat het zoeken naar een groeiketen eenvoudiger maakt. In een willekeurige graaf zijn het met name de *kringen van oneven lengte* die voor complicaties zorgen.

### Voorbeeld 6.4



Beschouw de koppeling  $M = \{(2, 3), (4, 5)\}$ . Als we een wisselketen vanuit knooppunt 1 gaan opbouwen, dan krijgen we bij de knooppunten 4 en 5 het probleem dat beide een uiteinde zijn van dezelfde tak van  $M$ . De groeiketen  $C = [1, 2, 3, 4, 5, 6]$  moet dus met een aangepaste methode worden bepaald.

Een wisselkring  $B$  bestaande uit een oneven aantal takken heet een *bloem*. Een bloem bevat precies één vrij knooppunt m.b.t.  $M \cap B$  en dit knooppunt heet de *basis* van de bloem. In Voorbeeld 6.4 is  $B = [3, 4, 5]$  een bloem en deze heeft knooppunt 3 als basis.

In vergelijking met bipartiete grafen moeten er twee aanpassingen worden gedaan:

- a. bloemen moeten worden opgespoord;
- b. bloemen moeten na opsporing worden *verschrompeld* (zie verder).

Voor de opsporing van bloemen en het zoeken naar groeiketens gebruiken we een *wisselbos*; dit is een bos bestaande uit wisselbomen.

Een *wisselboom*  $T$  m.b.t. een koppeling  $M$  is een boom waarvoor geldt:

- (1) één knooppunt is vrij, zeg  $v_0$ , en heet de *wortel* van de boom;
- (2) alle ketens van  $T$  die in de wortel  $v_0$  beginnen zijn wisselketens.

Een boom is bipartiet, d.w.z. de knooppunten zijn te verdelen in *buitenpunten*  $V_1$  en *binnenpunten*  $V_2$ ; het vrije knooppunt  $v_0$  behoort tot de klasse van buitenpunten  $V_1$ ; omdat de ketens wisselpunten zijn, hebben alle knooppunten graad 2.

Bij de constructie van het wisselbos krijgen de knooppunten die bezocht worden *labels*:  $l_1[i] = j$  betekent dat knooppunt  $i$  in  $V_1$  zit en dat  $i$  vanuit een wortel via een wisselketen bereikbaar is met als laatste tak  $(j, i) \in M$ ;  $l_2[i] = j$  betekent dat knooppunt  $i$  in  $V_2$  zit en dat  $i$  vanuit een wortel via een wisselketen bereikbaar is met als laatste tak  $(j, i) \in E \setminus M$ .

Als we een bloem hebben, dan zijn de knooppunten van deze bloem, behalve de basis, zowel via een tak van  $M$  als via een tak van  $E \setminus M$  (door vanuit de basis de andere kant op te lopen) bereikbaar. Daarom krijgen deze knooppunten van de bloem zowel een  $l_1$ - als een  $l_2$ -label.

De constructie van het wisselbos gaat als volgt:

1. Start met ieder vrij knooppunt  $k$  als wortel van een aparte wisselboom en geef deze het label  $l_1[k] = 0$ .
2. De gelabelde knooppunten worden in de volgorde waarin ze gelabeld zijn verder onderzocht. Het onderzoek van een knooppunt  $i$  houdt het volgende in (hierbij wordt een knooppunt dat tot een bloem behoort beschouwd als een knooppunt met een  $l_1$ -label).

- a. Als  $i$  een  $l_1$ -label heeft:

Voor iedere  $(i, j) \in E \setminus M$  en  $i$  en  $j$  niet in dezelfde bloem:

als  $j$  geen label heeft:  $l_2[j] = i$ ;

als  $j$  een  $l_1$ -label heeft:

als  $i$  en  $j$  in dezelfde wisselboom zitten: er is een bloem gevonden;

anders: er is een groeiketen gevonden tussen beide bomen.

(als  $j$  een  $l_2$ -label heeft, dan doen we niets)

- b. Als  $i$  een  $l_2$ -label heeft:

Beschouw het knooppunt  $j$  zdd.  $(i, j) \in M$  en  $i$  en  $j$  niet in dezelfde bloem

(als  $i$  en  $j$  wel in dezelfde bloem, dan doen we niets)

als  $j$  geen label heeft:  $l_1[j] = i$ ;

als  $j$  een  $l_2$ -label heeft:

als  $i$  en  $j$  in dezelfde wisselboom zitten: er is een bloem gevonden;

anders: er is een groeiketen gevonden tussen beide bomen.

(als  $j$  een  $l_1$ -label heeft, dan doen we niets).



Als een groeiketen wordt gevonden, dan kan de koppeling worden uitgebreid en beginnen we daarna aan een volgende iteratie. Knooppunten van een bloem  $B$  worden opgenomen in een groeiketen als deze groeiketen de bloem bij de basis binnenkomt. Vervolgens kunnen we de knooppunten van de bloem op twee manieren bezoeken, want we kunnen de bloem op twee manieren rondlopen.

Als een bloem  $B$  gevonden is, dan laten we deze *verschrompelen*, d.w.z. de knooppunten van  $B$  worden vervangen door een nieuw pseudoknooppunt  $v_B$  en de takken  $(i, j)$  met  $i \in B$  en  $j \notin B$  worden vervangen door  $(v_B, j)$ . Merk op dat deze takken geen van alle tot de koppeling behoren, behalve de tak via welke we de bloem bij de basis binnenkomen. Laat  $G_B$  de graaf zijn die uit  $G$  ontstaat door  $B$  te verschrompelen en  $M_B$  de koppeling in  $G_B$  die aldus uit  $M$  ontstaat. De volgende stelling laat zien dat het verschrompelen geen invloed heeft op het bestaan van groeiketen.

**Stelling 6.14**  $G$  heeft een groeiketen m.b.t.  $M$  d.e.s.d. als  $G_B$  een groeiketen heeft m.b.t.  $M_B$ .

**Bewijs**

$\Rightarrow$  Laat  $w$  de basis zijn van bloem  $B$  en laat  $C$  een groeiketen in  $G$  zijn m.b.t. koppeling  $M$ . Neem aan dat  $C$  knooppunten van  $B$  bevat (anders is  $C$  ook een groeiketen in  $G_B$  m.b.t.  $M_B$ ). Laat  $C = [v_0, \dots, x, w, \dots, y, z, \dots, v_1]$  met  $y$  het laatste knooppunt van  $B$ . Dan is  $C_B = [v_0, \dots, x, v_B, z, \dots, v_1]$  een groeiketen in  $G_B$  m.b.t.  $M_B$ .

$\Leftarrow$  Laat  $C_B = [v_0, \dots, x, v_B, z, \dots, v_1]$  een groeiketen in  $G_B$  m.b.t.  $M_B$  zijn (als  $C_B$  pseudoknooppunt  $v_B$  niet bevat, dan is  $C_B$  ook een groeiketen m.b.t.  $M$ ). Stel  $(x, v_B) \in M_B$  (anders is  $(z, v_B) \in M_B$  en draai de keten dan om). In  $G$  geldt dat  $(x, w) \in M$  met  $w$  de basis van  $B$  en  $(y, z) \notin M$  voor zekere  $y \in B$ . Dit geeft de volgende groeiketen in  $G$ :  $C = [v_0, \dots, x, w, C_1, y, z, \dots, v_1]$  met  $[w, C_1, y]$  de wisselketen in  $B$  die in  $y$  eindigt met een tak van  $M$  (deze bestaat door de bloem in één van de twee richtingen te doorlopen). ■

In de implementatie van het algoritme zullen we het verschrompelen van de bloem niet expliciet uitvoeren; we doen alsof de bloem wordt verschrompeld. We houden voor ieder knooppunt bij tot welke "buitenste" bloem het behoort en deze bloem wordt geïdentificeerd door hun basis. Tevens worden aan de knooppunten van een bloem de ontbrekende labels gegeven.

We doen daartoe het volgende:

- (1) Initialiseer ieder knooppunt als "zijn eigen" bloem:  $B[i] = i, i \in V$ .
- (2) Als een bloem wordt gevonden:
  - a. bepaal de basis van de bloem, zeg knooppunt  $k$ ;
  - b. als  $i$  tot deze bloem behoort:  $B[i] = B[k]$ ;
  - c. geef de knooppunten van deze bloem de ontbrekende labels.

Ga hierna verder met het knooppunt waar we waren toen de bloem werd ontdekt.

### Opmerkingen:

1. De knooppunten  $i$  en  $j$  zitten in dezelfde bloem als  $B[i] = B[j]$ .
2. Het nagaan of de knooppunten  $i$  en  $j$  in dezelfde wisselboom zitten kan worden gedaan door vanuit  $i$  resp.  $j$  terug te lopen in de wisselboom (via alternerend de  $l_1$ - en  $l_2$ -labels) tot de wortels, zeg  $r[i]$  resp.  $r[j]$ . De knooppunten  $i$  en  $j$  zitten in dezelfde wisselboom als  $r[i] = r[j]$ .
3. Als een bloem wordt gevonden bij het onderzoek van de tak tussen  $i$  en  $j$ , dan is de basis te bepalen door vanuit  $i$  en  $j$  terug te lopen totdat een gemeenschappelijk knooppunt wordt gevonden. Op dezelfde manier (door terug te lopen) kan aan de knooppunten die worden tegengekomen het andere label worden gegeven.
4. Als er een groeiketen wordt gevonden, dan kan de koppeling worden uitgebreid en beginnen we met de volgende iteratie.
5. Als er geen groeiketen wordt gevonden, dan geldt nadat alle eventuele bloemen zijn bepaald en verschrompeld:
  - geen enkele  $i$  en  $j$  met beide een  $l_1$ -label grenzen aan elkaar, tenzij  $i$  en  $j$  tot dezelfde bloem behoren;
  - geen enkele  $i$  met een  $l_1$ -label grenst aan een  $j$  zonder label.

In een dergelijke situatie spreken we van een *Hongaars bos*. In een Hongaars bos heeft de huidige koppeling maximale cardinaliteit omdat er geen groeiketen bestaat (bewijs volgt later).

### Voorbeeld 6.5

In nevenstaande graaf zijn de takken van koppeling  $M$

met dubbele lijnen aangegeven:

$$M = \{(1, 2), (3, 6), (5, 9), (7, 10)\}.$$

$$B[i] = i, \quad i = 1, 2, \dots, 10.$$

De vrije knooppunten zijn 4 en 8, dus  $l_1[4] = l_1[8] = 0$ .

Onderzoek knooppunt 4:  $l_2[3] = l_2[7] = 4$ .

Onderzoek knooppunt 8:  $l_2[1] = l_2[5] = l_2[9] = 8$ .

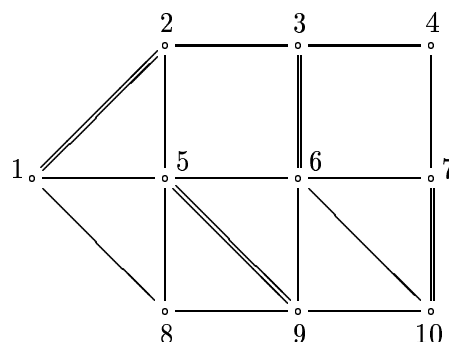
Onderzoek knooppunt 3:  $l_1[6] = 3$ .

Onderzoek knooppunt 7:  $l_1[10] = 7$ . Onderzoek knooppunt 1:  $l_1[2] = 1$ .

Onderzoek knooppunt 5:  $(5, 9) \in M$  en knooppunt 9 heeft een  $l_2$ -label en zit in dezelfde wisselboom: bloem  $[8, 5, 9]$  met basis 8;  $B[5] = B[9] = 8$ ;  $l_1[5] = 9$ ;  $l_1[9] = 5$ .

Onderzoek knooppunt 9: knooppunt 6 heeft ook een  $l_1$ -label en zit in een andere wisselboom: groeiketen  $C = [8, 5, 9, 6, 3, 4]$ .

De koppeling wordt nu uitgebreid tot de volmaakte koppeling  $M = \{(1, 2), (3, 4), (5, 8), (6, 9), (7, 10)\}$ .



*Algoritme 6.4: Algoritme voor een koppeling met maximale cardinaliteit in een willekeurige graaf*

1. a. Start met een willekeurige koppeling  $M$ .  
b.  $B[i] = i, i \in V; L = \emptyset$ .
2. Geef ieder vrij knooppunt  $i$  label  $l_1[i] = 0$  en plaats het op de lijst  $L$  van nog te onderzoeken knooppunten.

3. Als  $L = \emptyset$ : ga naar stap 8.  
 Anders: Kies het eerste knooppunt van  $L$ , zeg knooppunt  $i$ , en verwijder dit uit  $L$ .  
 Als  $i$  een  $l_1$ -label heeft: ga naar stap 4.  
 Als  $i$  een  $l_2$ -label heeft: ga naar stap 5.
4. Doe voor iedere  $(i, j) \in E \setminus M$  met  $B[i] \neq B[j]$  het volgende:  
 Als  $j$  geen label heeft:  $l_2[j] = i$  en plaats  $j$  achteraan op  $L$ .  
 Als  $j$  een  $l_1$ -label heeft:
  - a. Bepaal de wortels  $r[i]$  en  $r[j]$  van de wisselketens naar  $i$  resp.  $j$ .
  - b. Als  $r[i] \neq r[j]$ : ga naar stap 6.
  - c. Als  $r[i] = r[j]$ : ga naar stap 7.
 Ga naar stap 3.
5. a. Bepaal het unieke knooppunt  $j$  met  $(i, j) \in M$ .  
 b. Als  $B[i] \neq B[j]$ :  
 Als  $j$  geen label heeft:  $l_1[j] = i$  en plaats  $j$  achteraan op  $L$ .  
 Als  $j$  een  $l_2$ -label heeft:
  - (i) Bepaal de wortels  $r[i]$  en  $r[j]$  van de wisselketens naar  $i$  resp.  $j$ .
  - (ii) Als  $r[i] \neq r[j]$ : ga naar stap 6.
  - (iii) Als  $r[i] = r[j]$ : ga naar stap 7.
 c. Ga naar stap 3.
6. a. Bepaal de groeiketen  $C$ .  
 b.  $M := M \oplus C$ .  
 c. Verwijder alle labels en ga naar stap 1b.
7. a. Bepaal de (nieuwe) knooppunten  $B$  die tot de bloem behoren en de basis  $k$ .  
 b.  $B[i] = B[k]$ ,  $i \in B$ .  
 c. Bepaal voor alle  $i \in B \setminus \{k\}$  het nog ontbrekende label; indien dit een  $l_1$ -label is en  $i$  niet tot  $L$  behoort, dan plaatsen we  $i$  weer op  $L$ .  
 d. Ga verder in stap 4 of stap 5, afhankelijk waarvandaan we naar stap 7 zijn gesprongen.
8.  $M$  is een koppeling met maximale cardinaliteit en stop.

**Stelling 6.15** *Algoritme 6.4 is correct en heeft complexiteit  $\mathcal{O}(n^3)$ .*

### Bewijs

We eindigen in stap 8 met een Hongaars bos. Stel er is een groeiketen. Dan vinden we een  $i$  met een  $l_1$ -label dat grenst aan een  $j$  met eveneens een  $l_1$ -label, maar afkomstig uit een andere wisselboom. Dit komt echter niet voor in een Hongaars bos. Het algoritme is dus correct.

Omdat per iteratie de koppeling wordt uitgebreid, is het aantal iteraties  $\mathcal{O}(n)$ .

Beschouw een iteratie en splits het werk in het werk aan de bloemen en de rest.

Voor het werk aan de bloemen geldt:

- het aantal bloemen is  $\mathcal{O}(n)$ ;
- het bepalen van de basis: deze wordt bepaald door terug te lopen in twee ketens;

- de complexiteit hiervan is  $\mathcal{O}(n)$ ;
- het bepalen van de andere labels: deze worden bepaald door terug te lopen en de complexiteit hiervan is ook  $\mathcal{O}(n)$ ;
- het toekennen van de  $B$ -waarden: complexiteit  $\mathcal{O}(n)$ .

Het werk aan de bloemen is per iteratie dus  $\mathcal{O}(n^2)$ .

De overige werkzaamheden hebben dezelfde complexiteit als het zijwaarts zoeken, dus  $\mathcal{O}(m) = \mathcal{O}(n^2)$ , met daarnaast nog de volgende extra's:

- toekennen van labels aan de knooppunten: complexiteit  $\mathcal{O}(n)$ ;
- nagaan of twee knooppunten in dezelfde wisselboom zitten, d.w.z. de bepaling van de wortel van een wisselboom en dat wordt ook gedaan door terug te lopen; het werk hiervan heeft complexiteit  $\mathcal{O}(n)$  en dit gebeurt per iteratie  $\mathcal{O}(n)$  keer: per iteratie is het werk  $\mathcal{O}(n^2)$ ;
- nagaan of twee knooppunten in dezelfde bloem zitten: per knooppunt  $\mathcal{O}(1)$ , dus per iteratie  $\mathcal{O}(n)$ ;
- het bepalen van de groeiketen, zodra bekend is dat deze bestaat: complexiteit  $\mathcal{O}(n)$ .

Het overige werk per iteratie is  $\mathcal{O}(n^2)$ , zodat de complexiteit van het algoritme  $\mathcal{O}(n^3)$  is. ■

We zullen nu een dualiteitsresultaat afleiden. Dit is een generalisatie, van bipartiete naar algemene grafen, van Stelling 6.8.

Beschouw een collectie deelverz.  $W_i$ , met  $W_i \subseteq V$  en  $\#W_i$  oneven,  $1 \leq i \leq p$ . Als  $\#W_i = 1$ , zeg  $W_i = \{v\}$ , dan zeggen we dat  $W_i$  alle takken bedekt die incident zijn met  $v$  en dat  $W_i$  capaciteit 1 heeft; als  $\#W_i = 2k + 1$  ( $k \geq 1$ ), dan zeggen we dat  $W_i$  alle takken bedekt die beide eindpunten in  $W_i$  hebben en dat  $W_i$  capaciteit  $k$  heeft. Een dergelijke collectie heet een *oneven knooppuntenbedekking* als iedere tak van de graaf door de collectie wordt bedekt. De *capaciteit* van de collectie is de som van de capaciteiten van de deelverz. Gevraagd wordt een oneven knooppuntenbedekking te vinden met een minimale capaciteit.

Algoritme 6.4 lost dit probleem op. Beschouw het Hongaarse bos dat in de laatste iteratie is gevonden. Dan geldt voor iedere tak van de graaf precies één van de volgende situaties (hierin wordt een knooppunt dat tot een bloem behoort beschouwd als een knooppunt met een  $l_1$ -label):

- a. de beide uiteinden van de tak behoren tot een bloem;
- b. precies één uiteinde van de tak heeft een  $l_2$ -label;
- c. beide uiteinden hebben geen label.

Alle niet-gelabelde knooppunten zijn gebonden en voor iedere tak van de koppeling geldt dat de eindpunten ofwel beide gelabeld ofwel beide ongelabeld zijn. Het aantal niet gelabelde knooppunten is dus even, zeg  $2q$ .

We construeren als volgt een oneven knooppuntenbedekking:

- (1) Neem voor iedere buitenste bloem een aparte deelverz.
- (2) Neem voor ieder knooppunt met een  $l_2$ -label en niet behorend tot een bloem een aparte deelverz.

- (3) Voor de  $2q$  knooppunten zonder labels wordt de indeling als volgt gedaan:
- als  $q = 0$ : doe niets;
  - als  $q = 1$ : kies één van de twee ongelabelde knooppunten als een aparte deelverz.;
  - als  $q \geq 2$ : kies één van de  $2q$  ongelabelde knooppunten als een aparte deelverz. en de overige  $2q - 1$  knooppunten vormen tezamen ook een aparte verz.

**Voorbeeld 6.6**

In nevenstaande graaf zijn de takken van koppeling  $M$  met dubbele lijnen aangegeven:

$$M = \{(1, 9), (2, 3), (4, 7), (5, 6), (10, 11)\}.$$

Als we de laatste iteratie van het algoritme uitvoeren, krijgen we:

$$B[i] = i, \quad i = 1, 2, \dots, 11.$$

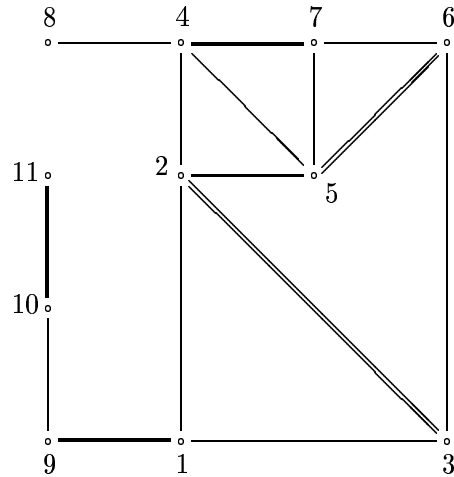
Het enige vrije knooppunt is 8, dus  $l_1[8] = 0$ .

Onderzoek knooppunt 8:  $l_2[4] = 8$ .

Onderzoek knooppunt 4:  $l_1[7] = 4$ .

Onderzoek knooppunt 7:  $l_2[5] = l_2[6] = 7$ .

Onderzoek knooppunt 5: knooppunt 6 heeft ook een



$l_2$ -label en zit in dezelfde wisselboom:  $B = [7, 5, 6]$  met knooppunt 7 als basis.

$$B[5] = B[6] = 7; \quad l_1[5] = 6, l_1[6] = 5. \quad l_2[2] = 5.$$

Onderzoek knooppunt 6:  $l_2[3] = 6$ .

Onderzoek knooppunt 2: knooppunt 3 heeft ook een  $l_2$ -label en zit in dezelfde wisselboom:  $B = [7, 5, 6, 2, 3]$  met knooppunt 7 als basis.

$$B[2] = B[3] = 7; \quad l_1[2] = 3, l_1[3] = 2. \quad l_2[1] = 2.$$

Onderzoek knooppunt 3: dit onderzoek levert geen toekenningen op.

Onderzoek knooppunt 1:  $l_1[9] = 1$ .

Onderzoek knooppunt 9:  $l_2[10] = 9$ .

Onderzoek knooppunt 10:  $l_1[11] = 10$ .

Onderzoek knooppunt 11: dit onderzoek levert geen toekenningen op.

De lijst  $L = \emptyset$  en de koppeling  $M$  heeft maximale cardinaliteit.

De constructie van een oneven knooppuntenbedekking met minimale capaciteit gaat als volgt:

$$W_1 = \{7, 5, 6, 2, 3\}; \quad W_2 = \{4\}; \quad W_3 = \{1\} \quad \text{en} \quad W_4 = \{10\}. \quad \text{Deze collectie heeft capaciteit } 2 + 1 + 1 + 1 = 5 \text{ en } \#M \text{ is ook } 5.$$

**Stelling 6.16** *Bovenstaande constructie van de collectie  $W_i$ 's geeft een oneven knooppuntenbedekking met minimale capaciteit.*

**Bewijs**

We merken allereerst op dat alle  $W_i$ 's een oneven aantal knooppunten hebben. Iedere tak heeft een eindpunt in de collectie, immers er geldt:

- òfwel de tak heeft minstens één eindpunt in een bloem;
- òfwel precies één eindpunt heeft een  $l_2$ -label;
- òfwel beide eindpunten hebben geen label en ook in dat geval geeft de constructie van de collectie een bedekking van deze tak.

Vervolgens merken we op dat voor iedere koppeling en iedere oneven knooppuntenbedekking geldt dat de capaciteit van de bedekking minstens het aantal elementen van de koppeling is, immers:

- iedere deelverz. die uit één element bestaat bedekt hoogstens één element van de koppeling;
- iedere deelverz. van  $2k + 1$  (met  $k \geq 1$ ) knooppunten bedekt hoogstens  $k$  elementen van de koppeling.

Voor de geconstrueerde bedekking geldt:

- een bloem met  $2k + 1$  knooppunten heeft capaciteit  $k$  en bevat  $k$  takken van de koppeling;
- iedere tak van de koppeling die ook voorkomt in de Hongaarse boom en niet geheel in een bloem zit wordt bedekt door een apart knooppunt;
- de  $q$  takken van de koppeling die niet in de Hongaarse boom zitten worden door één of twee deelverz. bedekt die tezamen capaciteit  $q$  hebben. ■

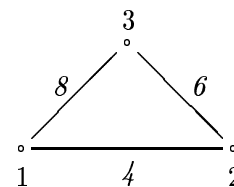
## 6.7 Koppeling met maximaal gewicht in een willekeurige graaf

In deze paragraaf is aan iedere tak  $(i, j)$  een gewicht  $w_{ij}$  toegekend. We zijn geïnteresseerd in een koppeling met maximaal gewicht. Zonder beperking der algemeenheid mogen we aannemen dat  $w_{ij} \geq 0$  voor alle  $(i, j) \in E$ .

Als we - naar analogie met het vinden van een koppeling met maximaal gewicht in een bipartiete graaf - het probleem willen oplossen met lineaire programmering, dan is de bijbehorende LP-formulering:

$$\max \left\{ \sum_{(i,j) \in E} w_{ij} x_{ij} \mid \begin{array}{l} \sum_j x_{ij} \leq 1, \quad i \in V \\ x_{ij} \geq 0, \quad (i, j) \in E \end{array} \right\} \quad (6.8)$$

Deze formulering is echter niet geschikt, omdat in een willekeurige graaf de eigenschap van totale unimodulariteit niet geldt en daarom de optimale oplossing niet geheeltallig hoeft te zijn, en daarom dus niet hoeft te corresponderen met een koppeling. Zo heeft nevenstaande graaf (naast de takken staan de gewichten) als optimale LP-oplossing:  $x_{12} = x_{13} = x_{23} = \frac{1}{2}$  met waarde 9. De koppeling met maximaal gewicht is echter:  $M = \{(1, 3)\}$  met gewicht 8.



Ook hier zorgen de oneven kringen weer voor het probleem. We kunnen proberen dit op te lossen door voor iedere kring met een oneven aantal, zeg  $2k + 1$ , takken te eisen dat de som van de desbetreffende  $x$ -variabelen hoogstens  $k$  is. Dit geeft de nieuwe LP-formulering:

$$\max \left\{ \sum_{(i,j) \in E} w_{ij} x_{ij} \mid \begin{array}{l} \sum_j x_{ij} \leq 1, \quad i \in V \\ \sum_{(i,j) \in C_k} x_{ij} \leq r_k \quad \text{voor alle oneven kringen } C_k \\ x_{ij} \geq 0, \quad (i,j) \in E \end{array} \right\}, \quad (6.9)$$

waarbij  $r_k = \frac{1}{2}[\#C_k - 1]$ .

We willen het probleem oplossen met een polynomiale methode. Het aantal beperkingen in formulering (6.9) kan echter exponentieel zijn, zodat dit ons niet helpt. We zullen langs een andere weg een polynomiale procedure ontwikkelen die een geheel-tallige oplossing van (6.9) oplevert. In essentie kan dit worden gedaan door niet vanaf het begin voor alle oneven kringen  $C_k$  te eisen dat  $\sum_{(i,j) \in C_k} x_{ij} \leq \frac{1}{2}[\#C_k - 1]$ , maar alleen voor die we tegenkomen en moeilijkheden blijken te veroorzaken. Dit kan worden gedaan via het duale lineaire programmeringsprobleem van (6.9), zoals we ook bij het vinden van een koppeling met maximaal gewicht in een bipartiete graaf gebruik maakten van het duale probleem. Dit duale probleem van (6.9) luidt als volgt:

$$\min \left\{ \sum_i u_i + \sum_k r_k y_k \mid \begin{array}{l} u_i + u_j + \sum_k \text{ met } (i,j) \in C_k y_k \geq w_{ij}, (i,j) \in E \\ u_i \geq 0, i \in V; y_k \geq 0 \text{ voor alle oneven kringen } C_k \end{array} \right\}. \quad (6.10)$$

Bij de methode die we bespreken is het niet nodig om alle oneven kringen  $C_k$  expliciet aan te geven. In iedere iteratie hebben we een  $x$  en een  $(u, y)$  zdd.

$$x \text{ een } (0,1)\text{-vector is die correspondeert met een koppeling: voldoet aan (6.9);} \quad (6.11)$$

$$(u, y) \text{ toelaatbaar is voor (6.10);} \quad (6.12)$$

$$x_{ij} \cdot [u_i + u_j + \sum_k \text{ met } (i,j) \in C_k y_k - w_{ij}] = 0, (i,j) \in E; \quad (6.13)$$

$$y_k \cdot [r_k - \sum_{(i,j) \in C_k} x_{ij}] = 0 \text{ voor alle oneven kringen } C_k; \quad (6.14)$$

Uit de theorie van de lineaire programmering weten we dat de oplossingen  $x$  en  $(u, y)$  optimaal zijn voor (6.9) resp. (6.10) als ook geldt:

$$u_i \cdot [1 - \sum_j x_{ij}] = 0, \quad i \in V. \quad (6.15)$$

We kunnen de methode starten met  $x = 0$ ,  $y = 0$  en  $u_i = \frac{1}{2} \cdot \max_{k,l} w_{kl}$ .

Een *iteratie* verloopt nu als volgt:

1. Beschouw de graaf  $G'$  met alle knooppunten (bloemen doen als verschrompelde knooppunten mee) en met de takken  $(i, j)$  waarvoor geldt dat  $u_i + u_j + \sum_k \text{ met } (i,j) \in C_k y_k = w_{ij}$ , waarbij de kringen  $C_k$  de oneven kringen zijn die als zodanig ontdekt zijn.
2. Gebruik het algoritme uit de vorige paragraaf om een koppeling met maximale cardinaliteit te bepalen in  $G'$ , totdat we in één van de volgende drie gevallen terecht komen:
  - a. Er wordt een groeiketen gevonden: breid dan de koppeling uit; merk op dat (6.11) t/m

(6.14) blijven gelden en dat voor de uiteinden van de groeiketen nu ook (6.15) geldt.

- b. Er wordt een bloem ontdekt: verschrompel de bloem en zet het zoeken naar een groeiketen voort.
- c. Er bestaat geen groeiketen, d.w.z. een Hongaarse boom wordt gevonden; de variabelen  $(u, y)$  van het duale probleem worden aangepast (zie verder) zdd. (6.11) t/m (6.14) blijven gelden en  $G'$  meer takken krijgt, waarna het zoeken naar een groeiketen wordt voortgezet.

De *aanpassing van de duale variabelen* geschiedt als volgt (we maken onderscheid tussen gewone knooppunten  $v_i$  en pseudoknooppunten  $B_k$ , ontstaan door het verschrompelen van een bloem; de knooppunten in een bloem worden geacht hetzelfde label te hebben als het pseudoknooppunt).

$$u_i = \begin{cases} u_i - \delta & \text{als } v_i \text{ een } l_1 - \text{label heeft} \\ u_i + \delta & \text{als } v_i \text{ een } l_2 - \text{label heeft} \end{cases}$$

$$y_k = \begin{cases} y_k + 2\delta & \text{als } B_k \text{ een } l_1 - \text{label heeft} \\ y_k - 2\delta & \text{als } B_k \text{ een } l_2 - \text{label heeft} \end{cases}$$

Hierbij moet  $\delta$  zodanig worden bepaald dat (6.11) t/m (6.14) blijven gelden en  $G'$  meer takken krijgt. Uit de niet-negativiteit van de variabelen  $u$  en  $y$  volgt dat  $\delta \leq \min(\delta_1, \delta_2)$ , waarbij  $\delta_1 = \min\{u_i \mid v_i \text{ heeft een } l_1 - \text{label}\}$  en  $\delta_2 = \frac{1}{2} \cdot \min\{y_k \mid B_k \text{ heeft een } l_2 - \text{label}\}$ .

Omdat  $(u, y)$  toelaatbaar moet blijven, moeten we ook nog rekening houden met een aantal andere mogelijkheden. We onderscheiden daartoe de volgende gevallen:

- a.  $(i, j)$  zit in een bloem:  $u_i + u_j + \sum_k y_k$  met  $(i, j) \in C_k$   $y_k$  blijft onveranderd, immers:  $u_i$  en  $u_j$  veranderen beide met  $\delta$ , maar de desbetreffende  $y_k$  verandert tegengesteld met  $2\delta$ .
- b.  $(i, j)$  zit niet in een bloem, maar is wel een tak van  $G'$ :  $u_i + u_j + \sum_k y_k$  met  $(i, j) \in C_k$   $y_k$  blijft onveranderd, immers:  $u_i$  verandert met  $\delta$  en  $u_j$  verandert tegengesteld met  $\delta$ .
- c.  $(i, j)$  zit niet in een bloem en is ook geen tak van  $G'$ :  $u_i + u_j + \sum_k y_k$  met  $(i, j) \in C_k$   $y_k = u_i + u_j > w_{ij}$ . In de volgende gevallen wordt  $u_i + u_j$  niet kleiner: als  $i$  en  $j$  beide niet gelabeld zijn, of als  $i$  en/of  $j$  een  $l_2$ -label heeft.  
 Als  $i$  een  $l_1$ -label heeft en  $j$  geen label heeft, dan wordt  $u_i + u_j$   $\delta$  kleiner.  
 Als  $i$  en  $j$  beide een  $l_1$ -label hebben, dan wordt  $u_i + u_j$   $2\delta$  kleiner.

Hieruit volgt dat moet gelden dat  $\delta \leq \min(\delta_3, \delta_4)$ , waarbij:

$$\delta_3 = \min\{u_i + u_j - w_{ij} \mid v_i \text{ heeft een } l_1 - \text{label en } v_j \text{ heeft geen label}\};$$

$$\delta_4 = \frac{1}{2} \cdot \min\{u_i + u_j - w_{ij} \mid v_i \text{ en } v_j \text{ hebben } l_1 - \text{labels en behoren niet tot dezelfde bloem}\}.$$

Neem nu  $\delta = \min(\delta_1, \delta_2, \delta_3, \delta_4)$ . Ga zelf na dat  $\delta > 0$ .

Als  $\delta = \delta_1$ , dan krijgen alle vrije knooppunten  $u$ -waarde 0, zodat (6.15) geldt en dus een optimale oplossing is verkregen.

Als  $\delta = \delta_2$ , dan wordt  $y_k = 0$  voor zekere  $k$  bij oneven kring  $C_k$ . Vervang dit pseudoknooppunt door de oorspronkelijke oneven kring met bijbehorend deel van de koppeling en ga verder met



het zoeken naar een groeiketen.

Als  $\delta = \delta_3$ , dan wordt de bijbehorende tak  $(i, j)$  aan  $G'$  toegevoegd en wordt deze tak bij het verder zoeken naar een groeiketen opgenomen in een wisselboom.

Als  $\delta = \delta_4$ , dan wordt de bijbehorende tak  $(i, j)$  aan  $G'$  toegevoegd en ontstaat hierdoor bij het verder zoeken naar een groeiketen een bloem.

*Algoritme 6.5: Algoritme om een koppeling met maximaal gewicht te bepalen in een willekeurige graaf*

1. Start met  $M = \emptyset$ ,  $u_i = \frac{1}{2} \cdot \max_{k,l} w_{kl}, i \in V, k = 0$ .
2. a. Laat  $E' = \{(i, j) \mid u_i + u_j + \sum_k \text{met } (i,j) \in C_k y_k = w_{ij}\}$ .  
 b. Pas het algoritme om een maximale koppeling te vinden toe op de graaf  $G'$  met takken  $E'$  totdat:
  - fwel een groeiketen  $C$  wordt gevonden: ga naar stap 3;
  - fwel een bloem  $B$  wordt gevonden: ga naar stap 4;
  - fwel een Hongaars bos wordt gevonden: ga naar stap 5.
3. a.  $M := M \oplus C$  (in een bloem de goede route nemen).  
 b. Als er geen vrij knooppunt is: stop (de koppeling heeft maximaal gewicht).  
 Anders: ga naar stap 2.
4. a.  $k := k + 1$ ;  $y_k = 0$ ;  $B_k := B$ .  
 b. Verschrompel bloem  $B_k$  tot  $v_{n+k}$ , pas  $E'$  aan en ga in stap 2 verder.
5. a.  $\delta_1 = \min\{u_i \mid v_i \text{ heeft een } l_1 - \text{label}\}$ ;  $\delta_2 = \frac{1}{2} \cdot \min\{y_k \mid B_k \text{ heeft een } l_2 - \text{label}\}$ ;  
 $\delta_3 = \min\{u_i + u_j - w_{ij} \mid v_i \text{ heeft een } l_1 - \text{label en } v_j \text{ heeft geen label}\}$ ;  
 $\delta_4 = \frac{1}{2} \cdot \min\{u_i + u_j - w_{ij} \mid v_i \text{ en } v_j \text{ hebben } l_1 - \text{labels en behoren niet tot dezelfde bloem}\}$ ;  
 $\delta = \min(\delta_1, \delta_2, \delta_3, \delta_4)$ .  
 b.  $u_i = \begin{cases} u_i - \delta & \text{als } v_i \text{ een } l_1 - \text{label heeft} \\ u_i + \delta & \text{als } v_i \text{ een } l_2 - \text{label heeft} \end{cases}$ ;  $y_k = \begin{cases} y_k + 2\delta & \text{als } B_k \text{ een } l_1 - \text{label heeft} \\ y_k - 2\delta & \text{als } B_k \text{ een } l_2 - \text{label heeft} \end{cases}$   
 c. Als  $\delta = \delta_1$ :  $M$  is een koppeling met maximaal gewicht (stop).  
 Als  $\delta = \delta_2$ , dan wordt  $y_k = 0$  voor zekere  $k$  bij oneven kring  $C_k$  en vervang dit pseudo-knooppunt door de oorspronkelijke oneven kring met bijbehorend deel van de koppeling, en ga verder met het zoeken naar een groeiketen.  
 Als  $\delta = \delta_3$  of  $\delta_4$ , dan wordt de bijbehorende tak  $(i, j)$  aan  $G'$  toegevoegd en wordt verder gegaan in de wisselboom.

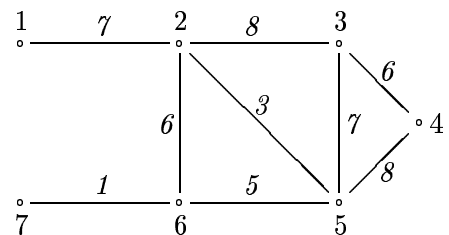
### Voorbeeld 6.7

*Iteratie 1:*

$M = \emptyset$ ;  $u_i = 4, 1 \leq i \leq 7$ ;  $k = 0$ .  $E' = \{(2, 3), (4, 5)\}$ .

Er wordt een groeiketen gevonden:

$C = [2, 3]$ ;  $M = \{(2, 3)\}$ .



*Iteratie 2:*

$E' = \{(2, 3), (4, 5)\}$ . Er wordt een groeiketen gevonden:  $C = [4, 5]$ ;  $M = \{(2, 3), (4, 5)\}$ .

*Iteratie 3:*

$E' = \{(2, 3), (4, 5)\}$ . Er wordt een Hongaarse boom gevonden, waarin alleen de knooppunten 1, 6 en 7 een label hebben, en wel een  $l_1$ -label.  $\delta_1 = 4$ ,  $\delta_2 = \infty$ ,  $\delta_3 = 1$ ,  $\delta_4 = \frac{7}{2}$ :  $\delta = 1$ .

$u_1 = u_6 = u_7 = 3$ .

$E' = \{(2, 3), (4, 5), (1, 2)\}$ . Er wordt een Hongaarse boom gevonden: de knooppunten 1, 3, 6 en 7 hebben een  $l_1$ -label en knooppunt 2 heeft een  $l_2$ -label.

$\delta_1 = 3$ ,  $\delta_2 = \infty$ ,  $\delta_3 = 1$ ,  $\delta_4 = \frac{5}{2}$ :  $\delta = 1$ .  $u_1 = u_6 = u_7 = 2$ ;  $u_3 = 3$  en  $u_2 = 5$ .

$E' = \{(2, 3), (4, 5), (1, 2), (3, 5)\}$ . Er is weer een Hongaarse boom, waarin de knooppunten 1, 3, 4, 6 en 7 een  $l_1$ -label hebben en knooppunt 2 een  $l_2$ -label heeft.

$\delta_1 = 2$ ,  $\delta_2 = \infty$ ,  $\delta_3 = \infty$ ,  $\delta_4 = \frac{1}{2}$ :  $\delta = \frac{1}{2}$ .  $u_1 = u_6 = u_7 = \frac{3}{2}$ ;  $u_3 = \frac{5}{2}$ ,  $u_4 = \frac{7}{2}$ ,  $u_2 = \frac{11}{2}$ ,  $u_5 = \frac{9}{2}$ .

$E' = \{(2, 3), (4, 5), (1, 2), (3, 5), (3, 4)\}$ . Nu wordt de bloem  $B_1 = [3, 4, 5]$  gevonden;

$k = 1$ ;  $y_1 = 0$ ;  $v_8 = \{3, 4, 5\}$ .

$E' = \{(1, 2), (2, 8)\}$ . Dit levert de Hongaarse boom met alleen de knooppunten 1, 2 en 8 gelabeld: 1 en 8 een  $l_1$ -label, 2 een  $l_2$ -label.  $\delta_1 = \frac{3}{2}$ ,  $\delta_2 = \infty$ ,  $\delta_3 = \infty$ ,  $\delta_4 = \frac{1}{2}$ :  $\delta = \frac{1}{2}$ .

$u_1 = u_6 = u_7 = 1$ ,  $u_3 = 2$ ,  $u_4 = 3$ ,  $u_2 = 6$ ,  $u_5 = 4$  en  $y_1 = 1$ .

$E' = \{(1, 2), (2, 8), (5, 6)\}$ . Nu wordt een groeiketen gevonden:  $C = [1, 2, 8, 6]$ , d.w.z. in de oorspronkelijke graaf  $[1, 2, 3, 4, 5, 6]$ , zodat  $M = \{(1, 2), (3, 4), (5, 6)\}$ .

*Iteratie 4:*

$E' = \{(1, 2), (2, 3), (3, 4), (4, 5), (3, 5), (5, 6)\}$ . Dit geeft een Hongaarse boom met alleen knooppunt 7 gelabeld, met een  $l_1$ -label.  $\delta_1 = 1$ ,  $\delta_2 = \infty$ ,  $\delta_3 = 1$ ,  $\delta_4 = \infty$ :  $\delta = 1$ .  $u_7 = 0$ .

Omdat  $\delta = \delta_1$ , is de koppeling  $M$  een koppeling met maximaal gewicht.

**Stelling 6.17** *Algoritme 6.5 is correct en heeft complexiteit  $\mathcal{O}(n^4)$ .*

### Bewijs

Als het algoritme stopt is aan (6.15) voldaan, immers: òfwel we stoppen in stap 3b en dan is  $\sum_j x_{ij} = 1$  voor alle  $i \in V$ , òfwel we stoppen in stap 5c en dan is  $u_i = 0$  voor alle vrije knooppunten.

Tijdens het algoritme wordt in iedere iteratie gewerkt aan de constructie van wisselbomen met als extra werk het verschrompelen van bloemen, de bepaling van  $\delta$  en de aanpassing van de duale variabelen  $u$  en  $y$ .

Als  $\delta = \delta_1$ , dan stopt het algoritme: dit kan maar één keer gebeuren.

Als  $\delta = \delta_2$ , dan wordt een pseudoknooppunt met een  $l_2$ -label vervangen door een oneven kring. Omdat er hoogstens  $n$  pseudoknooppunten kunnen zijn, kan dit per iteratie maar  $n$  keer gebeuren (omdat bij het ontstaan van bloemen het desbetreffende knooppunt een  $l_1$ -label heeft, is het niet mogelijk dat een bloem in één iteratie verschrompeld wordt en weer in een oneven kring wordt omgezet).

Als  $\delta = \delta_3$ , dan wordt bij het vervolg minstens één knooppunt meer gelabeld: dit kan dus per iteratie ook maar hoogstens  $n$  keer gebeuren.

Als  $\delta = \delta_4$ , dan wordt bij het vervolg een pseudoknooppunt gemaakt en dit kan per iteratie ook maar hoogstens  $n$  keer gebeuren.

Het algoritme heeft hoogstens  $\frac{1}{2}n$  iteraties (per iteratie wordt de koppeling groter) en iedere iteratie heeft hoogstens  $n$  deeliteraties. Per deeliteratie is de complexiteit gelijk aan de complexiteit van één iteratie van het algoritme om een koppeling met maximale cardinaliteit te vinden, wat volgens Stelling 6.15  $\mathcal{O}(n^2)$  is. De totale complexiteit van algoritme 6.5 is dus  $\mathcal{O}(n^4)$ . ■

## 6.8 Opgaven

### Opgave 6.1

Is de volgende uitspraak waar of niet waar?

”Voor iedere koppeling  $M_1$  is er een maximale koppeling  $M_2$  met  $M_1 \subseteq M_2$ ”.

### Opgave 6.2

- Geef een voorbeeld van een graaf  $G$  met  $\alpha(G) < \rho(G)$ .
- Geef een voorbeeld van een graaf  $G$  met  $\mu(G) < \tau(G)$ .

### Opgave 6.3

Zij  $G = (V_1 \cup V_2, E)$  een bipartiete graaf. Laat  $\sigma(G) = \max_{W_1 \subseteq V_1} [\#W_1 - \#\Gamma(W_1)]$ , met  $\Gamma(W_1) = \{j \in V_2 \mid (i, j) \in E \text{ voor een } i \in W_1\}$ . Toon aan dat  $\mu(G) = \#V_1 - \sigma(G)$ .

### Opgave 6.4

Zij  $G = (V_1 \cup V_2, E)$  een bipartiete graaf. Stel dat  $\min_{v \in V_1} \delta(v) \geq \max_{w \in V_2} \delta(w)$ . Toon aan dat  $\mu(G) = \#V_1$ .

### Opgave 6.5

Zij  $A$  een  $n \times n$  (0,1)-matrix met in iedere rij en iedere kolom precies  $k$  1'en. Toon aan dat  $A = P_1 + P_2 + \dots + P_k$  met  $P_i$  een permutatiematrix,  $i = 1, 2, \dots, k$ .

### Opgave 6.6

Zij  $M_1$  en  $M_2$  twee koppelingen in een bipartiete graaf  $G = (V_1 \cup V_2, E)$ . Toon aan: ”Er bestaat een koppeling  $M \subseteq M_1 \cup M_2$  zdd.  $M$  de knooppunten van  $V_1$  bindt die t.o.v.  $M_1$  gebonden zijn en de knooppunten van  $V_2$  die t.o.v.  $M_2$  gebonden zijn”.

### Opgave 6.7

Zij  $G = (V_1 \cup V_2, E)$  een bipartiete graaf en laat  $k = \max_{v \in V_1 \cup V_2} \delta(v)$ .

- Toon aan dat er een koppeling is die alle knooppunten met graad  $k$  bindt.
- Toon aan dat er disjuncte koppelingen  $M_1, M_2, \dots, M_k$  zijn zdd.  $E = M_1 \cup M_2 \cup \dots \cup M_k$ .

### Opgave 6.8

- Toon aan dat een boom hoogstens één volmaakte koppeling kan hebben.
- Toon aan, zonder gebruik te maken van de Stelling van Tutte, dat een boom een volmaakte koppeling heeft d.e.s.d. als  $o(W) = 1$  voor iedere  $W$  die uit  $V$  ontstaat door één knooppunt weg te laten.

### Opgave 6.9

Toon aan dat een 3-reguliere graaf zonder bruggen een volmaakte koppeling heeft.

### Opgave 6.10

Zij  $G = (V_1 \cup V_2, E)$  een bipartiete graaf. Bewijs of geef een tegenvoorbeeld van de bewering: "Er is een één-éénduidig verband tussen de knooppuntenbedekkingen van  $G$  en de hoekpunten van (6.2)".

### Opgave 6.11

- Bewijs dat iedere  $r$ -reguliere bipartiete graaf een volmaakte koppeling heeft.
- Bewijs dat iedere  $r$ -reguliere bipartiete graaf  $r$  disjuncte volmaakte koppelingen heeft.
- Geef voor iedere  $r \geq 2$  een  $r$ -reguliere graaf die geen volmaakte koppeling heeft.

### Opgave 6.12

Toon rechtstreeks aan dat resultaat C (van paragraaf 6.2) uit resultaat B volgt.

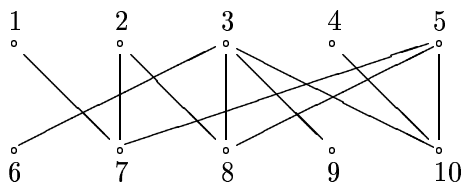
### Opgave 6.13\*

Bewijs resultaat D uit paragraaf 6.2 rechtstreeks.

Aanwijzing: Gebruik inductie naar het aantal elementen van  $X$ .

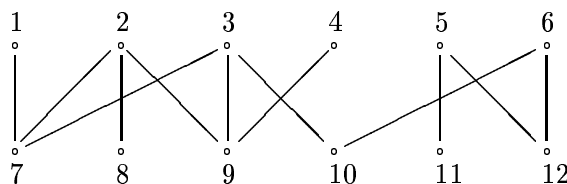
### Opgave 6.14

Vind een koppeling met maximale cardinaliteit en een knooppuntenbedekking met minimale cardinaliteit in nevenstaande bipartiete graaf.



### Opgave 6.15

Vind een koppeling met maximale cardinaliteit en een knooppuntenbedekking met minimale cardinaliteit in nevenstaande bipartiete graaf.



### Opgave 6.16

Toon aan dat een maximale gewogen koppeling geen koppeling met maximale cardinaliteit hoeft te zijn.

### Opgave 6.17

Beschouw het volgende probleem (het *toewijzingsprobleem*): in de volledige bipartiete graaf  $K_{n,n}$  is aan iedere tak  $(i, j)$  een getal  $c_{ij}$  toegekend, de *kosten*; gevraagd wordt een volmaakte koppeling te bepalen met minimale kosten.

Toon aan dat het toewijzingsprobleem equivalent is met het gewogen koppelingsprobleem.

### Opgave 6.18

Toon het volgende aan: Als tijdens het algoritme van de Hongaarse methode de koppeling  $M$  wordt vervangen door de koppeling  $M' = M \oplus C$ , waarbij  $C$  een groeiketen is in  $G'$ , dan geldt dat  $M'$  een groter gewicht heeft dan  $M$ .

### Opgave 6.19

Bepaal een koppeling met maximaal gewicht in de volledige bipartiete graaf  $K_{5,5}$ . De gewichten staan in nevenstaande  $5 \times 5$  matrix  $W$ .

$$\begin{pmatrix} 5 & 6 & 4 & 8 & 5 \\ 8 & 8 & 6 & 3 & 7 \\ 9 & 5 & 6 & 4 & 9 \\ 6 & 6 & 5 & 8 & 5 \\ 0 & 8 & 7 & 8 & 7 \end{pmatrix}$$

### Opgave 6.20

Beschouw het volgende algoritme voor het bepalen van een koppeling met maximaal gewicht in een bipartiete graaf.

1.  $M = \emptyset$ .

2. Bepaal een groeiketen  $C$  met minimale lengte t.a.v. de lengtefunctie  $l_{ij} = \begin{cases} w_{ij} & \text{als } (v_i, v_j) \in M \\ -w_{ij} & \text{anders} \end{cases}$   
(als er geen groeiketen is: stop).

3.  $M := M \oplus C$  en ga naar stap 2.

Voer de volgende opdrachten uit:

- Pas dit algoritme toe op de bipartiete gewogen graaf uit Opgave 19.
- Toon de correctheid aan door aan te tonen dat in iteratie  $k$  de koppeling  $M$  maximaal gewicht heeft t.o.v. alle koppelingen met cardinaliteit  $k$ .
- Bewijs dat er geen ronden van negatieve lengte kunnen zijn.
- Toon aan dat de complexiteit  $\mathcal{O}(n^4)$  is.

### Opgave 6.21

Een ski-leraar heeft  $n$  paar ski's voor  $n$  skiërs. Hoe kan de leraar het beste de ski's verdelen zdd. de som van de verschillen (in absolute waarde) tussen de lengtes van de ski's en de lengtes van de skiërs minimaal is?

**Opgave 6.22**

Bepaal een Gale-Shapley koppeling voor de volgende voorkeursmatrices ( $M$  voor de mannen en  $V$  voor de vrouwen):

$$M = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 4 & 3 & 2 \\ 2 & 1 & 3 & 4 \\ 4 & 2 & 3 & 1 \end{pmatrix} \quad \text{en} \quad V = \begin{pmatrix} 3 & 4 & 2 & 1 \\ 3 & 1 & 4 & 2 \\ 2 & 3 & 4 & 1 \\ 3 & 2 & 1 & 4 \end{pmatrix}.$$

**Opgave 6.23**

Veronderstel dat de voorkeursmatrix  $V$  van de vrouwen op dezelfde wijze is opgesteld als die van de mannen, d.w.z.  $v_{ij} = k$  d.e.s.d. als man  $k$  bij vrouw  $i$  op de  $j$ -de plaats komt.

Geef een procedure met complexiteit  $\mathcal{O}(n^2)$  om deze  $V$ -matrix om te zetten in de gewenste vorm, d.w.z. in een matrix  $V'$  met  $v'_{ij} = k$  d.e.s.d. als man  $j$  bij vrouw  $i$  op de  $k$ -de plaats komt.

**Opgave 6.24**

Het begrip Gale-Shapley koppeling is een optimaliteitscriterium vanuit het mannen standpunt: we zouden het *mannen-optimaal* kunnen noemen. Analoog heet een stabiele koppeling *vrouwen-optimaal* als iedere vrouw in en andere stabiele koppeling niet beter af is.

- Stel een algoritme op om een vrouwen-optimale koppeling te bepalen.
- Pas dit algoritme toe op de voorkeursmatrices uit Opgave 22.
- Toon aan dat als een koppeling zowel mannen- als vrouwen-optimaal is dit de enige stabiele koppeling is.

**Opgave 6.25**

Bestaat er voor  $n = 4$  een Gale-Shapley koppeling waarin alle mannen en alle vrouwen aan hun tweede keuze zijn gekoppeld? Verklaar uw antwoord.

**Opgave 6.26**

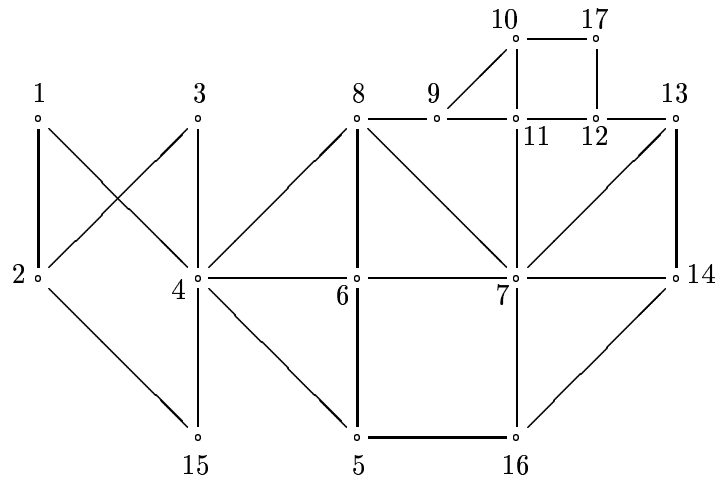
Vier mannen willen een herendubbel (tennis) gaan spelen, d.w.z. dat er twee koppels van elk twee mannen worden gevormd. Iedere man geeft met de cijfers 1, 2 en 3 zijn voorkeur voor de drie mogelijke partners aan. Een indeling heet stabiel als geen twee mannen liever met elkaar dubbelen dan met hun huidige partner. Bestaat er altijd een stabiele indeling? Verklaar uw antwoord.

**Opgave 6.27**

Geef  $n \times n$  voorkeursmatrices  $M$  en  $V$  zdd. het aantal iteraties om een Gale-Shapley koppeling te bepalen  $n^2 - n + 1$  is.

### Opgave 6.28

Bepaal een koppeling met maximale cardinaliteit en een oneven knooppunten-bedeeking met minimale cardinaliteit in nevenstaande graaf. Start met de koppeling  $M = \{(1, 2), (3, 4), (5, 6), (7, 8), (9, 10), (11, 12), (13, 14)\}$ .



### Opgave 6.29

Het *bin-packing* probleem is als volgt:

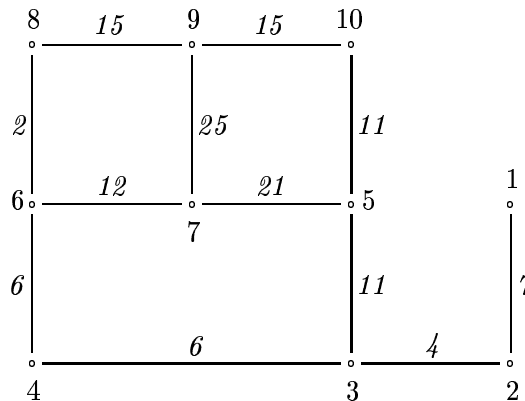
Gegeven zijn  $n$  hoeveelheden  $c_1, c_2, \dots, c_n$  met  $0 < c_i \leq 1$  voor  $1 \leq i \leq n$ .

Bepaal een verdeling van deze hoeveelheden in zo min mogelijk deelverz. zdd. iedere deelverz. een totale hoeveelheid van hoogstens 1 heeft.

Toon aan dat als  $c_i > \frac{1}{3}$  voor  $1 \leq i \leq n$ , dan kan het probleem worden geformuleerd als het vinden van een koppeling met maximale cardinaliteit.

### Opgave 6.30

Bepaal een koppeling met maximale gewicht in nevenstaande graaf.



### Opgave 6.31

Beschouw de volgende heuristiek voor het vinden van een koppeling met maximaal gewicht.

1.  $M = \emptyset$ .
2. a. Kies de tak met het grootste gewicht in de huidige graaf en voeg deze aan  $M$  toe.  
b. Laat de eindpunten van de in onderdeel a gekozen tak weg evens de takken die met deze eindpunten incident zijn.
4. Als er nog minstens twee knooppunten over zijn: ga naar stap 2.  
Anders: stop.

(1) Pas dit algoritme toe op de graaf uit de vorige opgave.

(2) Wat is de complexiteit van dit algoritme?

# Hoofdstuk 7

## Literatuur

### Hoofdstuk 1

Busacker, R.G. and T.L. Saaty: *Finite graphs and networks: an introduction with applications*, Prentice-Hall, Englewood Cliffs (1974).

Deo, N.: *Graph theory with applications to engineering and computer science*, Prentice-Hall, Englewood Cliffs (1974).

Swamey, M.N.S. and K. Thulasiraman: *Graphs, networks and algorithms*, Wiley, New York (1981).

### Hoofdstuk 2

Aho, A.V., J.E. Hopcroft and J.D. Ullman: *Data structures and algorithms*, Addison-Wesley, Reading (1983).

Baase, S.: *Computer algorithms: introduction to design and analysis*, Addison-Wesley, Reading (1988).

Mehlhorn, K.: *Data structures and algorithms 2: Graph algorithms and NP-completeness*, Springer, Berlin (1984).

Sedgewick, R.: *Algorithms*, second edition, Addison-Wesley, Reading (1988).

### Hoofdstuk 3

Ajuha, R.K., T.L. Magnanti and J.B. Orlin: *Network flows: theory, algorithms and applications*, Prentice Hall, Englewood Cliffs (1993).

Lawler, E.L.: *Combinatorial optimization: networks and matroids*, Holt, Rinehart and Winston, New York (1976).

Mehlhorn, K.: *Data structures and algorithms 2: Graph algorithms and NP-completeness*, Springer, Berlin (1984).

Sedgewick, R.: *Algorithms*, second edition, Addison-Wesley, Reading (1988).

### Hoofdstuk 4

Ajuha, R.K., T.L. Magnanti and J.B. Orlin: *Network flows: theory, algorithms and applications*, Prentice Hall, Englewood Cliffs (1993).

Bazaraa, M.S. and J.J. Jarvis: *Linear programming and network flows*, Wiley, New York (1977).



- Busacker, R.G. and T.L. Saaty: *Finite graphs and networks: an introduction with applications*, McGraw-Hill, New York (1965).
- Edmonds, J. and R.M. Karp: *Theoretical improvements in algorithmic efficiency for network flow problems*, Journal of the ACM 19 (1972) 248-264. Ford, L.R. and D.R. Fulkerson: *Flows in networks*, Princeton University Press, Princeton (1962).
- Lawler, E.L.: *Combinatorial optimization: networks and matroids*, Holt, Rinehart and Winston, New York (1976).
- Murty, K.G.: *Network programming*, Prentice Hall, Englewood Cliffs (1992).
- Papadimitriou, C.H. and K. Steiglitz: *Combinatorial optimization: algorithms and complexity*, Prentice Hall, Englewood Cliffs (1982).
- Schrijver, A. : *A course in combinatorial optimization*, Lecture notes (1997).
- Syslo, M.M., N. Deo and J.S. Kowalik: *Discrete optimization algorithms with Pascal programs*, Prentice Hall, Englewood Cliffs (1983).

### Hoofdstuk 5

- Ajuha, R.K., T.L. Magnanti and J.B. Orlin: *Network flows: theory, algorithms and applications*, Prentice Hall, Englewood Cliffs (1993).
- Bazaraa, M.S. and J.J. Jarvis: *Linear programming and network flows*, Wiley, New York (1977).
- Bertsekas, D.P.: *Network optimization: continuous and discrete models*, Athena Scientific, Belmont (1998).
- Bertsimas, D.P. and J.N. Tjittjklis: *Introduction to linear optimization*, Athena Scientific, Belmont (1997).

### Hoofdstuk 6

- Evans, J.R. and E. Minieka: *Optimization algorithms for networks and graphs*, Second edition, Marcel Dekker, New York (1992).
- Hopcroft, J.E. and R.M. Karp: *A  $n^{\frac{5}{2}}$  algorithm for maximum matching in bipartite graphs*, SIAM Journal of Computing 2 (1973) 225-231.
- Lawler, E.L.: *Combinatorial optimization: networks and matroids*, Holt, Rinehart and Winston, New York (1976).
- Lovász, L. and M.D. Plummer: *Matching theory*, Annals of Discrete Mathematics, Volume 29, North Holland, Amsterdam (1986).
- Papadimitriou, C.H. and K. Steiglitz: *Combinatorial optimization: algorithms and complexity*, Prentice Hall, Englewood Cliffs (1982).
- Reichmeider, P.F.: *The equivalence of some combinatorial matching theorems*, Polyganal, Washington (1984).
- Swamey, M.N.S. and K. Thulasiraman: *Graphs, networks and algorithms*, Wiley, New York (1981).