**stichting mathematisch centrum**

MC

AFDELING MATHEMATISCHE BESLISKUNDE BC 14/75 AUGUSTUS

B.J. LAGEWEG

# KORTSTE-PADPROBLEMEN

**2e boerhaavestraat 49 amsterdam**

*Printed at the Mathematical Centre, 49, 2e Boerhaavestraat, Amsterdam.*

*The Mathematical Centre, founded the 11-th of February 1946, is a non-profit institution aiming at the promotion of pure mathematics and its applications. It is sponsored by the Netherlands Government through the Netherlands Organization for the Advancement of Pure Research (Z.W.O), by the Municipality of Amsterdam, by the University of Amsterdam, by the Free University at Amsterdam, and by industries.*

AMS(MOS) subject classification scheme (1970): 05C35, 90B35

# KORTSTE-PADPROBLEMEN

B.J. LAGEWEG
*Mathematisch Centrum, Amsterdam*

Samenvatting

Het bepalen van het kortste pad tussen twee gegeven punten in een graaf wordt geformuleerd als een lineair programmeringsprobleem. Hiervan uitgaande worden algoritmen voor het kortste-padprobleem besproken, waarbij speciaal aandacht wordt geschonken aan het opsporen van negatieve circuits. Het lineaire toewijzingsprobleem wordt opgelost als een reeks van kortste-padproblemen; omgekeerd wordt de formulering van het kortste-padprobleem als een toewijzingsprobleem gebruikt voor postoptimalisatie van kortste paden en het bepalen van M-kortste en M disjuncte paden.

## 1. INLEIDING

Als G een *gerichte graaf* is, met een getal $d_{ij}$ toegekend aan kant (i,j), dan is één van de meest fundamentele problemen op G: bepaal een *kortste pad* van een gegeven beginpunt naar een gegeven eindpunt. Onder kortste pad verstaan we in dit geval een pad zo dat de som van de lengtes $d_{ij}$ van de kanten op dat pad minimaal is.

De getallen $d_{ij}$ kunnen ook de capaciteit van kant (i,j) weergeven. Onze interesse gaat uit naar een zodanig pad van een beginpunt naar een eindpunt dat de minimale capaciteit van een kant in het pad zo groot mogelijk is.

Als $d_{ij}$ de betrouwbaarheid is van kant (i,j), dat is de kans dat een signaal dat in punt i wordt uitgezonden, in punt j wordt ontvangen, dan kunnen we geïnteresseerd zijn in paden met een maximale betrouwbaarheid.

Deze problemen zijn alle voorbeelden van kortste-padproblemen. Daarnaast kan het kortste-padprobleem opduiken als onderdeel van ingewikkelder combinatorische problemen, zoals het toewijzingsprobleem, het handelsreizigersprobleem en het Chinese postbodeprobleem.

We zullen de volgende notatie en definities gebruiken.

Een *graaf* G = (V,E) wordt gedefinieerd door een verzameling *punten* V = {1,2,...,n} en een verzameling van ongeordende paren E. Een ongeordend paar (i,j) in E, genaamd *kant*, is een element van V × V. Een *gerichte graaf* wordt op dezelfde wijze gedefinieerd, zij het dat E uit geordende paren bestaat: kant (i,j) loopt van i naar j. Een kant van het type (i,i) heet een *lus*. In een *gewogen* gerichte graaf is aan iedere kant (i,j) een getal $d_{ij}$ gehecht, de *lengte* of het *gewicht* van de kant.

We zullen i.h.a. aannemen dat de graaf G volledig is. Als G' = (V,E') niet volledig is, dan kunnen we G' zonder bezwaar uitbreiden tot een volledige gewogen graaf G, als volgt:

$$d_{ij} = \begin{cases} \text{lengte van } (i,j) \text{ in } G', & (i,j) \in E' \\ \infty \qquad\qquad\qquad\qquad , & (i,j) \in E - E'. \end{cases}$$

We beperken ons tot gerichte grafen; ongerichte grafen komen in paragraaf 7 kort aan de orde.

Een (gericht) *pad* van punt i naar punt j is een rij kanten van de vorm $(i,i_1),(i_1,i_2),(i_2,i_3),.....,(i_k,j)$. Als beginpunt i en eindpunt j identiek zijn, heet het pad een *circuit*. De lengte van een pad is de som van de lengtes van zijn kanten. Andere definities van lengte, b.v. voor een pad met minimale capaciteit of maximale betrouwbaarheid zijn mogelijk[2,9]; wij zullen ons beperken tot de eerstgenoemde definitie.

## 2. FORMULERING

We zoeken in graaf G het kortste pad van punt 1 naar alle andere punten van G. We nemen daarbij aan dat er een pad met eindige lengte van punt 1 naar elk ander punt bestaat.

We definiëren: $y_j$ = lengte van het kortste pad van 1 naar j. Als de graaf geen circuits met een negatieve lengte heeft, is $y_1 = 0$. Als punt i de voorganger is van punt j op een kortste pad van 1 naar j, dan is $y_j = y_i + d_{ij}$. Bovendien moet dan voor elk punt $k \neq i$ gelden: $y_j \leq y_k + d_{kj}$. De lengtes $y_j$ van kortste paden voldoen dus aan de volgende vergelijkingen, het stelsel van Bellman[1]:

$$(1) \qquad \begin{cases} y_1 = 0 \\ y_j = \min\{y_i + d_{ij} \mid i \neq j\} & , j=2,\dots,n. \end{cases}$$

Omgekeerd, als elk circuit van G een positieve lengte heeft, dan kan bij een willekeurige oplossing $(y_j^*)$ van (1) een opspannende boom met wortel 1 worden gevonden, en wel zo dat de lengte van het pad van 1 naar j in die boom gelijk is aan $y_j^*$. Het pad van 1 naar j bepaalt men door een punt k te zoeken met $y_j^* = y_k^* + d_{kj}$; het pad van 1 naar j in de boom is dan het pad van 1 naar k verlengd met kant (k,j). Tevens kan bewezen worden dat onder deze veronderstellingen stelsel (1) een unieke oplossing heeft. Omdat de kortste-padlengtes $(y_j)$ voldoen aan (1) moet deze unieke oplossing gelijk zijn aan $(y_j)$. Het bepalen van de lengtes $(y_j)$ komt dus neer op het oplossen van de functionaalvergelijkingen (1).

Een andere formulering van het kortste-padprobleem is als een stroomprobleem. We stellen de lengte van kant (j,1), $j \neq 1$, gelijk aan $d_{j1} = 0$. Het probleem

$$(2) \qquad \begin{aligned} &\text{Minimaliseer} && \sum_i \sum_j d_{ij} x_{ij} \\ &\text{onder} && \sum_j x_{ji} - \sum_i x_{ij} = 0 && , i=1,\dots,n \\ &&& x_{i1} = 1 && , i=2,\dots,n \\ &&& x_{ij} \geq 0 && , 1 \leq i,j \leq n, \end{aligned}$$

heeft als optimale oplossing de boom van kortste paden van punt 1 naar alle andere punten, mits de graaf geen negatieve circuits heeft. Het duale probleem van (2) is:

$$\text{Maximaliseer } \sum_{j=2}^{n} y_j$$

$$(3) \qquad \text{onder} \quad y_j - y_i \le d_{ij} \qquad ,1\le i,j\le n$$

$$y_1 = 0$$

Een oplossing voor het lineaire programmeringsprobleem (3) kan als volgt worden geconstrueerd:

<u>Relaxatieprocedure voor het kortste-padprobleem</u>:

1. Initialiseer: $y_1 = 0$;
   $y_j = \infty$, $j=2,\ldots,n$.
2. Als aan alle bijvoorwaarden van (3) is voldaan, stop.
3. Bepaal een kant $(i,j)$ met $y_j - y_i > d_{ij}$;
   Stel $y_j = y_i + d_{ij}$ en ga naar stap 2.

Als de relaxatieprocedure in stap 2 eindigt, is de eindoplossing $(y_j)$ een optimale oplossing van (3): met behulp van $(y_j)$ kan een boom van paden met wortel 1 in G worden geconstrueerd, waarin het pad van 1 naar j lengte $y_j$ heeft. Deze boom geeft een toegelaten oplossing van (2) met waarde $\sum y_j$, zodat $(y_j)$ op grond van de dualiteitsstelling uit de lineaire programmering inderdaad een optimale oplossing van (3) is en $y_j$ de lengte van een kortste pad van 1 naar j.

In de relaxatieprocedure is de keuze van het paar $(i,j)$ in stap 3 vrij. De meeste kortste-padmethoden kunnen worden gezien als een nadere bepaling van stap 3 van de relaxatieprocedure, gebruik makend van de specifieke eigenschappen van de graaf waarop kortste paden worden berekend.

## 3. ACYCLISCHE GRAFEN

Een graaf $G = (V,E)$ is *acyclisch* als hij geen circuits bevat. Omdat we aannemen dat vanuit 1 elk ander punt van de graaf bereikt kan worden, heeft G in geen geval kanten van de vorm $(j,1)$. De volgende procedure levert de kortstepadlengtes voor een acyclische graaf:

1. Initialiseer: $y_1 = 0$;
$y_j = \infty$, $j=2,\ldots,n$;
$T = V$.

2. Als $T = \phi$, stop.

3.1. Kies een punt $i \in T$, waarvan alle voorgangers al eerder gekozen zijn, d.w.z. $\{j \mid (j,i) \in E\} \cap T = \phi$.

3.2. Relaxeer kant $(i,j)$ voor alle opvolgers van i, i.e.
$$y_j := \min\{ y_j, y_i + d_{ij}\} \text{ als } (i,j) \in E.$$

3.3. $T := T - \{i\}$; ga naar stap 2.

Omdat G geen circuits heeft, is in stap 3.1 altijd een punt i te vinden en eindigt de procedure na n keer de loop 2 - 3.3 te hebben doorlopen. Na stap 3.2 is voor alle kanten $(i,j)$ uitgaande van i voldaan aan $y_j - y_i \le d_{ij}$; in het vervolg van de algoritme blijft hieraan voldaan omdat de enig mogelijke verandering een afname van $y_j$ is.

Het aantal operaties (vergelijkingen en optellingen) van de algoritme wordt bepaald door de stappen 3.1 en 3.2. In een goede implementatie van de algoritme wordt iedere kant één keer opgepakt voor 2 vergelijkingen en 1 optelling: de orde van de methode is $O(n^2)$.

De waarde van de lengtes $d_{ij}$ doet niet terzake. Als men de berekening uitvoert met $-d_{ij}$ vindt men langste paden van punt 1 naar de andere punten. Het langste -padprobleem is voor acyclische grafen dus efficiënt oplosbaar, i.e. oplosbaar door een algoritme waarvan het aantal operaties polynomiaal begrensd is in de probleemparameters.

## 4. GRAFEN MET KANTEN VAN POSITIEVE LENGTE

De eigenschap dat de lengtes van alle kanten in de graaf G positief zijn, wordt geëxploiteerd in de algorithme van Dijkstra[5]:

1. Initialiseer: $y_1 = 0$;
   $y_j = \infty$, $j=2,\ldots,n$;
   $T = V$.

2. Als $T = \phi$, stop.

3.1. Kies $i \in T$ zo dat $y_i = \min\{ y_j \mid j \in T\}$.
3.2. Relaxeer kant $(i,j)$ voor alle opvolgers $j$ in $T$ van punt $i$, i.e.
$$y_j := \min\{ y_j, y_i + d_{ij}\} \text{ als } (i,j)\in E \text{ en } j\in T.$$
3.3. $T := T - \{i\}$; ga naar stap 2.

In stap 3.1 wordt ieder punt i eenmaal gekozen. Op het moment van de keuze van punt i geldt: $\max\{ y_j \mid j\notin T\} \le y_i \le \min\{ y_j \mid j\in T\}$.
Voor $j \in T$ is aan $y_j - y_i \le d_{ij}$ voldaan na stap 3.2; omdat $y_i$ niet meer veranderd en $y_j$ alleen kan afnemen, blijft de ongelijkheid vervuld.
Voor $j \notin T$ geldt $y_j \le y_i$, zodat $y_j - y_i \le 0 \le d_{ij}$; hier wordt gebruikt dat de lengtes van alle kanten positief zijn.

De orde van de procedure is $O(n^2)$, evenals voor acyclische grafen. In een handige implementatie zijn voor een volledige graaf $\frac{1}{2}n^2$ optellingen en $n^2$ vergelijkingen nodig.

## 5. HET ALGEMENE GEVAL: DE BELLMAN-YEN ALGORITME

Als een graaf zowel circuits als kanten met negatieve lengte bevat, zijn de voorgaande procedures niet bruikbaar. Bellman[1] en Yen[20] hebben een procedure ontwikkeld die iteratief de functionaalvergelijkingen (1) oplost.

We veronderstellen dat geen negatieve circuits optreden. Dan bestaat het kortste pad $(1,i_1),(i_1,i_2),\ldots,(i_{k-1},i_k)$ van 1 naar $i_k$ uit tenhoogste n-1 kanten. We spreken van een *omslag* in het pad als een kant $(i,j)$ gevolgd wordt door een kant $(j,k)$, waarbij het teken van $i-j$ verschilt van het teken van $j-k$. Een kortste pad bevat ten hoogste $n-2$ omslagen.

Definiëren we $y_j^{(k)}$ als de lengte van het kortste pad van 1 naar j met minder dan k omslagen, dan wordt $y_j = y_j^{(n-1)}$ gevonden uit het stelsel:

$$\begin{cases} y_j^{(0)} = d_{1j} \\ y_j^{(k+1)} = \min\{\ y_j^{(k)},\ \min\{\ y_i^{(k+1)} + d_{ij} \mid i<j\}\ \} & ,\ k \text{ even} \\ y_j^{(k+1)} = \min\{\ y_j^{(k)},\ \min\{\ y_i^{(k+1)} + d_{ij} \mid i>j\}\ \} & ,\ k \text{ oneven.} \end{cases}$$

Om $y_j^{(n-1)}$ te bepalen zijn ruwweg $\frac{1}{2}n^3$ optellingen en $\frac{1}{2}n^3$ vergelijkingen nodig: de orde van de algoritme is $O(n^3)$. In termen van de relaxatieprocedure valt op merken dat ieder ongelijkheid van (3) $\frac{1}{2}n$ keer gerelaxeerd wordt, waarbij de volgorde van de relaxaties van te voren vastligt.

## 6. KORTSTE PADEN TUSSEN ALLE PAREN PUNTEN

Kortste paden tussen alle paren punten kunnen worden berekend door achtereenvolgens ieder punt als oorsprong te kiezen en de kortste paden van die oorsprong naar de andere punten te bepalen. Het is echter mogelijk alle kortste paden tegelijk te berekenen.

De algoritme van Floyd[7] is gebaseerd opde eigenschap dat een kortste pad van i naar j, zeg $(i,i_1),(i_1,i_2),\ldots,(i_p,j)$, ten hoogste n-2 *tussenpunten* $i_1$, $i_2,\ldots,i_p$ heeft. Als k zo'n tussenpunt is, dan bestaat het kortste pad van i naar j uit het kortste pad van i naar k, gevolgd door het kortste pad van k naar j. Definiëren we $y_{ij}^{(k)}$ als de lengte van het kortste pad van i naar j, met tussenpunten uit de verzameling $\{1,2,\ldots,k\}$, dan kunnen we $y_{ij}^{(n)}$ iteratief bepalen uit:

$$\begin{cases} y_{ij}^{(0)} = d_{ij} \\ y_{ij}^{(k)} = \min\{\ y_{ij}^{(k-1)},\ y_{jk}^{(k-1)} + y_{kj}^{(k-1)}\ \}, & k=1,\ldots,n. \end{cases}$$

De algoritme van Floyd vergt ruwweg $n^3$ optellingen en $n^3$ vergelijkingen, en is dus van dezelfde orde als de Bellman-Yen-procedure. De methode is bruikbaar voor alle grafen die geen negatieve circuits bevatten.

## 7. NEGATIEVE CIRCUITS

Als een graaf negatieve circuits bevat, is de lengte van het kortste pad tussen 2 punten i en j, waarin ieder punt van de graaf willekeurig vaak mag optreden, niet gedefinieerd. Brengen we de restrictie aan dat ieder punt maximaal eenmaal in het pad mag voorkomen, dan is het kortste pad wel gedefinieerd; dit nieuwe kortste-padprobleem is echter tenminste zo moeilijk als het handelsreizigersprobleem.

Een handelsreizigersroute op een graaf G, i.e. het kortste circuit van G dat ieder punt van G precies één keer bevat, kan worden verkregen door een dergelijk kortste-padprobleem op te lossen. Daartoe voegen we aan G een punt, zeg v, toe en vervangen iedere kant (j,1) door een kant (j,v) met lengte $d_{jv} = d_{j1}$; vervolgens worden alle afstanden verminderd met een getal M, dat zo groot is gekozen, dat een pad met k + 1 kanten altijd korter is dan een pad met k kanten. Het kortste pad van 1 naar v, waarin ieder pad slechts eenmaal mag voorkomen, levert de oplossing van het handelsreizigersprobleem.

Het bestaan van een efficiënte algoritme voor dit type kortste-padprobleem, - of voor het langste-padprobleem met dezelfde restrictie op de punten in een pad -, zou betekenen dat ook het handelsreizigersprobleem, en vele moeilijke combinatorische problemen efficiënt zouden kunnen worden opgelost. Algoritmen voor dit probleem moeten dan ook worden gezocht in de sfeer van de impliciete aftelling.

Als we kortste paden gaan berekenen in een graaf die kanten van negatieve lengte en circuits bevat, dan is het zaak om na te gaan of de graaf negatieve circuits bevat. We kunnen hiervoor dezelfde algoritmen gebruiken die de kortste paden berekenen.

Bellman en Yen hanteren de volgende eigenschap: als het kortste pad van 1 naar j k-1 omslagen bevat, dan is het kortste pad in de $k^{de}$ iteratie gevonden en verandert de lengte van het pad van 1 naar j niet meer na de $k^{de}$ iteratie, bij afwezigheid van negatieve circuits. Het bestaan van een kortste pad met k-1 omslagen impliceert het bestaan van kortste paden met k-2, k-3,...,0 omslagen. Derhalve mag in de $k^{de}$ iteratie voor ten hoogste n-k getallen $y_j^{(k)}$ gelden: $y_j^{(k)} < y_j^{(k-1)}$, k=2,...,n. In het bijzonder geldt $y_j^{(n)} = y_j^{(n-1)}$ voor alle j. Als tijdens de algoritme van Bellman-Yen aan voorgaande stelling niet is voldaan, bevat de graaf een negatief circuit. Eenvoudig is in te zien dat de graaf geen negatieve circuits bevat als in een iteratie alle lengtes van paden niet veranderen, d.i. als geldt $y_j^{(k)} = y_j^{(k-1)}$ voor alle j in een iteratie k, $2 \le k \le n$.

Opmerking: De eigenschap dat tenminste k getallen $y_j^{(k)}$ in de $k^{de}$ iteratie onveranderd blijven, kan ook worden gebruikt om het aantal operaties in de algoritme van Bellman-Yen verder te beperken tot $n^3/4$ optellingen en $n^3/4$ vergelijkingen. Voor k even wordt $y_j^{(k+1)}$ dan bepaald als:

$$y_j^{(k+1)} = \min\{\, y_j^{(k)},\ \min\{\, y_i^{(k+1)} + d_{ij} \mid i<j,\ y_i^{(k+1)} < y_i^{(k-1)} \} \}.$$

De algoritme van Floyd ontdekt negatieve circuits middels de $y_{jj}^{(k)}$. Stellen we de afstanden $d_{jj} = \infty$, dan bevat het netwerk een negatief circuit d.e.s.d. als $y_{jj}^{(k)} < 0$ voor een j, $1 \le j \le n$, en een k, $1 \le k \le n-1$. Zijn er geen negatieve circuits, dan is $y_{jj}^{(n)}$ de lengte van het kortste circuit door punt j.

Ofschoon de algoritmen van Bellman-Yen en Floyd de aanwezigheid van negatieve circuits kunnen aantonen, zijn zij allereerst bedoeld om kortste paden te berekenen. Als men alleen uit is op het ontdekken van negatieve circuits, zijn andere methoden beschikbaar[8, 15].

Het kortste pad-probleem op een ongerichte graaf is te herleiden tot een kortste-padprobleem op een gerichte graaf, als alle lengtes $d_{ij}$ niet-negatief zijn. In dat geval vervangen we elke ongerichte kant (i,j) door een gerichte kant (i,j) en een gerichte kant (j,i), beide met dezelfde lengte als de ongerichte kant. De aldus gevormde gerichte graaf bevat geen negatieve circuits; kortste paden kunnen met Dijkstra of Floyd worden berekend.

Is de lengte van kant (i,j) negatief, dan ontstaat op deze manier een negatief circuit $i \to j \to i$. We kunnen het probleem nu aanpakken met een speciale algoritme van Tobin[18] van de orde $O(n^3)$, of door het te formuleren als een geheeltallig lineair programmeringsprobleem. In deze laatste aanpak wordt het kortste pad tussen 1 en n gevonden uit:

$$\begin{array}{lll} \text{Minimaliseer} & \sum\limits_{i \ne j} c_{ij} x_{ij} & \\ \text{onder} & \sum\limits_{j \ne 1} x_{1j} = 1 & \\ & \sum\limits_{j \ne i} x_{ij} + 2x_{ii} = 2, & 2 \le i \le n-1; \\ & \sum\limits_{j \ne n} x_{nj} = 1 & \\ & \qquad x_{ij} \in \{0,1\}. & \end{array}$$

Dit gewogen 2-koppelingsprobleem kan met een algoritme van Edmonds[6] worden opgelost. Mits geen negatieve ongerichte circuits voorkomen, bestaat een oplossing uit een kortste pad tussen 1 en n, plus lussen op de punten niet op dit kortste pad.

## 8. HET TOEWIJZINGSPROBLEEM ALS EEN KORTSTE-PADPROBLEEM

Zij gegeven een bipartite graaf $B = (I,J,E)$, waarbij $I = \{1,2,...,n\}$, $|J| = n$; kant $(i,j) \in E \subset I \times J$ heeft gewicht $a_{ij}$. Het (lineaire) *toewijzingsprobleem* op graaf B luidt: bepaal een volledige *koppeling* K, i.e. een verzameling kanten uit E zo dat ieder punt van B op precies één kant van K ligt, met minimaal gewicht. Ofwel:

$$
(4)\qquad
\begin{array}{lll}
\text{Minimaliseer} & \sum_i \sum_j a_{ij}x_{ij} & \\
\text{onder} & \sum_j x_{ij} = 1, & \text{voor alle } i; \\
 & \sum_i x_{ij} = 1, & \text{voor alle } j; \\
 & x_{ij} \in \{0,1\}, & \text{voor alle } i \text{ en } j.
\end{array}
$$

Een oplossing $(x_{ij})$ van (4) correspondeert met een koppeling K bestaande uit alle kanten waarvoor $x_{ij} = 1$, en vice versa.

Het duale probleem van (4) is:

$$
(5)\qquad
\begin{array}{lll}
\text{Maximaliseer} & \sum_i u_i + \sum_j v_j & \\
 & u_i + v_j \le a_{ij}, & \text{voor alle } i \text{ en } j.
\end{array}
$$

We construeren een oplossing $(x_{ij},u_i,v_j)$ van (4) en (5) door middel van een aantal kortste-padproblemen op grafen met niet-negatieve afstanden. Uit de gevonden kortste paden wordt een koppeling K geconstrueerd; gelijktijdig wordt met behulp van de bijbehorende lengtes een duale oplossing $(u_i,v_j)$ gevormd. Uit de constructie volgt dat het resultaat $(x_{ij},u_i,v_j)$ voldoet aan

$$
(6)\qquad x_{ij} = 1 \quad \Rightarrow \quad u_i + v_j = a_{ij},
$$

zodat de gevonden oplossing optimaal is op grond van de dualiteitsstelling van de lineaire programmering.

Algoritme voor het toewijzingsprobleem:

0. Initialiseer: $K_0 = \phi$ ($K_0$ is de lege koppeling);

   $I_0 = \phi$;

   $v_j = 0$, voor alle $j$;

   $k = 1$.

1. Construeer de gewogen gerichte graaf $B_k$, waarop het $k^{de}$ kortste-padprobleem speelt:

$$B_k = (I_k, J, E_1 \cup E_2), \text{ waarbij}$$
$$I_k = I_{k-1} \cup \{k\},$$
$$E_1 = \{ (i,j) \mid i \in I_k,\ j \in J,\ (i,j) \in E-K_{k-1}\},$$
$$E_2 = \{ (j,i) \mid (i,j) \in K_{k-1}\},$$
$$u_k = \min\{ a_{kj} - v_j \mid j \in J\},$$
$$d_{ij} = \begin{cases} a_{ij} - u_i - v_j, & (i,j) \in E_1, \\ 0, & (i,j) \in E_2. \end{cases}$$

2. Bereken met behulp van de algoritme van Dijkstra het kortste pad van punt $k$ naar de andere punten van $B_k$. Voor de lengtes $y_j$ van de kortste paden geldt:

$$(7) \quad \begin{cases} y_k = 0 \\ y_i = y_j, \text{ als } (i,j) \in K_{k-1} \\ y_j = y_i + a_{ij} - u_i - v_j, \text{ als } (i,j) \text{ in het kortste pad van } k \text{ naar } j \\ y_j \leq y_i + a_{ij} - u_i - v_j, \text{ als } (i,j) \in E_1. \end{cases}$$

3. Bepaal een punt $j_0 \in J$ dat niet op de koppeling $K_{k-1}$ ligt en dat het dichtst bij punt $k$ ligt, d.w.z. kies $j_0$ zo dat

$$y_{j_0} = \min\{ y_j \mid j \in J,\ (i,j) \notin K_{k-1} \text{ voor alle } i \in I_k\}.$$

   Zij $P$ het kortste pad van $k$ naar $j_0$.

4. Construeer koppeling $K_k$. $K_k$ bevat alle kanten van $K_{k-1}$ die niet op het kortste pad $P$ liggen, en alle kanten van $P$ die niet tot $K_{k-1}$ behoren:

$$K_k = \{(i,j) \mid (i,j) \in K_{k-1},\ (j,i) \notin P\} \cup \{(i,j) \mid (i,j) \in P-E_2\}.$$

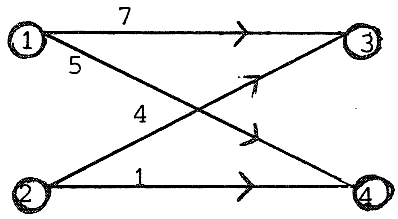
5. Bereken nieuwe duale variabelen $u_i$ en $v_j$:

$$u_i := u_i - y_i \quad , \text{ voor alle } i \in I_k;$$
$$v_j := v_j + y_j \quad , \text{ voor alle } j \in J.$$

6. Als $k < n$, stel dan $k := k + 1$ en ga naar stap 1.

   Anders stop: koppeling $K_n$ is volledig.

Voorbeeld: $2 \times 2$ toewijzingsprobleem.

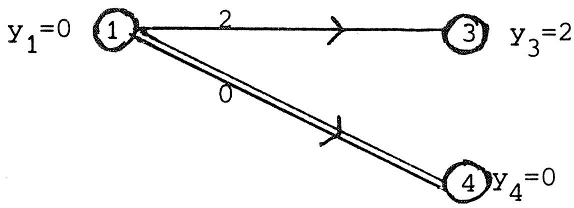



graaf B

Iteratie 1:

$u_1=5$ $\quad v_3=0$

$v_4=0$

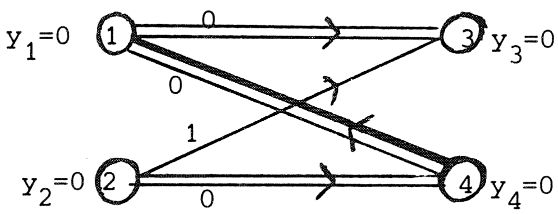



graaf $B_1$

Iteratie 2:

$u_1=5$ $\quad v_3=2$

$u_2=1$ $\quad v_4=0$



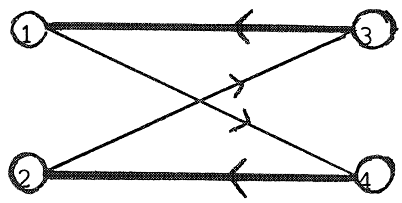

graaf $B_2$

Oplossing:

$u_1=5$ $\quad v_3=2$

$u_2=1$ $\quad v_4=0$

$x_{14}= x_{23}=1$





$(i,j)\in P, d_{ij}=d$ $\qquad$ $(i,j)\in K_{k-1}$

Kanttekeningen bij de toewijzingsalgoritme:

1. Na de $k^{de}$ uitvoering van stap 5 is voldaan aan de bijvoorwaarden van het duale probleem (5) en aan de optimaliteitsvoorwaarden (6) voor alle paren $(i,j)$ met $i \in I_k$ en $j \in J$, op grond van (7). Bijgevolg zijn in de volgende iteratie alle afstanden $d_{ij}$ in graaf $B_{k+1}$ niet-negatief, zodat de kortste-padalgoritme van Dijkstra gebruikt kan worden.
2. Pad P bestaat uit een oneven aantal kanten, zeg 2p+1, waarvan p kanten afkomstig uit $E_2$ en p+1 kanten uit $E_1$. Derhalve heeft $K_k$ één kant meer dan $K_{k-1}$, zodat na n iteraties een volledige koppeling $K_n$ is gevonden, samen met waarden $(u_i, v_j)$ die aan (5) en (6) voldoen: $K_n$ is optimaal.
3. De algoritme is van de orde $O(n^3)$, omdat de algoritme van Dijkstra in stap 2 die n keer wordt uitgevoerd, van de orde $O(n^2)$ is.

## 9. HET KORTSTE-PADPROBLEEM ALS TOEWIJZINGSPROBLEEM

Zij G een volledige gerichte graaf op de punten $\{1,2,...,n\}$; kant $(i,j)$ heeft lengte $d_{ij}$. We zoeken in G een kortste pad van punt 1 naar punt n.

We kunnen dit kortste-padprobleem opvatten als een lineair toewijzingsprobleem op een volledige bipartite graaf $B = (I, J, A)$; $I = \{1,2,...,n-1\}$, $J = \{2,3,...,n\}$, kant $(i,j) \in A$ heeft lengte $a_{ij} = 0$ als $i=j$ en lengte $a_{ij} = d_{ij}$ als $i \neq j$. Het toewijzingsprobleem is:

$$\begin{array}{lll} \text{Minimaliseer} & \sum_i \sum_j a_{ij} x_{ij} & \\ \text{onder} & \sum_{j \in J} x_{ij} = 1 & \text{, voor alle } i \in I; \\ & \sum_{i \in I} x_{ij} = 1 & \text{, voor alle } j \in J; \\ & x_{ij} \in \{0,1\} & \end{array} \tag{8}$$

De kanten $(i,j)$ waarvoor $x_{ij}=1$ in een oplossing van (8), vormen in de graaf G een pad van 1 naar n en een aantal circuits. Als er geen negatieve circuits in G voorkomen, zal de optimale oplossing van (8) bestaan uit een kortste pad van 1 naar n, plus lussen op de punten niet op dat kortste pad.

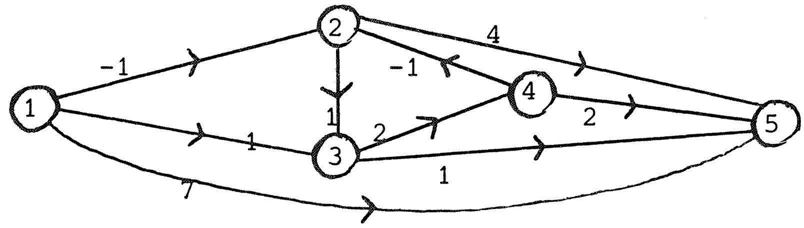
Als we (8) aanpakken met de in de vorige paragraaf geschetste algoritme hebben we een aanpak voor het kortste-padprobleem van de orde $O(n^3)$, gelijk aan de orde van de algoritme van Bellman-Yen. De methode is vooral van belang voor postoptimalisatie[19].

Stel we beschikken over een optimale oplossing $(x_{ij}, u_i, v_j)$ van (8) en het duale probleem van (8). Als gegevens betrekking hebbende op één punt van de graaf G veranderen, kan in $O(n^2)$ operaties een nieuw optimum worden berekend, zoals onderstaande voorbeelden illustreren.

Voorbeeld 1.

De lengte van de kanten uitgaande van punt $i_0$ verandert. We laten nu in de optimale koppeling corresponderende met $(x_{ij})$ de kant incident met punt $i_0$ weg en stellen $u_{i_0} = \min \{a_{i_0 j} - v_j \mid j \in J\}$. We beschikken dan over een koppeling met n-2 kanten en een duale oplossing die voldoen aan (5) en (6). Door één keer de stappen 2 - 5 van de toewijzingsalgoritme uit te voeren (orde $O(n^2)$) wordt weer een volledige koppeling van n-1 kanten gevonden.

Hetzelfde mechanisme kan worden gebruikt om na te gaan of G negatieve circuits bevat. We lossen eerst probleem (8) op, waarbij $a_{1n} = -K$, K een voldoende groot positief getal, gesteld is (fig. 1 en 2). G heeft d.e.s.d. geen




Figuur 1 Graaf G (m.u.v. kanten met lengte ∞)

| | $v_j$ | 0 | 1 | 1 | 0 |
|---|---|---|---|---|---|
| $u_i$ | i \ j | 2 | 3 | 4 | 5 |
| -K | 1 | -1 | 1 | ∞ | -K |
| 0 | 2 | 0 | 1 | ∞ | 4 |
| -1 | 3 | ∞ | 0 | 2 | 1 |
| -1 | 4 | -1 | ∞ | 0 | 2 |

Figuur 2 Toewijzingsprobleem om vast te stellen of G negatieve cycles bevat.

| | $v_j$ | 0 | 1 | 3 | 2 |
|---|---|---|---|---|---|
| $u_i$ | $i \backslash j$ | 2 | 3 | 4 | 5 |
| -1 | 1 | -1 | 1 | ∞ | 7 |
| 0 | 2 | 0 | 1 | ∞ | 4 |
| -1 | 3 | ∞ | 0 | 2 | 1 |
| -3 | 4 | -1 | ∞ | 0 | 2 |

Figuur 3 Toewijzingsprobleem om kortste pad $P^{(0)}$ van 1 naar n te bepalen; $P^{(0)} = (1,2),(2,3),(3,5)$ met lengte $\ell(P^{(0)})=1$.

negatieve circuits als de optimale oplossing van (8) waarde -K heeft. Vervolgens geven we $a_{1n}$ de oorspronkelijke waarde terug, stellen $x_{1n} = 0$ en berekenen in $O(n^2)$ operaties een volledige koppeling(fig.3).

Voorbeeld 2.

Punt i verdwijnt uit de graaf. We schrappen nu in het toewijzingsprobleem rij i en kolom i behorende bij punt i. Als $x_{ii}=1$, d.w.z. i ligt niet op het kortste pad van 1 naar n, is de koppeling in het resulterende probleem nog steeds volledig. Anders heeft de koppeling één kant te weinig; door één kortste-padberekening wordt weer een volledige en optimale koppeling gevonden.

Voorbeeld 3.

Punt w wordt toegevoegd aan de graaf G. We breiden het toewijzingsprobleem uit met een rij w en een kolom w en stellen $v_w = \min\{ a_{iw} - u_i \mid i \in I\}$ en $u_w = \min\{ a_{wj} - u_j \mid j \in J \cup \{w\}\}$. De uitgebreide duale oplossing $(u_i, v_j)$ voldoet aan de bijvoorwaarden van het duale probleem; de koppeling wordt volledig gemaakt door één kortste-padberekening met punt w als oorsprong.

Voorbeeld 4.[11,21]

Het kortste pad $P^{(0)}$ van punt 1 naar punt n is bekend, zeg $(i_0=1,i_1)$, $(i_1,i_2),\ldots,(i_{p-1},i_p=n)$. We zijn geïnteresseerd in de *kortste afwijking* $P_k^{(0)}$ van pad $P^{(0)}$, gedefinieerd als het kortste pad zonder herhaling van punten dat tot en met punt $i_k$ $(0 \le k < p)$ samenvalt met $P^{(0)}$ en kant $(i_k, i_{k+1})$ niet bevat.

$P_k^{(0)}$ kan worden bepaald door:

1. de kanten in het kortste pad $P^{(0)}$ vóór punt $i_k$ verplicht te stellen;
2. kant $(i_k,i_{k+1})$ uit de graaf te verwijderen;
3. het resterende toewijzingsprobleem optimaal op te lossen door één kortste-padberekening, uitgaande van de optimale oplossing voor $P^{(0)}$.

Het kortste pad onder de kortste afwijkingen $P_0^{(k)}$ $(0\le k<p)$ is het op één na kortste pad $P^{(1)}$ van alle paden van 1 naar n. We kunnen op een soortgelijke manier de kortste afwijkingen $P_0^{(1)},\dots,P_q^{(1)}$ van pad $P^{(1)}$ bepalen; daarbij moeten de eisen die golden bij de berekening van $P^{(1)}$, opnieuw worden gerespecteerd. Het op-twee-na-kortste pad $P^{(2)}$ is het kortste pad onder de paden $P_k^{(0)}$ $(0\le k<p,\ P_k^{(0)}\ne P^{(1)})$ en $P_k^{(1)}$ $(0\le k<q)$. Zo doorgaande vindt men achtereenvolgens $P^{(0)}$, $P^{(1)}$, ..., $P^{(M)}$,... , waarbij $P^{(M)}$ het op-M-na-kortste pad van 1 naar n is.

Toegepast op graaf G (fig. 1) vinden we:

kortste pad $P^{(0)} = (1,2),(2,3),(3,5)$ $\qquad \ell(P^{(0)})=1$

kortste afwijkingen van $P^{(0)}$:

$$\begin{cases} P_0^{(0)} = (1,3),(3,5) & \ell(P_0^{(0)})=2 \\ P_0^{(0)} = (1,2),(2,5) & \ell(P_1^{(0)})=3 \\ P_2^{(0)} = (1,2),(2,3),(3,4),(4,5) & \ell(P_2^{(0)})=4 \end{cases}$$

$$\ell(P^{(1)}) = \min\{\ \ell(P_0^{(0)}),\ \ell(P_1^{(0)}),\ \ell(P_2^{(0)})\} = 2$$

kortste afwijkingen van $P^{(1)} = P_0^{(0)}$:

$$\begin{cases} P_0^{(1)} = (1,5) & \ell(P_0^{(1)})=7 \\ P_1^{(1)} = (1,3),(3,4)\ (4,5) & \ell(P_1^{(1)})=5 \end{cases}$$

$$\ell(P^{(2)}) = \min\{\ell(P_1^{(0)}),\ \ell(P_2^{(0)}),\ \ell(P_0^{(1)}),\ell(P_1^{(1)})\} = 3$$

kortste afwijkingen van $P^{(2)} = P_1^{(0)}$: geen

$$\ell(P^{(3)}) = \min\{\ \ell(P_2^{(0)}),\ \ell(P_0^{(1)}),\ \ell(P_1^{(1)})\} = 4$$

kortste afwijkingen van $P^{(3)} = P_2^{(0)}$: geen

$$\ell(P^{(4)}) = \min\{\ \ell(P_0^{(1)}),\ \ell(P_1^{(1)})\} = 5$$

kortste afwijkingen van $P^{(4)} = P_1^{(1)}$:

$P_2^{(2)} = (1,3),(3,4),(4,2),(2,5)$ $\qquad \ell(P_2^{(4)})=6$

$\ell(P^{(5)}) = \min\{ \ell(P_0^{(1)}), \ell(P_2^{(4)})\} = 6$

kortste afwijkingen van $P^{(5)} = P_2^{(4)}$: geen

$\ell(P^{(6)}) = \min\{ \ell(P_0^{(1)})\} = 7$

kortste afwijkingen van $P^{(6)} = P_0^{(1)}$: geen.

Het is prettig om kortste paden op een dergelijke manier te kunnen genereren, als we geinteresseerd zijn in een kortste pad dat een bepaalde eigenschap bezit; bijvoorbeeld het pad bevat tenminste a punten uit een verzameling A, ten hoogste b punten uit een verzameling B, en de afstand op het pad tussen 2 punten uit een verzameling $C \supset \{1,n\}$ is ten hoogste c. We genereren eenvoudigweg kortste paden naar stijgende lengte, totdat we een pad treffen met de gewenste eigenschap.

Voorbeeld 5.[17]

In betrouwbaarheidsproblemen is het vaak nodig om te beschikken over 2 disjuncte, d.w.z. zonder gemeenschappelijke tussenpunten, paden van punt 1 naar punt n, zo dat de som van hun lengtes minimaal is. Als we het kortste pad van 1 naar n hebben bepaald door oplossing van het toewijzingsprobleem (zie fig. 3), kunnen we 2 disjuncte paden met minimale totale lengte vinden door het toewijzingsprobleem uit te breiden met een rij 1' en een kolom n'. We stellen $a_{1'j} = a_{1j}$ $(1<j<n)$, $a_{in'} = a_{in}$ $(1<i<n)$, $a_{ij} = \infty$ voor de overige cellen in rij 1' en kolom n', $u_{1'} = u_1$ en $v_{n'} = v_n$, en bepalen een volledige koppeling in het aldus uitgebreide toewijzingsprobleem, uitgaande van de eindoplossing van het oorspronkelijke probleem (fig. 4). Als deze koppeling een eindig gewicht heeft, representeert hij 2 paden van $\{1,1'\}$ naar $\{n,n'\}$ zonder gemeenschappelijke punten, plus lussen op de punten niet in één van deze paden. Deze 2 paden corresponderen met 2 disjuncte paden in G van minimaal totaal gewicht.

In het algemeen kunnen uit M disjuncte paden door toevoeging van een rij en een kolom aan het toewijzingsprobleem M+1 disjuncte paden worden geconstrueerd, of kan worden geconstateerd dat M het maximum aantal disjuncte paden van 1 naar n is (fig. 5).

| | $v_j$ | 0 | 2 | 6 | 5 | 5 |
|---|---|---|---|---|---|---|
| $u_i$ | i \ j | 2 | 3 | 4 | 5 | 5' |
| -1 | 1 | -1 | 1 | ∞ | 7 | ∞ |
| -1 | 2 | 0 | 1 | ∞ | 4 | 4 |
| -4 | 3 | ∞ | 0 | 2 | 1 | 1 |
| -6 | 4 | -1 | ∞ | 0 | 2 | 2 |
| -1 | 1' | -1 | 1 | ∞ | ∞ | ∞ |

Figuur 4  2 disjuncte paden: (1,2),(2,5') en (1',3),(3,5)

| | $v_j$ | 0 | 2 | 9 | 8 | 8 | 8 |
|---|---|---|---|---|---|---|---|
| $u_i$ | i \ j | 2 | 3 | 4 | 5 | 5' | 5'' |
| -1 | 1 | -1 | 1 | ∞ | 7 | ∞ | ∞ |
| -4 | 2 | 0 | 1 | ∞ | 4 | 4 | 4 |
| -7 | 3 | ∞ | 0 | 2 | 1 | 1 | 1 |
| -9 | 4 | -1 | ∞ | 0 | 2 | 2 | 2 |
| -1 | 1' | -1 | 1 | ∞ | ∞ | ∞ | ∞ |
| -1 | 1'' | -1 | 1 | ∞ | ∞ | ∞ | ∞ |

Figuur 5  3 disjuncte paden: (1,5); (1',3),(3,5'') en (1'',2),(2,5')

LITERATUUR


1. R.E. Bellman, "On a Routing Problem," *Quart. Appl. Math.* 16 (1958), 87-90.
2. R. Cuninghame-Green, "Minimax Algebra," *Technical Memorandum No. 70, 83, 86, 88* (1975), Twente University of Technology, Enschede.
3. B. Dorhout, "Het Lineaire Toewijzingsprobleem: Vergelijking van Algoritmen," *Report BN 21* (1973), Mathematisch Centrum, Amsterdam.
4. S.E. Dreyfus, "An Appraisal of Some Shortest-Path Algorithms," *Operations Res.* 17 (1969), 395-412.
5. E.W. Dijkstra, "A Note on Two Problems in Connexion with Graphs," *Numerische Mathematik* 1 (1959), 269-271.
6. J. Edmonds & W.R. Pulleyblank, "The Matching Problem and the Blossom Algorithm," University of Waterloo, Waterloo (to appear).
7. R.W. Floyd, "Algorithm 97, Shortest Path," *Comm. ACM* 5 (1962), 345.
8. M. Florian & P. Robert, "A Direct Search Method to Locate Negative Cycles in a Graph," *Management Sci.* 17 (1971), 307,310.
9. M. Gondran, "Algèbre Lineaire et Cheminement dans un Graphe," *Report HI 1137/02* (1973), Electricité de France, Clamart.
10. E.L. Lawler, *Combinatorial Optimization: Networks and Matroids*, Holt Rinehart and Winston, New York (to appear).
11. E.L. Lawler, " A Procedure for Computing the K Best Solutions to Discrete Optimization Problems and its Application to the Shortest Path Problem," *Management Sci.* 18 (1972), 401-405.
12. E. Minieka, "On Computing Sets of Shortest Paths in a Graph," *Comm. ACM* 17 (1974), 351-353.
13. K.G. Murty, "An Algorithm for Ranking all the Assignments in Increasing Order of Cost," *Operations Res.* 16 (1968), 682-687.
14. A.R. Pierce, "Bibliography on Algorithms for Shortest Path, Shortest Spanning Tree, and Related Circuit Routing Problems," *Networks* 5 (1975), 129-149.
15. M.A. Pollatschek & B. Avi-Itzhak, "An Efficient Shortest-Route Algorithm," *Angewandte Informatik* 11 (1974), 477-482.
16. P.M. Spira, "A New Algorithm for Finding All Shortest paths in a Graph of Positive Arcs In Average Time $O(n^2 \log^2 n)$," *SIAM Journ. on Comp.* 2 (1973), 28-32.
17. J.W. Suurballe, "Disjoint Paths in a Network," *Networks* 4 (1974), 125-145.
18. R.L. Tobin, "Finding a Minimal Weight Path in an Undirected Network," *Technical Report* (1974), University of Michigan, Ann Arbor.

19. A.Weintraub, "The Shortest and the K-Shortest Routes as Assignment Problems," *Networks* 3 (1973), 61-73.

20. J.Y. Yen, "An Algorithm for Finding Shortest Routes from All Source Nodes to a Given Destination in General Networks," *Quart. Appl. Math.* 27 (1970), 526-530.

21. J.Y. Yen, "Finding the K Shortest Loopless Paths in a Network," *Management Sci.* 17 (1971), 712-716.