Hoofdstuk 8

Stroomnetwerken

Een stroomnetwerk is een gerichte graaf met **twee speciale knopen**, een **source (producent)** en een **sink (verbruiker)**. De producent levert een materiaal met een gelijkmatig debiet, de verbruiker verwerkt dat materiaal met hetzelfde debiet. Het materiaal stroomt van source naar sink via de verbindingen van de graaf. Elke knoop van de graaf is bereikbaar vanuit de producent, evenals de verbuiker bereikbaar is vanuit elke knoop.

Elke gerichte verbinding is een soort pijpleiding met een bepaalde capaciteit. De knopen zijn enkel kruispunten van pijpleidingen. De stroom door de knopen is *conservatief*, in is gelijk aan uitstroom.

8.1 Maximalestroomprobleem

Het maximalestroomprobleem is het eenvoudigste probleem in stroomnetwerken. Er moet zoveel mogelijk materiaal van de producent naar de verbruiker stromen, zonder de capaciteit van de verbindingen te overschrijden.

Methode van Ford-Fulkerson.
Dit is een methode, geen algoritme. De methode is iteratief, en bij elke iteratie neemt de stroom vanuit de producent toe, tot het maximum bereikt wordt.

De capaciteit gaan we definiëren als een positief getal $c(i,j)$, de stroom is $s(i,j)$ en wordt beperkt door die capaciteit.

$$\textit{Nettostroom } f = \sum \left(s(p,j) - s(j,p) \right)$$

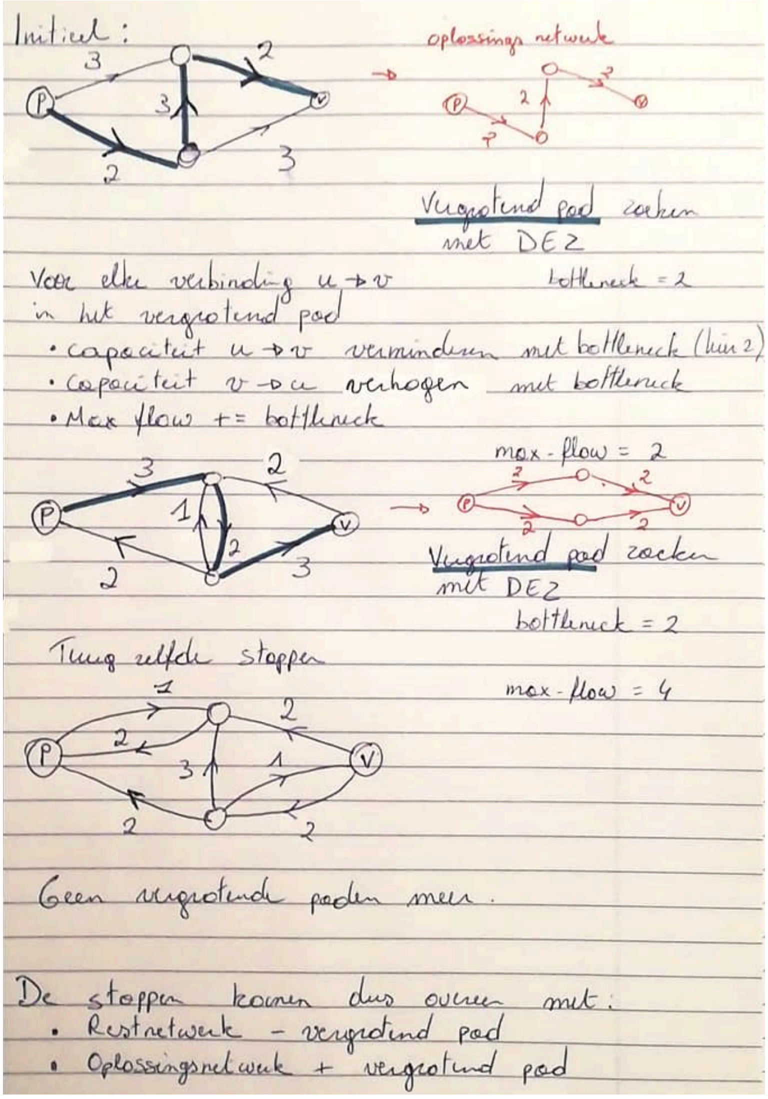
Er moet dus de som genomen worden over alle knopen j waarvoor een verbinding (p,j) of (j,p) bestaat. De waarde f is de *netwerkstroom* die maximaal moet worden.

- Alle stromen voor alle mogelijke knopenparen noemt men de *stroomverdeling*. De methode van Ford-Fulkerson start met een stroomverdeling die overal nul is.
- Bij elke iteratie wordt de huidige stroomverdeling gebruikt, samen met de verschillende capaciteiten, om overzicht te krijgen op mogelijke stroomtoenames. Dit overzicht is een nieuw stroomnetwerk, met dezelfde knopen, maar niet noodzakelijk dezelfde verbindingen en capaciteiten: **het restnetwerk**.
- In het restnetwerk wordt dan een weg van producent naar verbruiker gezocht waar materiaal kan stromen. Dit is de **vergrotende weg**. Met die extra stroom wordt het oorspronkelijke netwerk aangepast.
- Als er geen vergrotende weg meer gevonden wordt, stopt de iteratie en is de netwerkstroom maximaal.

```
Als R het restnetwerk is, S
het oorspronkelijk netwerk en
O het initiële netwerk met
stroomverdeling nul, dan
krijgen we het volgende:

R = S - O

while (vergrotend pad bestaat
in R) {
   O += vergrotend pad;
   R -= vergrotend pad;
}
```



Om aan te tonen dat meneertje Ford en meneertje Fulkerson, beiden meneertjes, want toen mochten dames nog niet naar school en dienden ze enkel als huishoudhulp en seksobjecten, gelijk hadden is er het begrip **snede (cut)**.

Een snede *(P,V)* van een samenhangende graaf is een verzameling verbindingen die de knopenverzameling in twee niet-ledige stukken P en V verdeelt. Een verbinding *(i,j)* behoort dus tot de snede als *i* in P zit en *j* in V (of omgekeerd). Wij hebben alleen nut voor de sneden waarbij *p (producent)* in P zit en *v (verbruiker)* in V zit.

De capaciteit *c(P,V)* is de som van alle capaciteiten van de verbindingen in de snede. De nettostroom *f(P,V)* is de som van de voorwaartse stromen *s(i,j)* min de som van de achterwaartse stromen *s(j,i)*.

Logischerwijs wordt *f* begrensd door *c*, voor elke snede *(P,V)*.

De vorige formule van de nettostroom wordt nu de volgende:

$$Nettostroom\ f = \sum_{i \in P} \sum_{j \in V} \ (s(p,j) - s(j,p)) = f(P,V)$$

We bekomen uiteindelijk volgende drie equivalente eigenschappen:

1. De netwerkstroom f is maximaal.
2. Er is geen vergrotende weg meer te vinden in het restnetwerk.
3. De netwerkstroom f is gelijk aan *een* snede in de oorspronkelijke graaf.

Er zijn hiervoor soorten implementaties gegeven in de cursus:

- **Performantie afhankelijk van de capacitieten (.)(.).** De performantie van dit soort implementaties wordt niet enkel bepaald door de graaf (*n* en *m*), maar ook door de grootte van de capaciteiten. Hun performantie is *pseudopolynomiaal*.
    - Stel dat C de grootste capaciteit voorstelt. De maximale stroom is dan O(nC), stel je een producent en verbruiker voor, met ertussen een kolom knopen, allemaal rechtstreeks verbonden met *p* en *v*, en met capaciteit C.
Het aantal iteraties in Ford-Fuckerson wordt dan ook O(nC).
Een restnetwerk bepalen is O(m), daarin de vergrotende weg zoeken (DEZ of BEZ) is ook O(m). Dit geeft een totaal van *O(nmC)*. Dit is enkel goed voor kleine C-waarden.
    - Als men steeds de juiste vergrotende weg kiest, die met de grootste stroomtoename, dan wordt het aantal iteraties O(m lg C). Om die weg te vinden gebruik je een variatie op Dijkstra (O(m lg n)), totaal krijgen we dan $O(m^2 \lg n \lg C)$.
    - Een eenvoudige, snellere variant op het vorige heet *"capacity scaling"* en volgt volgende observaties:
        - Een vergrotende weg met stroomtoename van minstens *c*, of besluiten dat er geen dergelijke weg bestaat, is O(m).
        - Stel dat er geen enkele dergelijke weg gevonden wordt, dan is de minimale snedecapaciteit van het restnetwerk *mc.* We halveren nu de ondergrens, en gaan op zoek naar wegen met een stroomtoename van minstens *c/2*. Er zijn maximaal *2m* dergelijke wegen.
Indien we beginnen met ondergrens $c = 2^{\lg C}$, en telkens halveren om te eindigen bij 1, dan wordt de maximale stroom inderdaad bereikt met O(m lg C) iteraties.

    Dit geeft een totale performantie van $O(m^2 \lg C)$.

- **Performantie onafhankelijk van de capaciteiten.** Als de vergrotende weg steeds het *minimum aantal verbindingen heeft*, dan stijgt de lengte van de vergrotende wegen na hoogstens m iteraties. Aangezien de maximale lengte *n-1* is, volstaan *O(nm)* iteraties. Elke iteratiestap gebruikt BEZ, en is dus O(m). Zo wordt de totale performantie $O(nm^2)$.

Een grote nadeel van algoritmen die grootste wegen zoeken is dat de stroomtoename telkens van p naar v moet, wat in het slechtste geval O(n) is. Dit kan echter gefixt worden door de recentere **preflow-push-methode** te gebruiken.

Deze methode vindt de maximale stroom door stroomtoename langs *individuele verbindingen*, de push-operatie, in plaats van langs een vergrotende weg. De stroom is tijdens de uitvoering van het algoritme niet langer conservatief (in ≠ uit). De instroom kan groter zijn dan de uitstroom, dan is er tijdelijk *preflow*. Knopen met een dergelijk overschot noemen we *actieve knopen*.

Zolang er actieve knopen zijn, voldoet de oplossing niet. De basisoperatie van deze methode is het selecteren van een actieve knoop en zijn overschot proberen weg te werken langs zijn buren. Als er geen actieve knopen meer zijn, dan voldoet de stroom en is die bovendien maximaal.

Enkele implementaties:

- Eenvoudig: *O(n2m)*
- Fifo preflow-push: *O(n3)*
- Highest label preflow-push: $O(n^2\sqrt{m})$
- Excess-scaling: $O(nm + n^2 \lg C)$

8.2 Verwante Problemen

- **Meerdere producenten en meerdere verbruikers.** Men wil dan de gezamenlijke nettostroom van alle producenten maximaliseren. Om dit eenvoudig op te lossen kan men een fictieve *totaalproducent* en een fictieve *totaalverbruiker* invoeren. Vanuit de totaalproducent komen verbindingen met *onbeperkte capaciteit* naar de producenten. Zelfde verbindingen worden ingevoerd tussen de verbruikers en de totaalverbruiker.

- **Capaciteitsbeperking in knopen.** Om opnieuw een gewoon stroomnetwerk te bekomen *ontdubbeld men elke knoop, en wordt een verbinding voorzien tussen de originele knoop en zijn dubbelganger, met de capaciteit van de knoop*.

- **Ongericht stroomnetwerk.** Elke verbinding kan dan vervangen worden door *een paar verbindingen, één in elke richting*. Beide verbindingen krijgen de originele capaciteit.

- **Ondergrens voor de stroom.** Hierbij onderzoekt men eerst of een netwerkstroom wel mogelijk is. Indien ja, transfeert men die daarna in een maximale stroom. Beide fasen vereisen de oplossing van een gewoon stroomnetwerk, met enkel bovengrenzen voor de stromen.

- **Meerder soorten materiaal.** Dit heet een *multicommodity network*. Voor elke soort is er één producent, en ook één verbruiker. In elke knoop is de stroom voor elke soort *apart conservatief* (een soort kan niet omgezet worden in een andere). De gezamenlijke stroom van alle materialen mag de capaciteit echter niet overschrijden.

- **Minimalekostenprobleem.** Naast de capaciteit kan ook een *kost per stroomeenheid* aan elke verbinding toegekend worden.

8.3 Meervoudige Samenhang in Grafen

De definities voor enkelvoudige en dubbel samenhang en lijnsamenhang kunnen veralgemeend worden:

- Wanneer er tussen elk paar knopen *k* onafhankelijke wegen bestaan (in beide richtingen voor gerichte grafen), zonder gemeenschappelijke *knopen,* dan is deze graaf **k-connected**. Bij een sterk samenhangde graaf is *k=1*. Bij een dubbel samenhangende graaf is *k=2*.
- Wanneer tussen elk paar knopen *k* onafhankelijke wegen bestaan (in beide richtingen voor gerichte grafen), zonder gemeenschappelijke *verbindingen*, dan is deze graaf **k-edge-connected**.

Eenvoudige vormen van samenhang (*k=1, 2, 3*) kunnen efficiënt via DEZ onderzocht worden. Voor meervoudige samenhang doet men beroep op stroomnetwerken. Er is immers een nauw verband tussen maximale stroomnetwerken en *minimale sneden*:

Als men een maximale netstroom vindt, heeft men meteen een
minimale snede gevonden - *max-flow min-cut stelling*

Als alle capaciteiten 1 zijn, dan is de *c(P,V)* gelijk aan het aantal verbindingen van knopen in P naar knopen in V. Elke vergrotende weg laat een stroomtoename van 1 toe, en satureert meteen een weg van *p* naar *v*, die één van de verbindingen van de minimale snede gebruikt.

Samenhang in een graaf is gegeven door de stelling van Menger. Er zijn vier versies: gericht, ongericht, meervoudige samenhang, meervoudige lijnsamenhang.

Het minimum aantal verbindingen dat moet verwijderd worden
om een knoop *v* van een gerichte graaf onbereikbaar te maken vanuit een
andere knoop *p* is gelijk aan het maximaal aantal lijnonafhankelijke wegen
van *p* naar *v* (zonder gemeenschappelijke verbindingen).
- *Meervoudig lijnsamenhangende gerichte graaf*

Deze stelling volgt uit de eigenschappen van een stroomnetwerk met capaciteit één. Beide aantallen zijn immers gelijk aan de capaciteit van een minimale snede (P,V).

- De maximale stroom loopt over een maximaal aantal lijnonafhankelijke wegen van *p* naar *v*. Dat aantal is gelijk aan de maximale stroom, en dus ook aan *c(P,V)*.
- Een minimale verzameling verbindingen M, die *v* onbereikbaar maakt vanuit *p* vormt een snede (P,V). Aangezien die verzameling minimaal is, is M een minimale snede, met een capaciteit gelijk aan het aantal verbindingen in M.

Om een verband te leggen tussen onafhankelijke wegen *zonder gemeenschappelijke knopen* en vergrotende wegen ontdubbelt men alle knopen en voegt men er een verbinding met capaciteit één tussen. Zo kan er maar één vergrotend pad gebruik maken van elke knoop. Om de knopen te vinden die *v* onbereikbaar maken vanuit *p* wanneer je ze wegneemt kan men de capaciteit van alle andere verbindingen onbeperkt groot maken.

Stelling van Whitney past de eigenschap toe op elk paar knopen van een graaf (ook vier versies).

Een graaf is k-(edge-)connected als er tenminste k knopen (verbindingen) moeten verwijderd worden om hem te doen uiteenvallen.

Om dus de samenhangsgraad k van een graaf te vinden kan men dus een maximalestroomprobleem oplossen voor elk knopenpaar en het minimum van al die oplossingen nemen. De performantie wordt dan $O(n^3m)$.