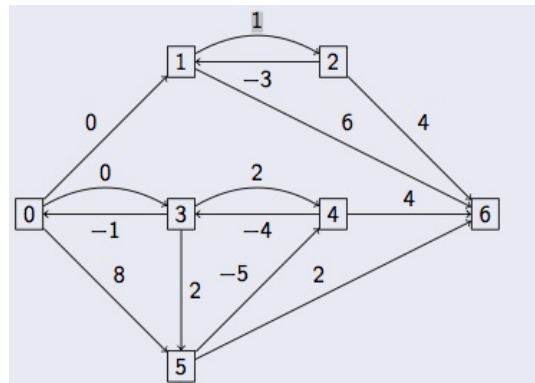


## Plan-It project planning voor Shell.

Plan-It is een software pakket dat met name door Shell gebruikt wordt voor het plannen van activiteiten op de werkvloer.

Gegeven een olieplatform met  $N$  activiteiten. De activiteiten kunnen gedeeltelijk onafhankelijk van elkaar worden uitgevoerd. Gedeeltelijk zijn er volgordebependingen: je kunt olie niet transporteren voordat deze uit de grond gehaald is. Verder heeft elke activiteit een bepaalde duur. Gevraagd is om een schema op te stellen dat de totale projectduur minimaliseert. Voor dit simpele model is er ook een simpele oplossing via de constructie van een gerichte graaf, waarbij een pijl van  $A$  naar  $B$  aangeeft dat activiteit  $A$  moet zijn uitgevoerd vóórdat  $B$  begint. Dit is in BK2 aan de orde geweest.

In de praktijk zijn er veel gedetailleerdere volgordebependingen: bijv.  $B$  mag pas een gegeven aantal tijdseenheden beginnen nadat  $A$  begonnen is: dit kan te maken hebben met een bepaalde overlap van handelingen die nodig zijn om de activiteit te starten. Dergelijke bependingen kun je ook met een gerichte graaf modelleren. Hoe? Dat staat geïllustreerd in nevenstaand voorbeeld: de 1 bij de pijl  $1 \rightarrow 2$  betekent dat taak 2 *minstens* 1 tijdseenheid *na* taak 1 moet beginnen; de  $-3$  bij de pijl  $2 \rightarrow 1$  betekent dat taak 1 minstens  $-3$  tijdseenheden na 2 moet beginnen, m.a.w. taak 2 moet *hooguit* 3 tijdseenheden *na* 1 beginnen. Het niet-bestaan van een ronde met positieve lengte is nodig en voldoende opdat er een projectplanning gemaakt kan worden zonder schending van de tijdsbependingen [1].



Veronderstel nu dat het starten van werk aan taak  $j$  op tijdstip  $t$  een bedrag  $w_{jt}$  kost. In [1] staat ook een minimale-snedelgoritme gegeven waarmee een project planning met minimale kosten bepaald wordt. Hierbij wordt gebruik gemaakt van een vantevoren bepaalde bovengrens op de duur van het project. Kun je hiermee ook de totale projectduur minimaliseren?

**In de praktijk..** In de praktijk is het zelfs nog gecompliceerder: elke activiteit vereist hulpmiddelen (grondstoffen, gespecialiseerde arbeiders), en van elk type hulpmiddel is maar een beperkte hoeveelheid beschikbaar (capaciteitsbependingen). Onder deze voorwaarden wordt het probleem een integer lineair programmeringsprobleem in de klasse **NP**. Praktisch niet oplosbaar dus.

**Heuristieken** Plan-IT moet capaciteitsbependingen wel inbouwen. Heuristieken zijn derhalve nodig. Eén daarvan is gebaseerd op het feit dat het integer LP zonder de capaciteitsbependingen totaal unimodulair is. Je hoeft dus slechts het LP zonder integraliteitseis op te lossen.

We zetten nu de capaciteitsbependingen als ‘soft constraint’ in de doelfunctie, met willekeurig gekozen coëfficiënten (‘Lagrange multiplicatoren’). Dit geeft als oplossing een project planning die hopelijk niet al te veel capaciteitsbependingen schendt. Vervolgens ‘haal’ je de verkregen project planning door een volgende heuristiek die een project planning oplevert, die wel aan de capaciteitsbependingen voldoet. In een lopend afstudeeronderzoek zijn verschillende heuristieken geïmplementeerd.

Een probleem dat zich voordoet, is hoe de Lagrange multiplicatoren gekozen moeten worden. Een methode is om ze willekeurig te trekken uit  $[0, 1]$ , en dat maar flink vaak te herhalen. Opmerkelijk is dat dit maar heel zelden leidt tot een optimale project planning. Het is niet duidelijk, waarom dat zo is.

**Optimaliteit** Een belangrijke vraag waarop de ontwerpers van Plan-IT antwoord willen hebben is: hoe goed zijn de uitkomsten van verschillende heuristieken?

**Project** Het project kan de volgende onderdelen omvatten:

1. Het bestuderen van [1], i.h.b. het genoemde minimale-snedevalgoritme, Lagrange relaxatie en zogenaamde  $\alpha$ -completion times. De laatste zijn heuristieken waarmee een project planning verkregen kan worden die aan de beperkingen voldoet.
2. Bestuderen van mogelijke heuristieken in [2,3] en eventueel implementatie hiervan.
3. Analyse van de vraag hoe de Lagrange multiplicatoren gekozen moeten worden.
4. Onderzoek naar ‘resource levelling’ heuristieken, waarmee het verbruik van de hulpmiddelen gelijkmatiger over de duur van het project kunnen worden verdeeld.
5. Numerieke analyse van testgevallen om de kwaliteit van een heuristiek te beoordelen.

## Referenties

- [1] R.H. Möhring, A.S. Schulz, F. Stork, M. Ütz (2003). Solving Project Scheduling Problems by Minimum Cut Computations.
- [2] K. Neumann, Ch. Schwindt, J. Zimmermann (2003). Project Scheduling with Time Windows and Scarce Resources: Temporal and Resource-Constrained Project Scheduling with Regular and Nonregular Objective Functions. Springer-Verlag.
- [3] N. Christofides, R. Alvarez-Valdes and J.M. Tamarit (1987). Project scheduling with resource constraints: A branch and bound approach. European Journal of Operational Research **29**, 262–273.

Floske Spieksma  
spieksma@math.leidenuniv.nl  
Bachelorproject voor het AS&B seminarium  
voorjaar 2012