

STICHTING
MATHEMATISCH CENTRUM
2e BOERHAAVESTRAAT 49
AMSTERDAM

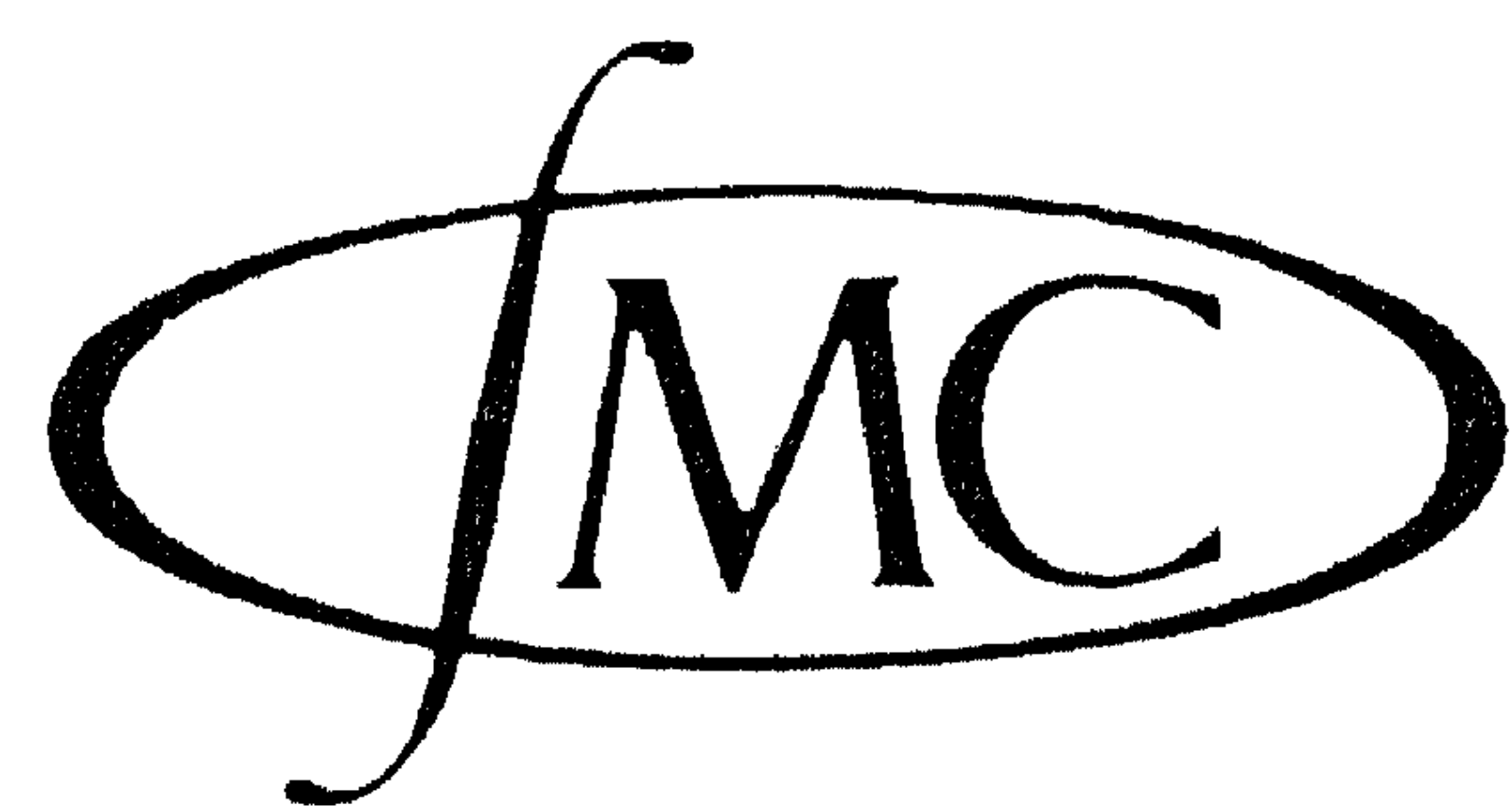
AFDELING MATHEMATISCHE STATISTIEK

S 391 (SP 107)

Een indelingsprobleem

door

Jac.M. Anthonisse



december 1967

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

Summary

A branch and bound algorithm is given to solve the following problem:

To each pair of elements (i,j) from a set $X = \{1, \dots, n\}$ a number r_{ij} with $r_{ij} \geq 0$, $r_{ij} = r_{ji}$ and $r_{ii} = 0$ has been assigned.

Find a prescribed number of disjoint subsets $P_1, \dots, P_m$ from X , such that

$$X = \bigcup_{h=1}^m P_h,$$

and $z = \sum_{h=1}^m z_h$ is minimized,

where

$$z_h = \sum_{i < j} (r_{ij} | i \in P_h, j \in P_h).$$

Experiments indicate that an optimal solution is usually found in a small number of iterations, but the verification may be rather time consuming.

The algorithm may be used to find the minimum value of m for which a partitioning of X with $z = t$ exists. The algorithm appears to be efficient for finding this 'chromatic number of a graph'.

Inleiding

De te beschrijven algorithmen werden, naar aanleiding van een praktijkprobleem, onder leiding van prof.dr. G. de Leve ontwikkeld.

Alle berekeningen werden, geprogrammeerd in ALGOL 60, uitgevoerd door de Electrologica X8-rekenautomaat.

Dit artikel bevat allereerst een definitie van de problemen waarvoor de algorithmen toepasbaar is. Daarna worden de resultaten van enkele experimenten gegeven. Na een algemene beschouwing van aftel-algorithmen wordt de algorithmen besproken. Hierop volgt een voorbeeld. Tenslotte wordt aangegeven hoe de algorithmen voor de oplossing van een met het oorspronkelijke verwant probleem kan worden gebruikt.

Probleemstelling

Een voorbeeld van het te bespreken type indelingsproblemen wordt gevonden bij het samenstellen van college roosters. Bij het verdelen van de colleges over de beschikbare uren zal er onder meer naar worden gestreefd dat colleges die voor nagenoeg hetzelfde gehoor zijn bestemd tijdens verschillende uren worden gegeven. Om verschillende indelingen van de colleges met elkaar te vergelijken is een criterium nodig. Een mogelijk criterium is het aantal hoor-uren dat verloren gaat als twee colleges tegelijkertijd worden gegeven en er studenten zijn die beide colleges zouden willen volgen. Is dit criterium gekozen dan zal worden gezocht naar een indeling waarvoor het criterium minimaal is, een dergelijke indeling heet optimaal.

Zij $X = \{1, \dots, n\}$ de verzameling in te delen colleges (elk college duurt 1 uur), laten m uren beschikbaar zijn, zij P_h de verzameling colleges die tijdens het h -de uur worden gegeven ($h = 1, \dots, m$), rij r_{ij} = het aantal studenten dat zowel college i als college j wil volgen ($1 \leq i \neq j \leq n$) en rij $r_{ii} = 0$ ($i = 1, \dots, n$).

Een oplossing van het nu te formuleren mathematische probleem bepaalt dan een indeling van de colleges die, in zekere zin, optimaal is.

Aan elk paar (i, j) ($1 \leq i, j \leq n$) elementen uit een verzameling $X = \{1, \dots, n\}$ is een getal r_{ij} met $r_{ij} \geq 0$, $r_{ij} = r_{ji}$ en $r_{ii} = 0$ toegevoegd.

Bepaal een voorgeschreven aantal disjuncte deelverzamelingen $P_1, \dots, P_m$ van X zó, dat

$$X = \bigcup_{h=1}^m P_h \quad (1)$$

en

$$z = \sum_{h=1}^m z_h \quad \text{minimaal} \quad (2)$$

waarbij

$$z_h = \sum_{i < j} (r_{ij} | i \in P_h, j \in P_h) \quad (h = 1, \dots, m). \quad (3)$$

Vele besliskundige problemen kunnen in bovenstaande vorm worden gebracht of zijn met het geformuleerde probleem verwant, doch ook in de grafentheorie wordt een verwant probleem gesteld:

bepaal de minimale waarde van m waarvoor een indeling met $z = 0$ bestaat.

Deze minimale waarde van m heet dan het kleurgetal van een door de r_{ij} bepaalde graaf.

Resultaten

Het gestelde probleem kan op verschillende manieren worden geformuleerd als een lineair programmeringsprobleem waarbij de variabelen slechts de waarde 0 of 1 mogen aannemen. Het aantal variabelen en voorwaarden stijgt echter zo snel met n dat deze problemen voor $n > 15$ praktisch onoplosbaar zijn.

Het is echter mogelijk gebleken een aftel-algorithme (branch and bound algorithm) te ontwikkelen waarmee ook grotere problemen kunnen worden opgelost.

Alvorens de algorithme te beschrijven zullen de resultaten van enkele experimenten worden gegeven.

Voor elk van de waarden $n = 5, 10, 15, 20$ werden (voor vaste ρ) 5 matrices (r_{ij}) gegenereerd met behulp van a-selecte trekkingen uit de verdeling:

$$P(\underline{x} = 0) = \rho$$

$$P(\underline{x} = i) = \frac{1-\rho}{n} \quad (i = 1, \dots, n).$$

Hierbij was $\rho = .2, .4, .6, .8$, zodat in totaal 80 matrices werden beschouwd. Voor elke matrix werden de indelingsproblemen met $m = 2, 3, \dots, n-1$ opgelost.

Tabel 1 vermeldt de waargenomen rekentijden en de minimale waarden van z voor de 5 matrices die met $n = 10, \rho = .4$ werden gegenereerd.

Tabel 2 vermeldt dezelfde gegevens als tabel 1, voor $n = 20, \rho = .2$.

Voor elk probleem worden 2 rekentijden vermeld:

t_s = voor het vinden van de oplossing benodigde tijd

t_t = totale tijd.

Tabel 3 geeft een samenvatting van alle waargenomen rekentijden. Voor elke n vermeldt de tabel de waargenomen maximale t_s en de waargenomen maximale t_t .

Alle tijden zijn gegeven in seconden.

m	matrix 1			matrix 2			matrix 3			matrix 4			matrix 5		
	z	ts	tt	z	ts	tt	z	ts	tt	z	ts	tt	z	ts	tt
2	37	.16	.90	45	.39	1.04	38	.17	.95	21	.16	.72	40	.16	.81
3	6	.68	.94	10	.34	1.05	7	.13	.39	4	.15	.95	6	.14	.20
4	0	.29	.30	4	.46	2.91	0	.11	.12	0	.11	.12	3	.12	.19
5	0	.10	.11	0	1.08	1.09	0	.09	.10	0	.10	.11	1	.09	.14
6	0	.07	.08	0	.45	.46	0	.07	.08	0	.07	.08	0	.08	.09
7	0	.05	.06	0	.05	.06	0	.06	.07	0	.06	.07	0	.05	.06
8	0	.04	.05	0	.03	.04	0	.04	.05	0	.03	.04	0	.03	.04
9	0	.01	.02	0	.02	.03	0	.01	.02	0	.02	.03	0	.02	.03

resultaten voor $n = 10$, $\rho = .4$

tabel 1.

m	matrix 1			matrix 2			matrix 3			matrix 4			matrix 5		
	z	ts	tt	z	ts	tt	z	ts	tt	z	ts	tt	z	ts	tt
2	636	.73	13.32	650	.74	12.76	621	1.76	12.77	535	1.76	12.70	636	.74	12.26
3	293	3.48	83.78	319	1.69	88.70	320	1.66	92.51	244	1.66	76.39	322	.70	91.07
4	162	3.37	323.82	172	.70	281.93	166	16.21	280.86	123	.68	148.93	178	26.99	331.72
5	93	6.23	451.39	81	12.18	111.73	96	5.36	194.10	62	12.84	80.77	92	3.12	179.98
6	44	1.39	141.67	48	1.36	76.13	50	19.92	79.20	30	2.74	27.73	47	2.21	41.18
7	22	2.00	62.99	21	.54	22.64	29	12.92	63.68	13	2.03	10.50	20	7.91	12.42
8	9	3.12	33.00	9	1.81	34.05	11	8.34	12.15	4	.52	2.68	8	.50	4.37
9	1	1.89	4.57	3	16.64	25.40	3	1.65	1.82	1	.47	2.59	3	.47	2.51
10	0	3.51	3.51	0	7.59	7.60	1	.41	.58	0	.97	.98	0	1.25	1.26
11	0	2.22	2.23	0	3.17	3.18	0	.36	.37	0	.37	.38	0	1.04	1.04
12	0	.31	.32	0	1.03	1.04	0	.31	.31	0	.32	.32	0	.72	.73
13	0	.27	.28	0	.76	.77	0	.26	.27	0	.26	.27	0	.27	.28
14	0	.22	.23	0	.52	.53	0	.21	.22	0	.22	.23	0	.22	.23
15	0	.17	.18	0	.18	.19	0	.16	.17	0	.17	.18	0	.17	.18
16	0	.13	.13	0	.14	.14	0	.12	.13	0	.13	.14	0	.13	.14
17	0	.09	.10	0	.10	.11	0	.09	.10	0	.09	.10	0	.10	.10
18	0	.06	.07	0	.07	.08	0	.07	.08	0	.07	.07	0	.06	.06
19	0	.04	.04	0	.04	.05	0	.04	.05	0	.04	.04	0	.03	.04

resultaten voor n = 20, $\rho = .2$

tabel 2.

n	5	10	15	20
tt	.11	2.91	17.83	451.39
ts	.09	1.08	4.94	26.99

waargenomen maximale rekentijden

tabel 3.

n \ ρ				
	.2	.4	.6	.8
5	.04	.04	.05	.01
10	.21	.14	.15	.20
15	.49	.85	.76	.53
20	1.26	1.87	1.65	1.03
25	2.92	3.18	3.08	3.32
30	6.11	5.59	3.96	3.61
35	7.84	8.32	9.31	7.23

rekentijden kleurgetal

tabel 4.

In elk van de beschouwde gevallen werd de optimale oplossing van het indelingsprobleem in een gering aantal iteratiestappen gevonden. Nadat een indeling is gevonden onderzoekt de algoritme of een indeling met een lagere z -waarde bestaat. Bestaat een zodanige indeling niet, dan is de gevonden indeling een optimale indeling. Bij vele van de beschouwde indelingsproblemen werd het grootste deel van de totale rekentijd gebruikt voor deze verificatie van de reeds gevonden optimale indeling.

Uitgaande van zekere veronderstellingen betreffende de matrices (r_{ij}) zou op grond van ervaringen met de algoritme wellicht een stop-regel kunnen worden ontwikkeld die aangeeft wanneer de berekeningen mogen worden afgebroken omdat, behoudens een tevoren vastgestelde kans, een optimale indeling is gevonden.

Voor een aantal matrices werd de minimale waarde van m bepaald waarvoor een indeling met $z = 0$ bestaat. Tabel 4 vermeldt de waargenomen maximale rekentijden in seconden. Voor elke combinatie van n en ρ werden 5 matrices gegenereerd.

Aftel-algorithmen

In het nu volgende is S de verzameling van alle toegelaten oplossingen van een gegeven minimaliseringsprobleem.

Zij S een eindige verzameling en f een op S gedefinieerde reële functie. In principe kan een punt uit S waarvoor f minimaal is, worden bepaald door f voor elk punt van S te berekenen. Wegens praktische bezwaren is deze methode veelal onuitvoerbaar, bijvoorbeeld als S zeer veel punten bevat.

Op grond van de volgende overweging kan het berekenen van f voor alle punten van S in vele gevallen worden vervangen door het berekenen van f in een relatief klein aantal punten van S .

Zij z^* een bovengrens voor de minimale waarde van f , dus

$$\min_{s \in S} f(s) \leq z^*,$$

zij S' een deelverzameling van S , en g een ondergrens voor f beperkt tot S' , dus

$$g \leq f(s') \text{ voor alle } s' \in S'.$$

Als $z^* < g$ dan geldt:

$$\min_{s \in S} f(s) < f(s') \text{ voor alle } s' \in S',$$

dus een punt van S waarvoor f minimaal is behoort niet tot S' . Dit betekent dat f niet hoeft te worden berekend voor de tot S' behorende punten.

De aftel-algorithmen hebben globaal de volgende stappen gemeen:

1. splits S , op een voorgeschreven wijze, in een aantal niet-lege, disjuncte deelverzamelingen S' en bereken voor elk van die deelverzamelingen, volgens een voorgeschreven principe, een ondergrens g voor f (dus $g \leq f(s')$ voor alle $s' \in S'$); ^{*)}

) Het is niet altijd bekend of voor een gegeven probleem een oplossing bestaat. Als $S = \emptyset$ is stap 1 onuitvoerbaar. Veelal wordt niet S maar een niet-lege verzameling S^ met $S \subset S^*$ gesplitst. Wordt voor een deelverzameling $S' \subset S^*$ gevonden dat $S' \cap S = \emptyset$ dan wordt aan S' de ondergrens $g = +\infty$ toegevoegd.

2. kies, op voorgeschreven wijze, een gegenereerde nog niet verder gesplitste deelverzameling, splits deze zoals hierboven voor S beschreven en bereken de bij de deelverzamelingen behorende ondergrenzen, zet dit proces voort tot een deelverzameling wordt gekozen die slechts één punt bevat;
3. zij s het gevonden punt, bereken $f(s)$, definieer $z^* = f(s)$ en vervang de bij het punt behorende ondergrens door $f(s)$;
4. kies een gegenereerde, nog niet verder gesplitste deelverzameling met ondergrens $g < z^*$, splits deze verzameling zoals in stap 1 voor S beschreven en bereken de bij de deelverzamelingen behorende ondergrenzen, zet dit proces (stap 4) voort tot:
 óf geen deelverzameling kan worden gekozen, de algoritme is beëindigd,
 óf een deelverzameling wordt gekozen die uit één punt bestaat, ga verder bij 3.

In de te beschrijven algoritme voor het indelingsprobleem heeft het principe voor het berekenen van de bij een deelverzameling behorende ondergrens g van f de eigenschap:

$$g = f(s)$$

als de deelverzameling uit het punt s bestaat. Bovendien heeft elk indelingsprobleem een oplossing, dus $S \neq \emptyset$. Op grond van deze eigenschappen kan de algoritme dan iets eenvoudiger worden beschreven:

1. stel $z^* = +\infty$, en beschouw S als een gegenereerde deelverzameling met $g < \infty$;
2. kies een gegenereerde, niet verder gesplitste, deelverzameling met $g < z^*$, splits deze in een aantal niet-lege, disjuncte deelverzamelingen en bereken de bijbehorende ondergrenzen, zet dit proces voort tot:
 voor alle gegenereerde, niet verder gesplitste, deelverzamelingen is $g \geq z^*$, de algoritme is beëindigd,
 of een uit één punt bestaande deelverzameling wordt gekozen;
3. zij s het gevonden punt, vervang de huidige waarde van z^* door $f(s)$, ga verder bij 2.

Voor een gegeven probleem kunnen veelal verschillende aftel-algorithmen worden geconstrueerd, er zijn verschillende criteria mogelijk om (in 2. en 4.) een gegenereerde deelverzameling te kiezen, en een gekozen deelverzameling kan op verschillende manieren worden gesplitst.

Het is nuttig een algoritme zó in te richten dat de minimale waarde van f in het algemeen na een klein aantal iteratiestappen wordt gevonden. Immers dan is ook z^* minimaal, en kunnen deelverzamelingen snel worden overgeslagen. Dit zal vooral efficiënt zijn als zó wordt gesplitst dat het mogelijk is een ondergrens te berekenen die slechts weinig van de minimale waarde van f (beperkt tot de deelverzameling) verschilt.

Een veel gebruikte splitsingsmethode voor het geval S uit een deelverzameling van een N -dimensionale ruimte bestaat ($N \geq 2$) is de splitsing door partiële oplossingen. Een punt $\underline{s}$ van S (een oplossing) kan worden beschreven door zijn coördinaten:

$$\underline{s} = (\underline{s}_1, \dots, \underline{s}_N)$$

of door het stelsel

$$s_i = \underline{s}_i \quad (i = 1, \dots, N). \quad *)$$

Een stelsel

$$s_i = \underline{s}_i \quad (i = 1, \dots, k) \text{ met } k < N$$

heet een partiële oplossing als het kan worden aangevuld tot een oplossing, dat wil zeggen, er bestaan zodanige getallen $\underline{s}_{k+1}, \dots, \underline{s}_N$ dat

$$(\underline{s}_1, \dots, \underline{s}_k, \underline{s}_{k+1}, \dots, \underline{s}_N)$$

een oplossing is.

Een partiële oplossing wordt ook aangegeven door

$$(\underline{s}_1, \dots, \underline{s}_k).$$

*) Hier is s_i een variabele, $\underline{s}_i$ is een getal.

Elke partiële oplossing bepaalt een deelverzameling van S , en wel de verzameling van alle oplossingen die door aanvulling van de gegeven partiële oplossing kunnen ontstaan. Dit betekent dat twee verschillende partiële oplossingen disjuncte deelverzamelingen van S bepalen.

Een splitsing van S kan dan worden verkregen door voor vaste k alle partiële oplossingen $(\underline{s}_1, \dots, \underline{s}_k)$ te bepalen. Dit geldt in het bijzonder voor $k = 1$, de splitsing van S is:

$$S = \bigcup_l S_l$$

met

$$S_l = \{(s_1, \dots, s_N) \mid s_1 = \underline{s}_1^{(l)}\}$$

waarbij $\underline{s}_1^{(l)}$ voor $l = 1, 2, \dots$ alle partiële oplossingen $(\underline{s}_1)$ doorloopt.

Een gekozen deelverzameling S_l (dus l is nu vast) kan door toevoeging van een voorwaarde verder worden gesplitst:

$$S_l = \bigcup_r S_{lr}$$

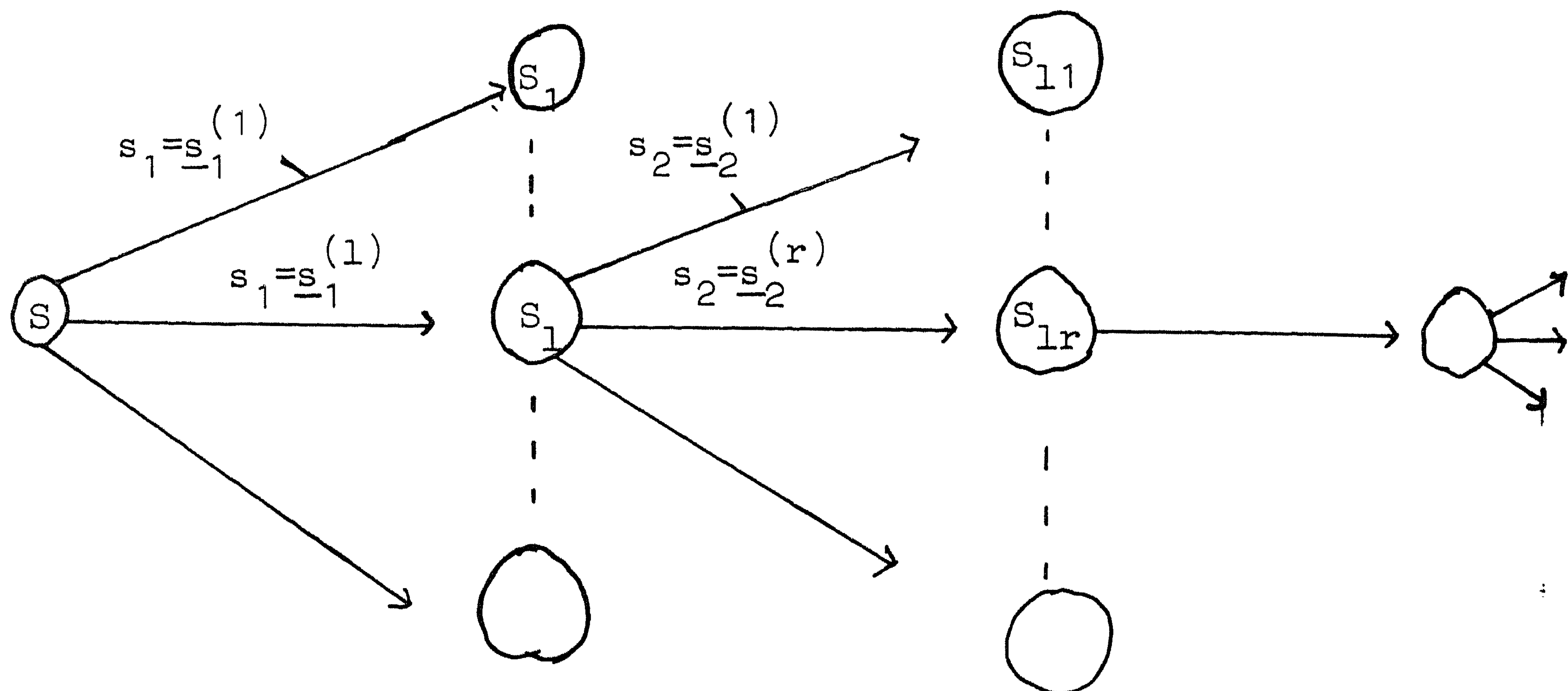
met

$$S_{lr} = \{(s_1, \dots, s_N) \mid s_1 = \underline{s}_1^{(l)}, s_2 = \underline{s}_2^{(r)}\}$$

waarbij $\underline{s}_2^{(r)}$ voor $r = 1, 2, \dots$ zodanige waarden doorloopt dat $(\underline{s}_1^{(l)}, \underline{s}_2^{(r)})$ een partiële oplossing is.

S_{lr} kan op analoge wijze verder worden gesplitst, etc.

Een en ander kan grafisch worden voorgesteld als in figuur 1.

Het splitsen van S

figuur 1.

Indelingsprobleem

Een aftel-algorithme voor het gestelde indelingsprobleem zal nu worden gegeven. Hierbij zal de terminologie van 'colleges' en 'beschikbare uren' worden gebruikt.

De in te delen verzameling X bestaat uit de n colleges.

Als voorbeeld zal een situatie met $n = 6$ en $m = 3$ worden besproken.

Tabel 5 bevat de betreffende waarden van r_{ij} .

Vanzelfsprekend is de optimale indeling onafhankelijk van de gebruikte nummering der beschikbare uren. De $m!$ indelingen die uit een gegeven indeling door hernummering kunnen worden gevormd zullen niet als verschillend worden beschouwd.

Vereenvoudigen

Het gestelde probleem n colleges optimaal te verdelen over m uren is alleen zinvol indien $1 < m < n$. Immers, is slechts 1 uur beschikbaar dan moeten alle colleges tegelijkertijd worden gegeven, zijn n of meer uren beschikbaar dan is het niet nodig dat twee colleges tijdens hetzelfde uur plaatsvinden.

	1	2	3	4	5	6
1	0	9	7	7	8	3
2	9	0	5	4	4	2
3	7	5	0	5	3	9
4	7	4	5	0	8	3
5	8	4	3	8	0	4
6	3	2	9	3	4	0

matrix (r_{ij})

tabel 5.

nummer	representatie			g_2	g_3	G_2	G_3
1	1	2	3	0	0	0	0
2	1	2	4	0	5	0	5
3	1	2	5	0	9	0	9
4	1	2	6	0	13	0	13
5	1	3	4	9	0	9	9
6	1	3	5	9	5	9	14
7	1	3	6	9	8	9	17
8	1	4	5	21	0	21	21
9	1	4	6	21	8	21	29
10	1	5	6	37	0	37	37

representaties en ondergrenzen

tabel 6.

Een indelingsprobleem heet onsamenhangend als de verzameling X van colleges kan worden gesplitst in twee niet-lege deelverzamelingen $X^{(1)}$ en $X^{(2)}$ met

$$i \in X^{(1)}, j \in X^{(2)} \implies r_{ij} = 0. \quad (4)$$

In dit geval kunnen de colleges $X^{(1)}$ en $X^{(2)}$ onafhankelijk van elkaar worden ingedeeld, zodat de oplossing van een onsamenhangend indelingsprobleem kan worden gevonden door twee indelingsproblemen, elk met een kleiner aantal colleges dan het oorspronkelijke, op te lossen.

De algoritme eist niet dat het probleem samenhangend is, uit praktische overwegingen kan het zinvol zijn de samenhangende delen van een gegeven probleem apart te beschouwen.

Men kan aantonen (zie [1]) dat een indeling met $z = 0$ bestaat als $m \geq b$ met

$$b = \max_{1 \leq i \leq n} \min(d_i + 1, i) \quad (5)$$

waarin

$$d_i = \sum_{j=1}^n (1 | r_{ij} > 0) \quad (i = 1, \dots, n)$$

en de colleges zó zijn genummerd dat

$$d_1 \geq d_2 \geq \dots \geq d_n.$$

Het is voor de algoritme niet nodig dat wordt onderzocht of aan $m \geq b$ is voldaan, de informatie dat een indeling met $z = 0$ bestaat zou echter kunnen worden gebruikt om zo'n indeling snel op te sporen.

Is een optimale indeling van n colleges over m ($m < n$) uren bekend, en wordt bij deze indeling gedurende x uren géén college gegeven, dan kunnen x colleges worden gekozen en aan de x uren toegevoegd, zó dat een indeling ontstaat waarbij elk van de beschikbare uren wordt gebruikt. De nieuwe indeling is ook optimaal, want z_h stijgt niet als een college uit P_h wordt verwijderd, en de x colleges worden elk aan een uur toegevoegd waarop nog geen college werd gegeven.

Hieruit volgt dat bij het zoeken naar een optimale indeling mag worden verondersteld dat gedurende elk van de beschikbare uren college zal worden gegeven.

Op grond van het bovenstaande geldt voor elke te onderzoeken indeling

$$P_h \neq \emptyset \quad (h = 1, \dots, m),$$

zodat gedefinieerd kan worden

$$k_h = \min_{1 \leq i \leq n} (i | i \in P_h),$$

dus k_h is de kleinste index uit de verzameling P_h . Omdat de nummering van de beschikbare uren niet voorgeschreven is mag worden verondersteld dat voor elk van de te onderzoeken indelingen geldt:

$$k_1 < k_2 < \dots < k_m,$$

zodat $k_1 = 1$ en $h \leq k_h \leq n-m+h$ ($h = 1, \dots, m$) (6)

voor elke indeling. De getallen $k_1, \dots, k_m$ heten de representanten van de indeling, m representanten vormen tezamen een representatie.

De algorithmen

De representanten van een indeling kunnen worden geïnterpreteerd als een partiële indeling van colleges, die op verschillende manieren tot een indeling van alle colleges kan worden aangevuld. Dit leidt tot een uit twee delen bestaande algorithmen:

1. het genereren van representaties,
2. het aanvullen van een gegeven representatie tot volledige indelingen.

De algorithmen zoekt naar steeds betere indelingen, d.w.z. indelingen met een lagere waarde van z dan die van alle reeds gevonden indelingen.

Hiertoe wordt een z^* ingevoerd, de algorithmen zoekt naar indelingen met $z < z^*$, is zo'n indeling gevonden dan wordt de huidige waarde van z^* vervangen door de z -waarde van die indeling. Bij het begin van de algorithmen, als nog geen indeling bepaald is, kan $z^* = +\infty$ worden gesteld.

Is aan $m \geq b$ (zie (5)) voldaan, dan kan met $z^* = 1$ worden begonnen. Voor elke partiële indeling van colleges kan een ondergrens z_0 voor de z -waarden van alle volledige indelingen die door aanvulling van de partiële indeling mogelijk zijn, worden berekend. Indien $z_0 \geq z^*$ is het niet nodig die aanvullingen uit te voeren. Omdat alle partiële indelingen met $z_0 < z^*$ moeten worden gegenereerd is het nuttig de algoritme zó in te richten dat de minimale waarde van z^* , d.w.z. een optimale indeling van alle colleges, in het algemeen na een gering aantal iteratiestappen wordt bereikt. Wegens $r_{ij} \geq 0$ kan de algoritme direct worden beëindigd zodra een indeling met $z = 0$ wordt gevonden.

Het genereren van representaties

Elk rijtje getallen $(k_1, k_2, \dots, k_m)$ dat voldoet aan (6) kan als representatie worden gebruikt. Dit betekent dat totaal $\binom{n-1}{m-1}$ representaties mogelijk zijn, want $k_1 = 1$ en $\{k_2, \dots, k_m\} \subset \{2, \dots, n\}$. Met behulp van de volgende regels worden de representaties in lexico-graphische volgorde gegenereerd (zie de in tabel 6 gegeven volgorde van representaties voor $n = 6, m = 3$):

eerste representatie: $k_1 = 1, k_2 = 2, \dots, k_m = m,$

'volgende' representatie: onderzoek voor $h = m, m - 1, m - 2, \dots$ de huidige representanten $(k_1, \dots, k_h, \dots, k_m)$, zij $h = h^*$ de eerstgevonden, d.w.z. maximale, index waarvoor geldt:

$$k_h < n - m + h.$$

Indien $h^* = 1$ is er geen volgende representatie, en de algoritme is beëindigd,

indien $h^* > 1$ wordt de volgende representatie $(k'_1, \dots, k'_m)$ gedefinieerd door

$$\left. \begin{aligned} k'_h &= k_h && \text{voor } h = 1, \dots, h^*-1; \\ k'_{h^*} &= k_{h^*} + 1 \\ k'_{h^*+i} &= k'_{h^*} + i && \text{voor } i = 1, \dots, m-h^*. \end{aligned} \right\} \quad (7)$$

Zij $(k_1, \dots, k_m)$ een gegeven representatie. Uit de definitie van k_h volgt dat elk college met een index i die voldoet aan

$$k_{h-1} < i < k_h \quad (h = 2, \dots, m)$$

niet aan een van de uren $(h, h+1, \dots, m)$ mag worden toegevoegd. Omdat dit college dus aan een van de uren $(1, 2, \dots, h-1)$ moet worden toegevoegd geeft dit i -de college een bijdrage van tenminste

$$\min_j (r_{ik_j} | j = 1, \dots, h-1)$$

tot z .

Naar aanleiding hiervan wordt voor elke representatie gedefinieerd (zie tabel 6):

$$g_2 = \sum_i \sum_j (r_{ij} | 1 \leq i < j < k_2),$$

$$g_h = \sum_i \{ \min_j (r_{ik_j} | j = 1, \dots, h-1) | k_{h-1} < i < k_h \} \\ (h = 3, 4, \dots, m)$$

en

$$G_h = g_2 + \dots + g_h \quad (h = 2, \dots, m).$$

G_h is een ondergrens voor z die slechts gebaseerd is op de representanten $k_1, \dots, k_h$. Alle representaties die deze representanten bevatten zullen bij aanvulling tot volledige indelingen uitsluitend indelingen met $z \geq G_h$ leveren. Zodat uit $G_h \geq z^*$ volgt dat alle representaties die de representanten $k_1, \dots, k_h$ bevatten mogen worden overgeslagen.

Door de lexicografische ordening van de representaties is dit eenvoudig te realiseren met een aanvulling op de voorschriften voor het bepalen van de 'volgende' representatie.

Zij $(k_1, \dots, k_m)$ een reeds bepaalde representatie.

Bereken $G_2, G_3, \dots, G_m$.

Indien $G_m < z^*$ is $G_h < z^*$ ($h = 2, \dots, m$) en mag deze representatie niet worden overgeslagen. Is echter $G_m \geq z^*$ dan bestaat een kleinste index $h = h'$ waarvoor geldt $G_h \geq z^*$. Als $h' = 1$ is de algorithmme beëindigd.

Is $k_{h'} < n - m + h'$ en $h' > 1$ kunnen de regels (7) worden toegepast met $h^* = h'$.

Is $k_{h'} = n - m + h'$ en $h' > 1$ dan bestaat een maximale index $h = h^*$ met $h < h'$ en $k_h < n - m + h$. De algorithmme is beëindigd als $h^* = 1$, als $h^* > 1$ kan (7) worden toegepast.

Is een nieuwe representatie $(k'_1, \dots, k'_m)$ gevonden dan kan met behulp van de bijbehorende $G'_2, \dots, G'_m$ worden onderzocht of deze wellicht ook moet worden overgeslagen.

Nummering van de colleges

De colleges kunnen zó worden genummerd dat

$$r_{12} \geq r_{ij} \quad (1 \leq i, j \leq n)$$

en

$$\left. \begin{aligned} \min_j (r_{ij} | j < i) &\geq \max_k \left\{ \min_j (r_{kj} | j < i) | k > i \right\} \\ \text{voor } i &= 3, \dots, n-1. \end{aligned} \right\} \quad (8)$$

Geldt in (8) het gelijktteken dan kan bijvoorbeeld aan het college met de grootste $\sum_{j=1}^n r_{ij}$ het laagste beschikbare nummer worden gegeven.

Relatie (8) kan ook worden geformuleerd als:

de minimale bijdrage tot z die ontstaat door het i -de en een van de colleges $1, 2, \dots, i-1$ tijdens hetzelfde uur te geven is niet kleiner dan de minimale bijdrage die ontstaat door een van de colleges $i+1, \dots, n$ gelijktijdig met een van de colleges $1, 2, \dots, i-1$ te geven.

Dit betekent dat de colleges met onderling relatief grote waarden voor r_{ij} in het algemeen lage indices zullen krijgen. Door de gekozen volgorde van de representaties zullen deze colleges tijdens de eerste iteratiestappen aan verschillende uren zijn toegevoegd, zodat verwacht mag worden dat een optimale indeling spoedig wordt gevonden.

Bij overgang van representatie $(k_1, \dots, k_{h^*}, \dots, k_m)$ op representatie $(k_1, \dots, k_{h^*-1}, k_{h^*+1}, k_{h^*+2}, \dots, k'_m)$ blijven $g_2, \dots, g_{h^*-1}$ ongewijzigd.

Het verschil tussen de nieuwe g_{h^*} en de oude g_{h^*} is

$$v = \min_j (r_{jk_{h^*}} | j = 1, \dots, h^*-1).$$

Omdat $v \geq \min_j (r_{jk_{h^*}} | j = 1, \dots, h^*-1) = \min_j (r_{k_{h^*}j} | j < k_{h^*})$ geldt op grond van (8) dat verwacht mag worden dat v in het algemeen groot zal zijn.

Hierdoor wordt een snelle toename van G_h bevorderd zo dat in de loop van de berekeningen veelal steeds grotere groepen representaties zullen worden overgeslagen.

Aanvulling tot volledige indeling

Voor een gegeven representatie kan het volgende tableau worden opgesteld:

		l_1 — — — — — l_i — — — — — l_{n-m}
k_1	0	$\vdots$
$\vdots$	$\vdots$	
k_h	0	$\vdots$
$\vdots$	$\vdots$	a_{hi}
k_m	0	

uitgangstableau

Hierbij vormen $(k_1, \dots, k_m)$ de representatie en doorloopt l_i ($i=1, \dots, n-m$) de indices die niet in de representatie voorkomen. De elementen van het tableau zijn gedefinieerd door:

$$a_{hi} = \begin{cases} r_{k_h l_i} & \text{als } l_i > k_h \\ +\infty & \text{anders.} \end{cases} \quad (9)$$

Bovenstaand uitgangstableau zal worden getransformeerd tot tableaux voor partiële indelingen:

		l_1 l_i l_{n-m}
P_1	z_1	
$\vdots$	$\vdots$	$\vdots$
P_h	z_h	a_{hi}
$\vdots$	$\vdots$	
P_m	z_m	

algemeen tableau

De $P_1, \dots, P_m$ bepalen een partiële indeling met $k_h \in P_h$ ($h = 1, \dots, m$), het uitgangstableau is een speciaal geval ($P_h = \{k_h\}$ ($h = 1, \dots, m$)) van het algemene tableau. De elementen van het algemene tableau worden gedefinieerd door:

$$z_h = \sum_{i < j} (r_{ij} | i, j \in P_h) \quad (h = 1, \dots, m)$$

$$a_{hi} = \begin{cases} \sum_j (r_{l_i j} | j \in P_h) & \text{als } l_i \text{ aan } P_h \text{ mag worden toegevoegd,} \\ +\infty & \text{als } l_i \text{ niet aan } P_h \text{ mag worden toegevoegd.} \end{cases} \quad (10)$$

De equivalentie van (9) met (10) voor het uitgangstableau is eenvoudig na te gaan.

De z-waarde van de partiële indeling is dus $z = \sum_{h=1}^m z_h$, a_{hi} geeft aan met welk bedrag z_h , dus ook z , stijgt als l_i aan P_h wordt toegevoegd.

Zij $a_i = \min_h (a_{hi} | h = 1, \dots, m)$, dan leidt aanvulling van de partiële indeling tot een volledige indeling waarvan de z-waarde tenminste

$$z_0 = \sum_{h=1}^m z_h + \sum_i (a_i | l_i \text{ nog niet ingedeeld})$$

bedraagt.

Is $z_0 \geq z^*$ dan behoeft de partiële indeling niet te worden aangevuld.

Toevoeging van het nog niet ingedeelde college l_i aan P_h leidt tot een nieuwe partiële indeling waarvan het tableau wordt gegeven door:

		l_1 --- --- --- l_i --- --- l_j --- --- --- l_{n-m}
P_1 $P_h \cup \{l_i\}$ P_m	z_1 $z_h + a_{hi}$ z_m	$+\infty$ $a_{h1} + r_{l_1 l_i}$ --- $+\infty$ --- $a_{hj} + r_{l_j l_i}$ --- $a_{h,n-m} + r_{l_{n-m} l_i}$ $+\infty$

De rekenregels luiden:

breidt P_h uit met l_i ,

tel a_{hi} op bij z_h ,

tel $r_{l_j l_i}$ op bij a_{hj} voor $j = 1, \dots, n-m$,

vervang a_{ki} door $+\infty$ voor $k = 1, \dots, m$.

De keuze van de uitbreiding van een partiële indeling wordt als volgt gemaakt.

Zij h_i gedefinieerd door:

$$a_{h_i i} = \min_h (a_{hi} | h = 1, \dots, m),$$

zij h'_i gedefinieerd door:

$$a_{h'_i i} = \min_h (a_{hi} | h = 1, \dots, m; h \neq h_i),$$

dus $a_{h_i i}$ en $a_{h'_i i}$ zijn het kleinste respectievelijk het op een na kleinste element uit de i -de kolom van het tableau.

Definieer

$$d_i = a_{h'_i i} - a_{h_i i} \quad (i = 1, \dots, n-m | l_i \text{ nog niet ingedeeld}),$$

zij i^* bepaald door:

$$d_{i^*} = \max_i d_i,$$

en definieer $h^* = h_{i^*}$.

De uit te voeren uitbreiding is dan:

voeg l_{i^*} toe aan P_h^* .

Vanuit het uitgangstableau wordt met deze regels hetzij in $n-m$ stappen een volledige indeling met $z < z^*$ gevonden, en z^* kan worden verlaagd, of een partiële indeling met $z_0 \geq z^*$ bereikt. In het eerste geval wordt de gevonden volledige indeling genoteerd.

In beide gevallen wordt het tableau terug-getransformeerd. Is de toevoeging van l_{i^*} aan P_h^* de laatst uitgevoerde uitbreiding dan bestaat deze transformatie uit het ongedaan maken van die toevoeging en het vervangen van $a_{h i^*}$ door $+\infty$.

Op grond van het nieuwe tableau wordt een nieuwe benedengrens z_0 berekend. Als $z_0 < z^*$ kan l_{i^*} worden toegevoegd aan P_h^{**} , waarbij h^{**} wordt bepaald door:

$$a_{h^{**} i^*} = \min_h (a_{hi^*} | h = 1, \dots, m).$$

Wegens $a_{h i^*} = +\infty$ is de nieuwe partiële indeling ongelijk aan de partiële indeling waarvan het tableau werd terug-getransformeerd. Nu kunnen de voor het uitbreiden gegeven regels weer worden toegepast.

Als $z_0 \geq z^*$ leidt geen enkele aanvulling tot een volledige indeling met $z < z^*$. In dit geval wordt voor de indices h met $a_{hi^*} = +\infty$ en $k_h < l_{i^*}$

a_{hi^*} vervangen door $\sum_{i=j} (r_{ij} | i, j \in P_h)$,

zodat het tableau weer gelijk is aan het tableau op grond waarvan l_{i^*} aan P_h^* werd toegevoegd. Nu wordt nogmaals als boven omschreven terug-getransformeerd en onderzocht of andere dan de reeds uitgevoerde uitbreidingen tot een volledige indeling met $z < z^*$ leiden.

Is terugtransformeren onmogelijk, omdat $P_h = \{k_h\}$ ($h = 1, \dots, m$), dan zijn alle aanvullingen van de representatie expliciet of impliciet onderzocht, zodat de volgende representatie en het volgende uitgangstableau kunnen worden gegenereerd.

Voorbeeld

Met behulp van de in tabel 5 gegeven getallen r_{ij} voor 6 colleges zal de optimale indeling over 3 uren worden berekend. In de tableaux zullen de plaatsen waarop $+\infty$ behoort te staan blanco worden gelaten. De tableaux bevatten een extra regel waarin z en de getallen d_i staan vermeld. De gekozen uitbreiding wordt aangegeven door een $*$ in het tableau.

		$l_1=4$	$l_2=5$	$l_3=6$
$k_1 = 1$	$z_1 = 0$	7	8	3
$k_2 = 2$	$z_2 = 0$	4^*	4	2
$k_3 = 3$	$z_3 = 0$	5	3	9
	$z = 0$	$d_1=1$	$d_2=1$	$d_3=1$

$z_0 = 9$

tableau 1

		4	5	6
$P_1=\{1\}$	$z_1=0$		8	3
$P_2=\{2,4\}$	$z_2=4$		12	5
$P_3=\{3\}$	$z_3=0$		3^*	9
	$z = 4$		$d_2=5$	$d_3=2$

$z_0 = 10$

tableau 2

		4	5	6
1	0			3^*
2,4	4			5
3,5	3			13
	7			2

$z_0 = 10$

tableau 3

		4	5	6
1,6	3			
2,4	4			
3,5	3			
	10			

tableau 4

		4	5	6
1	0			
2,4	4			5
3,5	3			13
	7			8

$$z_0 = 12$$

tableau 5

		4	5	6
1	0		8	3
2,4	4		12	5
3	0			9
	4		4	2

$$z_0 = 15$$

tableau 6

		4	5	6
1	0	7	8	3
2	0		4	2
3	0	5	3	9
	0	2	1	1

$$z_0 = 10$$

tableau 7

		3	5	6
1	0	7	8	3
2	0	5	4	2
4	0		8	3
	0	2	4	1

$$z_0 = 11$$

tableau 8

		3	4	6
1	0	7	7	3
2	0	5	4	2
5	0			4
	0	2	3	1

$$z_0 = 11$$

tableau 9

		2	5	6
1	0	9	8	3
3	0		3	9
4	0			3
	0	∞	5	0

$$z_0 = 15$$

tableau 10

Tableau 1 is het uitgangstableau dat behoort bij de representatie (1,2,3). Door toevoeging van het n-de college aan het 2-de uur ontstaat tableau 2. Toevoeging van het 5-de college aan het 3-de uur geeft tableau 3. Door het overgebleven 6-de college in te delen ontstaat tableau 4, een volledige indeling met $z = 10$ is gevonden, dus nu wordt $z^* = 10$ gesteld. Terugtransformeren geeft tableau 5 (vergelijk met tableau 3), wegens $z_0 > z^*$ wordt

$a_{13} = \infty$ vervangen door $a_{13} = 3$ en nogmaals teruggetransformeerd. Het gevonden tableau 6 (vergelijk tableau 2) geeft weer $z_0 > z^*$, dus terugtransformeren tot tableau 7 (vergelijk tableau 1). Nog is $z_0 \geq z^*$, verder terugtransformeren is onmogelijk, de volgende representatie (zie tabel 6) is (1,2,4).

Tableau 8 is het bij (1,2,4) behorende uitgangstableau. Wegens $3 = l_1 < k_3 = 4$ is $a_{31} = +\infty$. Omdat $z_0 > z^*$ behoeft de representatie niet te worden aangevuld.

Tableau 9 is het uitgangstableau van representatie (1,2,5), ook deze representatie behoeft niet verder te worden onderzocht.

Representatie (1,2,6) kan direct worden overgeslagen wegens $G_3 > z^*$ (zie tabel 6), de volgende in aanmerking komende representatie is (1,3,4). Het bijbehorende uitgangstableau (tableau 10) geeft $z_0 > z^*$, dus representatie (1,3,5) wordt gegenereerd. Nu is $G_3 > z^*$, zodat representatie (1,3,6) wordt gegenereerd. Weer is $G_3 > z^*$, de volgende representatie geeft $G_2 > z^*$ en de algorithmen is beëindigd.

Kleurgetal

Voor het probleem: bepaal de minimale waarde van m waarvoor een indeling met $z = 0$ bestaat, mogen de $r_{ij} > 0$ vervangen worden door $r_{ij} = 1$. Wordt nu, voor gegeven m , de bovenbeschreven algorithmen uitgevoerd met $z^* = 1$ dan zal het resultaat zijn dat een indeling met $z = 0$ wordt gevonden, of de algorithmen wordt beëindigd zonder dat een indeling is gevonden. In het laatste geval bestaat, voor de gegeven m , geen indeling met $z = 0$.

Onder de vereenvoudigingen is een bovengrens b voor de gezochte waarde m^* van m gegeven, zodat $1 \leq m^* \leq b$. Met de definitie $x \underline{\cdot} y =$ het grootste gehele getal niet groter dan $\frac{x}{y}$ leiden de volgende rekenregels tot het gewenste resultaat:

1. Stel $low = 0$ en $up = b + 1$;
2. stel $m = low + (up - low) \underline{\cdot} 2$;
3. zoek, met $z^* = 1$ en de in 2. berekende m , een optimale indeling, indeling gevonden, verder bij 4., indeling bestaat niet, verder bij 5.,

4. stel $up = m$, verder bij 6.;
5. stel $low = m$;
6. als $up - low \leq 1$ verder bij 7., anders verder bij 2.;
7. $m^* = up$.

Deze procedure werkt snel omdat bij vaste m de eerstgevonden indeling optimaal is. Verder is z^* klein, dus als geen indeling met $z = 0$ bestaat behoeven slechts weinig representaties te worden gegenereerd om dit te detecteren.

Literatuur

1. D.J.A. Welsh, M.B. Powell,
An upper bound for the chromatic number of a graph and its application to time tabling problems.
The Computer Journal 10 (1967) 85-86.
2. K. Kirchgässner,
Die graphentheoretische Lösung eines nicht linearen Zuteilungsproblem.
Unternehmensforschung 9 (1965) 217-229.