

# Facility layout problemen met vaste oppervlaktes en variabele vorming

**Citation for published version (APA):**

Schuur, P. C. (1997). *Facility layout problemen met vaste oppervlaktes en variabele vorming*. (BETA publicatie : working papers; Vol. 12), (BETA publicatie : preprints; Vol. 17). Universiteit Twente.

**Document status and date:**

Gepubliceerd: 01/01/1997

**Document Version:**

Uitgevers PDF, ook bekend als Version of Record

**Please check the document version of this publication:**

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

**General rights**

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

[www.tue.nl/taverne](http://www.tue.nl/taverne)

**Take down policy**

If you believe that this document breaches copyright please contact us at:

[openaccess@tue.nl](mailto:openaccess@tue.nl)

providing details and we will investigate your claim.

**Facility layout problemen met vaste  
oppervlaktes en variabele vorming**

**P.C. Schuur**

~~PR-17~~

WF-12

BETA-publicatie : PR-17  
ISSN : 1386-9213; PR-17  
NUGI : 684  
Enschede : Oktober 1997  
Keywords : Facility layout, simulated annealing  
BETA-Research Programme : Operations Management of Supply/Distribution  
Processes  
Te publiceren in : ITW-nieuws

# Facility layout problemen met vaste oppervlaktes en variabele vormen

door: Peter Schuur  
Faculteit Technologie en Management,  
Universiteit Twente, Postbus 217, 7500 AE Enschede

Bij een *facility layout probleem (FLP)* gaat het erom een groep afdelingen, machines, o.i.d. op zo'n manier te situeren binnen een faciliteit (gebouw, fabriek, etc.) dat een door de gebruiker gedefinieerde doelfunctie wordt geoptimaliseerd. In een productie-omgeving bestaat de doelstelling doorgaans uit het minimaliseren van de material handling kosten, in het bijzonder de totale afstand waarover materiaal moet worden verplaatst. Aan de te realiseren layout worden bovendien allerlei meer kwalitatieve ontwerpbeperkingen opgelegd ingegeven door veiligheidseisen, lawaaioverlast, milieu-eisen, ergonomische eisen, etc.

In dit onderzoek beperken we ons tot de kwantitatieve aspecten: onze FLP-doelstelling is het bepalen van een layout van een groep afdelingen binnen een faciliteit zodanig dat over een gegeven periode de totale afstand afgelegd door personeel, producten, apparatuur, etc. minimaal is. Hierbij nemen we aan dat per periode het gemiddeld aantal trips tussen de verschillende afdelingen bekend is. Verder gaan we ervan uit dat alle te plaatsen afdelingen een *voorgeschreven oppervlakte* hebben, maar dat hun *vorm* nog *variabel* is. Om de discussie niet onnodig te compliceren nemen we verder aan dat alle te plaatsen afdelingen op dezelfde verdieping komen te liggen en dat aan deze verdieping geen ruimtebeperkingen worden opgelegd. Psychologisch komt dit overeen met het ontwerpen van een bungalow in een uitgestrekt weiland. Vandaar dat in de literatuur deze situatie wordt aangeduid met de term *greenfield*.

Relevante discretisaties van het FLP zijn  $\mathcal{NP}$ -lastig, wat zoveel betekent als dat je er niet op mag hopen voor een redelijk groot probleem binnen aanvaardbare rekentijd een exacte oplossing te vinden, hoe slim je algoritme ook is. Wel zijn er veel algoritmen bedacht die redelijk snel een bijna-optimale oplossing produceren. De meest recente van deze benaderingsalgoritmen zijn gebaseerd op simulated annealing en tabu search. Jammer genoeg representeren veel layout algoritmen afdelingen op een nogal simpele manier, meestal als cellen, d.w.z. vierkanten met hetzelfde oppervlak. Een gevolg is dat in een praktijk-toepassing de afmetingen en de precieze posities van de te plaatsen afdelingen nog moeten worden uitgedokterd door de gebruiker op basis van een of andere cellulaire programma output.

In het heuristische algoritme dat we hieronder presenteren wordt deze tekortkoming voor een groot deel verholpen. We staan afdelingen toe van verschillende oppervlakte en willekeurige vorm. Elke afdeling wordt gerepresenteerd als een groep met elkaar verbonden cellen. Het vaste aantal cellen dat we in deze representatie gebruiken is een maat voor de voorgeschreven oppervlakte van de afdeling. Gedurende ons optimaliseringsproces, dat gebaseerd is op simulated annealing, veranderen de vormen van de afdelingen voortdurend. De uiteindelijke bijna-optimale layout die door ons algoritme geproduceerd wordt geeft de gebruiker niet alleen een blueprint voor de situering van de afdelingen, maar ook een voorstel voor de vorm van de afdelingen.

## Simulated annealing

Een stochastisch lokaal zoekalgoritme dat in staat is te ontsnappen uit lokale optima en op een elegante manier om te gaan met niet-toegelaten oplossingen is het simulated annealing algoritme. In feite is simulated annealing een flexibele, heuristische optimaliseringstechniek gebaseerd op de analogie tussen het simuleren van het annealing proces uit de vaste stof fysica en het oplossen van grootschalige combinatorische optimaliseringsproblemen.

In zijn gebruikelijke vorm, wanneer het zoekproces beperkt blijft tot toegelaten oplossingen, kan simulated annealing als volgt worden samengevat. Het algoritme start vanuit een willekeurige beginconfiguratie. In elke iteratie wordt, door de huidige configuratie licht te perturberen, een nieuwe configuratie gegenereerd. Het verschil in doelfunctiewaarde wordt vergeleken met een acceptatiecriterium dat alle verbeteringen zonder meer accepteert, maar ook, in beperkte mate, verslechtingen in doelfunctiewaarde toestaat.

Het optimaliseringsproces wordt gestuurd door een zogenaamde koelingparameter  $c > 0$ , die een grote invloed heeft op het acceptatiecriterium. Deze koelingparameter daalt langzaam naar nul gedurende het proces. In het begin, voor hoge waarden van  $c$ , wordt het acceptatiecriterium zo gekozen dat verslechtingen met een grote kans geaccepteerd worden. Naarmate het optimaliseringsproces voortschrijdt, wordt het acceptatiecriterium zodanig aangepast, dat de kans om verslechtingen te accepteren afneemt. Aan het eind van het proces loopt deze kans naar nul. Op deze manier krijgt het optimaliseringsproces de kans om te ontsnappen uit lokale optima. Het proces komt tot stilstand wanneer gedurende een voorgeschreven aantal iteraties geen verdere verbetering optreedt in de tot dan toe beste doelfunctiewaarde. Als de koelingparameter voorzichtig genoeg naar nul gebracht wordt, eindigt het proces in een bijna-optimale oplossing.

Voor veel combinatorische optimaliseringsproblemen geldt, dat een lokale zoekprocedure flexibeler kan worden gemaakt door oplossingen op te nemen die sommige van de probleemrestricties schenden. Om de slechte eigenschappen van een bepaalde niet-toegelaten oplossing tot uitdrukking te laten komen, wordt de doelfunctie aangevuld met een boete-term die de mate van schending van de restricties afstraft. Het simulated annealing algoritme biedt de mogelijkheid om de schending van de restricties op een dynamische manier af te straffen door de boete-term te laten afhangen van de koelingparameter  $c$ . Naarmate  $c$  afneemt, neemt de boete geassocieerd met een vaste niet-toegelaten oplossing geleidelijk aan toe. Op deze manier wordt uitgedrukt dat tegen het einde van het optimaliseringsproces een groeiende hekel aan niet-toegelaten oplossingen ontstaat. We merken nog op dat een simulated annealing aanpak alleen dan succesvol kan zijn als het verschil in doelfunctiewaarde tussen de huidige oplossing en de kandidaat-oplossing berekend kan worden op een simpele, bij voorkeur incrementele wijze.

## Formulering van het facility layout probleem

Om te beginnen presenteren we een formulering van het FLP waarmee gemakkelijk te werken valt in de setting van een heuristisch zoekalgoritme. Deze formulering kan worden opgevat als een cellulaire benadering van het werkelijke layout probleem. De details zijn als volgt.

Beschouw een facility layout probleem met  $D$  afdelingen. Laat  $A_i$  de vereiste oppervlakte voorstellen van afdeling  $i$  in de layout. Schrijf verder  $f_{ij}$  voor het gemiddeld aantal trips vanaf afdeling  $i$  naar afdeling  $j$ . Allereerst bepalen we een *grootste gemene deler* voor de afdelingsoppervlaktes, zodanig dat iedere afdelingsoppervlakte  $A_i$  bij benadering gelijk is aan een veelvoud  $n_i$  van deze g.g.d.-oppervlakte. Iedere afdeling  $i$  representeren we voortaan als een groep van  $n_i$  eenheidscellen. Zij  $N = \sum_{i=1}^D n_i$  het totaal aantal afdelingscellen. Voorzie nu het plaatsingsoppervlak van een gridstructuur opgebouwd uit potentiële afdelingscellen (zie Figuur 1). Schrijf  $(m, n)$ , met  $m$  en  $n$  geheeltallig, voor de cellen geassocieerd met het grid.

Een *plaatsing* is een toewijzing van de  $N$  afdelingscellen aan gridcellen. In een plaatsing mogen de afdelingscellen elkaar overlappen. Verder kunnen cellen behorend bij dezelfde afdeling los van elkaar liggen. Een *toegelaten layout* is een plaatsing waarin de toegewezen cellen elkaar niet overlappen, terwijl bovendien ieder tweetal cellen behorend bij dezelfde afdeling met elkaar verbonden is via een rectilineair pad van eigen afdelingscellen. Dit komt overeen met de huis-tuin-en-keuken situatie dat je om iets uit een hoek van je eigen kamer te pakken de kamer niet hoeft te verlaten.

Om de afstand tussen twee afdelingen in een plaatsing  $p$  te meten gaan we als volgt te werk. Voor elke afdeling  $i$  kiezen we een vaste afdelingscel  $e_i$  als ingangs- en uitgangscel. Laat  $p(e_i) = (m(e_i), n(e_i))$  de gridcel zijn waaraan de ingangscel  $e_i$  is toegewezen. Als de afstand  $d_{ij}(p)$  tussen twee afdelingen  $i$  en  $j$  nemen we de rectilineaire afstand tussen de middelpunten van de gridcellen waaraan de twee corresponderende ingangscellen zijn toegewezen, d.w.z.

$$d_{ij}(p) = |m(e_i) - m(e_j)| + |n(e_i) - n(e_j)|.$$

Ons facility layout probleem kunnen we nu als volgt formuleren:

Vind een toegelaten layout  $p$  die de volgende uitdrukking minimaliseert:

$$T(p) = \sum_{i,j=1}^D f_{ij} d_{ij}(p).$$

## Implementatie van het annealing algoritme

Laten we nu aangeven hoe we het bovenbeschreven annealing algoritme toepassen op het FLP. Essentieel voor de implementatie van het simulated annealing algoritme is de specificatie van een configuratieruimte, een te minimaliseren doelfunctie en een mechanisme om de huidige configuratie te modificeren. Verder zijn van belang de keuze van het acceptatiecriterium en het koelingsschema, waarvoor we verwijzen naar de literatuur.

### Configuraties en doelfunctie

Als configuraties kiezen we plaatsingen. Zoals eerder opgemerkt zijn dit willekeurige allocaties van afdelingscellen aan gridcellen. Hieronder zullen we een doelfunctie definiëren op zo'n manier dat de kwaliteit van iedere configuratie adequaat gemeten wordt. Dit betekent

ondermeer dat we storende zaken als overlap en onsamenshangendheid gaan afstraffen. Voor onze doelfunctie zal gelden: hoe lager de functiewaarde hoe beter de configuratie.

Laat een bepaalde plaatsing  $p$  gegeven zijn. Gewoonlijk wordt in situaties als deze de totale *overlap* als volgt bepaald: Bereken voor alle  $N(N-1)/2$  paren afdelingscellen de oppervlakte van de doorsnijding en sommeer daarna. Wij stellen een nieuwe definitie van overlap voor, die de overlap registreert zoals deze ondervonden wordt door een individuele cel. Ons basisidee is om de informatie betreffende overlap op te slaan in de gridcellen, waarbij we voor het gemak deze gridcellen maar even opvatten als de cellen van het werkblad van een spreadsheet. Op deze manier besparen we een aanzienlijke hoeveelheid rekentijd.

Met dit doel voor ogen definiëren we een werkblad-matrix  $W$  die de overlap-informatie opslaat. Voor iedere gridcel  $(m, n)$  geven we het totale aantal op deze cel geplaatste afdelingscellen aan met  $W(m, n)$ . Nu definiëren we de overlap  $O_{mn}(p)$  van een gridcel  $(m, n)$  als de graad van extra overdekking, d.w.z. als het totale aantal afdelingscellen geplaatst op cel  $(m, n)$  geteld vanaf 2, in formule

$$O_{mn}(p) = \max(0, W(m, n) - 1).$$

De totale overlap geassocieerd met de plaatsing  $p$  wordt nu gedefinieerd door

$$O(p) = \sum_{(m,n)} O_{mn}(p).$$

De totale *onsamenhangendheid* wordt als volgt bepaald: Associeer iedere afdeling  $i$  met een graaf  $G_i(p)$ , zo dat de knopen van  $G_i(p)$  de afdelingscellen zijn en twee knopen een kant delen dan en slechts dan als de rectilineaire afstand tussen de gridcellen waar ze op zijn geplaatst ten hoogste 1 is. Zij  $C_i(p)$  het aantal componenten van graaf  $G_i(p)$ . Dan wordt de totale onsamenshangendheid gegeven door

$$D(p) = \sum_i (C_i(p) - 1).$$

Voor de uiteindelijke doelfunctie kiezen we nu:

$$f_c(p) = T(p) + \gamma_1 A(p) + \gamma_2 \frac{O(p)}{c} + \gamma_3 \frac{D(p)}{c},$$

met  $T(p)$  de - eerder gedefinieerde - totale afstand,  $A(p)$  de oppervlakte van de omhullende rechthoek,  $O(p)$  de totale overlap,  $D(p)$  de totale onsamenshangendheid,  $\gamma_j$  schaalfactoren en  $c$  de koelingparameter. De constante  $\gamma_1$  wordt voldoende klein gekozen. De reden om de term  $\gamma_1 A(p)$  op te nemen in de doelfunctie is tweeledig: Aan de ene kant zorgt deze term ervoor dat van elke twee toegelaten layouts  $p_1$  en  $p_2$  met  $T(p_1) = T(p_2)$  die met de kleinste rechthoeksoppervlakte wordt gekozen. Aan de andere kant stuurt de term het annealing proces in de goede richting door te verhinderen dat afdelingscellen op drift raken.

De constanten  $\gamma_2$  en  $\gamma_3$  worden op experimentele wijze bepaald zodanig dat de afdelingscellen gedurende een substantieel deel van het optimaliseringsproces mogen overlappen en los van elkaar mogen liggen zonder daarvoor al te zeer gestraft te worden. Deze keuze voor de doelfunctie brengt met zich mee, dat overlap en onsamenshangendheid steeds strenger gestraft

worden naarmate  $c$  kleiner wordt. Wanneer  $c$  uiteindelijk bijna nul is, zullen alle geaccepteerde configuraties corresponderen met toegelaten layouts.

Om de hierboven geïntroduceerde concepten te illustreren beschouwen we de plaatsing van de twee afdelingen  $A$  en  $B$  weergegeven in Figuur 1. Voor iedere gridcel  $(m, n)$  waaraan een of meer afdelingscellen zijn toegewezen, is de waarde van de werkblad-matrix  $W(m, n)$  aangegeven in de linkerbovenhoek. De twee ingangscellen zijn herkenbaar aan een asterisk. In dit voorbeeld zijn er  $N = 9$  afdelingscellen, waarvan er twee paren overlappen, zodat  $O(p) = 2$ . Afdeling  $B$  is samenhangend, maar  $A$  bestaat uit drie componenten. Vandaar dat de totale onsamenhangendheid gelijk is aan  $D(p) = 2$ . De afstand tussen de twee afdelingen bedraagt  $d_{AB}(p) = 4$ .

	1 *	2	1	
	A	A B	B	
	1		2	
	A		A B	
		1	1 *	
		A	B	

FIGUUR 1. Plaatsing van twee afdelingen.

## Een modificatie mechanisme

Om vanuit de huidige configuratie een kandidaat-configuratie te genereren gebruiken we simpele lokale verstoringen zoals horizontale of verticale translaties van afdelingscellen over een aantal celeenheden en verwisselingen van twee afdelingscellen. Af en toe worden blokken bestaande uit met elkaar verbonden afdelingscellen getransleerd of verwisseld. De frequentie van optreden van deze verschillende types verstoringen laten we afhangen van de koelingparameter. In het begin van het proces gebruiken we vaak blokoperaties en celtranslaties over verscheidene celeenheden. Naarmate het optimaliseringsproces vordert, laten we de hoeveelheid blokoperaties afnemen ten gunste van de hoeveelheid celoperaties. Ook transleren we minder ver. Tegen het eind van het proces, gebruiken we alleen nog maar celverwisselingen en celtranslaties over één celeenheid.

Of we een kandidaat-configuratie al dan niet accepteren hangt af van de waarde van de koelingparameter, maar ook van  $\Delta f_c$ , de kosten van de kandidaat-configuratie minus die van de huidige. Duidelijk is dat

$$\Delta f_c = \Delta T + \gamma_1 \Delta A + \gamma_2 \frac{\Delta O}{c} + \gamma_3 \frac{\Delta D}{c}.$$

Hierin wordt  $\Delta T$ , het verschil in totale afstand, eenvoudig incrementeel berekend. Verder wordt  $\Delta A$ , het verschil in rechthoeksoppervlakte, bepaald door een (snelle) directe zoekmethode. Zoals eerder vermeld wordt de overlap-informatie opgeslagen in de werkblad-matrix  $W$ . Een rechtstreeks gevolg van het opslaan van deze informatie is, dat de evaluatie van  $\Delta O$ , het verschil in totale overlap, bijzonder eenvoudig wordt: Stel bijvoorbeeld dat we het verschil in totale overlap willen berekenen ontstaan door het verplaatsen van een afdelingscel van gridcel  $(m, n)$  naar gridcel  $(m', n')$ . Dan vinden we voor het verschil in totale overlap de volgende incrementele uitdrukking

$$\Delta O = \max(0, W(m', n')) + \max(0, W(m, n) - 2) - O_{m', n'} - O_{mn}.$$

Het bepalen van  $\Delta D$  is het meest tijdrovend. We verkrijgen deze waarde door de onsamenhangendheid van de kandidaat-afdelingsgraaf te berekenen via een standaard graaftheoretisch algoritme.

## Performance van het algoritme

Je kunt je afvragen of voor een stochastisch algoritme met zoveel vrijheidsgraden als dit, het concept rekentijd zinvol is in te vullen. In de praktijk worden voor elke nieuwe probleeminstantie alle regelknoppen opnieuw ingesteld en na een aantal proefruns net zolang gewijzigd als nodig is om het algoritme lekker te laten lopen. Laten we niettemin een ruwe indicatie geven voor het tijdsgedrag. Hiertoe hebben we het algoritme getest op een reeks random probleeminstanties, in grootte variërend van 30 tot 50 afdelingscellen. Voor elk van deze testproblemen draaiden we 30 simultane annealing runs. Het programma is zo ingericht dat de meestbelovende runs de meeste aandacht krijgen. Gemiddeld genomen verkregen we - op een bescheiden vierachtzesje - toegelaten layouts van hoge kwaliteit in ongeveer een half uur rekentijd.

Laten we de praktische mogelijkheden van het algoritme illustreren met de volgende bescheiden case. Een bedrijfskundefaculteit hier te lande besluit de vakgroep Operations Research onder te brengen in een - nieuw te bouwen - luxe bungalow. De zes onderzoekers plus hun bibliotheek dienen te worden gehuisvest in verschillende ruimtes op zo'n manier dat de totale dagelijkse loopafstand minimaal is. Hieronder geven we de verschillende afdelingen en hun vereiste oppervlakte (in eenheden van 1 AIO-kamer):

Afdelingscode	Functie	Vereiste oppervlakte
1	AIO	1
2	AIO	1
3	UD	2
4	UD	2
5	UHD	3
6	Professor	5
7	Bibliotheek	10

TABEL 1. Vereiste oppervlaktes voor de afdelingen.



De volgende tabel geeft in rij  $i$  en kolom  $j$  het aantal trips per week van afdeling  $i$  naar afdeling  $j$ :

	1	2	3	4	5	6	7
1	0	29	20	1	21	6	6
2	35	0	15	3	20	5	3
3	20	18	0	1	40	15	1
4	1	1	0	0	5	1	70
5	25	15	45	25	0	20	1
6	1	1	10	0	15	0	0
7	0	0	0	0	0	0	0

TABEL 2. Interactiematrix.

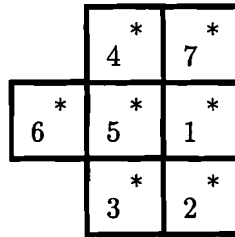
Het probleem is reeds zodanig gemodelleerd dat ons algoritme dadelijk aan de slag kan. De toegelaten layout die we uiteindelijk vinden is weergegeven in Figuur 2. Hierin duidt een kruisje op een lege cel. Zoals eerder markeert een asterisk een ingangscel.

X	4	4 *	7 *	7
5	5	5 *	1 *	7
6	6 *	3 *	2 *	7
6	6	3	7	7
6	7	7	7	7

FIGUUR 2. De beste layout die we konden vinden.

De totale afstand behorend bij deze layout bedraagt  $T(p) = 648$ . Zoals valt te verwachten, liggen de ingangscellen dicht bij elkaar. Verder worden de kleinere afdelingen tamelijk centraal gesitueerd en de grotere aan de periferie.

De beste layout gevonden voor het geassocieerde *gelijke oppervlakte* probleem wordt gegeven door Figuur 3. De totale afstand is hier  $T(p) = 636$ . De ingangscelconfiguratie van Figuur 2 verschilt van die van Figuur 3. We zien hieruit dat om het ongelijke oppervlakte layout probleem op te lossen, men niet kan volstaan met het oplossen van het geassocieerde *gelijke oppervlakte* probleem gevolgd door toevoegen van extra oppervlaktecellen.



FIGUUR 3. De beste layout gevonden voor het gelijke oppervlakte probleem.