

Optimaal Asfalteren met Dynamisch Programmeren

G.J. Franx*, A. van Harten, J.G.M. Kroon, G.T. Timmer*

Samenvatting

De Stafafdeling Wetenschappelijke Assessment (SWA) van de Dienst Informatie Verwerking (DIV, Rijswijk) geeft wetenschappelijke ondersteuning aan projecten bij diverse Diensten van Rijkswaterstaat. Eén van deze projecten betreft de optimalisatie van onderhoud aan rijkswegen, waarvoor de DIV in samenwerking met ORTEC-consultants de LP- en DP-modellen ontwikkelde die in dit artikel beschreven worden.

Wiskundig gesproken komt het probleem neer op de minimalisatie van de frees- en uitvulkosten die nodig zijn om het bestaande wegoppervlak weer zodanig vlak te maken, dat de eerste en tweede afgeleiden van het oppervlak aan zekere vlakheidseisen voldoen. Het artikel gaat met name in op de flexibiliteit van de methode t.a.v. de toegestane kostenfuncties en op de praktische aspecten van de implementatie, zoals het nauwkeurig berekenen van de tweevoudige kosten-integraal in een korte rekentijd.

DIV, Nijverheidsstraat 1, Postbus 5809, 2280 HV Rijswijk, 070-906628

*ORTEC Consultants, Groningenweg 6-02, 2803 PV Gouda, 01820-30588

1. Inleiding

Rijkswaterstaat is verantwoordelijk voor de kwaliteit van de Nederlandse rijkswegen. Om verkeerskundige redenen dient een weg voldoende "vlak" te zijn, hetgeen inhoudt dat er maxima gesteld zijn aan de helling en de kromming in de lengterichting, terwijl de dwarshelling overal voorgeschreven is en alleen in de lengterichting mag variëren. Er mogen dus b.v. geen "sporen" in het wegdek zitten. In het nu volgende staat x voor de lengte-, y voor de breedte-, en z voor de hoogte-coördinaat. We kunnen de vlakheidseisen nu als volgt wiskundig formuleren:

$$\left| \frac{\partial z}{\partial x} \right| \leq \epsilon_1 \quad (1.1)$$

$$\left| \frac{\partial^2 z}{\partial x^2} \right| \leq \epsilon_2 \quad (1.2)$$

$$\frac{\partial z}{\partial y} = C(x) \quad (1.3)$$

Om de vlakheid van de rijkswegen te controleren worden regelmatig hoogtemetingen verricht. Hierbij wordt op een groot aantal roosterpunten de hoogte van het wegdek bepaald. Meestal wordt om de 10 meter een z.g. dwarsprofiel opgemeten, dat op zich weer bestaat uit de hoogtes van 4 tot 10 dwarspunten.

Indien de geconstateerde overschrijdingen van de vlakheidseisen daartoe aanleiding geven, kan Rijkswaterstaat besluiten z.g. klein onderhoud te plegen. Er zijn dan twee mogelijke bewerkingen om de weg te reconstrueren: het frezen van de "te hoge" wegdelen en het uitvullen (met asfalt) van de "te lage" delen. Om constructie-technische redenen zijn beide bewerkingen aan een maximum gebonden (maximale freesdiepte f_{max} en uitvulhoogte u_{max}). Het probleem bij de reconstructie is dat niet éénduidig vaststaat welke wegdelen nu eigenlijk te hoog of te laag zijn. Er zijn in de regel vele oplossingen mogelijk die leiden tot een nieuw wegdek, dat weer aan de vlakheidseisen voldoet. De kosten van de verschillende oplossingen kunnen echter sterk uiteenlopen, zodat er behoefte bestaat aan een methode om de goedkoopste oplossing voor het reconstructie-probleem te vinden.

In praktijk worden wegreconstructies altijd per wegvak (ca. 1 km) uitgevoerd, zodat bij de uiteinden gelet moet worden op een goede aansluiting met de bestaande weg.

We kunnen het wegreconstructie-probleem dus als volgt wiskundig formuleren: Vind het oppervlak dat tegen de laagste kosten aan de vlakheidseisen voldoet, continu aansluit bij het bestaande wegoppervlak, en dat nergens meer dan u_{max} hoger of f_{max} lager ligt dan het oorspronkelijke wegoppervlak.

In eerste instantie is er van uitgegaan dat de kosten van zowel frezen als uitvullen evenredig zijn aan de te frezen en te vullen volumes. Met deze veronderstelling bleken twee oplosmethodes geschikt te zijn om een gediscrètiseerde versie van het probleem aan te pakken: Lineaire Programmering (LP) en Dynamische Programmering (DP) [3]. In het vervolg van dit artikel zullen eerst deze twee methodes toegelicht worden. Omdat de kostenmodellering van deze twee modellen in de praktijk nog te grof blijkt te zijn, zal daarna uitvoerig ingegaan worden op de wijze waarop binnen de DP-aanpak verfijningen van de kostenmodellering gerealiseerd kunnen worden.

2. De LP-aanpak

Om het optimalisatieprobleem wiskundig hanteerbaar te maken, reduceren we het probleem tot een discreet probleem. We beschouwen alleen de meet-roosterpunten en doen alsof alle punten van het wegoppervlak dezelfde hoogte hebben als het dichtst-bijgelegen roosterpunt. Het wegoppervlak wordt dus behandeld als ware het een soort tegeloppervlak, waarvan de tegels op onderling verschillende hoogtes liggen. De tegel op meetpunt (x_i, y_j) heeft een oppervlak ter grootte van $d_{i,j}$, een gemeten hoogte $m_{i,j}$, en als beslisvariabelen de uitvulhoogte $u_{i,j}$ en de freesdiepte $f_{i,j}$, die samen leiden tot de nieuwe hoogte $z_{i,j} = m_{i,j} + u_{i,j} - f_{i,j}$.

Met behulp van het Lagrange interpolatie polynoom [5] voor drie steunpunten vinden we een adequate schatting voor de eerste en tweede afgeleide in het punt (x_i, y_j) :

$$\left| \frac{\partial z}{\partial x} \right| = \left| \frac{z_{i,j} - z_{i-1,j}}{x_i - x_{i-1}} + \frac{z_{i+1,j} - z_{i,j}}{x_{i+1} - x_i} - \frac{z_{i+1,j} - z_{i-1,j}}{x_{i+1} - x_{i-1}} \right| \leq \epsilon_1 \quad (2.1)$$

$$\left| \frac{\partial^2 z}{\partial x^2} \right| = \left| \frac{2z_{i-1,j}}{(x_i - x_{i-1})(x_{i+1} - x_{i-1})} - \frac{2z_{i,j}}{(x_i - x_{i-1})(x_{i+1} - x_i)} + \frac{2z_{i+1,j}}{(x_{i+1} - x_i)(x_{i+1} - x_{i-1})} \right| \leq \epsilon_2 \quad (2.2)$$

Omdat de dwarshellingen vast liggen, worden (2.1) en (2.2) slechts langs één lengte-as getest. De keuze van deze z.g. referentie-as of hoofd-as is arbitrair, b.v. $y = y_1$.

De constructietechnische en dwarshellings-eisen geven als randvoorwaarden:

$$0 \leq u_{i,j} \leq u_{max} \quad (2.3)$$

$$0 \leq f_{i,j} \leq f_{max} \quad (2.4)$$

$$z_{i,j} - z_{i,1} = C(x_i)[y_j - y_1] \quad (2.5)$$

Indien het uitvullen α per m^3 , en het frezen β per m^3 kost, dan komt het probleem voor een wegdek van L lengte- en M breedte-punten dus neer op het minimaliseren van $\sum_{i=1}^L \sum_{j=1}^M d_{i,j}(\alpha u_{i,j} + \beta f_{i,j})$ onder de lineaire randvoorwaarden (2.1),..., (2.5). hetgeen door een standaard LP-pakket opgelost kan worden.

3. De DP-aanpak

Een blijvend nadeel van de LP-aanpak is het feit dat we alleen met lineaire kostenfuncties kunnen werken. Met name niet-convexe en discontinue functies zijn met een LP-aanpak moeilijk te behappen. In deze paragraaf zal ingegaan worden op een Dynamische Programmerings modellering, die een grotere klasse van kostenfuncties toestaat.

Bij de DP-aanpak wordt het probleem nog een stap verder gediscretiseerd. De freesdiepte of vulhoogte op de hoofdas kunnen we bij deze aanpak slechts met een beperkte nauwkeurigheid –zeg δ – kiezen. We hebben bij elk punt op de hoofdas dus de keuze uit $K = \frac{u_{max} + f_{max}}{\delta} + 1$ mogelijke acties. Omdat de dwarshelling voorgeschreven is, liggen alle acties op een gegeven dwarsprofiel vast, zodra de actie op de hoofdas gekozen is. Als het rooster uit L lengte- en M breedte-punten bestaat, dan is elke onderhoudsstrategie dus volledig bepaald door de gekozen hoofdas-acties (k_1, \dots, k_L) , met $k_i \in \{1, \dots, K\}$. Er zijn dus K^L mogelijke onderhoudsstrategieën.

Gelukkig hoeven we deze niet allemaal door te rekenen om de beste strategie te vinden. Indien we aan uiteinde A van het te onderhouden wegvak beginnen en successievelijk naar het andere uiteinde B toewerken, dan blijkt dat de toegestane vervolgstategieën om van een zeker bereikt punt x_i tot B te komen slechts afhangen van de laatste twee gekozen acties: k_i en k_{i-1} (in resp. x_i en x_{i-1}). Dit is een gevolg van het feit dat $\frac{\partial z}{\partial x}$ en $\frac{\partial^2 z}{\partial x^2}$ bepaald worden door een kwadratische interpolatie over drie opeenvolgende steunpunten. Door de keuze van k_i en k_{i-1} wordt dus volledig vastgelegd welke waarden voor k_{i+1} zijn toegestaan, onafhankelijk van (k_1, \dots, k_{i-2}) .

Om goed aan te sluiten bij de bestaande weg, willen we in uiteinde B uitkomen op een gegeven hoogte, d.w.z. k_L is voorgeschreven. De minimale kosten F waarmee we in punt B ($=x_L$) op de voorgeschreven hoogte kunnen aankomen, bedragen:

$$F = \min_{k_{L-1}, k_{L-2}} \{F_{L-1}(k_{L-1}, k_{L-2}) + T(k_L, k_{L-1}, k_{L-2}) + c_L(k_L)\}, \quad (3.1)$$

waarin:

$$\begin{aligned} F_i(k_i, k_{i-1}) &= \text{minimale kosten om (bij } x_i) \text{ in toestand } (k_i, k_{i-1}) \text{ te komen} \\ c_i(k_i) &= \text{de kosten van actie } k_i \text{ op het dwarsprofiel met } x - \text{coördinaat } x_i \\ T(k_i, k_{i-1}, k_{i-2}) &= \begin{cases} 0, & \text{als het actietrio } k_i, k_{i-1}, k_{i-2} \text{ voldoet aan (2.1), \dots, (2.4)} \\ \infty, & \text{als het actietrio } k_i, k_{i-1}, k_{i-2} \text{ niet voldoet.} \end{cases} \end{aligned}$$

De kostenfuncties $c_i(k_i)$ en de testfunctie $T(k_i, k_{i-1}, k_{i-2})$ volgen rechtstreeks uit de probleemstelling. Voor $F_i(k_i, k_{i-1})$ kunnen we een recursieve formule opstellen, waarmee bepaald kan worden wat de minimale kosten zijn om (bij x_i) in toestand (k_i, k_{i-1}) te komen, gegeven de minimale kosten om (bij x_{i-1}) in toestand (k_{i-1}, k_{i-2}) te komen voor alle $(k_{i-1}, k_{i-2}) \in \{1, \dots, K\} \times \{1, \dots, K\}$:

$$F_i(k_i, k_{i-1}) = \min_{k_{i-2}} \{F_{i-1}(k_{i-1}, k_{i-2}) + T(k_i, k_{i-1}, k_{i-2}) + c_i(k_i)\} \quad (3.2)$$

De werkwijze is nu als volgt: Om in punt A ($=x_1$) goed aan te sluiten bij de bestaande weg, mogen we hier alleen een voorgeschreven actie $k_1=k_{start}$ kiezen. We definiëren daarom:

$$F_2(k_2, k_1) = \begin{cases} c_1(k_{start})+c_2(k_2), & \text{voor } k_1=k_{start}, k_2=1, \dots, K \\ \infty, & \text{voor } k_1 \neq k_{start}, k_2=1, \dots, K \end{cases} \quad (3.3)$$

Met de recursie-formule (3.2) kan F_i successievelijk voor $i=3, \dots, L-1$ bepaald worden. (Het is met een kleine aanpassing ook mogelijk om behalve de continuïteitseis in beide uiteinden A en B tevens de voorwaarden (2.1) en (2.2) te eisen. Daartoe moeten ook de punten x_0 en x_{L+1} opgemeten worden.)

Het is handig om tijdens de recursie-berekeningen direct te onthouden wat de optimale acties waren om in een gegeven toestand te komen. Daartoe wordt voor elk lengte-punt x_i de matrix $P_i(k_i, k_{i-1})$ van optimale acties ter hoogte van x_{i-2} opgeslagen, die als volgt geïnterpreteerd dient te worden:

$$P_i(k_i, k_{i-1}) = \text{waarde van } k_{i-2} \text{ waarvoor } F_i(k_i, k_{i-1}) \text{ het minimum aanneemt.}$$

4. Verfijningen van de kostenmodellering in de DP-methode

4.0 Algemeen

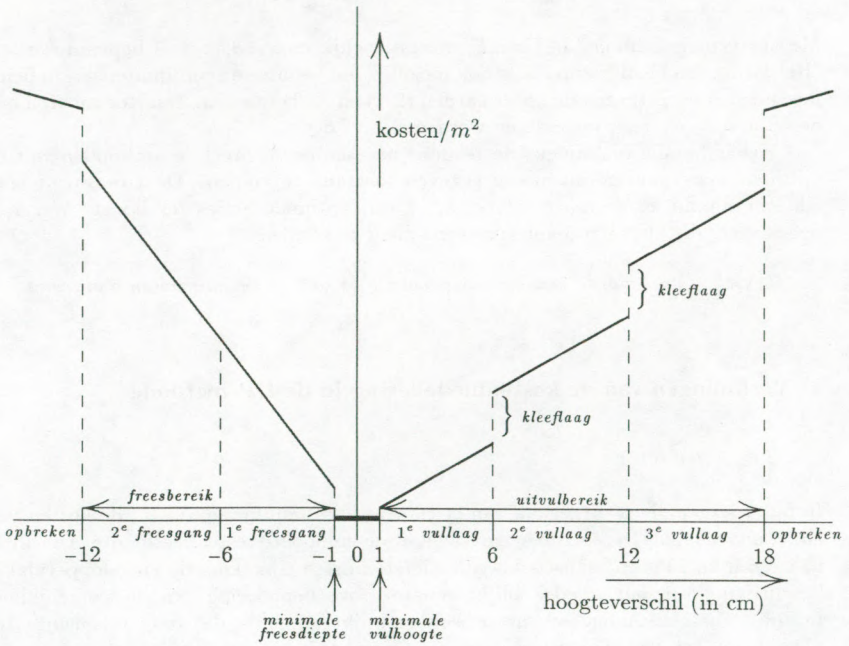
In het voorgaande is uitgegaan van een lineaire kostenfunctie, d.w.z. de kosten van zowel uitvullen als frezen zijn evenredig met de vulhoogte resp. freesdiepte. Dit blijkt in werkelijkheid geen realistische veronderstelling te zijn. Ook de "tegeloppervlak"-beschrijving van het wegdek blijkt een te grove benadering van de werkelijkheid te zijn. Uit eenvoudige rekenvoorbeelden blijkt namelijk dat deze rekenwijze tot afwijkingen van wel 20% van de frees- en/of vul-volumes kan leiden.

In deze sectie wordt aangegeven op welke wijze in een acceptabele rekentijd de kostenintegraal nauwkeurig berekend kan worden bij een ruime klasse van kostenfuncties.

4.1 De representatie van de kostenfunctie

Dynamische programmering staat in principe elke kostenfunctie toe die slechts afhankelijk is van het locale hoogteverschil tussen het nieuwe en het oude weggoppervlak.

De meest flexibele manier om een dergelijke functie te representeren in het geheugen van de computer is een tabel, die van elke toegestane actie op de hoofdas de corresponderende kosten weergeeft. Helaas is het echter zo, dat alleen op de hoofdas de bewerkingshoogtes slechts uit een discrete verzameling gekozen kunnen worden, daarbuiten is een volledig gesloten interval (tussen het frees- en het uitvulmaximum) toegestaan gebied. Er zal voor deze punten dus in ieder geval geïnterpoleerd moeten



Figuur 1: Typisch voorbeeld van een discontinue kostenfunctie

worden tussen de waarden uit de tabel. Een probleem dat hierbij kan optreden is het feit dat we ook discontinue kostenfuncties willen toestaan, zodat het interpoleren tussen twee mogelijke "hoofd-as-acties" (uit de tabel) een hachelijke zaak wordt.

Dit probleem is eenvoudig te ondervangen. In de praktijk blijken de in aanmerking komende kostenfuncties altijd stuksgewijs lineair te zijn, met een eindig aantal discontinuïteits- en/of "knik"-punten. Anders geformuleerd: deze functies bestaan uit een eindig aantal rechte lijnstukjes. We kunnen daarom volstaan met het opslaan van de coördinaten van de begin- en eindpunten van al deze lijnstukjes. Een dergelijke representatie van de kostenfunctie vergt zeer weinig geheugenruimte en stelt ons in staat om in korte rekentijd met grote nauwkeurigheid een willekeurige functiewaarde te bepalen. Wegens deze goede eigenschappen is voor deze kostenfunctie-representatie gekozen. In figuur 1 is een realistisch voorbeeld van een dergelijke kostenfunctie geschetst.

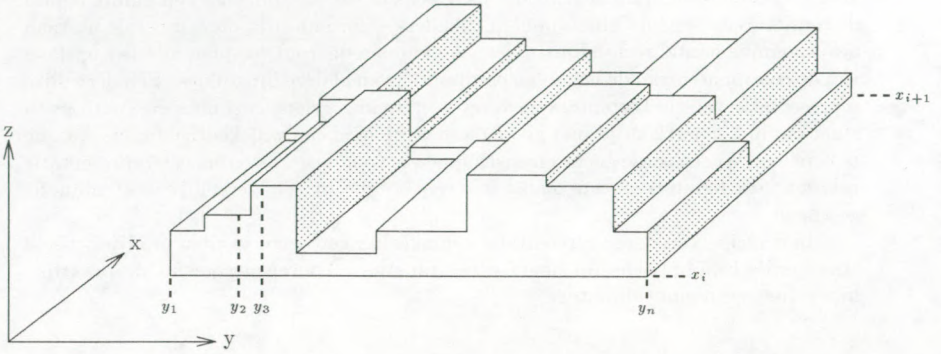
In principe kan deze representatie gemakkelijk gewijzigd worden in bijvoorbeeld stuksgewijs kwadratische (of hoger orde) functies. Vooralsnog geeft "de praktijk" hier echter geen aanleiding toe.

4.2 De interpolatiemethode

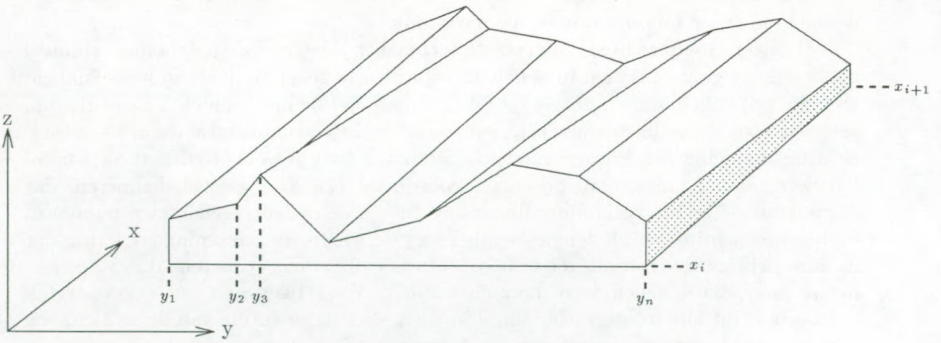
De meetgegevens van het "oude" wegoppervlak zijn slechts beperkt tot een raster van meetpunten waar de hoogte gemeten is. Indien we nauwkeuriger berekeningen willen maken dan de genoemde "tegeloppervlak" methode (zie figuur 2), dan moeten we wel eerst een zinvolle interpolatiemethode kiezen om de tussenliggende hoogtes te definiëren. We dienen daarbij rekening te houden met het feit dat de ligging van de meetpunten niet toevallig is, maar dat met name de breedtepunten zodanig gekozen worden dat de spoorvorming zo goed mogelijk vastgelegd wordt door de metingen. Er wordt daarom getracht de meetpunten zoveel mogelijk overeen te laten komen met de maxima en de minima van de spoorvorming.

De meer "sophisticated" interpolatietechnieken (hogere orde polynomen, splines) hebben de neiging extremen tussen de meetpunten te generen, die er in werkelijkheid niet zijn (zij vallen immers zoveel mogelijk samen met de meetpunten). Er is daarom gekozen voor de veiligste en tevens eenvoudigste interpolatiemethode in de dwarsrichting, namelijk een lineaire interpolatie tussen twee opeenvolgende meetpunten. Op deze manier kunnen we bij elke x -coördinaat een dwarsprofiel definiëren, dat bestaat uit de rechte verbindingslijnstukjes tussen de opeenvolgende meetpunten in de breedte-richting. In de lengte richting van de weg is de kromming zó gering dat in deze richting de tussenliggende hoogtes zonder bezwaar eveneens d.m.v. een lineaire interpolatie gedefinieerd kunnen worden. We krijgen dan een wegoppervlak dat bestaat uit alle rechte verbindingslijnen (in de lengterichting van de weg) tussen twee opeenvolgende dwarsprofielen, zoals geschetst in figuur 3.

Merk op, dat bij een rechthoekig raster de interpolatie-volgorde (eerst lineair in de y -richting, dan in de x -richting, of omgekeerd) niet uitmaakt.



Figuur 2: De LP-interpolatiemethode ("tegeloppervlak")



Figuur 3: De verfijnde DP-interpolatiemethode (bij vaste y linear in x en vice versa)

4.3 De integratiemethode

De gekozen kostenfunctie-representatie en interpolatiemethode bieden ons een ruime keuze aan integratiemethoden om de kosten-integraal K_i voor het wegtrajectje tussen x_i en x_{i+1} te berekenen. Met de notatie uit figuur 3 luidt deze:

$$K_i = \int_{x_i}^{x_{i+1}} \int_{y_1}^{y_n} k(h(x, y)) dy dx \quad (4.1)$$

met: $h(x, y) =$ hoogte die opgevuld ($h > 0$) of weggefreesd ($h < 0$) moet worden
 $k(h) =$ kosten per m^2 bij bewerkingshoogte h

In principe zijn er vier methoden om deze tweevoudige integraal uit te rekenen:

1. Volledig analytisch (in zowel x- als y-richting)
2. In de x-richting analytisch en de y-richting numeriek
3. In de y-richting analytisch en de x-richting numeriek
4. Volledig numeriek (in beide richtingen)

Methode 1

De vorm van de kostenfunctie en de interpolatiemethode stellen ons in principe in staat om de integraal volledig analytisch aan te pakken. Dit heeft natuurlijk als voordeel dat we in één keer (zonder iteraties) een zeer nauwkeurige uitkomst kunnen genereren. Toch kleven er aan deze methode een aantal praktische bezwaren, die tot lange rekentijden leiden:

- a. Het integratie-gebied moet opgesplitst worden naar de verschillende definitiegebieden van de kosten-functie (deze is immers stuksgewijs lineair).
- b. De analytische formule is lang, en vergt derhalve op zichzelf al veel rekentijd.
- c. De integraal vertoont een aantal singulariteiten. Deze zijn welliswaar te verhelpen, maar dat kost wederom extra rekentijd.
- d. De kosten zijn nu niet meer alleen afhankelijk van de laatste actie, maar tevens van de voorlaatste, d.w.z. $c_i(k_i)$ wordt in (3.1)...(3.3) vervangen door $c_i(k_i, k_{i-1})$. Als we bij elk lengte-punt een keuze hebben uit K acties dan moeten we de integraal dus K^2 keer uitrekenen. Dit was in de "tegeloppervlak" methode slechts K keer. De rekentijd gaat dus kwadratisch toenemen met de nauwkeurigheid.

Methode 2

Indien elk dwarsprofiel is opgemeten in n meetpunten (met y -coördinaten y_1, \dots, y_n), dan kunnen we de bewerkingshoogte $h(x, y)$ tussen x_i en x_{i+1} afleiden uit

$D_i(y)$ = bestaande dwarsprofiel ter hoogte van x_i

$D'_i(y)$ = nieuwe dwarsprofiel ter hoogte van x_i

$H_i(y) = D'_i(y) - D_i(y)$ = verschilprofiel

Omdat in beide richtingen lineair geïnterpoleerd wordt, geldt:

$$h(x, y) = H_i(y) + \frac{H_{i+1}(y) - H_i(y)}{x_{i+1} - x_i} (x - x_i) = a_i(y)x + c_i(y), \text{ met: } \begin{cases} a_i(y) = \frac{H_{i+1}(y) - H_i(y)}{x_{i+1} - x_i} \\ c_i(y) = H_i(y) - a_i(y)x_i \end{cases}$$

Met deze notatie worden de kosten voor het wegtrajectje tussen x_i en x_{i+1} :

$$K_i = \int_{y_1}^{y_n} \int_{x_i}^{x_{i+1}} k(a_i(y)x + c_i(y)) dx dy = \int_{y_1}^{y_n} \frac{1}{a_i(y)} \int_{H_i(y)}^{H_{i+1}(y)} k(h) dh dy = \int_{y_1}^{y_n} \frac{1}{a_i(y)} [K(h)]_{H_i(y)}^{H_{i+1}(y)} dy \quad (4.2)$$

De primitieve $K(h)$ van de kostenfunctie $k(h)$ kan eveneens eenvoudig in het geheugen gerepresenteerd worden, omdat het een stuksgewijs kwadratische functie betreft. We hoeven deze primitieve dus aan het begin van het programma slechts éénmaal uit te rekenen, waarna deze eenvoudig in het geheugen opgeslagen kan worden. Bij het berekenen van alle oppervlakte-integralen hoeft dan alleen nog maar in de y -richting numeriek geïntegreerd te worden, b.v. met de Newton-Cotes methode [5]. In zijn totaliteit blijkt deze methode veel efficiënter dan methode 1. Alleen bezwaar d blijft bestaan: alles moet nog steeds K^2 keer uitgerekend worden.

Methode 3

We kunnen natuurlijk niet ontkomen aan het feit dat de kosten afhangen van zowel de laatste als de voorlaatste actie, maar we kunnen wel ontkomen aan het feit dat alles K^2 keer uitgerekend moet worden. Dan moeten we kiezen voor methode 3, d.w.z. eerst analytisch integreren in de dwarsrichting en vervolgens numeriek in de lengterichting. Mathematisch beschouwd verloopt deze methode geheel analoog aan methode 2. Het grote verschil is echter van praktische aard, n.l. de dwarsintegralen die analytisch uitgerekend zijn, kunnen voor meerdere kostenberekeningen gebruikt worden. Zo kan de analytische dwarsintegraal die ter hoogte van x_{i+1} voor zekere actie $k_{i+1}=n$ is uitgerekend, gebruikt worden voor de kostenbepaling van alle toegestane actie-paren (k_i, k_{i+1}) met $k_{i+1}=n$. (k_i = gekozen actie op beslissingspunt x_i .) Dit levert een flinke besparing in rekentijd op t.o.v. methode 2.

Methode 4

Het numeriek integreren in zowel x- als y-richting heeft hetzelfde voordeel als methode 3 en lijdt evenmin aan de bezwaren van methode 1. Het analytisch bepalen van de dwarsintegralen van methode 3 verloopt echter zodanig efficiënt dat deze methode sneller en bovendien nauwkeuriger is, zodat deze verkozen wordt boven methode 4.

4.4 Het aantal interpolatiepunten voor de numerieke integratie

Indien we in de Newton-Cotes integratie kiezen voor slechts één interpolatiepunt, dan komt deze methode overeen met de z.g. Simpson-regel [5], die we als volgt noteren:

$$\int_0^a f(x)dx \approx S_0^a [f(x)dx] \stackrel{\text{def}}{=} \frac{a}{6} \{f(0) + 4f(\frac{1}{2}a) + f(a)\} \quad (4.3)$$

Integratiemethode 3 komt met de Simpson-regel dus neer op de volgende benadering:

$$K_i = \int_0^a \int_0^b k(h(x, y))dy dx \approx S_0^a \left[\int_0^b k(h(x, y))dy dx \right] \quad (4.4)$$

Als we aannemen dat de plaatsen op het wegdek waar een discontinuïteits- of "knik"-punt van de kostenfunctie aangenomen wordt, homogeen verdeeld liggen tussen de meetpunten, dan geeft de Simpson-benadering in verwachting het juiste resultaat. Dit kunnen we inzien door de kostenfunctie $k(z)$ op te vatten als een lineaire combinatie van functies van het type:

$$\pi_p(z) = \{z-p\}^+ = \begin{cases} z-p, & z > p \\ 0, & z \leq p \end{cases} \quad \text{en:} \quad \sigma_p(z) = \{\text{sgn}(z-p)\}^+ = \begin{cases} 1, & z > p \\ 0, & z \leq p \end{cases}$$

Indien we p opvatten als een stochast die homogeen verdeeld is op $(0, a)$, dan geldt:

$$\mathbf{E} \int_0^a \pi_p(z)dz = \mathbf{E} S_0^a [\pi_p(z)dz] \quad (4.5)$$

$$\text{en:} \quad \mathbf{E} \int_0^a \sigma_p(z)dz = \mathbf{E} S_0^a [\sigma_p(z)dz]. \quad (4.6)$$

Bij vaste y is $z = h(x, y)$ lineair in x . Dus is $k(h(x, y))$ te schrijven als een lineaire combinatie van $\pi_p(x)$ en $\sigma_p(x)$ -functies, zodat met (4.5) en (4.6) volgt:

$$\mathbf{E} S_0^a [k(h(x, y))dx] = \mathbf{E} \int_0^a k(h(x, y))dx. \quad (4.7)$$

Daar verwisseling van integratievolgorde is toegestaan in beide leden van (4.4), geldt:

$$\mathbf{E}S_0^a \left[\int_0^b k(h(x,y)) dy dx \right] = \int_0^b \mathbf{E}S_0^a [k(h(x,y)) dx] dy = \mathbf{E} \int_0^b \int_0^a k(h(x,y)) dx dy = \mathbf{E}K_i \quad (4.8)$$

Dit resultaat is minder triviaal dan het op het eerste gezicht lijkt. Vergelijking (4.5) geldt b.v. niet voor de trapeziumregel ($\frac{a}{2}\{f(0) + f(a)\}$), die derhalve in verwachting niet het juiste resultaat voor de kostenintegraal zal opleveren.

5 Realisatie en evaluatie

De LP-methode werd gerealiseerd op een Sperry mainframe (DIV) met een standaard LP-pakket, terwijl de (verfijnde) DP-methode in FORTRAN werd geprogrammeerd voor een PC. Als testprobleem werd een 300m lang wegvak van de A12 met beide methodes doorgerekend. De hoogtes van het wegdek waren om de 10 meter op 6 breedtepunten opgemeten. Voor zover lineaire kostenfuncties werden gebruikt, waren de oplossingen identiek (op de rekennauwkeurigheid na). Afhankelijk van de nauwkeurigheid (aantal toegestane acties binnen $u_{max}=12\text{cm}$ en $f_{max}=-4\text{cm}$) varieerde de rekentijd van het DP-programma tussen 2 seconden (bij een nauwkeurigheid van 10mm) en 1 minuut (1mm). De DP-methode was daarmee in alle gevallen sneller dan de LP-aanpak. Een belangrijker voordeel van DP is natuurlijk de veel grotere flexibiliteit in kostenfuncties en het feit dat de rekentijd slechts lineair toeneemt met de lengte van het door te rekenen wegvak.

De convergentiesnelheid van de numerieke integraal voldeed volledig aan de verwachtingen van sectie 4.4. Bij slechts 1 interpolatiepunt (Simpson) was de nauwkeurigheid al zodanig goed, dat de verschillen met de hogere orde Newton-Cotes interpolaties volstrekt irrelevant waren (beter dan een promille).

Referenties:

- [1] G.J. Franx, A. v.Harten, J.G.M. Kroon, G.T. Timmer, M. Waltmans, M. v.d. Wolf: *Kosten optimalisatie voor de reconstructie van wegen m.b.v. Dynamische Programmering*, Dienst Informatie Verwerking (Rijkswaterstaat) SWA 89/012
- [2] A. van Harten, G.T. Timmer, M. Waltmans, M. v.d. Wolf: *O.R.-technieken voor de optimale reconstructie van wegen*, Dienst Informatie Verwerking (Rijkswaterstaat) SWA 87/071
- [3] F.S. Hillier, G.J. Lieberman: *Introduction to Operations Research*, Holden-Day, 1986
- [4] Projectgroep REKMAT 87: *Materiewensen voor de reconstructie-programmatuur REKON 87 - REKON 2000*, WO-R-86.030
- [5] J. Stoer: *Einführung in die Numerische Mathematik 1*, Springer Verlag, 1976

ontvangen 12- 9-1989
geaccepteerd 10- 9-1992