



Centrum voor Wiskunde en Informatica
Centre for Mathematics and Computer Science

M.W.P. Savelsbergh

Interactieve methoden voor de routing van voertuigen

Afdeling Mathematische Besliskunde en Systeemtheorie

Notitie OS-N8404

December

Interactive methods for vehicle routing

Bibliotheek
Centrum voor Wiskunde en Informatica
Amsterdam

Interactieve Methoden voor de Routing van Voertuigen

M.W.P. Savelsbergh

Centrum voor Wiskunde en Informatica, Amsterdam

In het *standaard* voertuig-routeringsprobleem (VRP) moet een aantal voertuigen vanuit een centraal depot een aantal klanten bezoeken. Elk voertuig heeft een beperkte capaciteit en elke klant heeft een bepaalde vraag. Het probleem is routes voor de voertuigen te vinden zodat alle klanten worden bezocht, de totale vraag van de klanten in de route de capaciteit van het betreffende voertuig niet overschrijdt, en de totale lengte van de routes minimaal is.

Dit probleem en talloze variaties treden op in een grote verscheidenheid van praktische situaties. Overheid en industrie besteden een aanzienlijk deel van hun budget aan distributie- en transportactiviteiten. Een effectief routeringssysteem kan enorme besparingen in tijd, geld en energie opleveren.

VRPs behoren tot de lastige problemen in de combinatorische optimalisering. Resultaten uit de complexiteitstheorie geven aan dat er waarschijnlijk geen snelle optimaliseringsmethoden bestaan, zodat het gebruik van benaderingsalgoritmen onvermijdelijk lijkt.

De klassieke VRP-heuristieken construeren routes op grond van eenvoudige, intuïtieve criteria; soms wordt daarna nog een groot aantal mogelijkheden tot lokale verbetering van de routes nagegaan. De resulterende oplossingen zijn over het algemeen matig van kwaliteit. De computerimplementatie is altijd *off-line*, d.w.z. zonder mogelijkheid tot menselijk ingrijpen in de berekeningen.

Recentelijk zijn krachtiger VRP-heuristieken ontwikkeld op basis van formele *wiskundige modellen* en met gebruikmaking van *interactieve methoden*. Optimaliseringsmethoden voor deze modellen vergen (te) veel tijd, maar zij kunnen tevens benaderingsalgoritmen suggereren. Op deze wijze zijn enkele heuristieken ontworpen die effectiviteit en efficiëntie verrassend combineren, d.w.z. uitstekende oplossingen produceren in weinig rekentijd.

Het succes van *interactieve technieken* viel samen met bovengenoemde ontwikkelingen en met de beschikbaarheid van in- en uitvoerfaciliteiten voor rastergrafische beeldstations. Terwijl de computer oplossingen aandraagt is de menselijke beslisser bezig het gehele oplossingsproces *on-line* te sturen. Niet alleen leidt dit tot betere oplossingen, maar het vergroot ook de kans op acceptatie van de inschakeling van computers in praktische situaties.

1980 Mathematics Subject Classification: 68V20, 69G30, 69G12. *69K36*

Trefwoorden: voertuig-routeringsprobleem, interactieve methode, optimalisering, gebruikersinterface, kleurengrafiek, Graphical Kernel System.

Opmerking: Dit onderzoek wordt ondersteund door de Stichting voor de Technische Wetenschappen (project CWI-22.0348).

1. Inleiding

Door een enorme toename in de kosten van fysieke distributie en de economische depressie van het laatste decennium is er een groeiende vraag naar beslissingsondersteunende systemen en optimaliseringsmethoden voor distributiemanagement. Ondanks de grote verscheidenheid van distributieproblemen zijn de basiscomponenten vrijwel altijd een aantal voertuigen met vaste eigenschappen (capaciteiten, snelheden, etc.) en een verzameling orders voor het vervoer van objecten (producten, schoolkinderen, etc.) tussen gespecificeerde afhaal- en besteladressen. Bij het *voertuig-routeringsprobleem* gaat het erom te bepalen welke orders toegewezen worden aan ieder voertuig en welke route ieder voertuig moet volgen bij het afwerken van de orders zodanig dat de kosten geminimaliseerd worden. (Voor nadere informatie over het voertuig-routeringsprobleem zie [1].)

Tegenwoordig is er een aantal softwarepakketten op de markt voor het oplossen van voertuig-routeringsproblemen, waarvan sommige zijn gebaseerd op geavanceerde technieken uit de mathematische programmering. Het gebruik van deze pakketten leidt in de praktijk tot aanzienlijke besparingen op de distributiekosten. Bijna alle op dit moment beschikbare pakketten werken *off-line*. Hiermee bedoelen wij dat de gebruiker geen invloed kan uitoefenen op het oplossingsproces. Het programma fungeert als *black box*: na het specificeren van de invoergegevens krijgt de gebruiker, bij beëindiging van het programma, de verzameling te rijden routes. In het dagelijks gebruik blijkt het vaak, uit praktische overwegingen, noodzakelijk om in deze routes nog enige veranderingen aan te brengen. Om deze reden werken vele ontwikkelaars van voertuig-routeringspakketten aan interactieve uitbreidingen die het mogelijk maken achteraf met routes te manipuleren. Hoewel deze uitbreidingen bijdragen tot een grotere flexibiliteit, zijn wij van mening dat betere resultaten behaald kunnen worden als de interactie tussen computer en gebruiker start bij het begin van het oplossingsproces. Wij stellen voor de *black box* open te breken en de krachten van mens en computer te bundelen om zo het beste resultaat te verkrijgen.

Met dit artikel willen wij twee punten naar voren brengen. Ten eerste geven wij aan dat een interactieve aanpak uitermate geschikt is voor het voertuig-routeringsprobleem en ten tweede benadrukken wij het belang van een gebruikersinterface gebaseerd op kleurengrafiek. In de volgende sectie geven wij een formele definitie van het probleem. Daarna bespreken wij de voordelen van een interactieve aanpak van het voertuig-routeringsprobleem en zeggen wij iets over de invloed van een dergelijke aanpak op de te gebruiken optimaliseringsalgoritmen. Een van de belangrijkste onderdelen van een interactief systeem is de gebruikersinterface. Een duidelijke en gebruikersvriendelijke interface draagt niet alleen bij tot een snellere acceptatie van het systeem, maar is tevens voor een groot deel verantwoordelijk voor de kwaliteit van de verkregen resultaten. In de laatste secties vertellen wij iets over onze kijk op een gebruikersinterface en over onze ervaringen met het implementeren van een interactief pakket voor voertuig-routeringsproblemen, waarbij wij gebruik hebben gemaakt van het Graphical Kernel System (GKS) [6].

2. Het voertuig-routeringsprobleem

In het *standaard* voertuig-routeringsprobleem (VRP) moet een aantal voertuigen vanuit een centraal depot een aantal klanten bezoeken. Elk voertuig heeft een beperkte capaciteit en elke klant heeft een bepaalde vraag. Het probleem is routes voor de voertuigen te vinden zodat alle klanten worden bezocht, de totale vraag van de klanten in de route de capaciteit van het betreffende voertuig niet overschrijdt, en de totale lengte van de routes minimaal is. Om een precieze beschrijving van het probleem te kunnen geven voeren wij de volgende notatie in.

Constanten

- K = aantal voertuigen.
 n = aantal klanten dat bezocht moet worden. Klanten worden geïndiceerd van 1 t.e.m. n en index 0 staat voor het centrale depot.
 b_k = capaciteit (gewicht of volume) van voertuig k .
 a_i = grootte van de aflevering aan klant i .
 c_{ij} = afstand van klant i tot klant j .

Variabelen

$$y_{ik} = \begin{cases} 1 & \text{als klant } i \text{ bezocht wordt door voertuig } k \\ 0 & \text{anders} \end{cases}$$

$$x_{ijk} = \begin{cases} 1 & \text{als voertuig } k \text{ gaat van klant } i \text{ naar klant } j \\ 0 & \text{anders} \end{cases}$$

Formulering van het voertuig-routeringsprobleem

minimaliseer

$$\sum_{ijk} c_{ij} x_{ijk} \quad (1)$$

zodanig dat

$$\sum_i a_i y_{ik} \leq b_k, \quad k = 1, \dots, K, \quad (2)$$

$$\sum_k y_{ik} = \begin{cases} K, & i = 0, \\ 1, & i = 1, \dots, n, \end{cases} \quad (3)$$

$$\sum_i x_{ijk} = y_{jk}, \quad j = 0, \dots, n, \quad k = 1, \dots, K, \quad (4)$$

$$\sum_j x_{ijk} = y_{ik}, \quad i = 0, \dots, n, \quad k = 1, \dots, K, \quad (5)$$

$$\sum_{(i,j) \in S \times S} x_{ijk} \leq |S| - 1, \quad S \subseteq \{1, \dots, n\}, \quad 2 \leq |S| \leq n - 1, \quad k = 1, \dots, K. \quad (6)$$

Ingebed in deze formulering bevinden zich twee bekende combinatorische optimaliseringsproblemen. Restricties (2)-(3) zijn de restricties van een *gegeneraliseerd toewijzingsprobleem* en zorgen ervoor dat het depot onderdeel is van iedere route, dat iedere klant bezocht wordt door een voertuig, en dat de lading toegewezen aan een voertuig de capaciteit niet overschrijdt. Als de y_{ik} vastliggen en voldoen aan (2)-(3), dan definiëren de restricties (4)-(6), voor ieder voertuig k , een *handelsreizigersprobleem* voor de klanten toegewezen aan voertuig k .

Hoewel de boven gegeven formulering erg compact is, is het aantal variabelen en restricties enorm. Verder zijn beide optimaliseringsproblemen erg nauw met elkaar verbonden en moeten zij in een optimaliseringsalgoritme simultaan worden opgelost. Met de huidige technieken is dit onmogelijk, zelfs voor problemen van zeer beperkte omvang. Daarom moeten wij onze toevlucht nemen tot heuristische methoden.

Een van de eerste en meest gebruikte heuristieken is bekend geworden als de *besparingenmethode* [3]. In eerste instantie nemen wij aan dat iedere klant individueel bezocht wordt door een voertuig. Veronderstel nu dat wij twee klanten i en j samenvoegen en bezoeken met één voertuig. Dit elimineert een voertuig en resulteert in een besparing in afstand van

$$(2c_{0i} + 2c_{0j}) - (c_{0i} + c_{ij} + c_{j0}) = c_{0i} + c_{0j} - c_{ij}.$$

Voor ieder mogelijk paar klanten i en j is er een bijbehorende besparing s_{ij} . We ordenen deze

besparingen in afnemende volgorde en, boven aan de lijst beginnend, voegen wij klanten i en j samen tenzij een van de restricties wordt geschonden.

Een recentere aanpak is afkomstig van Fisher & Jaikumar [4]. Zij scheiden het gegeneraliseerde toewijzingsprobleem en het handelsreizigersprobleem en creëren op die manier een *twee-fasemethode*. In de eerste fase wordt een toewijzing van klanten aan voertuigen verkregen door het oplossen van een gegeneraliseerd toewijzingsprobleem met een doelstellingsfunctie die de lengte van de handelsreizigersroutes van de voertuigen langs de klanten benadert. In de tweede fase, nadat de toewijzing is gemaakt, wordt een route verkregen voor ieder voertuig langs de aan dat voertuig toegewezen klanten door het oplossen van een handelsreizigersprobleem.

Christofides, Mingozzi & Toth [2] ontwikkelden een branch & bound methode die in staat is kleine probleeminstanties optimaal op te lossen. Zoals het geval is met alle branch & bound algoritmen is de effectiviteit volledig afhankelijk van de kwaliteit van de grenzen die gebruikt worden om de zoekboom te beperken. De grens die zij gebruiken is gerelateerd aan *through-q-routes*. Een through- q -route is de kortste route met lading q zonder lussen, beginnend en eindigend in het depot, en onderweg een gegeven klant bezoekend.

3. Interactieve optimalisering

Er zijn twee belangrijke redenen om een interactieve aanpak van het voertuig-routeringsprobleem te prefereren boven een off-line aanpak. De ene gebaseerd op algoritmische, de andere op praktische overwegingen.

Het voertuig-routeringsprobleem (zoals boven beschreven) behoort tot de klasse van NP-lastige problemen [7]. Deze klasse bevat problemen die naar alle waarschijnlijkheid inherent moeilijk zijn [5]. Dit houdt in dat alleen kleine probleeminstanties optimaal op te lossen zijn met een aanvaardbare hoeveelheid rekenwerk. Verder is de formulering uit de vorige sectie gebaseerd op de eenvoudigste verschijningsvorm van het voertuig-routeringsprobleem. Praktische problemen zijn vaak moeilijker door de aanwezigheid van extra restricties, zoals maximale routelengte voor de voertuigen en tijdsrestricties bij de klanten. Deze beide observaties geven aan dat wij ons niet moeten vastleggen op het vinden van een optimale oplossing, maar ons moeten concentreren op het vinden van een 'goede' oplossing met een aanvaardbare hoeveelheid rekenwerk. Om dit te bewerkstelligen gebruiken wij heuristische methoden. Bij het ontwerpen van een heuristische methode voor het voertuig-routeringsprobleem kunnen wij profiteren van de meetkundige structuur in samenhang met het ruimtelijk inzicht van de mens. Een planner kan de computer sturen door deelverzamelingen aan te geven die waarschijnlijk bijna-optimale oplossingen bevatten en deelverzamelingen uit te sluiten die waarschijnlijk geen bijna-optimale oplossingen bevatten. Deze manier van mens-machine interactie kan de praktische complexiteit van een probleeminstantie aanzienlijk verkleinen.

Een tweede en zeker niet minder belangrijke eigenschap van interactieve systemen is de flexibiliteit, waardoor menselijk ingrijpen in de besluitvorming mogelijk wordt. Hieronder wordt een aantal praktische aspecten van het voertuig-routeringsprobleem beschreven waarbij dit essentieel is:

- Een belangrijke beperking van off-line systemen voor voertuig-routeringsproblemen is het feit dat zij geen rekening kunnen houden met de flexibiliteit van de gegevens. Capaciteitsrestricties en tijdsrestricties hebben een zekere flexibiliteit. Neem bijvoorbeeld de tijdsrestricties bij een klant; tien minuten later arriveren dan het einde van het tijdvenster hoeft niet automatisch te leiden tot ontoelaatbaarheid van de route. Sommige klanten zullen er geen bezwaar tegen maken zolang het niet iedere dag voorkomt. Een planner kan zo'n situatie eenvoudig beoordelen maar het is ondoenlijk om dergelijke overwegingen op te nemen in de software.
- Door geen enkele controle over het oplossingsproces toe te staan, is het heel goed mogelijk dat de routes van dag tot dag veel verschillen. Dit kan theoretisch gezien wel leiden tot minder gereden kilometers en kortere rijtijden, maar, in de praktijk, door onvoldoende kennis van bepaalde gebieden door de chauffeurs, zou het juist kunnen leiden tot een toename in de werkelijke rijtijd. Bij een interactieve aanpak kan de planner dergelijke aspecten tegen elkaar afwegen voor hij een beslissing neemt.

- In off-line systemen is het veel moeilijker te reageren op onverwachte gebeurtenissen (spoedorders, onderhoud aan wegen, etc).
- Naast de voor de hand liggende criteria van afstand, tijd en benuttingsgraad zijn er ook andere doelstellingsfuncties, die moeilijk precies te definiëren zijn maar, die een belangrijke rol spelen bij planningsbeslissingen (gelijke werkverdeling voor chauffeurs, voorkeuren van de chauffeurs, etc.).

Een interactieve planningsmethode voor voertuig-routing moet dan ook niet alleen bestaan uit geavanceerde mathematische subroutines voor deelproblemen, maar er moeten ook plaatsen zijn waar menselijk ingrijpen mogelijk en nuttig is. Wij hebben gekozen voor een twee-fasemethode gebaseerd op de aanpak van Fisher & Jaikumar [4]. Speciaal in de toewijzingsfase kan een planner het oplossingsproces gunstig beïnvloeden. De functie die de afleveringskosten benadert wordt verkregen met behulp van *kiemroutes* ("seed routes") en de kosten van het invoegen van klanten in deze kiemroutes. Een kiemroute is een kunstmatige route die in eerste instantie bestaat uit het depot en een *kiempunt* ("seed point"), waarbij het kiempunt een gebied aangeeft waarvan verwacht wordt dat het bezocht gaat worden door één voertuig. De kiempunten worden gekozen door de planner gebaseerd op zijn kennis en intuïtie. Hij kan iteratief experimenteren met verschillende kiemroutes en onmiddellijk het effect zien op de kosten en routeringsbeslissingen. In de routeringsfase kan de planner geheel naar eigen inzicht routes maken of de door de computer gegenereerde routes aanpassen.

4. Gebruikersinterface met behulp van kleurengrafiek

Zoals al eerder is aangegeven leidt een interactieve aanpak van voertuig-routeringsproblemen tot grotere flexibiliteit en betere resultaten. Een heel belangrijk onderdeel van een interactief planningssysteem zal de gebruikersinterface zijn. Wij denken dat een grafische gebruikersinterface een noodzaak is. Het hoofdargument voor het gebruik van grafische eindstations is hun effectiviteit bij het tonen van informatie. Gegevens die anders in alfanumerieke vorm zouden moeten worden weergegeven kunnen nu, met behulp van het eindstation, gebruikt worden om een meetkundige representatie van het probleem op te bouwen, die èn overzichtelijk èn informatief is.

Het effect van een grafische gebruikersinterface zal nog sterker zijn als gebruik wordt gemaakt van kleurengrafiek. Kleurengrafiek biedt de mogelijkheid om op een eenvoudige manier onderscheid aan te brengen tussen de verschillende routes door ieder een andere kleur te geven. Voor problemen met een klein aantal routes levert dit slechts een klein voordeel op, maar voor problemen met een groot aantal routes wordt het een noodzaak.

Een meetkundige representatie is echter slechts één kant van het probleem. De gebruiker moet ook de mogelijkheid hebben een deel van de enorme hoeveelheid alfanumerieke gegevens te bekijken.

Om het scherm voor de gebruiker zo rustig mogelijk te houden hebben wij het verdeeld in drie gebieden:

OPDRACHT INFO:	Dit gebied is gereserveerd voor het tonen van opdrachten;
GRAFISCHE INFO:	Dit gebied is gereserveerd voor het tonen van de meetkundige representatie;
ALFANUMERIEKE INFO:	Dit gebied is gereserveerd voor het tonen van allerlei alfanumerieke informatie.

Wij hebben gekozen voor een menu-gestuurde interactie. Menu-gestuurde interactie heeft het voordeel dat op ieder tijdstip alle op dat moment mogelijke opdrachten voor de gebruiker duidelijk zichtbaar zijn en het voorkomt dat de gebruiker niet-toegelaten opdrachten geeft.

De mogelijke opdrachten zijn verdeeld over vier menu's. Er zijn twee hoofdmenu's. Het eerste bevat de opdrachten die gebruikt kunnen worden in de toewijzingsfase en het tweede de opdrachten die gebruikt kunnen worden in de routeringsfase. Daarnaast zijn er nog twee menu's voor de ondersteunde functies, die op ieder willekeurig moment kunnen worden gebruikt. Eén van deze menu's bevat de functies die zorgdragen voor het tonen van gewenste informatie, het andere

menu bevat de functies die de inhoud van het grafische info-gebied definiëren. De opsplitsing van opdrachten in vier groepen heeft er toe geleid dat de ruimte op het scherm, nodig voor het tonen van de opdrachten, is afgenomen omdat ten hoogste twee menu's tegelijkertijd zichtbaar moeten zijn.

Een deelverzameling van de opdrachten in de routeringsfase geeft de planner de mogelijkheid om met routes te manipuleren. Te allen tijde kan de planner klanten heen en weer schuiven en routes veranderen. Routemanipulatie heeft plaats met behulp van verwisselingstechnieken [9]. De planner heeft twee mogelijkheden: hij kan een deel van de route verplaatsen of van richting veranderen. Een locale verwisseling is opgebouwd uit twee stappen. In de eerste stap geeft de planner aan welk deel van de route hij wil verplaatsen dan wel van richting wil veranderen. Dit resulteert in het volgende:

- tonen van de voorgestelde verwisseling in het grafische info gebied (de oorspronkelijke situatie blijft ook zichtbaar!);
- tonen van informatie betreffende veranderingen in gereden kilometers, rijtijd, wachttijd, lading en toelaatbaarheid.

Pas in de tweede stap beslist de planner of de voorgestelde verwisseling daadwerkelijk moet worden uitgevoerd. Op deze manier is de planner in staat om een aantal verschillende verwisselingen te bekijken en te vergelijken alvorens er één uit te voeren en is hij tevens beschermd tegen het maken van vergissingen bij de invoer.

Om de planner de mogelijkheid te bieden onderdelen van de probleeminstantie apart te beschouwen zijn twee ondersteunende functies gecreëerd. De eerste stelt de planner in staat een bepaald gebied uit te vergroten en de tweede stelt hem in staat de zichtbaarheid van routes tijdelijk te veranderen. Beide ondersteunende functies zijn zeer nuttig bij het manipuleren van routes.

5. GKS

Voor de implementatie van een gebruikersinterface met kleurengrafiek hebben wij gebruik gemaakt van het grafische pakket GKS ontwikkeld op het CWI [8]. GKS is een grafisch basispakket voor toepassingen die tweedimensionale plaatjes genereren op vector-grafische of raster-grafische eindstations. Het ondersteunt mens-machine interactie door het leveren van basisfuncties voor grafische invoer en beeldsegmentatie.

Het GKS-concept van een *segment*, het opslagmechanisme, blijkt uitermate goed bruikbaar in deze applicatie. Segmenten zijn de eenheden voor manipulatie en transformatie. Manipulatie omvat creatie, vernietiging en naamsverandering. Transformatie omvat translatie, rotatie, aanpassing van zichtbaarheid, aanpassing van detecteerbaarheid en het doen oplichten. Daarnaast is het met behulp van de *pick identifier* mogelijk, binnen een segment, groepen uitvoerprimitieven te onderscheiden. Deze tweede identificatielaag wordt geleverd om de segmentadministratie te reduceren voor applicaties waar plaatjes zijn opgebouwd uit een groot aantal kleinere componenten die allen onderscheiden moeten worden voor invoerdoeleinden, maar waar de noodzaak voor manipulatie minder belangrijk is. Maar dat is precies de situatie in de applicatie die wij op dit moment aan het beschrijven zijn. In het voertuig-routeringsprobleem is er een groot aantal klanten die verdeeld worden over een relatief klein aantal routes. Het meest voor de hand liggende is dan ook om een segment te creëren voor elke route en iedere klant binnen zo'n route een eigen pick identifier te geven. Deze constructie heeft tevens tot gevolg dat een aanroep van de REQUEST PICK functie in één keer en de klant en de route waar deze deel van uitmaakt identificeert.

De *segmentattributen* spelen ook een belangrijke rol in deze applicatie. De detecteerbaarheid van een segment geeft ons de mogelijkheid de invoer te beheersen en de gebruiker te beschermen tegen foutieve aanroepen van de REQUEST PICK functie. Verder zijn de ondersteunende functies die de inhoud van het grafische info gebied definiëren bijna geheel gebaseerd op de segmentattributen. De zichtbaarheid van een segment en dus de zichtbaarheid van de corresponderende route kan veranderd worden door het aanpassen van het zichtbaarheidsattribuut.

Door het aanpassen van de huidige segmenttransformatie en de huidige normalisatietransformatie kunnen plaatjes worden uitvergroot. De segmenttransformatie vergroot de al aanwezige uitvoer uit en de normalisatietransformatie de nieuw gegenereerde uitvoer.

Onze keuze voor een menu-gestuurde interactie wordt in GKS verwezenlijkt door het *CHOICE logical input device*. (Om op een uniforme wijze, en onafhankelijk van de gebruikte hardware, om te kunnen gaan met alle mogelijke vormen van invoer introduceert GKS zes logical input device klassen.) GKS laat het aan de applicatieprogrammeur over om voor ieder logical input device het geschikste *prompt* type te definiëren (met behulp van een prompt geeft het applicatieprogramma aan dat het invoer verwacht), en een van de mogelijkheden voor het *CHOICE logical input device* is een prompt die een aantal teksten toont, die te zamen een menu vormen.

Voor ieder logical input device kent GKS drie stijlen van interactie: *REQUEST*, *SAMPLE* en *EVENT*. De huidige implementatie kent alleen de synchrone *REQUEST*. Bij deze interactiestijl wordt het programma onderbroken op het moment dat het om invoer vraagt tot het moment waarop de gebruiker via een signaal aangeeft dat de invoer beëindigd is. De beide andere interactiestijlen laten ook asynchrone invoer toe en zijn daarom beter geschikt voor complexere interactie. De afwezigheid van asynchrone invoerfaciliteiten heeft ons echter niet beperkt in de ontwikkeling van de interactieve applicatie die wij ons voorstelden.

Het huidige systeem wordt uitgetest op een AED512, een eenvoudig raster-grafisch eindstation van 500 bij 500 pixels (picture elements). De AED512 ondersteunt op geen enkele manier moderne eindstation eigenschappen zoals interne beeldsegmentatie, transformaties en clipping en filling algoritmen. Al deze acties worden daarom uitgevoerd op de hoofdcomputer, een VAX 11/750. Dit houdt echter niet in dat de interactie traag is. Doordat de uitvoerprimitieven die door het applicatie programma gegenereerd worden niet al te complex zijn en een snelle hoofdcomputer gebruikt wordt is de responstijd acceptabel. In de toekomst als een modern scherm met ingebouwde GKS-functies wordt gebruikt zal de interactiesnelheid aanzienlijk omhoog gaan en de respons spectaculaire vormen aannemen.

Omdat GKS onafhankelijk van de gebruikte configuratie gedefinieerd is, zal de enige verandering bij het overgaan naar andere apparatuur de vervanging van het scherm type zijn in de *OPEN WORKSTATION* functie.

6. Conclusie

Wij hebben een aantal argumenten gegeven ten gunste van een interactieve aanpak voor het voertuig-routeringsprobleem. Deze argumenten zijn gebaseerd op overwegingen die betrekking hebben op algoritmische aspecten, flexibiliteit en gebruikersvriendelijkheid. Wij hebben verder aangegeven dat bepaalde optimaliseringsmethoden, ontwikkeld voor het voertuig-routeringsprobleem, beter geschikt zijn voor mens-machine interactie dan andere en hebben een *cluster first-route second* aanpak beschreven. Een belangrijk onderdeel van een interactief systeem is de gebruikersinterface. Wij stellen dat kleurengrafiek de basis moet vormen van een dergelijke gebruikersinterface voor voertuig-routeringspakketten. Een goede gebruikersinterface zorgt niet alleen voor een snelle acceptatie van het systeem, het speelt ook een vitale rol tijdens het oplossingsproces. Wij hebben daarnaast onze ervaringen met het implementeren van een gebruikersinterface gebaseerd op kleurengrafiek, waarbij we gebruik hebben gemaakt van het grafische pakket GKS, besproken. Het bleek dat voor de applicatie beschreven in dit artikel de meeste functies eenvoudig konden worden opgebouwd met behulp van de GKS basisfuncties.

7. Literatuur

1. L.D. Bodin, B.L. Golden, A. Assad & M. Ball (1983). The state of the art in routing and scheduling of vehicles and crews. *Computers and Oper. Res.* 10, 63-212.
2. N. Cristofides, A. Mingozzi & P. Toth (1980). Exact algorithms for the vehicle routing problem based on spanning tree and shortest path relaxation. *Math. Programming* 19, 255-282.
3. G. Clarke & J.W. Wright (1964). Scheduling of vehicles from a central depot to a number of delivery points. *Oper. Res.* 12, 568-581.
4. M.L. Fisher & R. Jaikumar (1981). A generalized assignment heuristic. *Networks* 11, 109-124.
5. M.R. Garey & D.S. Johnson (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman, San Fransisco.
6. ISO (1982). *Graphical Kernel System (GKS) - Functional Description*, Standard ISO/DIS 7942.
7. J.K. Lenstra & A.H.G. Rinnooy Kan (1981). Complexity of vehicle routing and scheduling problems. *Networks* 11, 221-227.
8. D.S.H. Rosenthal & P.J.W. ten Hagen (1982). *GKS in C*, Report IW 204/82, Mathematisch Centrum, Amsterdam.
9. M.W.P. Savelsbergh (1984). *Local Search in Routing Problems with Time Windows*. Report OS-R8409, Centre for Mathematics and Computer Science, Amsterdam.