

## En funktionel dynamisk vejvalgsmodel

Otto Anker Nielsen

Professor, Center for Trafik- & Transport-forskning. (CTT)  
Danmarks Tekniske Universitet (DTU), Bygn. 115, st.tv.  
DK-2800 Lyngby, Denmark  
e-mail: [oan@ctt.dtu.dk](mailto: oan@ctt.dtu.dk), Tlf: 4525 1514, Fax: 4593 6412

Rasmus Dyhr Frederiksen

Center for Trafik- & Transport-forskning. (CTT)  
Danmarks Tekniske Universitet (DTU), Bygn. 115, st.tv.  
DK-2800 Lyngby, Denmark  
e-mail: [rdf@ctt.dtu.dk](mailto: rdf@ctt.dtu.dk), Tlf: 4525 148614, Fax: 4593 6412

### ABSTRAKT

Med den stigende trængsel – især i Hovedstadsområdet – er det klart, at statiske modeller ofte ikke tilstrækkeligt præcist beskriver trafikafviklingen. Dette har i særlig grad betydning for modeller, der benyttes i forbindelse med vurdering af forslag til forbedring af vejnettets kapacitet – f.eks. omlægning af signalanlæg, udvidelse af vejkapacitet, forbedring af ramper, etc. Der er således et behov for et funktionelt alternativ til de traditionelle statiske vejvalgsmodeller.

Inspireret af dette samt forskning i udlandet, skitseres i det følgende principperne for en stokastisk dynamisk flerklasse vejvalgsmodel, og de indledende erfaringer fra implementering og test af denne model beskrives.

# 1 INDLEDNING

I danske trafikmodeller har man primært benyttet statiske vejvalgsmodeller. Her antages det;

- 1) at trafikken er rektangulært fordelt over et bestemt tidsrum (at trafikmængden f.eks. er konstant i myldretiden),
- 2) at alle ture afvikles over dette tidsrum (f.eks. at en tur fra Gilleleje til Kbh.'s centrum på 1 time med start kl. 8.50 afvikles inden for myldretiden 7.00-9.00),
- 3) at rejsetiden er konstant i tidsrummet (hvis samme tur startes kl. 6.50 er der fortsat ingen kø i Kbh. centrum, når turen når derind ved 8-tiden), og at
- 4) kødannelsen på den enkelte vejstrækning er uafhængig af de øvrige vejstrækninger (at der ikke sker tilbagestuvning af trafikpropper).

Med den stigende trængsel – især i Hovedstadsområdet – er det klart, at sådanne modeller ofte ikke tilstrækkeligt præcist beskriver trafikafviklingen. Dette har i særlig grad betydning for modeller, der benyttes i forbindelse med vurdering af forslag til forbedring af vejnettets kapacitet – f.eks. omlægning af signalanlæg, udvidelse af vejkapacitet, forbedring af ramper, etc. Der er således et behov for et funktionelt alternativ til de traditionelle statiske vejvalgsmodeller.

Dynamiske vejvalgsmodeller kan groft deles i to klasser; 1) analytisk baserede modeller, der sædvanligvis kun kan løses for ret små trafiknet (uanset mængden af regnekraft), og 2) simuleringsbaserede modeller, der kan løses for medium store net, men som kræver meget stor regnekraft. Funktionelle dynamiske vejvalgsmodeller er primært af den sidste type. Men sådanne modeller kræver ofte en stor ressourceindsats at implementere. Og som regel kan de kun benyttes for en delmængde af det samlede trafiknet, f.eks. Motorring 3. De skal i givet fald linkes til en mere overordnet statisk model ved hjælp af en heuristik.

Nyere forskning ved *University of Montreal* har vist, at deterministiske dynamiske vejvalgsmodeller kan løses ved hjælp af successive gennemsnits metode for relativt store net. Modellen finder en bruger-ligevægt, hvor ingen bilist kan finde en hurtigere rute end den fundne med det ønskede starttidspunkt og den ønskede destination. En sådan model er testet for det overordnede vejnet i Stockholm. En ulempe er imidlertid at en forholdsvis detaljeret simulering af trafikafviklingen på den enkelte strækning umuliggør, at modellen kan køre på f.eks. hele Hovedstadsområdets trafiknet.

Inspireret af dette arbejde, skitseres i det følgende principperne for en stokastisk dynamisk flerklasse vejvalgsmodel, tænkt som en udvidelse af den klassiske statiske model. Målet er at kunne modellere dynamiske trængselseffekter på et tilstrækkeligt detaljeringniveau, på en måde der er nok mindre detaljeret end ved en mikrosimuleringstilgang, men som samtidig også kan anvendes på større og mere detaljerede modelnet.

## 2 OVERORDNET METODEBESKRIVELSE

Helt overordnet kan den nedenfor beskrevne metode beskrives som:

- Baseret på (stokastisk) bruger ligevægt (alle bilister optimerer deres egen nytte)
- Multiple trafikant-klasser håndteres
- En udvidet statistisk assignment model, hvor trafik beskrives i intervallet, i stedet for et gennemsnit for modelperioden
- Implementeret vha. Successive Gennemsnits Metoder (Method of Successive Averages, MSA).

Detaljerne i hvordan en sådan metode kan implementeres, beskrives nærmere nedenfor.

### 2.1 Hoved-algoritme

Den klassiske MSA-algoritme for statistiske assignment modeller kan kort beskrives således:

Trin 1: Initialisering

Trin 2: Opdatering af rejsemodstande

Trin 3: Alt-eller-intet udlægning af ture

Trin 4: Trinlængde fastlægges:  $\xi(n) := 1/n$  (eller ved optimering)

Trin 5: Opdatering:

$$Ta(n) := (1 - \xi(n)) \cdot Ta(n-1) + \xi(n) \cdot Ta(tmp), \forall a$$

Trin 6: Stop kriterium eller sæt  $n:=n+1$  og gå til trin 2

Denne metode er i grove træk en iterativ metode, hvor der ud fra gentagne alt-eller-intet trafik-udlægning dannes et løbende gennemsnit. Efter et vist antal skridt, eller hvis et givet stop-kriterium er opfyldt, stoppes beregningerne.

Denne overordnede algoritme modificeres til følgende:

1. For hvert turformål, tidsinterval og startzone
  - a. Korteste vej beregnes, hvor generelle rutevalgspræference opdateres løbende, og tidsafhængigt.
  - b. Ruten gennemløbes baglæns, og følgende opdateres:
    - Ruten justeres med hensyn lukkede tidsintervaller
    - Først Ind Først Ud (FIFO) checkes, dvs. at køer overholdes
    - Trafikmængder opdateres (vha. løbende MSA-gennemsnit)
    - Kapaciteten af de berørte kanter genberegnes for hvert tidsinterval
2. Alle ikke benyttede kanter opdateres vha. MSA

### 3. Stop hvis stop-kriterium er sandt ellers gå til 2

De to mest komplekse problemer i algoritmen ovenfor, er hvorledes man håndterer udlægningen og opdatering af trafikstrømmene henover tidsintervaller, samt beregning af korteste vej i et netværk med tidsafhængige vægte. Begge dele behandles mere detaljeret nedenfor.

## 2.2 Rute-valgs præferencer

Som beskrivelse af bilisternes rutevalgspræference benyttes i første omgang det udtryk og de koefficienter der blev estimeret i Havnetunnel-modellen (Nielsen & Frederiksen, 1999), hvor nytte  $U$  af alternativ  $i$  var:

$$U_i = \beta_c \text{Omkostning} + \beta_{\text{ff}} \text{Tid}_{\text{ff}} + \beta_{\text{tcon}} \text{Tid}_{\text{con}} + \varepsilon_i$$

Hvor omkostning kan være længde • kr./km. eller f.eks. road pricing,  $\text{tid}_{\text{ff}}$  er fri køretid og  $\text{tid}_{\text{con}}$  er ekstra køretid som følge af trængsel. Dertil kommer et fejllid, der i anvendelsessammenhæng er summen af fejllid på strækninger  $a$  (arcs) langs ruten  $R$ :

$$U_R = V_R + \varepsilon_R = \\ \beta_c \text{RoadPricing} + \beta_l \text{Length} + \\ \beta_{\text{ff}} \text{Time}_{\text{ff}} + \beta_{\text{tcon}} \text{Time}_{\text{con}} + \sum_{a \in R} \varepsilon_a$$

Der blev estimeret parametre for hver af de 5 trafikant-klasser: bolig-arbejde, erhverv, fritid, varebiler og lastbiler. Parametrene er også estimeret som stokastiske variable.

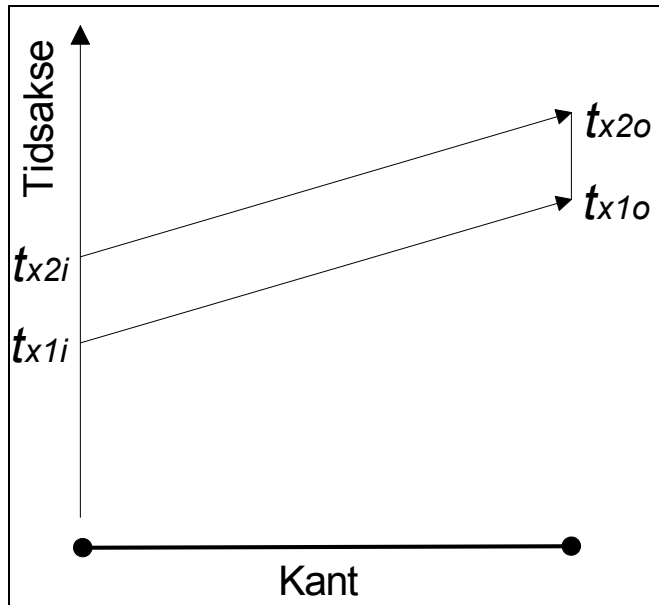
Modellen kan let generaliseres til en hvilken som helst lineær nyttefunktion, hvor de enkelte koefficienter også kan følge statistiske fordelinger. Med en kompakt notation kan nytten da beskrives ved:

$$U_i = \sum_j (\beta_j + \xi_j) X_{ji} + \varepsilon_i$$

Hvor  $\xi_j$  er fordelingen af koefficient  $j$  for variabel  $X_j$ . I estimationskontekst formuleres alternativerne typisk som diskrete valg. Mens de i anvendelseskontekst er ruter, der hver er summen af variabler (tid, længde, etc.) og et kant-afhængigt fejllid.

## 2.3 Beskrivelse af trafik i intervaller

Idéen i modellen er, at tidsdimensionen i trafiknettet beskrives tabellarisk for den enkelte kant i vejnettet, hvor dagen segmenteres i en række tidsintervaller. Hver kants tidsinterval  $x$  er fast ved indkørsel  $i$  (in) på kanten  $[t_{x1i}; t_{x2i}]$ , mens slutintervallet er trafikafhængigt og afhængig af længde og hastighed på kanten  $[t_{x1o}; t_{x2o}]$ , hvor  $o$  angiver udkørsel (out) fra kanten. Se figur 1.



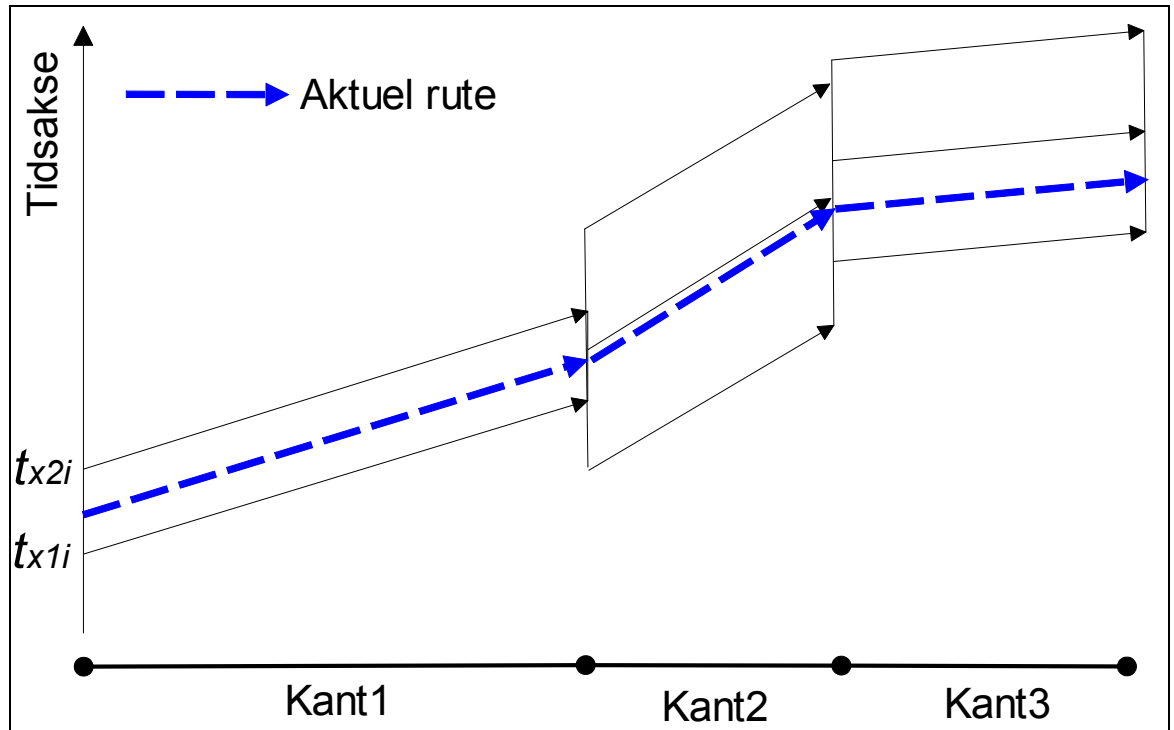
Figur 1 Tidssegmentering af kant

## 2.4 Trafikudlægning

Trafikken lægges i den enkelte iteration ud på trafiknettet som trinvis udlægning af segmenter af OD-matricen som funktion af tidspunktet.

OD matricen har således formen  $T_{ijkx}$  hvor  $i, j$  er fra til centroide,  $k$  er eventuelt turformål og  $x$  er angivelsen af det relevante tidsinterval.

Det givne starttidspunkt er fastlagt som midtpunktet i tidsintervallet (fra  $x1$  til  $x2$ ), men ruten benytter derefter aktuelle tidspunkter for de enkelte kanter i grafen (se eksemplet i figur 2).



Figur 2 Eksempel på rute gennem kanter/tidsintervaller.

Hver gang korteste vejs algoritme når en knude, søges relevante tidsintervaller for de næste kanter i beregningsnetten. Hvis kapaciteten for en kant er opbrugt, må næste tidspunkt benyttes.

## 2.5 Opdatering af tidsafhængige kanttider

Ved backward pass (udlægningen af trafikken i beregningsnettet) opdateres trafikmængder og hastigheder, og derved tidspunkterne for out-tidspunktet for kanten. Denne operation er tidskrævende, men essentiel for at det hele virker – i statiske assignmentmodeller foregår dette i løkken uden for AON. Men i en dynamisk model vil dette give anledning til, at ruter, der senere er ikke mulige, ofte vil blive valgt (især i de indledende iterationer). Ved en inkremental udlægning af trafik som funktion af tidsintervaller, og med simultan MSA opdatering, reduceres dette problem.

Opdateringen foretages som (forklaring på notation følger):

$$(1) T_{akx(tmp)} = T_{akx(tmp)} + T_{ijkx}$$

$$(2) T_{akx(n)} := (1 - \xi_{(n)}) \cdot T_{akx(n-1)} + \xi_{(n)} \cdot T_{akx(tmp)}$$

$$(3) t_{akx(n)} = f(t_{(0)ax}, T_{ax(n)}) \quad \forall k$$

(ad 1)  $T_{akx(tmp)}$  er strækningsbelastningen for den del af OD-matricen  $T_{ijk}$  for den pågældende iteration, der indtil videre er udlagt. Det bemærkes, at alle OD-relationer ( $i, j$ ) og trafik fra tidsintervaller i matricen før det pågældende tidspunkt kan have bidraget til strækningsbelastningen. I den aktuelle iteration, lægges en bestemt række af turmatricen for et bestemt turformål  $k$  og med ture startende på et bestemt tidspunkt  $x$

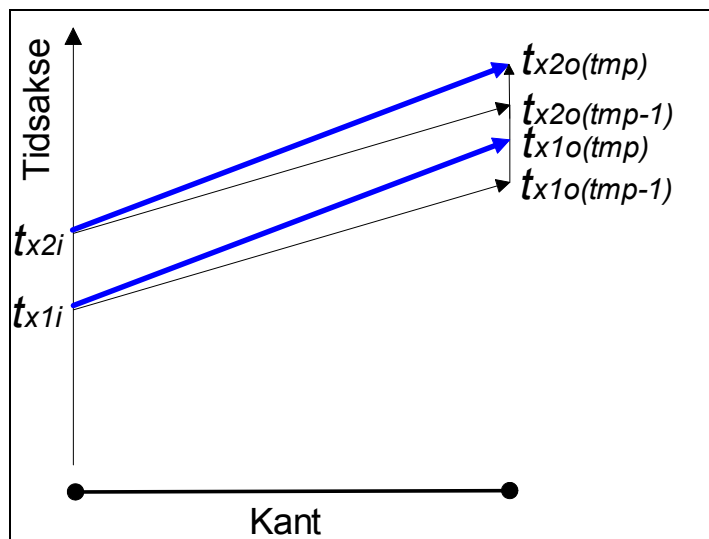
(f.eks. 7.30) ud på nettet. Men kantopdateringen  $T_{akx(tmp)}$  sker for det tidspunkt  $x$  (f.eks. 8.44,14, hvor den pågældende tur har nået den pågældende kant).  $x$  for hhv. matrix og kant er altså ikke ens. ( $tmp$ ) starter med at være NULL før den pågældende iteration (der er endnu ikke lagt trafik ud her).

(ad 2)  $T_{akx(n)}$  er den MSA-opdaterede strækningsbelastning. Denne opdateres således flere gange – nemlig i forbindelse med udlægningen af den enkelte rute. At opdateringen kan gennemføres simultant skyldes, at der holdes styr på den temporære belastning i den særskilte  $tmp$  variabel. Til gengæld er det ikke nødvendigt at opdatere hele nettet, men kun for de kanter, der ligger langs ruten, hvorved beregningskompleksiteten ikke er meget større end netudlægningen selv (sammenlignet med endnu en ydre løkke i de fleste dynamiske assignmentmodeller).

Da opdateringen sker i forbindelse med den enkelte rute/trafikudlægning, vil resultatet efter den sidste rute være, at alle kanter hvor der har været mindst én rute er MSA-opdateret. Men ikke alle kanter i nettet er opdateret. Derfor er det efterfølgende nødvendigt at foretage en opdatering af trafikmængder for kanter, der ikke er opdateret, dette inkluderer nedjustering af hastigheden og efterfølgende FIFO check (check af at køer overholdes, og der ikke foretages umulige overhalinger).

(ad 3) Da de forskellige turformål påvirker rejsehastighederne for hinanden skal alle rejsetider opdateres – ikke alene for det aktuelle turformål, men også for de turformål det påvirker og visa versa.

Figur 3 illustrerer princippet i tidsopdateringen for en enkelt kant, hvor der er lagt mere trafik ud.



Figur 3 Princippet i tidsopdateringen

## 2.6 FIFO Check

Hvis FIFO betingelsen (First in First Out) for de efterfølgende tidspunkter på en kant er overtrådt, opdateres disse. FIFO må checkes for alle turformålene ved udlægning af ruten fra et bestemt turformål (index  $k$  er således udeladt i sektion 2.3).

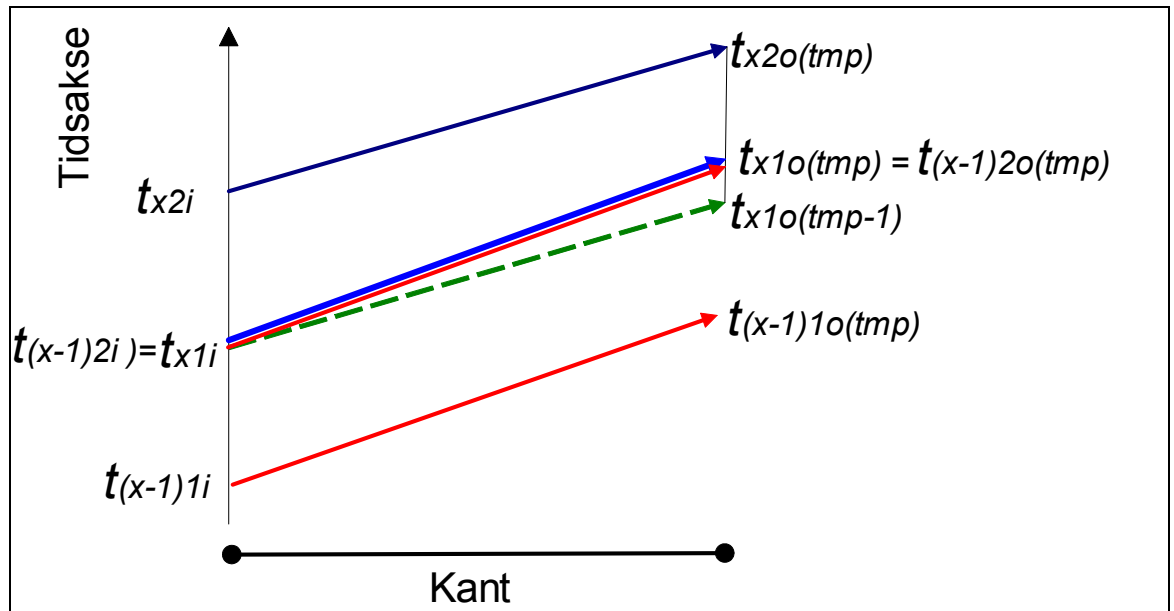
Først checkes om FIFO for kanten for det pågældende tidsinterval er overtrådt, dvs.:

Hvis  $t_{x1o} < t_{(x-1)2o}$  så  $t_{x1o} := t_{(x-1)2o}$

Figur 3 illustrerer denne operation, hvor  $tmp$  refererer til tidelementerne efter opdatering og  $tmp-1$  før opdatering.

Opdateringen kan endvidere påvirke sluttidspunktet i ekstreme tilfælde:

Hvis  $t_{x2o} < t_{x1o}$  så  $t_{x2o} := t_{x1o}$



Figur 4 Mekanisme ved check af FIFO for tidligere tidsinterval

Det bemærkes, at "bredden i tid" for tidsintervallet langs kanten kan blive asymmetrisk, og at den gennemsnitlige passagetid derfor skal udregnes ud fra middelværdien af ind og out tidspunktet, dvs.:

$$t_x = (t_{x2o} + t_{x1o})/2 - (t_{x2i} + t_{x1i})/2$$

Dernæst checkes om FIFO for kant tidsintervallet efter det pågældende tidsinterval er overtrådt, dvs.:

(1) Hvis  $t_{(x+1)1o} < t_{x2o}$  så  $t_{(x+1)oi} := t_{x2o}$

Det er altså det efterfølgende tidsinterval, der opdateres, *ikke* det tidsinterval, hvor den aktuelle tur forløber (er der kø spærres en bil for de biler, der kommer senere).

Opdateringen kan ydermere påvirke sluttidspunktet på de efterfølgende tidsintervaller i ekstreme tilfælde:



(2) Hvis  $t_{(x+1)2o} < t_{(x+1)1o}$  så  $t_{(x+1)2o} := t_{(x+1)1o}$

Da denne opdatering kan påvirke følgende tidsintervaller fortsættes operation (1) og (2) for  $X := X+1$ , så længe, der har været anledning til opdateringer.

## 2.7 Test af kapacitetsrestriktioner ved generering af træ for korteste vej

Generelt effekterne af kapacitetsrestriktioner vha. den klassiske BPR-formel (sammenhæng mellem trafikmængde og hastighed), evt. kombineret med en fast lav hastighed når kapacitetsudnyttelsen når 100%; da kapacitetsgrænsen ved dette punkt er karakteriseret ved, at den maksimale trafik afvikles, vil det være mere korrekt at lade hastigheden gå mod uendeligt over dette punkt. F.eks. ved en justering af BPR-formlen over kapacitetsgrænsen.

Men derudover vil den enkelte kant have en max-kapacitet, der kendetegner total kø, når denne grænse nås, lukkes kanten for efterfølgende trafikudlægninger (indtil MSA evt. har neddæmpet trafikken igen til under denne grænse). Testet af om kapacitetsgrænsen er nået udregnes som:

$$\frac{\sum_k l_k \cdot E(T_{akx})}{\#spor} \leq l_a \quad , \quad \text{hvor} \quad E(T_{akx}) = \begin{cases} T_{akx} & , \quad \text{hvis } (t_{akx2i} - t_{akx1i} + t_{akx20} - t_{akx10})/2 < t_{akx} \\ T_{akx} := T_{akx} \cdot \frac{t_{akx}}{(t_{akx2i} - t_{akx1i} + t_{akx20} - t_{akx10})/2} & , \quad \text{ellers} \end{cases}$$

Eller mere kompakt:

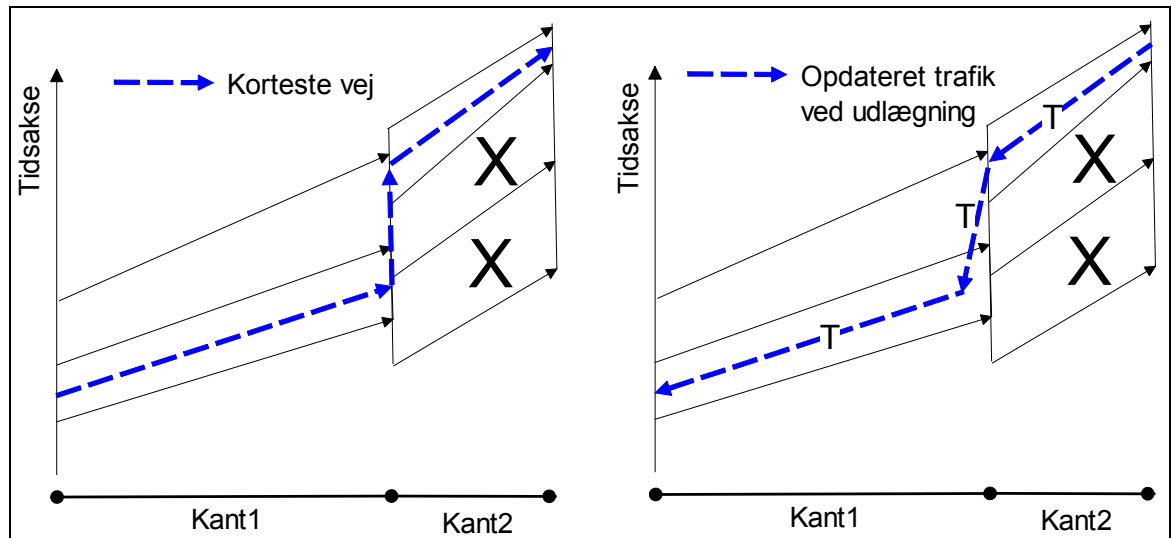
$$\frac{\sum_k l_k \cdot T_{akx} \cdot \min \left\{ 1, \frac{t_{akx}}{(t_{akx2i} - t_{akx1i} + t_{akx20} - t_{akx10})/2} \right\}}{\#spor} \leq l_a$$

For hver køretøjklasse udregnes klassens belastning af kanten som længden (inkl. afstand mellem køretøjer) multipliceret med antallet af det gennemsnitlige antal køretøjer på kanten. Den samlede køretøjslængde divideres med antal kørespor. Resultatet skal være mindre end vejkantens længde, for at bilerne kan være der. Hvis tidsintervallet for kanten er større end den tid, det aktuelt tager at gennemkøre kanten, skal der korrigeres for, at bilen kun optager kapacitet noget af tiden. Dette sker i udregningen af  $E(T_{akx})$ .

Disse formler sikrer en nogenlunde rimelig beskrivelse af trafikafviklingen. En mere korrekt beskrivelse vil være baseret på flowteori, hvilket dog også øger beregningsomfanget. Forskellen mellem de to tilgange vil dog blive testet senere. Ydermere kunne trafikbeskrivelsen ske via meso-simulering, som i University of Montreals model. Men dette vil være meget regnetungt, og ikke funktionelt på et net af samme størrelse som i AKTA/Trængselsprojektet.

## 2.8 Udlægning af trafik under hensyntagen til venten i knuder

Hvis der i en given knude ventes på ledig kapacitet på frabringer kanten, antages det, at denne venten sker på tilbringerkanten. Trafikken ad en given rute kan derfor spærre kapacitet på flere tidsintervaller på tilbringerkanten (tilbagestuvning ved kødannelse). Alle disse tidsintervaller på tilbringerkanten tillægges derfor trafikken. En tilsvarende opdatering vil gælde, hvis krydsforsinkelser implementeres – altså at kø fra en svingbevægelse kan stuve tilbage på den tilstødende tilkant.



**Figur 5** Trafikopdatering af kanter ved lukkede frabringerkanter (tilbagestuvning ved kø). X markerer tidsintervaller, hvor der ikke er mere kapacitet. Den ankommende rute fra kant 1 må derfor vente 2 tidsintervaller foran kant 2, før det er muligt at kunne komme videre ad kant 2t. Ved netudlægning lægges trafik ud på begge tidsintervaller på kant 1 (markeret med T på højre side af figuren), men kun på den benyttede tidsinterval på kant 2.

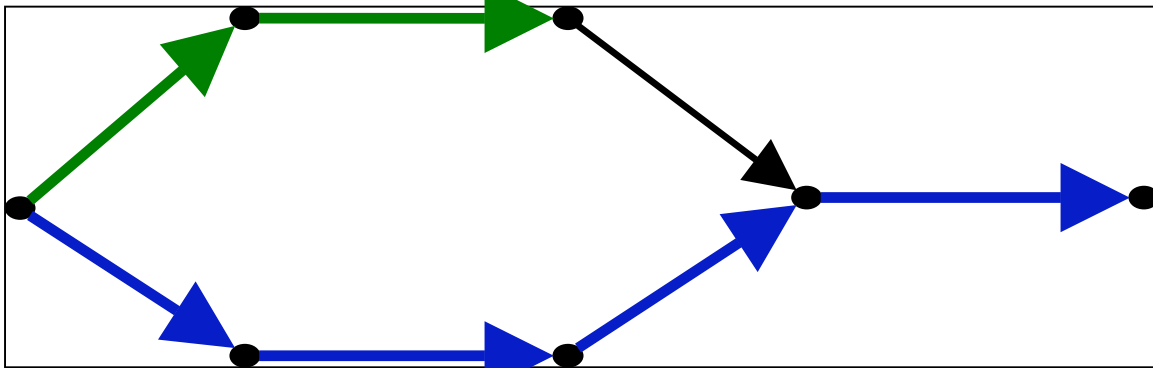
I princippet vil der kunne være ruter, der ankommer senere end den aktuelle tilbringerkant (kant 1), og som kører videre ad en anden frabringerkant end kant 2 på et tidligere tidspunkt end den viste rute (forudsat at kapaciteten på kant 1 ikke er nået).

Dette vil på kanter, hvor der ikke er overhalingsmulighed (enkeltsporede kanter, eller 2-sporede veje med fuldt optrukne linier) i princippet kunne overtræde FIFO. Imidlertid vil en sådan tilkant hurtigt fyldes op, så kapaciteten nås.

## 3 BEREGNING AF KORTESTE RUTER I ET DYNAMISK NETVÆRK

En væsentlig forudsætning for at opnå tilfredsstillende regnetider med modellen, er at kunne beregne korteste veje på en effektiv måde. I statiske modeller er det ikke noget problem, idet vægtene på kanterne i nettet er faste. Til denne situation kan benyttes en række simple og effektive korteste vej algoritmer, hvor af den mest kendte formodentligt

er *Dijkstras Algoritme*. En korteste rute til en knude, er også en del af den korteste rute for enhver efterfølgende knude der nås via en rute, der passerer den pågældende knude,

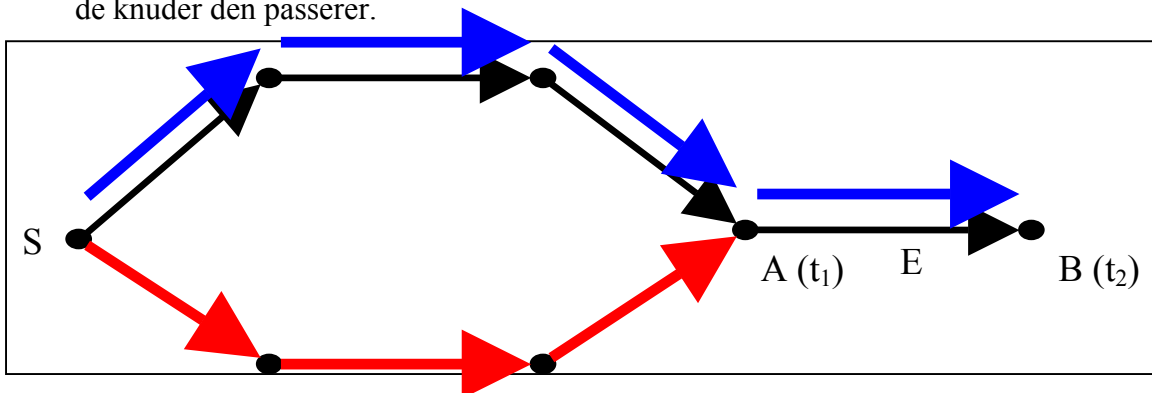


**Figur 6** Et klassisk netværk med faste vægte. En korteste rute til en knude er lig den korteste rute til den forrige knude + en kant

Anderledes forholder det sig med beregning af korteste vej i et net, hvor vægtene er tidsafhængige, det vil sige at vægten på hver kant i princippet er en funktion af ankomsttidspunktet. Dersom den korteste vej blot var den der gav tidligste ankomst, var dette ikke noget større problem. Men i det aktuelle tilfælde er vægten på hver kan desværre en generaliseret omkost, der består af en række sammenvejede omkostningskomponenter, f.eks. rejsetid, køtid, roadpricing pris og længde. Og den korteste rute mellem to punkter, er den rute der minimerer den samlede generelle omkostning, defineret som summen af omkostningerne for kanterne i ruten.

Dette har flere konsekvenser:

- Enhver rute indeholder nu en tidslig dimension.
- Den rute der ankommer først til en knude, er ikke nødvendigvis den med den korteste generelle omkostning
- En korteste rute til en knude, overlapper ikke nødvendigvis med de korteste ruter til de knuder den passerer.



**Figur 7** I et netværk med generaliserede tidsafhængige vægte, har de korteste vej til hhv. A og B ikke nødvendigvis noget til fælles.

Ofte hænder ovennævnte problemstilling i begyndelsen af myldretiden. Her er der tilfælde, hvor den langsomste rute til en knude har den bedste nytte (mindste generaliserede nytte). Det kan f.eks. skyldes:

- At den langsomste rute er kortere (f.eks. byvej versus omfartsvej)
- At den langsomste rute er billigere (ikke-betalingsvej, etc.)
- At den langsomste rute har mindre trængsel; mange bilister foretrækker at køre længere ruter for at undgå trængsel, idet de ikke kan lide at køre i kø, eller endog fordi de oplever det som om det er hurtigere at køre den langsomme rute

Imidlertid kan det være, at der senere ad ruten optræder en kø, som den hurtigere rute til knuden ville nå at undgå. Og derfor er den alligevel bedre.

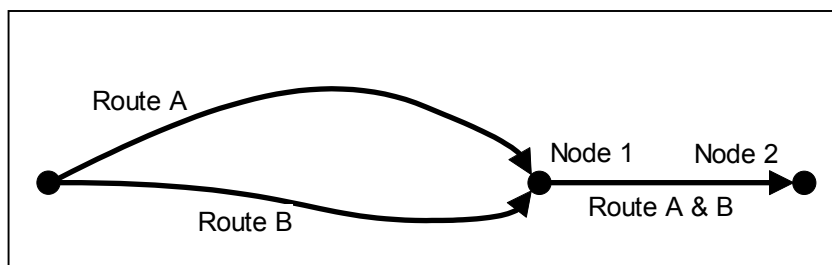
Ikke dominerede alternative ruter som i dette eksempel optræder typisk i begyndelsen af myldretiden.

Typisk i slutningen af myldretiden, kan det imidlertid forekomme, at den hurtigste rute med den bedste nytte (laveste rejsemodstand) til en given knude alligevel ikke er den optimale videre i vejnettet. Dette kan f.eks. skyldes:

- Det tager næsten samme tid til den endelige destination p.g.a. en kø op til en flaskehals, der er ved at blive afviklet (det ekstreme tilfælde er en færgekø, hvor det er lige meget om man kommer ½ time før eller 5 min. før, hvis man kommer med samme færge)
- Lavere nytte i den videre rute for meget trængselshadende bilister
- Eller lavere nytte i den videre rute på grund af trængselsbetaling (road pricing)
  - ⇒ Dette gør ruterne ikke-optimal samlet set, selvom den indtil en bestemt knude både var hurtigst og havde bedst nytte

Et eksempel på sidstnævnte kan være en bilist med følgende nyttefunktion (ren vejafgift på trængsel på 1 kr. per minuts trængsel):

$$\text{tid}_{\text{fr}} \cdot 40 \text{ DKK/H} + \text{tid}_{\text{con}} \cdot 60 \text{ DKK/H} + \text{længde} \cdot 0,7 \text{ DKK/km.} + \text{time}_{\text{cc}} \cdot 1 \text{ DKK/min}$$



**Figur 8** Eksempel på rutevalgsproblem.

Tabel 1 viser beregningen af rejsetid og nytte for de to ruter på figur 8. Til knude 1 er rute A hurtigst (21 minutter versus 25), og den har samtidig mindst rejsemodstand (størst nytte), nemlig 21 kr. versus 23. Den dominerer således rute B. Men i det videre forløb, hvor de to ruter følger samme strækning, er A stadig hurtigst (36 minutter versus 37), men har lavere nytte, fordi en større andel af tiden har forløbet i trængsel, ligesom der har været roadpricing i tilsvarende større omfang. Rute B har således bedst nytte (40,5 kr.

rejsemodstand mod 44,5), og må således foretrækkes efter den pågældende bilists nyttefunktion.

	Rute	Fri køretid	Trængsels tid	total tid	længde	Road pricing	Nytte
Rute til	A	21	0	21	10	0	-21,0
Knude 1	B	25	0	25	9	0	-23,0
Rute fra	A	8	7	15	6	7	-23,5
Knude 1 til 2	B	8	4	12	6	4	-17,5
Total til	A	29	7	36	16	7	-44,5
knude 2	B	33	4	37	15	4	-40,5

**Tabel 1** Beregning af tid og generaliseret omkostning (nytte) for to ruter.

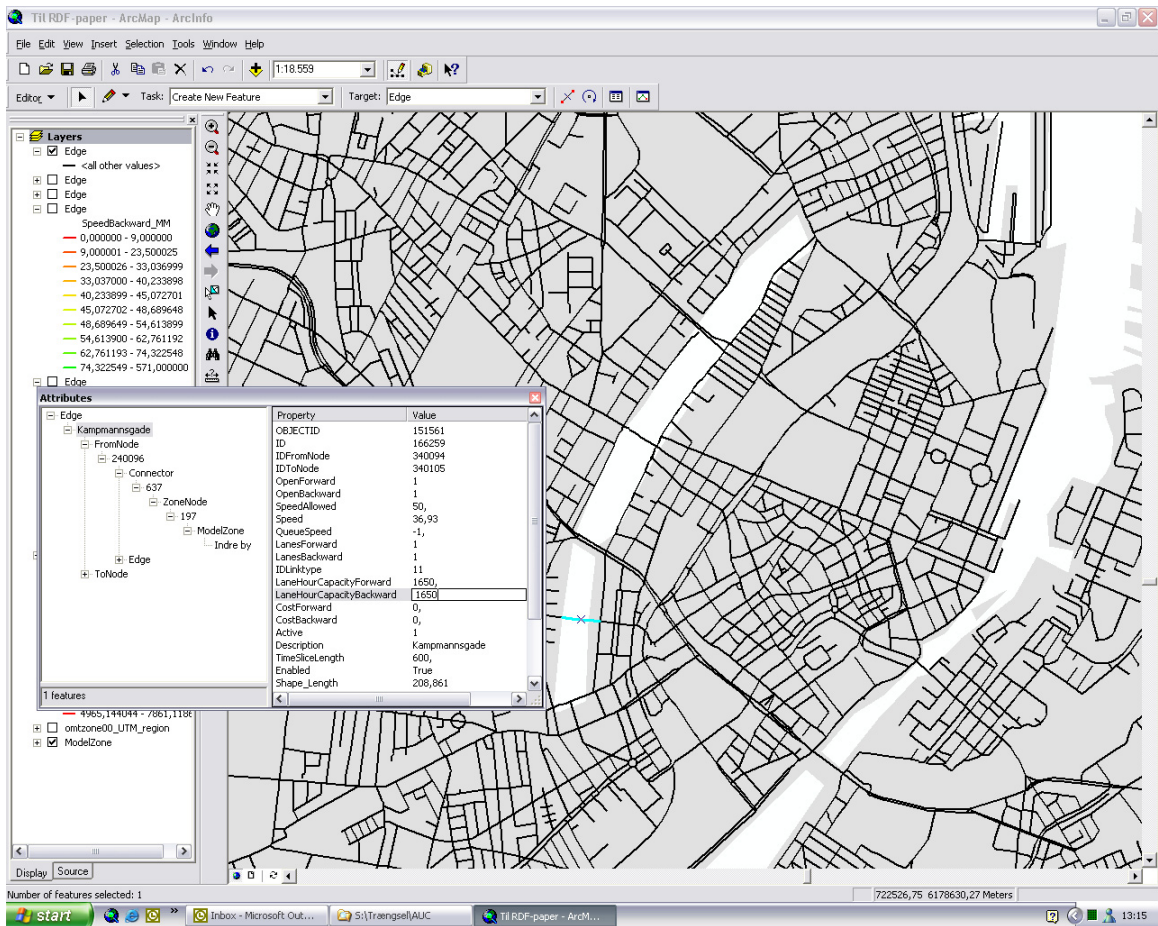
I første omgang er ruteberegning forsøgt løst med en iterativ algoritme der tilnærmelsesvist finder den korteste vej. Men da denne tilgang er relativ tidskrævende, er korteste vej beregning nu ved at blive ændret til en tilgang, der bygger på følgende elementer:

- Simultan afsøgning af ruter med tidslig dimension, fra en knude, til alle andre
- Hvor ruter krydser, beholdes kun ruter efter et heuristisk kriterium, f.eks. de tre hurtigste og de tre med bedst nytte, eller ruter afvises hvis de afviger mere end en bestemt værdi i tid eller nytte.

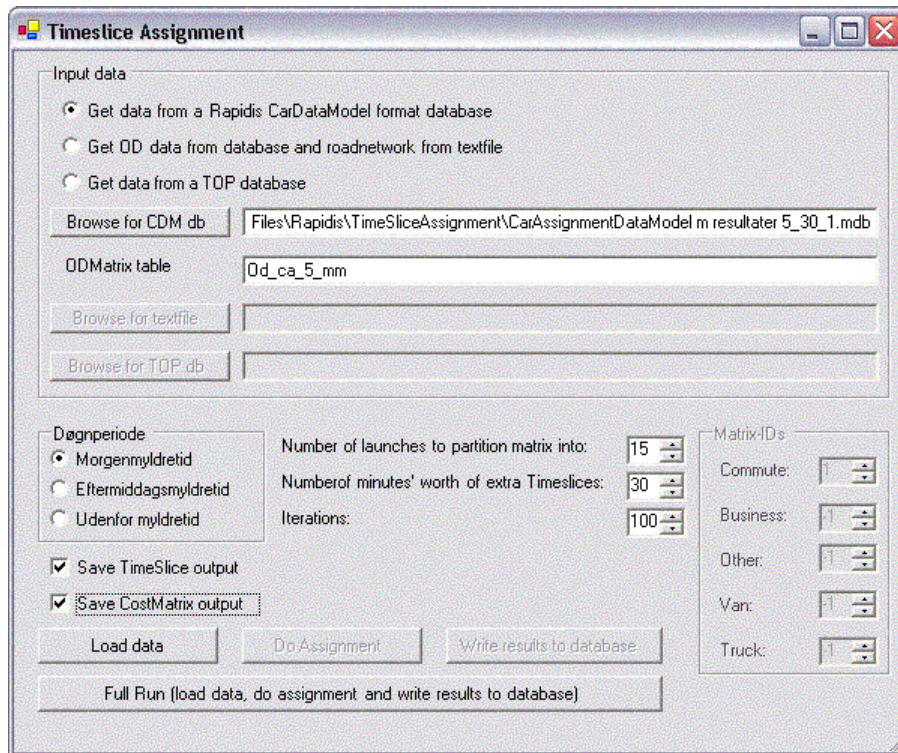
## 4 TEKNISK IMPLEMENTERING

Meget kortfattet kan den tekniske implementering beskrives således:

- Data håndteres i ArcGIS, vha. en datamodel mht. relationer, netværk og regler for datas korrekthed
- Der er implementeret et yderst simpelt bruger-interface, de fleste parametre specificeres i databasen.
- Beregningsprogrammet er implementeret i C#, under Microsofts .Net platform.



Figur 9 Data håndteres vha. ArcGIS



*Figur 10 Et simpelt bruger-interface er implementeret*

## 5 DATAGRUNDLAG

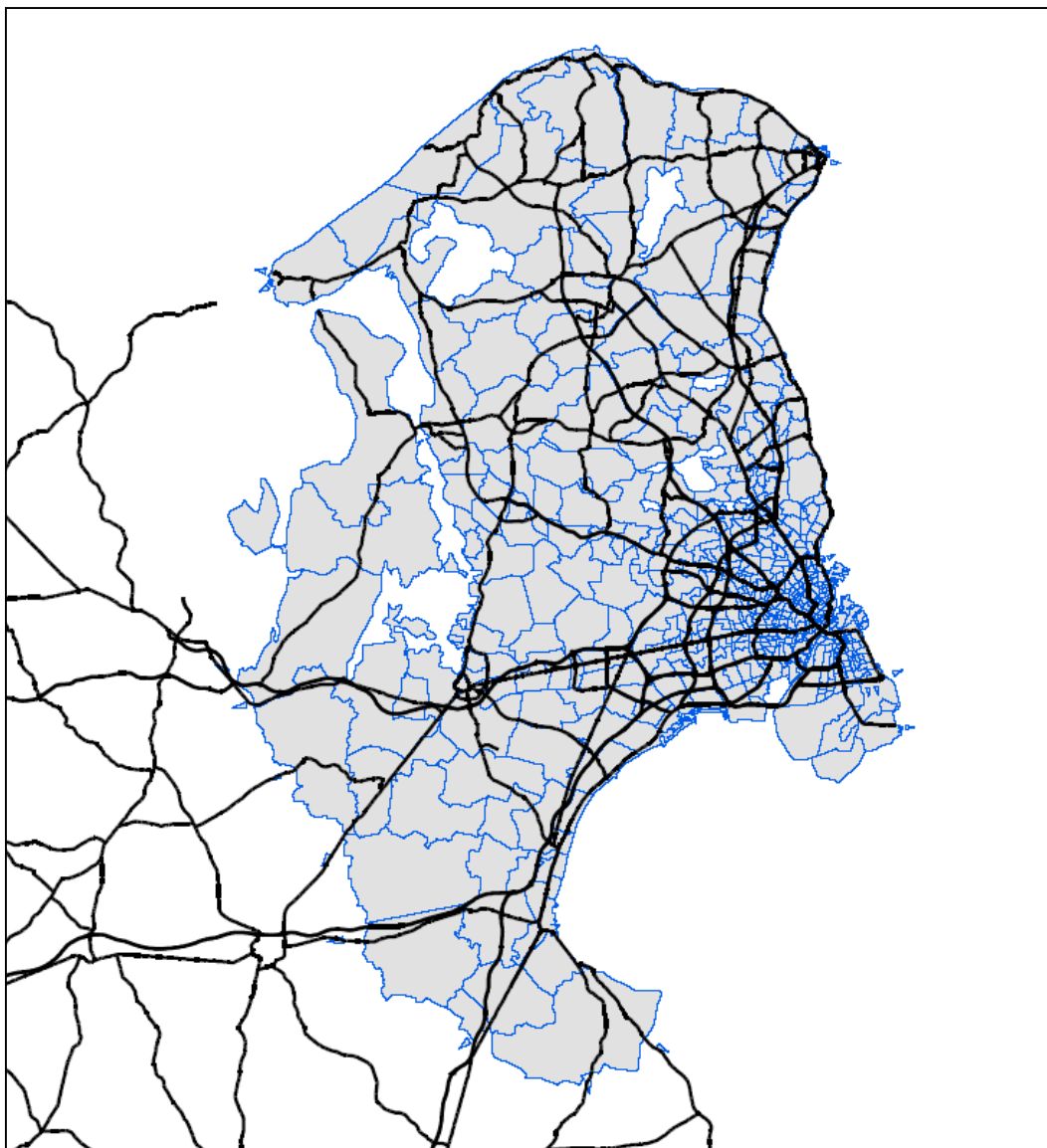
Som datagrundlag benyttes modelzoner og turmatricer fra OTM<sup>1</sup> 4.0 modellen (Ørestadsselskabets Trafikmodel, opbygget af TetraPlan A/S). Denne indeholder 618 modelzoner, 5 trafikant-klasser og 5 døgn-perioder.

Som vejnet benyttes et model-vejnet opbygget på CTT, på basis af KRAK's vejnet, hvor mindre veje og stier er frasortet. Dette modelnet indeholder 25.000 strækninger.

Grunden til at dette net blev valgt var, at der derved kan sammenlignes med køretidsmålinger i det Københavnske AKTA projekt (Nielsen, 2003a), der via trængselsprojektet linkes op til målinger af trafik og hastighed på Vejdirektoratets TRIM-net og Københavns Kommunes TrafMIL net. Dette sikrer et godt grundlag til kalibrering og validering af metoden. Forskellige artikler ved årets trafikdage beskriver de to projekter.

Som standard tidssegmentering for modellen benyttes 10-minutters intervaller.

<sup>1</sup> se evt. <http://www.m.dk/trafikprognose.pdf> for en overordnet beskrivelse



*Figur 11 OTM's modelzoner benyttes*

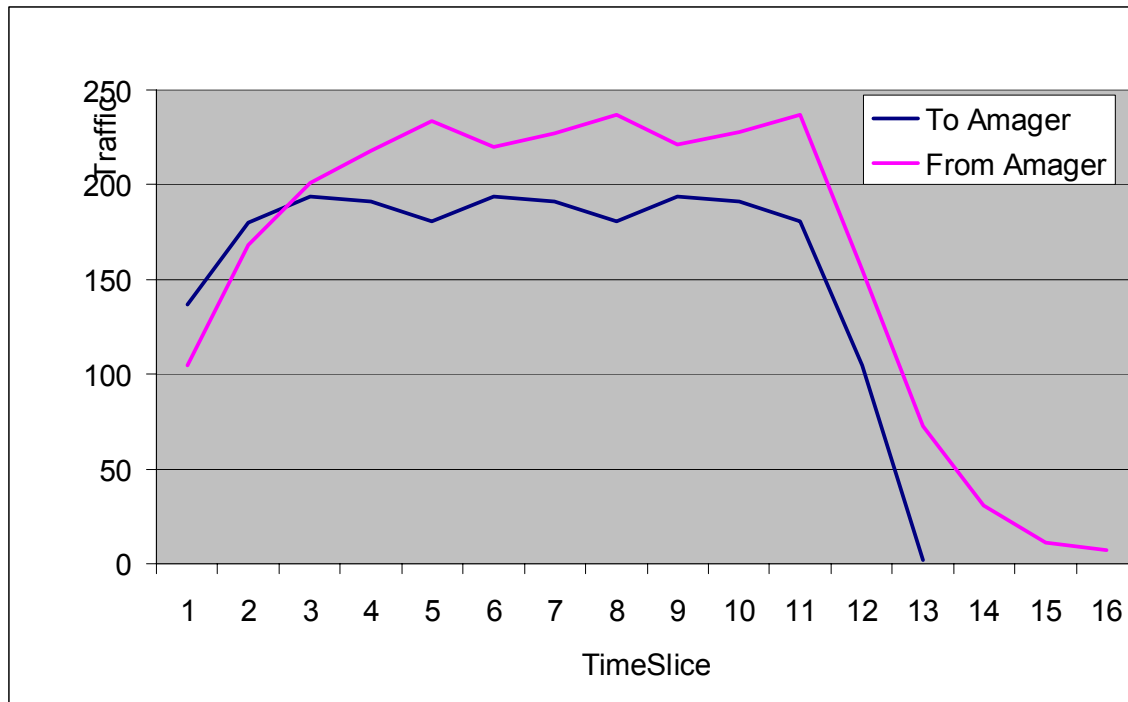
## 6 EKSEMPLER PÅ RESULTATER

Modellen er ved at blive testet, og en færdigkalibreret model kan således ikke præsenteres. Det skyldes også, at den er udviklet i et forskningsprojekt, der har fokuseret på metodeudviklingen, og implementeringsmæssige spørgsmål, mens der ikke har været ressourcer til en fuldskala kalibrering og validering af modellen. Imidlertid har det som nævnt ved hjælp af eksisterende datakilder og GIS været muligt at etablere et fuldskala modelnet med tilhørende zonestruktur og turmatricer, hvilket har dannet grundlag for de tests, der præsenteres i det følgende.

Den første test lagde simpelt en morgenmyldretidsmatrix (7.00-9.00) ud på vejnettet i Hovedstadsområdet. Matricen blev segmenteret i 10-minutters intervaller, der hver blev tildelt 1/12 trafik. Som det ses stiger trafikken i starten af myldretiden. Dette skyldes, at



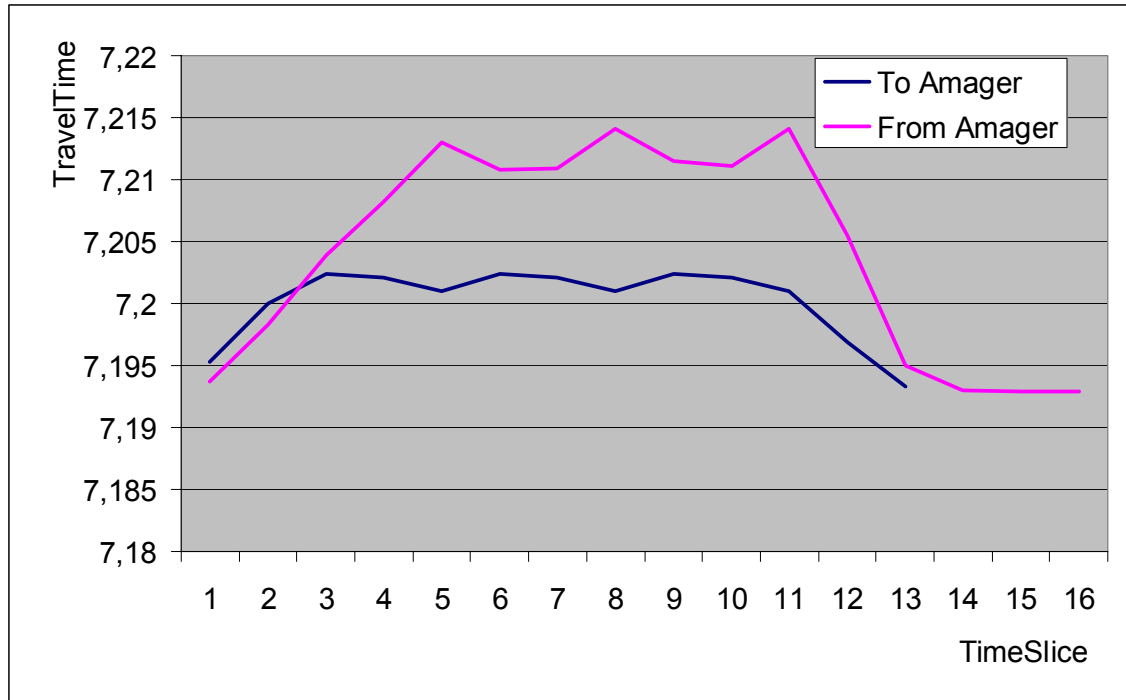
en del af turene ikke når Knippelsbro før et stykke henne i myldretiden f.eks. en tur fra Dragør til centrum, når ikke til Knippelsbro på de første 10 min. Efter en vis tid nås et rimeligt stabilt niveau (hvilket i dette simple tilfælde skyldes at tidsslicene er lige store over myldretiden). I slutningen af intervallet er der stadig lidt trafik tilbage, der passerer Knippelsbro.



**Figur 12** Eksempel på udlagt trafik, i dette tilfælde Knippelsbro.

Figur 13 viser de tilsvarende resultater, men her rejsetiden i de to retninger. I retningen mod København nås kapacitetsgrænsen næsten, hvor rejsetiderne her stiger temmelig meget. Dette er netop intentionen med en dynamisk model – at beskrive variation af såvel efterspørgsel som trafikmængder på strækningsniveau og hastigheder over tid.

Overheadene fra AUC – og i særlig grad fra det efterfølgende seminar om dynamiske vejvalgsmodeller på CTT – rummer også en billedsekvens, der viser trafikbelastninger som funktion af tiden i hele Hovedstadsområdet i samme test (de kan rekvireres via hovedforfatteren, oan@ctt.dtu.dk). Generelt viste de, at modellen selv uden kalibrering og med konstante tidsslices over myldretiden kunne give et ganske godt billede af trafikafviklingen i myldretiden. Eksempelvis kunne køerne ind mod byen på de store indfaldsveje godt ses.



*Figur 13 Rejsetider for samme eksempel som figur 12.*

## 7 STATUS OG KONKLUSION

Artiklen har præsenteret første version af en dynamisk vejvalgsmode, der kan køre i Hovedstadsområdet på et vejnet af samme høje detaljeringsgrad som eksisterende statistiske modeller (f.eks. Ørestadstrafikmodellen). Modellen har følgende trafikfaglige karakteristika:

- Den finder en stokastisk brugerligevægt, som tilnærmer dynamiske effekter via segmentering af tid i tidsslices. Herved kan modellen være nyttebaseret og den tager højde for overlappende ruter
- Den tager højde for rejsetidsvariationer på strækninger over tid som funktion af trafikmængder
- Den overholder first ind først ud betingelsen på strækningsniveau
- Den kan modellere køer og tilbagestuvning
- Den beskriver variation i rutevalg som funktion af turtidspunktet
- Den er ikke så detaljeret som mikro simuleringsmodeller, men har et tilstrækkeligt detaljeringsniveau til vurdering af trængsel i Hovedstadsområdet. Men i delområder kan den metodisk uproblematisk linkes til mikrosimuleringsmodeller (men i praksis kræver det et stort programmeringsarbejde)

I praksis har modellen følgende karakteristika:

- Den har håndterbare krav til data og kalibrering; således kunne en test-case etableres forholdsvis overkommeligt. Hvis matricerne ønsket estimeret på tidssliceniveau er dette et stort arbejde, men dette er et selvstændigt problem ud over udvikling af en rutevalgsmodel-
- Den kræver meget RAM, hvilket dog er inden for mulighedens grænser på moderne PC'ere
- Den er regnetung, men per tidsslice ikke voldsomt mere end en statisk model (ca. 2-3 gange langsommere, hvilket skal multipliceres med antal tidsslices i matricen). Dette er ikke uoverkommeligt i forhold til den meget højere detaljeringsgrad, der opnås dermed.
- Den kan håndtere langt større og mere detaljerede trafiknet end andre dynamiske vejvalgsmodeller.

Før modellen kan benyttes til praktisk planlægning forestår imidlertid et betydeligt praktisk og udviklingsmæssigt arbejde:

- Først og fremmest skal der arbejdes mere med at formulere flow-teoretiske betingelser, når trafikken når nær kapacitetsgrænsen. Der er her tale om trafikteknisk forskning, snarere end forskning i rutevalgsmodellen, idet den eksisterende algoritme kan håndtere de flowformler, man måtte ønske.
- Kriterier for at løse korteste-vej beregning under hensyntagen til såvel tid som nytte skal viderebearbejdes og testes. Herunder kan rutevalgskriterierne også estimeres ud fra observerede ruter i AKTA (Nielsen, 2003b).
- Der skal arbejdes med at tilføje stokastisk rutevalg i praksis
- Der skal evt. tilføjes modellering af krydsforsinkelser, f.eks. efter samme formler og metoder som i Ørestadsmodellen (Nielsen, m.fl. 1997).
- Modellen mangler at blive kalibreret og valideret på et praktisk case, f.eks. det her præsenterede for Hovedstadsområdet.

## 8 REFERENCER

M. Florian, M. Mahut and N. Tremblay, 2001, A Hybrid Optimization-Mesoscopic Simulation Dynamic Traffic Assignment Model, **Proceedings of the 2001 IEEE Intelligent Transport Systems Conference, Oakland**, pp. 120-123.

Nielsen, Otto Anker; Frederiksen, Rasmus Dyhr & Simonsen, Nikolaj (1997). SUE-rute valgsmodel med krydsmodellering. **Trafikdage på AUC**. Supplementsbndind, pp. 263-274.

Nielsen, Otto Anker & Frederiksen, Rasmus Dyhr (1999). En stokastisk flerklasse vejvalgsmode med fordelte koefficienter for tid og omkostning. **Trafikdage på AUC**. Vol. 2. pp. 483-494.

Nielsen, Otto Anker (2003a). Analyse af trængselsdata og hastigheder baseret på GPS-data. **Trafikdage på AUC**.

Nielsen, Otto Anker (2003b). Estimation af rutevalgsmodeiller baseret på GPS-data. **Trafikdage på AUC**.

J.H. Wu, Y. Chen and M. Florian, 1998a, The Continuous Dynamic Network Loading Problem: A Mathematical Formulation and Solution Method, **Trans. Res.-B**, Vol. 32, No. 3, pp.173-187.

J.H. Wu, M. Florian, Y.W. Xu and J.M. Rubio-Ardanaz, 1998b, A projection algorithm for the dynamic network equilibrium problem, **Traffic and Transportation Studies, Proceedings of the ICTTS'98**, pp. 379-390, Ed. By Zhaoxia Yang, Kelvin C.P. Wang and Baohua Mao, ASCE

Y.W. Xu, J.H. Wu and M. Florian, 1998, An Efficient Algorithm for the Continuous Network Loading Problem: a DYNALOAD Implementation, in **Transportation Networks: Recent Methodological Advances**, Ed. By M.G.H. Bell, Pergamon Press.

Y.W. Xu, J.H. Wu, M. Florian, P. Marcotte and D.L. Zhu, 1999, Advances in the Continuous Dynamic Network Problem, **Transportation Science**, Vol. 33, No. 4, pp. 341-353.