
Stimulering af Umnus Numsus

Project Report
Group A319

Aalborg University
Det Teknisk-Naturvidenskabelige Fakultet
Strandvejen 12-14
DK-9000 Aalborg



AALBORG UNIVERSITY
STUDENT REPORT

**Det Teknisk-Naturvidenskabelige
Fakultet AAU**
Strandvejen 12-14
DK-9000 Aalborg
<http://cs.aau.dk>

Title:

Stimulering af Umnus Numsus

Theme:

Simulering

Project Period:

2. Semester 2016, P2

Project Group:

A319

Participant(s):

Benjamin Jhaf Madsen
Jacob Sloth Thomsen
Alexander Umnus
Kim Larsen
Lasse Fisker Olesen
Niclas Struntze Bach
Rasmus Thomsen

Supervisor(s):

Anders Mariegaard

Copies: 5

Page Numbers: 25

Date of Completion:

March 9, 2016

Abstract:

Pasta ipsum dolor sit amet rotini
pasta al ceppo lagane spaghetti
penne lisce tagliatelle conchiglie.
Stringozzi ricciutelle capellini lasag-
nette pennoni lasagnette trenette
croxetti capelli d'angelo mafalde
farfalle ziti strozzapreti rotini. Tagli-
atelle sacchetti pasta al ceppo
spaghetti foglie d'ulivo capunti
tortiglioni vermicelloni fettuccine.
Penne zita gnocchi manicotti sac-
chetti fiorentine corzetti pasta al
ceppo stringozzi vermicelli fusilli
lanterne sacchetti fettucelle. Fiori
tuffoli fiori tuffoli capelli d'angelo
sagnarelli chifferi tuffoli ricciolini
cavatappi.

Benjamin Jhaf Madsen
<bjma15@student.aau.dk>

Kim Larsen
<klars15@student.aau.dk>

Jacob Sloth Thomsen
<jsth15@student.aau.dk>

Lasse Fisker Olesen
<llese15@student.aau.dk>

Alexander Umnus
<aumnus14@student.aau.dk>

Niclas Struntze Bach
<nbach14@student.aau.dk>

Rasmus Thomsen
<rkth15@student.aau.dk>

Forord

Læsevejledning

Terminologi

Contents

Preface	iii
1 Indledning	1
2 Metode	2
I Problemanalyse	3
3 Teori	4
3.1 Dijkstras Algoritme	4
3.2 A* Algoritmen	5
4 Problembeskrivelse	9
4.1 Problemets Relevans	9
4.1.1 Økonomi	9
4.1.2 Forurening	10
4.1.3 Livskvalitet	10
4.1.4 Nødsituationer	10
4.2 Eksisterende Modeller	10
5 Teknologianalyse	12
6 Interessentanalyse	13
6.1 Kollektiv trafik	13
6.1.1 Nordjyllands Trafikselskab (NT)	13
6.2 Erhvervskørsel	13
6.2.1 Transport og fragt	13
6.2.2 Taxiselskaber	13
6.3 Privatkørsel	14
6.3.1 Myldretidstrafik	14
6.3.2 Enkelte borger	14

6.4	Katastrofekørsel	14
6.4.1	Alarmberedskab	14
6.4.2	Politi	14
6.4.3	Ambulanceudrykning	14
6.4.4	Trafikuheld	15
6.5	Kommunale og statslig instanser	15
6.5.1	15
6.5.2	Vejarbejde	15
7	Problemformulering	16
II	Problemløsning	17
8	Løsningsforslag	18
9	Kravspecifikationer	19
9.0.1	Succeskriterier	19
9.0.2	•	19
10	Implementation	20
11	Diskussion	21
12	Konklusion	22
13	Perspektivering	23
	Bibliography	24
A	Appendix	25

Initierende Problem

Danske trafik modeller bliver ofte opbygget med et konkret formål i fokus og bliver svære at vedligeholde i fremtiden i de tilfælde hvor konteksten ændrer sig.

Arbejdsspørgsmål

- *Hvilke variabler bruges til at simulere trafik?*
- *Hvem har gavn af disse trafikmodellerings simulatorer?*
- *Hvordan fungerer de forskellige eksisterende software, trafik modellerings simulatorer?*

2

Metode

Part I

Problemanalyse

3.1 Dijkstras Algoritme

Dijkstras algoritme er en algoritme til at finde den korteste vej fra et bestemt punkt til et andet punkt. Disse punkter kan bland andet repræsentere den korteste vej mellem to forskellige byer. Dijkstras er en grådig algoritme, da den finder den mindste længde først og fortsætter således.[1] Dijkstras algoritme finder den korteste rute mellem 2 forskellige punkter i en simpel ikke-orienteret vægtet graf. Man kan se på figur ?? at der angives forskellige punkter A, B, C, D, E, Z. Hvis man skal fra punkt A til Z, så starter man ved A og derfor initialiseres A til at være 0. Algoritmen virker således, at alle punkter er uendeligt udover det punkt man befinder sig på. Algoritmen tager punkt fra punkt, så den starter med at se de grene som A har disse er $|AB| = 4$ og $|AC| = 2$. Her fra kan algoritmen ikke se videre end B og C. Den ser altid på det mindste tal, og derfor tager den $|AB| = 4$ og $|AC| = 2$, da begge disse tal er mindre end uendeligt. Herefter ser den efter hvilket af de nuværende tal som er mindst, hvilket er 2. Så derfor vælger den C som sit næste punkt. Algoritmen finder nu de næst tætteste punkt, ved at addere alle de tidligere ruter, som har den korteste rute fra A til det næste sæt af punkter. Her ser algoritmen ud fra C og hvilke grene C har. Dette er $|CB|$ $|CD|$ $|CE|$, dog er længden altid fra A, så derfor er længden fra A til E 12 da $2+10 = 12$. Længden til B er nu blevet 3 , da algoritmen ser på den mindste rute, så A til B er 3 , da $2+1 = 3$. Således fortsætter algoritmen indtil den rammer Z.[1]

```

1  procedure Dijkstras(G: weighted connected simple graph, with  $\leftarrow$ 
    all weights positive)
2  {G has vertices  $a = V_0, V_1, \dots, V_n = z$  and lengths  $w(V_i, V_j) \leftarrow$ 
    where  $w(V_i, V_j) = \text{infinity}$  if  $\{V_i, V_j\}$  is not an edge in G}
3  for (i = 1 to n)
4    L( $V_i$ ) = infinity
5  L(a) = 0
6  S = NULL
7  { the labels are now initialized so that the label of a is 0 and  $\leftarrow$ 
    all other labels are infinity, and S is the empty set }
8  while (z does not belong to S)
9    u = a vertex not in S with L(u) minimal
10   S = S  $\cup$  {u}
11   for (all vertices v not in S)

```

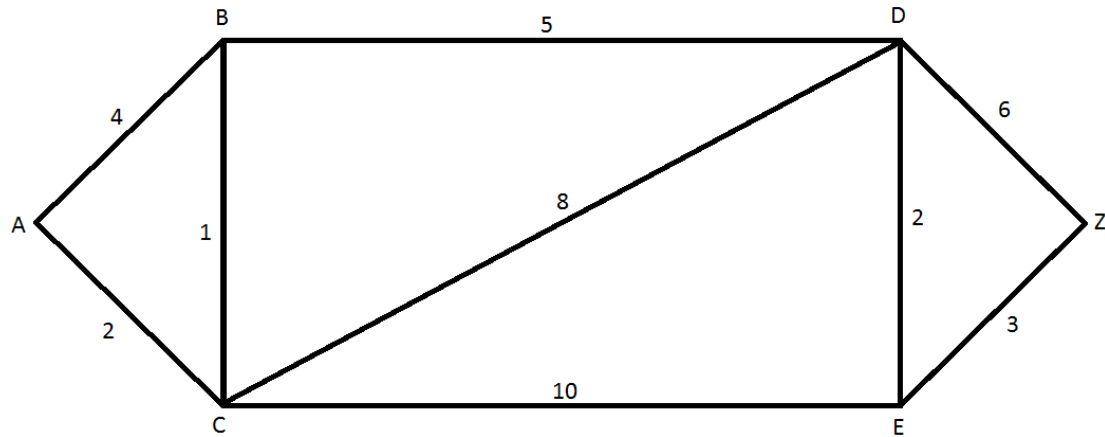


Figure 3.1: Graf til fremvisning af eksempel af Dijkstras algoritme i brug

	A	B	C	D	E	Z
	0	Inf	Inf	Inf	Inf	Inf
A	0	4	2	Inf	Inf	Inf
A, C	0	3	2	10	12	Inf
A, C, B	0	3	2	8	12	Inf
A, C, B, D	0	3	2	8	10	14
A, C, B, D, E	0	3	2	8	10	13
A, C, B, D, E, Z	0	3	2	8	10	13

Table 3.1: Dijkstra tabel

```

12   if (L(u) + w(u, v) < L(v) then L(v) = L(u) + L(u, v))
13   {this adds a vertex to S with minimal label and updates the ↵
    labels of vertices not in S}
14   return (L(z)) {L(z) = length of a shortest path from a to z}

```

Listing 3.1: Dijkstras angivet som eksempel i pseudo-kode

3.2 A* Algoritmen

Primært når det kommer til belægning af en dynamisk rute, foregår det ved at en enhed fortsætter hen i mod et mål indtil den når en forhindring. Dette er et ekstremt simpelt bevægelsesmønster og indebærer in vis in-effektivitet. Rent retorisk kunne man stille spørgsmålet om det ikke ville være smartere at planlægge en rute før man overhovedet bevæger sig.

A* er en algoritme til at beregne den korteste rute baseret på en række heuristiske datasæt. A* får input igennem en brugerlavet graf der indeholder en række datasæt for at algoritmen kan fungere. Først har vi distancen fra punkt til punkt, eksempelvis punkt 'A' til punkt 'B' som vi kalder for f.eks. 'H' og dernæst har vi et datasæt 'G' der indeholder bekostningen for at flytte fra en kant til en anden, denne variabel er bestemt på forhånd. Et virkelighedseksempel kunne være at man vil over på den anden side af en sø, så har man så muligheden for at svømme direkte eller gå uden om og det koster f.eks. 2 gange så meget at bevæge sig direkte igennem søen. Dette er givet ved 'G', hvor som sagt 'H' er den ultimative korteste længde til det bestemte slutpunkt. 'H' fungerer desuden for hvilket som helst punkt i et system og angiver *altid* den korteste vej til slutpunktet uanset forhindringer. Det skal også nævnes at 'H' ikke er påvirket af bevægelsesbekostningen, til at starte med, som 'G' angiver, dette kommer først senere. Til sidst har vi 'F' der er en sammenlagt værdi af både 'H' og 'G'. Dette gælder kun for hver kasse der flyttes til, hvori 'H' er angivet ved kassen man flytter tils 'H' værdi. Det kan vises således i formelen 3.1:

$$F(n) = G(n) + H(n) \quad (3.1)$$

En måde man kan visualisere A* på er f.eks. med et gitter-system som set i figur ???. Her kan vi se at vi har et start punkt (grøn) og et slutpunkts (blå). De kasser vi ikke kan bevæge os igennem er de røde kasser. Figuren angiver ingen heuristiske datasæt endnu.

Ud fra figuren kan vi begynde os at forestille hvordan A* fungerer. Når man bevæger sig fra kasse til kasse laver man 2 lister til at holde styr på hvor brikken har været. En liste til at holde styr på hvilke kasser man ikke har besøgt endnu og en liste der holder styr på hvilke man **har** besøgt. Når man flytter brikken skal man derfor angive hvilken kasse der nu skal på *besøgt* listen. Derfor som nævnt skal vi bruge information om hvor meget 'G' koster. Brikken skal nu til at flytte sig for at komme til slutpunktet. Dette kunne f.eks. være 10 point for at flytte sig i hvilken som helst retning, men man kunne også sagtens angive at diagonal bevægelse ville koste 12 point. Dvs. at ruten ændrer sig til måske ikke at være så direkte som den ellers kunne have været.

Der findes flere metoder man kan anvende A* på og en af dem vises her. Det vises her i den lille bid af pseudo-kode i Listing 3.2:

```

1  class A_Star
2  {
3      List<int> nodeIdentifierVisited = new List<int> { 7, 8, 9 , ←
          10, 11, 12, 13, 14, 21, 23, 24, 28, 29, 30, 31, 32};
4
5      Expand to new node from previous and move
6      Queue frontQueue = new Point();
7
8      when Point is acknowledged to move
9      change new Point to become Visited = true;
10
11     while(new Point == Visited)
12     {
13         move Point to nodeIdentifierVisited
14     }
15 }

```

Listing 3.2: A stjerne og pseudo-kode af brug af lister

1	2	3	4	5	6	7	8	9	10	11	12
13	14	15	16	17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32	33	34	35	36
37	38	39	40	41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70	71	72
73	74	75	76	77	78	79	80	81	82	83	84
85	86	87	88	89	90	91	92	93	94	95	96

Figure 3.2: A* gitter-system

Som set i figur ?? har vi vores liste givet ved kassernes nummerering. Nummereringen kører fra venstre mod højre én række ad gangen. Vi angiver at det tager 10 point at gå lodret og vandret én kasse ad gangen og 12 point at gå diagonalt. I figur ?? kan vi nu se de heuristiske datasæt angivet fra startpunktet (grøn). Hver enkel kasse omkringliggende startpunktet har deres '**H**' værdi angivet med lys-lilla tekst og bevægelsesomkostningen '**G**' fra startpunktet til kassen angivet i blå tekst.

Nu udregnes '**F**' værdien så f.eks. hvis vi går fra kasse 14 (startpunktet) til 15 skal vi lægge 10 ('**G**') og 8 ('**H**') sammen. Dette gør vi så for alle omkringliggende kasser for startpunktet. Dernæst går man til den laveste '**F**' værdi og gør helt det samme som før, derudover flyttes den nye kasse man står på til *besøgt* listen. Noget man skal være opmærksom på her er at man stadig skal sammenligne bevægelsesomkostningen fra den tidligere kasse til de kasser der også er relevante for den nye kasse man har flyttet sig til.

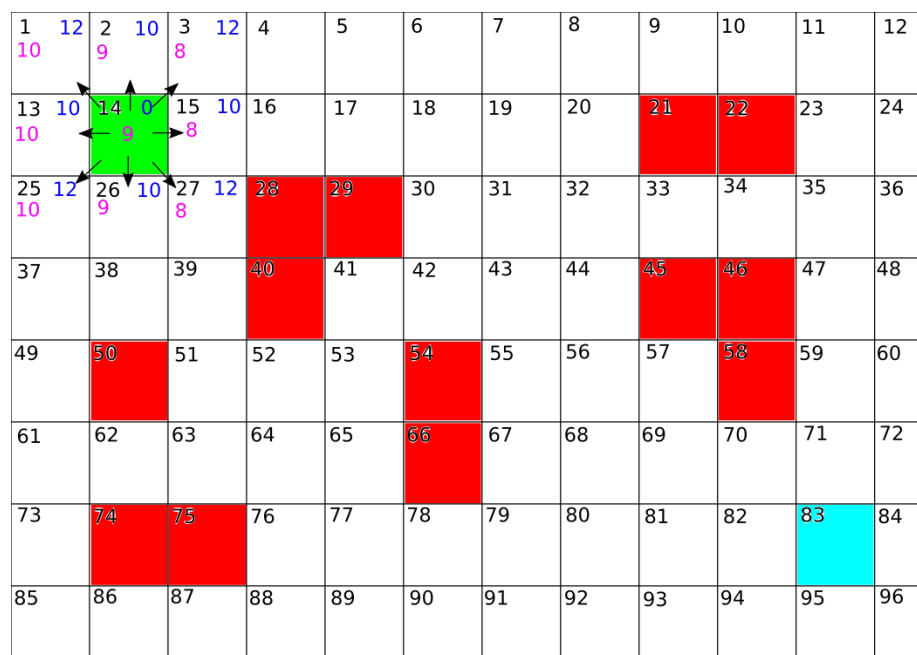


Figure 3.3: A* der viser bekostning af bevægelse fra startpunkt (grøn) til omkringliggende kasser ('G') angivet med blå farve samt 'H' angivet med lys-lilla

Problembeskrivelse

Trafiksystemet i Danmark undergår ofte udbygninger og ændringer, hvilket kan påvirke nærliggende vejnet. Det kan være svært at se hvordan disse ændringer vil påvirke trafikken, og derfor er der blevet opstillet forskellige modeller til at forudsige hvordan trafikken på vejnettet i fremtiden vil afvikle sig. Disse trafikmodeller bliver ofte opbygget med et konkret formål i fokus, og bliver derfor svære at vedligeholde i fremtiden i de tilfælde hvor konteksten ændrer sig. Konsekvensen af dette er at under en tredjedel af modellerne er blevet vedligeholdt, og at der ikke længere findes en model der dækker hele Danmark [6, s. 1-2].

4.1 Problemets Relevans

Formålet med trafikmodellerne er at forhindre trafikpropper og at sænke rejsetiderne. Uden modellerne er det besværligt at bestemme hvor der er problemer i vejnettet, og hvilken effekt nye veje vil have på trafikstrømmen. Trafikpropper har en effekt på landets økonomiske vækst, forurening, livskvalitet og tiden det tager for beredskaber og politi at nå frem.

4.1.1 Økonomi

Er man uheldig, kan man risikere at sidde fast i trafikken på vej til arbejdet. For at finde ud af hvilken effekt denne spildtid har på Danmarks økonomi, har Michael Knørr Skov og Karsten Sten Pedersen, der arbejder for konsulent firmaet COWI, analyseret 3 vejprojekter [4]. Vejprojekterne inkludere en tredje Limfjordsforbindelse, en ny motorvejsstrækning ved København, og en Forbindelse mellem Fyn og Als. Udfra COWI's beregninger vil disse tilføjelser spare danskere 25 tusinde timer dagligt, hvilket svarer omtrent til en værdi på 2500 millioner kroner årligt. Antager man at en fjerde del af denne tid bliver brugt på arbejde vil man opleve en BNP-vækst på 0,035%. Et velfungerende vejnet er dermed et vigtigt aspekt i forhold til at forbedre Danmarks økonomiske vækst.

4.1.2 Forurening

Billister er en af de største kilder af CO₂ forurening. Mængden af CO₂ der bliver udsluppet, afhænger af hastigheden billisterne kører. Ved en lav hastighed kan CO₂ udslippet per kilomet blive fordoblet, i forhold til at køre en stabil 50-130 km/t. I den anden ende, hvis man kører over de 130 km/t vil udslippet igen øges, da bilen er mindre effektiv i udnyttelsen af brændstoffet [3, s. 5-6].

4.1.3 Livskvalitet

En undersøgelse har vist at der er en sammenhæng mellem trafik densiteten, og stress niveauet på en individ der befærder sig i denne trafik. Udover at det kan være ubehageligt under kørslen, bliver stressen også ført med videre på arbejdet og til hjemmet. Stressen kan også føre til aggressiv kørsel og i værste tilfælde ender det med en ulykke [2, s. 2-3].

4.1.4 Nødsituationer

For den almindelige bilist, så kan trafikpropper være irriterende at skulle igennem, da det er tidskrævende. Men når det kommer til ambulancernes udrykning, og det kan have fatale konsekvenser for nogle patienter. Falck har oplyst at det koster ambulancerne 1-2 minutter i udrykningstid, når der er trafikprop. Konsekvenserne kan variere alt efter hvor alvorligt syg patienten er, og i værste tilfælde så er konsekvensen menneskeliv [5].

4.2 Eksisterende Modeller

De modeller der er vedligeholdet og stadig bliver brugt i dag, er meget forskellige i deres fokus. Der findes modeller som Senex, der analyserer godstrafikken mellem Danmark og Tyskland, der er en meget avanceret model til trafikafviklingen i hovedstadsområdet, og en masse mindre regionale og kommunale modeller [6, s. 2]. Forskellen på modellerne kan ses på detaljeringsgraden og hvor langt modellen kigger ud i fremtiden, hvor de mindre modeller har flere detaljer, men kun kigger få år ud i fremtiden, og vice versa for de større modeller. Trafikmodellerne er derfor delt op i 3 kategorier; strategiske, taktiske og operationelle modeller [6, s. 1].

Strategiske modeller er langsigtede modeller, men med færre detaljer. Manglen på detaljer er påkrævet, da det ellers vil blive for svært at anskaffe data'en, der skal bruges til at specificere alle forudsætningerne for modellens forudsigelser [6, s. 1]. Modeller af denne slags danner et billede over den internationale situation [6, s. 9]. Danmark benytter sig af en strategisk model, Trans-Tools, der blev udviklet i samarbejde med EU-kommissionen. Formålet med denne model er at forstå konsekvenserne af ændringer i det europæiske vejnetværk. Trans-tools hører også ind under taktiske modeller da den inkorporerer detaljer som for eksempel transportmiddelvalg [6, s. 10].

Taktiske modeller har i størstedelen af tilfældene et sigte mellem 3 og 20 år. I forhold til de strategiske modeller er detaljeringsgraden højere. Formålet med

disse modeller kan for eksempelvis være at finde ud af hvilke veje er belastede eller hvor lang tid en rejse vil tage [6, s1]. Taktiske modeller bliver brugt til at vise udviklingen i både internationale, nationale og regionale situationer [6, s. 9]. Modellerne der hører herunder er Senex, Storebæltsmodellen og Ørestadstrafikmodellen. Senex bruges til at vurdere tyske lastbilafgifter. Storebæltsmodellen bliver brugt til at vurdere takster og hvordan færgeudbuddet kan påvirke taksterne. Ørestadstrafikmodellen beskriver trafikken i Ørestad, og giver prognoser på hvordan en fremtidig stigning af antal biler vil påvirke vejnettet.

Operationelle modeller er kortsigtede modeller, og området man undersøger er meget afgrænset. Fordelen ved disse modeller er at den høje detaljerings grad kan give et mere præcist billede over situationen, dog kræver det at der skal bruges en masse data for at resultatet bliver realistisk [6, s. 1]. Herunder har vi kommunale og regionale modeller.

Udover disse er der mange andre modeller der blev opstillet, men ikke længere er vedligeholdt, herunder har vi eksempelvis Landstrafikmodellen, Hovedstadstrafikmodellen (HTM), national lastbilmodel, trafikafviklingsmodeller og mange andre [6, s. 8]. Gennem problemanalysen vil en af de vedligeholdte modeller og en af de ikke vedligeholdte modeller blive undersøgt ved hjælp af en teknologianalyse, for at finde ud af hvilke elementer er vigtige. Informationen fra teknologianalysen kan derefter bruges til at lave en trafikmodel der er nem at vedligeholde.

5

Teknologianalyse

Interessentanalyse

Dette kapitel vil afdække nogle af de interessenter som ville have et incitament til at se et program, der kan simulering en optimering af trafikken udviklet, samt de interessenter som vil have et incitament til at hindre eller miskrediterer udviklingen af dette. Der vil blive udvalgt instanser til at repræsentere de relevante interessenter til at afspejle et generelt billede i samfundet.

6.1 Kollektiv trafik

Dette afsnit har til formål at give et generelt billede af hvordan et simuleringsprogram vil være behjælpelig eller påvirke den kollektive trafik samt om det har et incitament til at hindre eller støtte udviklingen af programmet.

6.1.1 Nordjyllands Traffikselskab (NT)

6.2 Erhvervskørsel

Dette afsnit har til formål at give et generelt billede af hvordan et simuleringsprogram vil være behjælpelig eller påvirke den erhvervskørsel samt om det har et incitament til at hindre eller støtte udviklingen af programmet.

6.2.1 Transport og fragt

Dette afsnit har til formål at give et generelt billede af hvordan et simuleringsprogram vil være behjælpelig eller påvirke den transport og fragt samt om det har et incitament til at hindre eller støtte udviklingen af programmet.

6.2.2 Taxiselskaber

hold

6.3 Privatkørsel

Dette afsnit har til formål at give et generelt billede af hvordan et simuleringsprogram vil være behjælpelig eller påvirke privatkørsel samt om det har et incitament til at hindre eller støtte udviklingen af programmet.

6.3.1 Myldretidstrafik

Der er ingen tvivl om hvorvidt at myldretiden er det tidspunkt på dagen, hvor en simulering vil være til stop hjælp, og kunne afvikle et stort tidsforbrug. Myldretidstrafik propper er ofte forskyldt ved opbremsning og accelerations problemer. En undersøgelse foretaget i Japan viste hvordan trafikpropper opstod på trods af at alle forsøgspersoner, var blev bedt om at køre med en konstant hastighed i en cirkel. Det viser altså hvordan, at trafikpropper vil opstå i tæt trafikkeret områder, selvom der ikke vil opstå nogle problemer eller forhindringer i trafikken. En simulering der kan være med til at distribuere trafikken, således at alle ikke skal igennem de store knudepunkter i trafikken, kan være med til at hindre store trafikpropper.

<https://www.youtube.com/watch?v=Suugn-p5C1M>

<https://www.newscientist.com/article/dn13402-shockwave-traffic-jam-recreated-for-first-time/>

[http://iopscience.iop.org/article/10.1088/1367-](http://iopscience.iop.org/article/10.1088/1367-2630/10/3/033001/meta;jsessionid=438C580C1C2C3B3FB10D9B11878FE7D3.c3.iopscience.cld.iop.org)

[2630/10/3/033001/meta;jsessionid=438C580C1C2C3B3FB10D9B11878FE7D3.c3.iopscience.cld.iop.org](http://iopscience.iop.org/article/10.1088/1367-2630/10/3/033001/meta;jsessionid=438C580C1C2C3B3FB10D9B11878FE7D3.c3.iopscience.cld.iop.org)

6.3.2 Enkelte borger

Den enkelte borger i privatbiler er dem som udgøre den største del af trafikken, specielt omkring de tidspunkter hvor folk skal på arbejde, generelt om morgenstunden, samt når de skal hjem fra arbejde, generelt imellem eftermiddagen og aften, hvor det ofte er de tidspunkter hvor der er størst chance for at mængden af trafikulykker, uheld og trafikpropper opstår (find en kilde). eventuelt giv en sammenligning hvordan trafiktrykket kan sammenlignes med andre ting?

6.4 Katastrofekørsel

Dette afsnit har til formål at give et generelt billede af hvordan et simuleringsprogram vil være behjælpelig eller påvirke katastrofekørsel samt om det har et incitament til at hindre eller støtte udviklingen af programmet.

6.4.1 Alarmberedskab

hold

6.4.2 Politi

hold

6.4.3 Ambulanceudrykning

hold

6.4.4 Trafikuheld

Når der sker et trafikuheld, standser det hele trafikken indtil ambulanceredere har været og givet behandling til de tilskadekomne og der har været et opryddningshold. Sker der et uheld på en stærkt trafikeret vej, som motorvejen eller en hovedvej, kan det skabe store køer alt efter uhelds størrelsesomfang, samt hvor hurtig svartiden er på beredskabsfolket. Dette koster mange tusinder mennesker timer, selvom der bliver sagt over trafiksstyrelsesradioen, hvilke strækninger man bør undgå. Derfor vil en simulering der kan hjælpe med at distribuere trafikken efter ulykken således at man undgår store trafikpropper.

6.5 Kommunale og statslig instanser

Dette afsnit har til formål at give et generelt billede af hvordan et simuleringsprogram vil være behjælpelig eller påvirke de kommunale og statslige instanser samt om det har et incitament til at hindre eller støtte udviklingen af programmet.

6.5.1

6.5.2 Vejarbejde

Når der er vejarbejde optager de for det meste et helt vejspor eller mere, i værste tilfælde lukker de strækningen der arbejdes med. De veje som ikke er præget af meget trafik, vil ikke opleve den store påvirkning i trafikken, men større trafikeret veje, broer, tunneller og motorvejsstrækninger kan skabe trafikpropper når mængden af biler der skal passere forbliver forholdsvis uændret, mens vejpladsen er blevet reduceret med et spor eller mindre (find kilder på hvor meget trafikken kan påvirkes pga. vejarbejde eventuelt kig på hollands sommerferien). En simulering som kan vise hvordan

7

Problemformulering

Part II

Problemløsning

8

Løsningsforslag

9

Kravspecifikationer

9.0.1 Succeskriterier

9.0.2 ●

10

Implementation

11

Diskussion

12

Konklusion

13

Perspektivering

Bibliography

- [1] hans. *Game engines dissected*. Publisher, 1. edition, 2015.
- [2] Dwight A Hennessy og David L Wiesenthal. Traffic congestion, driver stress, and driver aggression. https://www.researchgate.net/profile/Dwight_Hennessy/publication/229863510_Traffic_congestion_driver_stress_and_driver_aggression/links/0deec53274dd4c9e88000000.pdf, 1999.
- [3] Matthew Barth og Kanok Boriboonsomsin. Traffic congestion and greenhouse gases. <https://escholarship.org/uc/item/3vz7t3db>, 2009.
- [4] Michael Knørr Skov og Karsten Sten Pedersen. Trafikprop. flere veje vil skabe større vækst. http://www.cowi.dk/menu/tema/infrastruktur-2030/cowi-i-medierne/Documents/Veje%20skaber%20v%C3%A6kst_Politiken%20analyse%2024052014.pdf, 2014.
- [5] Henrik Papsø. Trafikpropper kan koste menneskeliv. <http://www.tveast.dk/artikler/trafikpropper-kan-koste-menneskeliv>, 2007.
- [6] Danmarks TransportForskning. Trafikmodeller arbejdsnotat til infrastrukturkommissionen notat 3. http://www.transport.dtu.dk/~media/Institutter/Transport/forskning/publikationer/publikationer%20dtf/2007/arbejdsnotat_om_trafikmodeller_160507.ashx?la=da, 2007.

A

Appendix