



Göteborgs universitet

Naturvetenskaplig problemlösning

FIN 800

23 maj 2001

# Trafiksimulering med cellulära automater

-en jämförelse av koldioxidutsläpp i olika trafiksituationer

Lova Carlsson  
Emma Djärv  
Magdalena Lundh

**Handledare:**  
Peter Kumlin

## **Abstract**

Traffic simulation has a wide range of applications. It is used to predict and analyse traffic situations. The aim of this project is to present a flexible simulation model based on cellular automata. With this tool we estimate the amount of carbon dioxide released into the air in different traffic situations. The simulations and the calculations are performed with a MATLAB programme. Cellular automata is a grid of cells, where the cells are either filled or empty. The configuration of filled and empty cells is updated according to a set of rules. The new configuration is then updated according to the rules and so on. In our model the roads have two lanes, divided into cells. Each cell can be either occupied with a car or empty. A number of rules, modelled to reflect traffic rules and behaviour of drivers, determine the movements of the cars. We have calculated the carbon dioxide emission in three different traffic situations, free flow, traffic lights and with a wrecked car in the right lane. The highest emission of carbon dioxide appeared for the case of a wrecked car in the right lane, both during rush hour and normal traffic. Traffic lights during rush hour have no negative effect on the carbon dioxide emissions but during normal traffic it has. The technique with cellular automata proves to be flexible and our programme can be further developed.

# Innehåll

<b>1</b>	<b>Inledning</b>	<b>1</b>
<b>2</b>	<b>Programmets utformning</b>	<b>1</b>
2.1	Regler för körning rakt fram . . . . .	2
2.2	Omkörningsregler . . . . .	2
2.3	Regler vid trafikljus . . . . .	5
2.4	Regler för hinder i en fil . . . . .	5
<b>3</b>	<b>Beräkning av koldioxidutsläpp</b>	<b>6</b>
3.1	Approximationer av koldioxidutsläpp vid specifika hastigheter och accelerationer	6
3.2	Programmets beräkningsmetod för koldioxid . . . . .	7
<b>4</b>	<b>Simulering och resultat</b>	<b>8</b>
<b>5</b>	<b>Diskussion</b>	<b>9</b>

## 1 Inledning

Trafiksimulering används för att efterlikna situationer som kan uppstå i trafiken. Exempelvis används det vid försök att förbättra en trafik Korsning som redan existerar eller vid planering av nya vägar och väg Korsningar. Man kan även beräkna hur avgasutsläppen påverkas om trafikmiljön förändras.

Vid simulering används olika modeller beroende på syftet med simuleringen. En mikroskopisk simulering är detaljerad, där varje bil betraktas som en enhet med observerbar hastighet, acceleration och position, vilka beräknas vid varje tidsintervall. Vid makroskopisk simulering beskrivs trafiken istället som ett flöde där bilarna inte kan urskiljas från varandra. Den makroskopiska simuleringen går ofta fort, men den detaljerade informationen går förlorad. Till exempel kan differentialekvationer användas som grund för att approximera trafikflöde och trafiktäthet [1]. Då ges trafikflödet som en sannolikhetsfördelning som förändras i tid och rum.

Vi ville skapa en modell för att kunna simulera ett antal trafiksituationer och jämföra koldioxidutsläppen. För att göra detta valdes en mikroskopisk modell som grundar sig på cellulära automater. I USA används redan cellulära automater för att mikroskopiskt simulera stadstrafik och beräkna avgasutsläpp [2]. De används även för makroskopisk simulering [3]. Vid simulering med cellulära automater används ett rutnät med ändligt eller oändligt många rutor där varje ruta, även kallat cell, kan vara antingen besatt eller obesatt. Genom att sätta upp regler för förändringar kommer fördelningen av besatta och obesatta celler att utvecklas på ett visst sätt. Den ursprungliga konfigurationen av celler kommer på så sätt att ge upphov till en ny sammansättning genom att varje cell provas mot de bestämda uppdateringsreglerna. Reglerna kommer att avgöra om rutan i följande tidssteg kommer att vara besatt eller ej. Den nya generationen celler kommer på samma sätt att provas mot reglerna och en ny sammansättning uppstår. Detta upprepas ett bestämt antal gånger.

Vid simuleringarna undersöks hur olika hinder i trafiken, exempelvis trafikljus eller en havererad bil, påverkar utsläppen av koldioxid. I Sverige står trafiken för 35 % [4] av koldioxidutsläppen, ett problem som inte går att lösa med katalysatorer. Därför har vi försökt skapa en simuleringsmodell som gör det möjligt att bedöma trafiken, för att sedan kunna effektivisera trafikflödet och därmed minska bränsleförbrukningen.

I nästa avsnitt beskrivs simuleringsprogrammet och de trafiksituationer som kommer att simuleras. Ett par av trafikreglerna kommer där att presenteras i programkod för att kunna förstå programmet, som i sin helhet finns i appendix A. I avsnitt 3 beskrivs hur koldioxidutsläppen beräknas. Därefter följer resultat och diskussion.

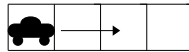
## 2 Programmets utformning

För att kunna simulera trafiksituationerna och göra beräkningar av koldioxidutsläpp skrev vi ett program i MATLAB. Vägavsnittet som användes var tvåfiligt och 1 km långt. Vägen beskrivs med en matris som består av två rader med 182 element vardera. Raderna i matrisen motsvarar filer som är uppdelade i lika många avsnitt, så kallade celler, där varje cell är 5,5 m lång. En besatt cell kommer i det här fallet att representera en bil, medan en obesatt cell betyder att det inte finns någon bil på vägavsnittet. Bilarna har möjlighet att öka respektive sänka farten, köra rakt fram samt byta fil. Reglerna är utformade så att de på ett så trovärdigt

sätt som möjligt ska modellera varje bilist körbeteende. Det antas att bilisterna vill ta sig från en punkt till en annan på kortast möjliga tid utan att krocka. De begränsas dock av en övre hastighet, samt av framförvarande bilar. Genom att låta bilarna som lämnar vägavsnittet återkomma i början hålls antalet bilar i systemet konstant. Uppdatering kommer att ske varje sekund.

## 2.1 Regler för körning rakt fram

Bilarna i högerfilen kan köra i två olika hastigheter, 20 km/h och 40 km/h. Om möjlighet finns kör bilarna i 40 km/h, vilket motsvarar att de flyttas två steg framåt per sekund. En möjlighet är när det är ledigt tre celler framåt vilket visas i Figur 1.



Figur 1: En bil i högerfilen kör fram två steg förutsatt att det är ledigt tre celler framåt.

Alternativet till regeln ovan är att det är ledigt i två celler framåt samt i den fjärde cellen framför, se Figur 2a. Då kan bilen i cell tre i nästa uppdatering flytta sig framåt och lämna plats åt bilen bakom. Denna regel och de två följande är gemensamma för höger- och vänsterfilen. Om det bara är ledigt i en cell framåt åker bilarna i 20 km/h, vilket visas i Figur 2b. Det gör de också när det är ledigt i två celler framåt men det finns en bil i både cell tre och fyra, eftersom bilen i cell tre inte har möjlighet att förflytta sig vid nästa uppdatering, se Figur 2c. Regeln finns för att undvika att bli stillastående i nästa sekvens. Denna regel illustreras även med dess programkod

```
elseif (B(2,n)==1) & (B(2,n+1)==0) & (B(2,n+2)==0) & (B(2,n+3)==1)
    & (B(2,n+4)==1)
    C(2,n+1)=1;
```

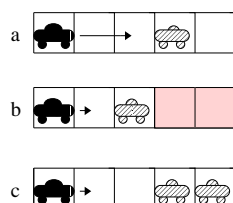
där B är matrisen som representerar den ursprungliga sammansättningen av celler. Nollorna på positionerna  $B(2,n+1)$  och  $B(2,n+2)$  betyder att det inte finns några bilar på dessa vägavsnitt. Bilen, här representerad av en etta, på position  $B(2,n)$  flyttas bara ett steg fram och sätts in i en ny matris C på position  $C(2,n+1)$ . Detta eftersom bilen på plats  $B(2,n+3)$  står stilla på grund av att det finns en bil i cell  $B(2,n+4)$ . Programkoden för övriga regler för att köra rakt fram i både högerfilen och vänsterfilen är uppbyggda på samma sätt.

Eftersom trafiken ofta flyter fortare i vänsterfilen ges bilarna där möjlighet att köra i ytterligare en hastighet, 60 km/h. För att få göra det måste det vara fritt i minst fyra celler framåt eller tre celler framåt samt i cellerna fem och sex, se Figur 3a och b. Den sistnämnda regeln förhindrar en kraftig inbromsning i sekvensen efter.

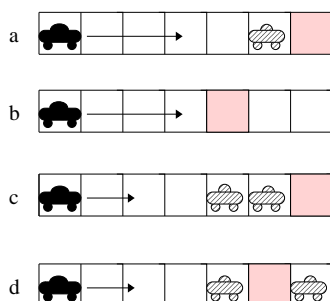
För att beskriva körning rakt fram i 40 km/h behövs nu ytterligare två regler. Om det är fritt i tre celler framför, upptaget i cell fyra och upptaget i antingen cell fem eller sex så kör bilen två celler framåt för att undvika alltför kraftiga inbromsningar, se Figur 3c och d.

## 2.2 Omkörningsregler

I programmet kan omkörning bara göras i vänsterfilen. Numera är det tillåtet att köra om även i högerfilen, men körbeteendet antas vara oförändrat. Omkörning sker antingen i 20 eller 40 km/h beroende på trafiksituationen. För att göra en omkörning måste det i samtliga

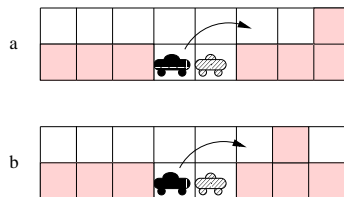


Figur 2: Regler för körning rakt fram i båda filerna. a) Bilen kör två celler fram om det är ledigt i två celler fram, upptaget i den tredje men ledigt i den fjärde. b) Bilen kör ett steg fram om det är ledigt en cell framåt men upptaget i den andra. c) Bilen kör ett steg om det är ledigt två steg fram men upptaget i cell fyra och fem. Skuggade rutor betyder att det inte har någon betydelse om de är tomma eller upptagna.



Figur 3: Regler för vänsterfilen. a) När det är ledigt fyra celler framåt kör bilen tre celler fram. b) Om det är ledigt tre celler fram och i cell fem och sex kör bilen tre celler fram. c,d) När det är ledigt i tre celler framåt men upptaget i cell fyra och i cell fem och/eller sex kör bilen fram två celler. Skuggad ruta betyder att det inte spelar någon roll om det finns en bil där eller inte.

fall vara fritt tre celler bakåt i vänsterfilen så att bilarna inte krockar. Om detta är uppfyllt finns det två möjligheter att köra om i den högre hastigheten 40 km/h, se Figur 4. Dessa är utformade så att en krock eller en kraftig inbromsning är omöjlig i sekvensen efter. Den första möjligheten till omkörning i den högre hastigheten är när cellen bredvid bilen och tre celler framåt i vänsterfilen är tomma, som i Figur 4a. Den andra möjligheten är när cellen bredvid och två celler framåt i vänsterfilen är tomma. Dessutom ska fjärde cellen framåt i vänsterfilen vara tom, så att en eventuell bil i den tredje cellen kan åka fram i nästa uppdatering, se Figur 4b.



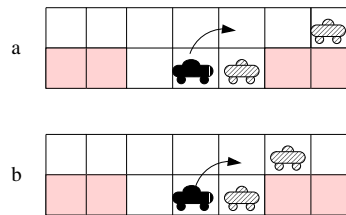
Figur 4: Omkörning i 40 km/h. a) Det är ledigt från tre celler bakåt i vänsterfilen till tre celler fram. b) Det är ledigt från tre celler bakåt till två celler fram i vänster filen. Dessutom måste den fjärde cellen framåt i vänsterfilen vara ledig för att en eventuell bil i cell tre ska kunna köra framåt. De rutor där det inte har någon betydelse om de är upptagna eller tomma är skuggade.

Även för omkörning i 20 km/h finns det två möjligheter, se Figur 5. Om det är fritt två celler framåt i vänsterfilen är det möjligt att köra om i 20 km/h. För att inte köra om samtidigt som bakomvarande och kanske snabbare bilar måste även cellen precis bakom vara tom, se Figur 5a. Denna regel skrivs med programkod på följande vis

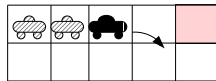
```
elseif (B(3,n)==1) & (B(3,n+1)==1) & (B(3,n-1)==0) & (B(2,n)==0)
& (B(2,n+1)==0) & (B(2,n+2)==0) & (B(2,n+3)==1)
& (B(2,n-1)==0) & (B(2,n-2)==0) & (B(2,n-3)==0)
C(2,n+1)=1;
```

där rad tre i matrisen B representerar högerfilen och rad två vänsterfilen. Bilen på position  $B(3,n)$  har en bil framför sig, men ledigt precis bakom samt jämte sig, tre rutor bakåt och två rutor framåt men upptaget i tredje rutan fram. Därför kan den köra om. Framfarten begränsas dock av bilen i ruta  $B(2,n+3)$  varför hastigheten endast blir 20 km/h. Bilen hamnar efter omkörningen i vänsterfilen en cell framåt. Det är även tillåtet att köra om i 20 km/h när det som i Figur 5b står en bil i den andra cellen snett framför men inte i den tredje. Bilen som står två celler snett framför kommer vid nästa uppdatering att flytta sig framåt och lämna plats åt bilen som kör om. Även här måste cellen precis bakom vara fri. Denna och övriga omkörningsregler skrivs med programkod på liknande sätt som den ovan beskrivna regeln.

För att förhindra att alla bilar samlas i vänsterfilen måste bilarna även kunna byta till högerfilen. När det blir kö på minst tre bilar i vänsterfilen byter därför den första bilen i kön fil förutsatt att det inte finns någon bil framför, bredvid eller två celler framåt eller bakåt i högerfilen, se Figur 6. Filbytet sker i 20 km/h.



Figur 5: Omkörning i 20 km/h. Cellen precis bakom den omkörande bilen och tre celler bakåt i vänsterfilen måste vara tomma för att förhindra att bilen blir påkörd bakifrån. a) Två celler fram i högerfilen ska vara lediga. b) Det är upptaget i cellen två steg fram i vänsterfilen men tomt i den tredje så att bilen i cell två kan förflytta sig framåt.



Figur 6: Kö i vänsterfilen medför att den första bilen i kön byter fil om det är ledigt framför den och i två celler bakåt till två celler fram i högerfilen. Filbytet sker i 20 km/h. När en ruta är skuggad har det ingen betydelse om den är upptagen eller tom.

## 2.3 Regler vid trafikljus

En av de trafiksituationer som simuleras är trafikljus. Vi sätter trafikljuset brevid vägen, i rad ett och fyra i matrisen B. När en bil närmar sig ett trafikljus kommer den att förbereda sig för att stanna genom att sänka hastigheten. Då trafikljuset slår om till rött och en bil befinner sig precis innan stannar bilen, se Figur 7a. Med programkod blir detta

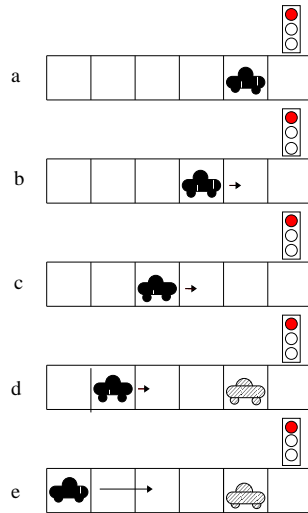
```
if (B(2,n)==1) & (B(1,n+1)==2)
C(2,n)=1;
end
```

Tvåan på plats B(1,n+1) symboliserar rött ljus och ettan på plats B(2,n) bilen. I den uppdaterade matrisen C står bilen kvar på samma position som i B. Plats B(2,n) är en position i vänsterfilen men det kunde lika gärna vara högerfilen eftersom samma regler gäller där. När det är ledigt i cellen framför men rött två celler fram kör bilen en cell framåt, se Figur 7b. Även när det är ledigt två celler fram och rött i den tredje kör bilen ett steg fram, se Figur 7c. Det gör den även om det står en bil tre celler fram och trafikljuset är fyra celler fram, se Figur 7d. Om trafikljuset är fem celler fram och det finns en bil i cell fyra kör bilen två celler fram, se Figur 7e. Den sista regeln behövs endast för vänsterfilen eftersom det inte går att köra lika fort i högerfilen.

## 2.4 Regler för hinder i en fil

Ett hinder i en fil kan vara ett mindre vägarbete eller en bil som har havererat. I programmet modelleras ett hinder av en besatt cell i högerfilen. Hindret placeras ut i en tom cell och är permanent under alla uppdateringarna. Alla regler för omkörning gäller också för att köra om hindret. Övriga regler som gäller ser ut som de fyra första reglerna för trafikljus, se Figur 7a, b, c och d, med den skillnaden att hindret finns på vägbanan.





Figur 7: Regler vid trafikljus. a) Bilen står stilla när det är rött ljus direkt framför. b) När det är ledigt en cell fram och rött ljus två celler fram åker bilen till cellen framför. c) Om det är ledigt två celler fram och trafikljuset visar rött i den tredje åker bilen också fram en cell. d) Om det är fritt till tre celler fram, där det står en bil och det är rött i den fjärde cellen åker bilen en cell fram. e) Denna regel behövs endast i vänsterfilen. Bilen åker två celler fram när det är ledigt tre celler fram, en bil i den fjärde och rött i den femte.

### 3 Beräkning av koldioxidutsläpp

Eftersom koldioxidutsläpp ska jämföras vid olika trafiksituationer är det fullt tillräckligt att approximera utsläppen vid olika hastigheter och accelerationer. Nedan beskrivs hur approximationerna går till samt hur värdena används i programmet.

#### 3.1 Approximationer av koldioxidutsläpp vid specifika hastigheter och accelerationer

För att räkna ut koldioxidutsläppen vid konstanta hastigheter använde vi formeln [5]

$$U(v) = 231 - 3,62v + 0,0263v^2 + \frac{2526}{v},$$

där  $v$  är hastigheten i km/h och  $U(v)$  ger koldioxidutsläppen i g/km, se Tabell 1. Formeln gäller endast för hastigheter mellan 5 och 130 km/h. Därför lät vi koldioxidutsläppen vid den lägsta hastigheten representera utsläpp vid tomgångskörning.

För att värdena ska passa in i programmet gjorde vi om dem från g/km till g/uppdatering. Eftersom en kilometer väg representeras av 182 celler delar vi värdena på utsläppen i g/km med 182 och multiplicerar detta med antalet framkörda rutor. Vid körning i 40 km/h blir det alltså

$$\frac{191 \text{ g/km}}{182 \text{ rutor/km}} \cdot 2 \text{ rutor} = 2,1 \text{ g.}$$

För tomgångskörning räknar vi med att bilen kör fram en fjärdedels ruta, men i programmet står bilen still.

Hastighet (km/h)	Utsläpp (g/km)
5	719
20	295
40	191
60	151

Tabell 1: *Koldioxidutsläpp vid konstanta hastigheter. 5 km/h får här representera tomgångskörning.*

Utsläppen vid acceleration beräknade vi genom att först räkna ut accelerationsarbetet med formeln  $W_a = \frac{mv^2}{2} - \frac{mv_0^2}{2}$ , där  $v_0$  är utgångshastigheten och där  $m$  är massan som sattes till 1000 kg. Detta omvandlade vi till kWh, där 1 kWh =  $3,6 \cdot 10^6$  J, och jämförde med bensen som har ett energiinnehåll på 8,7 kWh per liter [6]. Därefter multiplicerade vi antalet liter med 2360 g [7], vilket är mängden koldioxid som släpps ut per liter bensen. Detta läggs sedan till utsläppet för medelhastigheten av de två hastigheterna, se Tabell 2. Vid inbromsning kommer utsläppet att beräknas efter sluthastigheten. Utsläppet för acceleration från 20 till 40 km/h beräknades enligt följande

$$W_a = \frac{1000 \cdot 40^2}{2} - \frac{1000 \cdot 20^2}{2} = 46250 \text{ J},$$

vilket är detsamma som  $12,8 \cdot 10^{-3}$  kWh. Antalet liter bensen blir då

$$\frac{12,8 \cdot 10^{-3} \text{ kWh}}{8,7 \text{ kWh/l}} = 0,0039 \text{ l},$$

vilket ger ett koldioxidutsläpp på  $0,0039 \text{ l} \cdot 2360 \text{ g} = 3,5 \text{ g}$ . Medelhastigheten för 20 och 40 km/h, 30 km/h, ger dessutom ett utsläpp på  $(231 - 3,62 \cdot 30 + 0,0263 \cdot 30^2 + \frac{2526}{30}) \cdot \frac{2}{182} = 2,5 \text{ g}$ . Anledningen till att vi multiplicerar med två är att bilen kör två rutor framåt. Det totala utsläppet från bilen kommer i denna uppdatering därför att bli  $3,5 \text{ g} + 2,5 \text{ g}$ , det vill säga 6,0 g.

Utgångs- och sluthastighet (km/h)	Utsläpp (g)
0 → 20	3,6
0 → 40	7,9
0 → 60	14,3
20 → 40	6,0
20 → 60	12,5
40 → 60	8,6

Tabell 2: *Koldioxidutsläpp vid acceleration.*

### 3.2 Programmets beräkningsmetod för koldioxid

För att beräkna utsläppen från bilarna i våra simuleringar vill vi samla utsläppen för varje bil i en matris. För att kunna ta reda på en bils acceleration måste vi också veta hur fort

den körde i tidssekvensen före.

Samtidigt som en bil flyttas fram kommer en siffra för utsläppet att läggas på samma position i en ny matris,  $K$ . Efter varje uppdatering kommer dessa värden att överföras på en minnesmatris,  $k$ . Vid nästa uppdatering kommer värdena i  $k$  flyttas in i en matris  $Z$  på den position dit den tillhörande bilen flyttas i  $C$ . Detta illustreras med ett exempel på hur en regel kan se ut

```
elseif (B(3,n)==1) & (B(3,n+1)==0) & (B(3,n+2)==0) & (B(3,n+3)==0)
    C(3,n+2)=1;
    K(3,n+2)=2.1;
    Z(3,n+2)=k(3,n);
```

Eftersom bilen flyttas fram två steg i  $C$  kommer koldioxidutsläppet att vara 2,1 g. Detta värde kommer att sättas in i matrisen  $K$  på position (3,n+2), det vill säga samma plats som bilen hamnar på i  $C$ -matrisen. Samtidigt kommer bilens utsläpp från föregående tidssekvens att flyttas två steg framåt. Alla utsläpp som har med en viss bil att göra kommer på så vis att befinna sig på samma position som bilen själv.

Nu kan utsläppen jämföras för den aktuella tidssekvensen med den föregående och dra slutsatser om hur mycket bilen har ökat eller minskat hastigheten. Detta gör vi genom att subtrahera varje element i  $Z$  från samma element i  $K$  och sätta in skillnaden i en matris,  $D$ . Eftersom MATLAB inte räknar på skillnader med decimaler multiplicerar vi först matriserna  $Z$  och  $K$  med en faktor 10. I programkod blir detta

```
D(1:i,1:j)=10*K(1:i,1:j)-10*Z(1:i,1:j);
```

Från varje differens utläses hur stor accelerationen är och varje bil tilldelas ett värde på utsläppet. Dessa värden sätts in i en matris,  $A$ . Utsläppen från accelerationen adderas med värdena i  $K$ -matrisen och läggs in i en ny matris,  $Tot$ . Detta blir med programkod

```
Tot(1:i,7:j)=A(1:i,7:j)+K(1:i,7:j)
```

Anledningen till att vi endast summerar från element 7 är att element 1 till 6 förekommer två gånger i matriserna. Slutligen summeras utsläppen för hela uppdateringen och läggs ihop med utsläppen från de andra uppdateringarna.

När en bil kör framåt kommer även längden på den framkörda sträckan att läggas in i en matris,  $Q$ . Den totalt körda sträckan för alla bilar kommer sedan att summeras och vi kan beräkna koldioxidutsläpp per längdenhet. Genom att dividera utsläppen från alla bilarna i hela simuleringen med den sammanlagt körda sträckan för alla bilar erhålls ett snittvärde på hur mycket varje bil har släppt ut per kilometer.

## 4 Simulering och resultat

Simuleringarna kommer att göras med två olika initialmatriser där den ena symboliserar rusningstrafik och den andra glesare trafik. Dessa är 182 element långa och 4 element breda. I de två mittenraderna finns 132 respektive 70 slumpmässigt utplacerade bilar, medan ytterradererna är tomma förutom vid simulering av trafikljus. När simuleringen startar kommer alla bilar att stå stilla, vilket medför att de börjar med att accelerera. För att detta inte ska ge ett fel

i resultaten uppdateras matrisen 50 gånger innan beräkningarna av koldioxidutsläppen och den körda sträckan startar. Under varje simulering kommer matrisen att uppdateras ungefär 1000 gånger. Vid trafikljus kommer antalet uppdateringar att variera för att det ska bli ett jämnt antal intervall. Som exempel tar vi intervallet 20 sekunder rött och 25 sekunder grönt ljus. Vi söker ett heltal med vilket vi kan dividera 1000 med 45, som är den totala tiden för en period. Närmaste heltal är 22 och därför gör vi 990 uppdateringar. Vid fritt flytande trafik och vid ett hinder kommer vi dock att göra exakt 1000 uppdateringar. För att avgöra hur många uppdateringar som behövs gjordes försök med olika antal uppdateringar. Skillnaden mellan resultaten av 1000 och 5000 uppdateringar var endast 0,4 %. Därför bedömdes 1000 uppdateringar vara tillräckligt för att få goda värden.

Vid simuleringarna med trafikljus och hinder kommer utsläppen från fritt flytande trafik att användas som referens. I den första simuleringen användes matrisen som representerar rusningstrafik. Simuleringen av fritt flytande trafik gav att bilarna i genomsnitt släpper ut 313 gCO<sub>2</sub>/km. I Tabell 3 visas resultaten från simulering med olika trafikljusintervall. Där kan man se att det bara är vid tre av sju trafikljusintervall som bilarna släpper ut mer koldioxid än vid fritt flytande trafik. Simulering gjordes även med ett hinder i högerfilen. Vid denna simulering släppte bilarna i genomsnitt ut 507 g/km. Detta är 62 % mer än vid fritt flytande trafik.

Den andra simuleringen gjordes med glesare trafik. Den fritt flytande trafiken gav ett koldi-

Intervall trafikljus tid rött/grönt (s)	Koldioxidutsläpp (g/km)	Förändring (%)
20/25	342	9
24/30	321	3
28/35	307	-2
32/40	316	1
36/45	302	-4
40/50	295	-6
44/55	295	-6

Tabell 3: *Genomsnittligt koldioxidutsläpp vid rusningstrafik för olika trafikljusintervall. Förändringen är beräknad i förhållande till utsläppen vid fritt flytande trafik, vilket är 313 g/km.*

oxidutsläpp på 165 g/km. Då trafikljus med skiftande intervall lades in i programmet ökade utsläppen, till skillnad från simuleringen av rusningstrafik där endast hälften av intervallen gav upphov till ökat koldioxidutsläpp. Med ett hinder i högerfilen blev det genomsnittliga utsläppet 186 gCO<sub>2</sub>/km. Detta är en ökning på endast 13 % jämfört med den fritt flytande trafiken.

## 5 Diskussion

Vid simulering av trafikljus i rusningstrafik var det vid vissa intervall mindre utsläpp än vid simuleringen av fritt flytande trafik. Detta kan förklaras med att det uppstår köer även när

Intervall trafikljus tid rött/grönt (s)	Koldioxidutsläpp (g/km)	Förändring (%)
20/25	215	30
24/30	216	31
28/35	209	27
32/40	211	28
36/45	199	20
40/50	194	18
44/55	201	22

Tabell 4: *Genomsnittligt koldioxidutsläpp för gles trafik vid olika trafikljusintervall. Förändringen är beräknad i förhållande till fritt flytande trafik, som är 165 g/km.*

trafiken flyter fritt på grund av den täta trafiken. Med trafikljus blir köerna ännu längre men när det slår om till grönt kan bilarna köra i full hastighet. Den höga hastigheten kommer att ge ett mindre utsläpp per kilometer jämfört med den låga hastigheten som hålls i en bilkö. När vi sedan simulerade med glesare trafik gav alla trafikljussimuleringarna ett högre utsläpp än fritt flytande trafik. Där kan bilarna hålla en hög snitthastighet utan trafikljus, men med trafikljus blir det köer som ger höga utsläpp per kilometer. Hindret som sattes in i högerfilen gav ett påtagligt förhöjt utsläpp vid rusningstrafik. I glesare trafik var det inte lika stor skillnad jämfört med då det inte fanns något hinder på vägen. Detta på grund av att det med glesare trafik finns bättre omkörningsmöjligheter. Vid försök med andra konfigurationer av bilar i initialmatrisen erhöles resultat liknande de som presenteras i resultatavsnittet. Skillnaden var mindre än 10 %.

Slutsatsen av detta är att trafikljus i rusningstrafik endast har en marginell inverkan på koldioxidutsläppen per kilometer. Vid glesare trafik inverkar trafikljus däremot negativt. Vi har inte lyckats finna ett idealiskt intervall för att minimera koldioxidutsläppen, men har sett en tendens att längre intervall ger lägre utsläpp. Varken bilister eller fotgängare vill dock vänta på grönt någon längre tid, varför ett mellanting är att föredra.

Syftet med projektet var att skapa en bra och utvecklingsbar modell för trafiksimulering, inte att exakt kunna beräkna koldioxidutsläpp. Därför ansåg vi det tillräckligt att approximera värden på utsläpp vid olika hastigheter och acceleration. Detta räcker för att kunna jämföra medelvärden på utsläpp i olika trafiksituationer. Det finns dock en osäkerhet i dessa framräknade värden. Formeln för hastighetsberäkning baseras på uppmätta värden från olika typer av fordon. Bilarna har delats in beroende på storlek, bränsletyp, motor och så vidare [5], varefter de har körts i cykler som ska representera verkligheten [8]. Approximeringen av utsläpp vid acceleration är däremot teoretiskt beräknade. Vid beräkningarna av accelerationsarbetet har ingen hänsyn tagits till luftmotstånd eller rullmotstånd.

En svaghet hos modellen är de kraftiga hastighetsförändringarna. Bilarna kan accelerera från 0 km/h till 60 km/h på en sekund, något som inte är möjligt i verkligheten. Detta medför att accelerationen inte pågår under tillräckligt lång tid. Inbromsning kan inte ske lika hastigt, eftersom reglerna inte tillåter tvärbromsningar. Men även inbromsningarna är kraftigare än i realiteten. Detta skulle kunna åtgärdas genom att lägga till fler regler, men tiden har begränsat oss.

I vårt program följer alla bilar samma regler, men vid en utveckling av programmet är det möjligt att låta sannolikheter styra bilarnas körbeteenden. Till exempel kör inte alla bilister om även om detta är möjligt. För oss var dock individuella körbeteenden mindre viktigt, eftersom vårt mål var att jämföra medelvärden på utsläpp i olika situationer och inte att studera den enskilda bilen.

Att bilarna endast kan köra om i 20 km/h och 40 km/h kan tyckas onaturlig. Det skulle vara enkelt att lägga till ett antal regler för omkörning i 60 km/h, men då det inte förekommer någon mötande trafik låter vi omkörning ske i de två lägre hastigheterna. Bilen som byter fil kommer dessutom ha möjlighet att öka hastigheten till 60 km/h i nästa sekvens.

Huvudsyftet med detta projekt var att skapa en användbar modell för trafksimulering. Resultatet blev ett enkelt och flexibelt simuleringsverktyg som i ett större sammanhang skulle kunna utvecklas för att simulera mer komplexa trafksituationer. Ett första steg skulle kunna vara att ge förarna individuella egenskaper. För att göra detta så realistiskt som möjligt vore det dock önskvärt med en studie av förarens olika körsätt. Det som kan vara av intresse är acceleration, vilken växel som används och hur defensiv körningen är.

## Referenser

- [1] Personlig kommunikation med adj professor Ingmar Andréasson, Centre for TrafficSimulation Research, Kungliga Tekniska Högskolan
- [2] Los Alamos National Laboratory, California University *TRANSIMS Homepage* (<http://www-transims.tcasa.lanl.gov>) 20010523
- [3] P Wagner *Traffic simulations using cellular automata: comparison with reality* Universitat zu Koln 1995. Report No. 95.214
- [4] Malmberg Birgitta, Naturvardsverket *Sa paverkas miljon av trafiken* (<http://www.environ.se>) 20010517
- [5] Hickman J, Hassel D, Joumard R, Samaras Z, Sorenson S *Methodology for calculating transport emissions and energy consumption: Deliverable 22 for the project MEET* Transport Research Laboratory 1999. Report No. SE/491/98
- [6] Personlig kommunikation med professor Christian Azar, Fysisk Resursteori, Chalmers Tekniska Hogskola
- [7] Personlig kommunikation med lektor Gunnar Lanner, Vag och Vatten, Chalmers Tekniska Hogskola
- [8] Laboratory of Applied Thermodynamics Aristotle University of Thessaloniki, Samaras Z, Ntziachristos L *Methodologies for estimating air pollutant emissions from transport: Average hot emission factors for passenger cars and light duty trucks: Deliverable 7 for the project MEET* 1998. 3 LAT Report No. 9811