

중동고등학교
2013-2014학년도
소논문 탐구프로젝트 최종논문



SDS를 바탕으로 하는 교통모델 제작과, 간단한 활용

2014년 6월

중 동 고 등 학 교

20823 전찬호
21027 정현엽

SDS를 바탕으로 하는 교통모델 제작과, 간단한 활용

지도교사 이 희 영 (인)

이 논문을 소논문 탐구프로젝트 최종 논문으로 제출함



2014년 6월

중동고등학교

20823 전찬호

21027 정현엽

전찬호, 정현엽 2인의 소논문을 인준함.

심사위원 : _____ 인

심사위원 : _____ 인

중 동 고 등 학 교

2014 년 6 월 27 일

목 차

I. 서론	1
II. 이론적 배경	2
1. 차선 변경 결정	2
2. 차선 실행 결정	2
2. 가속	3
4. 이동	3
III. 기본 모델	4
1. 개요	4
2. 장소	5
3. 차	6
4. 그 외에 사용할 용어들	6
5. 사이클	6
5.1. 차선 변경 결정	7
5.2. 차선 변경 실행	10
5.3. 속도 변경	11
5.4. 위치 변경	13
IV. 연구 결과 및 해석	14
1. 모델을 이용한 실험의 개요	14
2. 세부적인 실험 결과	15
3. 종합적인 결과	19
4. 결과 해석 및 의의.....	20
V. 결론	21
1. 의의	21
2. 아쉬운 점 및 미래 연구	21
참고문헌	23
감사의 글	24

SDS를 바탕으로 하는 교통모델 제작과, 간단한 활용

전찬호

정현엽

중동고등학교

<요약>

An SDS (Sequential Dynamic System) based traffic model for micro situations was made and used for a real simulation test. Vehicles are constantly looking out for safety and are also calculating the best possible move for the next time step in the simulation. Behaviors such as trying to avoid trucks in their way, doing greedy lane changes resulting deceleration of vehicles behind were observed similar to real life ones.

주제어: SDS, traffic model, simulation

I. 서론

사람들은 도심 지역과 고속도로 위에서 많은 시간을 정체 된 상태로 보낸다. 그래서 사람들은 많은 시간을 의미 없이 허비하게 된다. 그렇기 때문에 교통 체증을 해소하려는 것은 현대 사회에서 매우 중요한 문제가 되었다. 따라서 지금까지 이를 해결하기 위한 다양한 연구가 많이 진행되어 왔고, 계속 진행 중이다. 도로 위에서 현재 위치에서 도착지 까지 가는 데의 과정은 같은 도로 위의 자동차들 수, 목적지까지 거쳐야 하는 교차로의 수, 교통 신호 체제, 다른 운전자들의 운전 형태 등, 여러 가지 요소들에 의해 영향을 받는다. 그리고 위에 제시된 것들을 포함하여 많은 조건들이 다양하게 조작이 가능하며, 어떻게 조작을 하는 것에 따라 교통 혼잡의 정도와 운전자들의 주행의 쾌적성 등이 달라질 수 있다. 이번 연구를 통해서 교통 모델을 만들어 보고 이를 이용하여 조건들을 어떻게 설정함으로써 막힘이 없는 도로 환경을 최대한 구현해 낼 수 있는 지에 대해 조금 더 다가갈 수 있을 것이다. 물론 벌써 지금까지 많은 연구들이 보다 더 다양하고 전문적이고 체계적으로 이루어졌으며 그것을 토대로 다양한 이론과 법칙들이 제시되어 있다. 따라서 우리는 이 연구를 통해서 조금이라도 교통 체제 연구에 도움이 된다면 더욱 좋지만, 이 모델을 만드는 것 자체에 의미를 둘 것이다.

II. 이론적 배경

교통 모델은 보통 연속적인 모델(continuous model)과 이산적인 모델(discrete model) 총 두 가지로 나눌 수 있다. 연속적인 모델은 보통 편미분방정식(이하 PDE: partial differential equation)을 기반으로 하고, 이산적인 모델은 보통 연속적 동적 시스템(이하 SDS: sequential dynamic system)과 세포 자동자(cellular automata)를 기반으로 한다.

또한 이산적인 모델은 셀 기반 모델(cell-based model)과 차 기반 모델(vehicle-based model)로 나눌 수 있으며, 셀 기반 모델은 도로를 작게 구획을 나누어서 말 그대로 도로를 이루는 셀에 정보를 저장시켜서 이 정보를 변화시킴으로써 도로 상태를 나타내는 모델이며, 차 기반 모델은 말 그대로 도로위의 차들에 중심을 두어 차들의 정보를 변화시킴으로써 도로 상태를 나타내는 모델이다.

가장 간단하고 기본적인 셀 기반 모델은 다음과 같은 세포자동자들로 구성된다.

- Φ1 -차선 변경 결정 (lane change decision)
- Φ2 -차선 변경 실행 (lane change execution)
- Φ3 -가속/감속 (acceleration/deceleration)
- Φ4 -이동 (movement)

따라서 $\Phi4 \circ \Phi3 \circ \Phi2 \circ \Phi1$ 를 반복적으로 실행시키며 한 사이클이 된다. 아주 간단한 예로 한 사이클이 1초이고, 도로의 한 셀을 7.5m로 생각하며, 속도 1은 1초당 한 셀을 이동하는 속도로 설정하여 생각하는 경우, 4개의 단계는 다음과 같이 간단히 생각할 수 있다.

1. 차선 변경 결정

바로 앞차가 있는 셀과 현재 차가 있는 셀 사이에 존재하는 셀의 개수(이하 $\Delta(i)$ 이라 한다.)가 최고속도(혹은 제한속도)보다 작고, 바로 옆 셀에 있을 경우의 $\Delta(i)$ (이하 $\Delta(j)$ 라 한다.)보다 작고 (여기까지는 차선변경의 합리성을 판단하고) 바로 옆 셀에 있을 경우 바로 뒤차가 있는 셀과 그 셀 사이의 셀의 개수 (이하 $\delta(j)$ 라 한다.)가 최고속도보다 크면, (안전성을 판단한다.) 차선변경 결정 상태를 1로 저장한다. 그렇지 않은 경우 0으로 저장한다.

2. 차선 변경 실행

차가 차선변경결정에서 차선을 변경할 것으로 결정하였다면, 그 차의 정보를 지우고 옆 셀에 그 차를 입력시킨다. 또한 동시에 한차선의 양쪽의 차선의 차가 그 차선으로 차선변경을 하려는 경우 두 차가 한 번에 같은 셀로 옮겨질 수도 있으므로 보통 짝수 번째 실행에서는 오른쪽으로만, 홀수 번째 실행에서는 왼쪽으로만 차선 변경을 함으로써 이러한 문제를 해결한다.

3. 가속

1. $v := \min(v+1, v_{\max}, \Delta)$ (가속), Δ 는 바로 앞차가 있는 셀과 자신이 있는 셀 사이에 있는 셀의 개수이다.
2. if [UniformRandom() < pbreak] and [$v > 0$], then $v := v - 1$ (확률적인 감속)과 같은 과정을 거친다. 따라서 매초마다 차와의 거리나 제한속도보다 현재속도가 작으면 1씩 증가시키며, 일정한 확률에 따라 1만큼 감속한다.

4. 이동

차가 있는 셀의 차가 속도가 영보다 크면 차를 삭제하고 셀에서 n 칸 뒤의 셀에 $n+1$ 만큼의 속도를 갖는 차가 있으면 그 차의 정보를 이 셀에 입력하며, 모두에 해당되지 않는 경우 셀의 정보를 그대로 둔다.

Ⅲ. 기본 모델

1. 개요

우리가 고안한 모델은 자동차를 기반으로 하는 모델이며, SDS를 바탕으로 하고, 세포자동차를 이용한다고 볼 수 있다. 기본적으로 이 모델을 장소, 차, 사이클로 구분할 수 있다. 장소에는 차가 움직일 수 있는 공간을 나타낸 것이고, 사이클은 차가 운영되는 방식이다. 장소에는 도로가 있으며, 도로의 차선의 수는 조정할 수 있으며, 그 길이도 조정할 수 있다. 또한 도로의 끝을 정지선으로 생각해서 멈추게 할 수도 있고 도로의 일부로 취급하여 그냥 지나가게 할 수도 있다. 차를 생성시킬 때의 초기 속도를 부여할 수 있다. 트럭의 비율도 조절할 수 있고, 차선을 변경할 정도도 조절할 수 있다.

그 외에도 변경할 수 있는 파라미터들이 있으며 이는 다음과 같다. (화면에서는 일부 대문자로 표시되어 있으나, 이 논문에서는 모두 소문자로 표기하였다.)

- a: 차들은 모두 차선변경을 $1/a$ 초안에 끝내도록 예상하고 차를 회전시킨다.(그리고 그 회전각에 따라 속도와 가속도가 분배된다.)
- c: 차선변경을 할 때 감속을 하는 정도를 나타내며, 클수록 감속을 덜하면 작을수록 감속을 더하게 된다.
- e: 차량이 속도가 굉장히 낮은 경우에도 정지하는 경우에도 적어도 유지하려는 거리를 나타낸다.
- f: 차는 제한속도에 가까워질수록 가속도가 줄어들게끔 디자인되어있다. 이때 f값을 늘리면 제한속도에 가까워질수록 가속도가 더 작아지고 f값을 줄이면 제한속도에 가까워져도 가속도가 별로 줄어들지 않는다.
- h: 차는 차선변경을 할 때 내 차선과 옆 차선의 앞차의 속도, 앞차와의 거리를 고려하여 차선변경을 한다. 이때 h값이 커질수록, 그 앞차와의 거리차이를 덜 중요하게 인식하고, 작아질수록, 더 중요하게 인식한다.
- k: 차는 앞차와 부딪치지 않도록 안전거리를 유지하려고 한다. k값이 커질수록 그 유지하려는 정도가 강하고, 작아질수록 그 정도가 약해진다. 쉽게 말해 차들이 더 대담해진다.
- q: 클수록 차선변경을 하려는 경향이 낮아지고, 작을수록 차선변경을 하려는 경향이 커진다.
- r: 클수록 운전자가 트럭이 앞에 있을 때 느끼는 부담감이 커져서 차선변경을 더 하려고 하고, 작을수록 트럭이 있어도 부담감을 덜 느낀다.
- z: 좌회전이나 우회전을 하기 위해 차가 좌회전 차선이나 우회전 차선으로 가려는 차선변경을 하는 경우가 있다. 이 경우 차선변경을 하고 싶은 마음은 있으나 (안전성의 이유로) 차선변경을 하기가 어려운 상황인 경우 차선변경을 할

기회를 엿보며 일정시간동안 기다리게 되는데 z 가 클수록 그 시간이 늘어나고 작아질수록 그 시간이 줄어든다.

- t_{\min} : 차선변경을 하려는 마음은 먹었으나(좌회전차선으로 가려는 경우, 단지 빨리 가고 싶어서 하려는 경우 모두 포함한다), 차선변경을 하기가 어려운 상황인 경우 어떤 상황이던지 차가 기다리는 최소의 시간을 의미한다. (단위는 (초)다.)
- t_{\max} : 차선변경을 해야 더 빨리 갈 수 있을 것 같아서 차선변경을 하려고 마음을 먹었으나, 차선변경을 하기가 어려울 때 차가 기다릴 최대의 시간을 의미한다. (단위는 (초)다.)
- Max Vision: 운전자의 시야이며 단위는 (m)이다. 만약 앞차와의 거리를 계산하려는데, 시야 내에 차가 없으면 그 값을 시야로 대체한다.
- Max angle(θ_{\max}): 차가 차선 변경 시 최대로 회전할 수 있는 각도이다.(단위는 $^{\circ}$ 이다)
- Max Accel(a_{\max}): 차가 낼 수 있는 최대 가속도이다. (단위는 m/s^2 이다)
- Min Accel(a_{\min}): 차가 낼 수 있는 최소 가속도이다. 즉 차의 제동의 한계이다. (단위는 m/s^2 이다)
- Init Accel: 차를 생성할 때에 부여하는 초기 가속도이다. (단위는 m/s^2 이다)
- Max Speed(v_{\max}): 도로의 제한속도이다. (단위는 m/s 이다)
- Init Speed: 차를 생성할 때에 부여하는 초기 속도이다. (단위는 m/s 이다)
- Fast Forward: 시뮬레이션의 빠르기이다. 1을 입력하면, 실제속도로 보여주고, 3을 입력하면 3배로 보여주며, 단위시간은 3배가 된다. 따라서 이 수가 커지면 실험결과는 금방 얻을 수 있으나 그 결과는 정확성이 더 떨어질 수 있다.
- Truck Ratio: 생성되는 차들 중 트럭의 비율이다.
- Min Lane Change Y: 도로의 차선변경을 못하게 할 초반 길이이다. 단위는 (m)이며, 예를 들어 100으로 두면 첫 100m이내에서는 차선변경을 하지 않는다.
- Road Length: 도로 길이이다. 단위는 (m)이다.

이 시뮬레이션은 rawgit.com/justinjuncl/TrafficSimulation/master/index.html에서 할 수 있다.

2. 장소

차들은 항상 장소에 위치한다. 장소는 도로, 생성기, 소멸기가 있다. 교차로는 시도하였으나, 아직 도입하지는 못하였다. 도로에는 차선이 있고, 차는 그 위에서 움직인다. 각 차선은 고유한 길이와 제한속도가 있으며, 그 차선 위의 차들이 위치한 순서대로 1차원 행렬로 저장되어있다. 이를 통해 어떤 차든지 바로 앞차를 쉽게 찾을 수 있다. 도로에는 이러한 차선의 개수와 시작점, 종점에 대한 정보가 들어있다. 도로의 시작점과 종점은 각각 차 생성기와 소멸기이며, 교차로를 만들게

되는 경우 교차로와 만나는 곳일 수도 있다. 생성기는 도로의 시작점에서 차를 랜덤으로 만들어내며, 소멸기는 도로의 종점을 통과하는 차들을 사라지게 한다.

3. 차

차는 변하지 않는 여러 가지 데이터들이 있으며 이에겐 차의 너비, 길이, 최대가속도, 최소가속도, 앞차와 유지하고 싶은 최소 거리 등이 속한다. 그리고 계속해서 변하는 데이터도 있는데, 이에겐 차의 위치, 차선변경을 할 방향, 마음 등이 있다. 차는 매 사이클마다 다른 차들의 정보를 바탕으로 이러한 데이터들이 결정되며 이를 바탕으로 차가 움직인다고 볼 수 있다.

도로 위의 차의 위치는 (차선, x , y)로 저장되며 필요에 따라서는 (차선, y)의 형태로 x 값을 생략하여 보여주기도 할 것이다. 차선은 1번째 차선이면 0이 2번째 차선이면 1이 저장된다. x 는 -(차선 폭)과 +(차선 폭)사이의 값이며 한 차선 내에서 차선에 수직인 방향의 좌표이고, y 는 0과 (차선 길이)사이의 값이며 차선과 평행한 방향의 좌표이다. 참고로 y 좌표는 차의 가장 뒷부분을 기준으로 한다. 그래서 차의 앞부분의 좌표는 y 좌표 + 차의 길이로 구할 수 있다.

또한 일정한 비율로 트럭이 생성되는데, 트럭의 경우, 제한속도, 최대가속도를 낮게 하고, 최소가속도는 크게 한다. 또한 트럭이 바로 앞에 있다면 일반 운전자들은 부담스러워하며, 차선을 변경하려는 경향이 조금 더 커진다.

4. 그 외에 사용할 용어들

- i : 차가 갈 방향을 나타내며, 그 값이 -1, 0, 1이면 각각 왼쪽, 가운데, 오른쪽으로 갈 것을 나타낸다.
- Δ : 바로 앞차까지의 거리로 차의 앞부분의 위치와 앞차의 뒷부분의 위치와의 거리로 계산한다. Δ_i 는 차가 있는 차선의 i 번째 방향에 있는 차선에 있다고 가정할 시에의 Δ 이다.
- γ : 바로 뒤차까지의 거리이다. γ_i 는 차가 있는 차선의 i 방향에 있는 차선에 있다고 가정할 시에의 γ 이다.
- s_l 또는 안전거리: $\frac{81}{625}v^2$
- dt : 각 사이클 사이의 시간 간격

5. 사이클

모든 차들은 시뮬레이션을 거치면서, 크게 2가지, 작게는 4가지의 사이클(세포자동차)을 거친다. 차는 판단(현재 상황을 바탕으로 다음의 속도, 차선변경결

정 등)하고, 실행(실제로 이동)을 한다. 각 판단과 실행은 도로방향(y방향)과 도로와 수직인 방향(x방향)에 대한 사이클로 총 4개의 사이클이 존재한다. 자세히는 다음과 같이 생각할 수 있다.

1. 차선 변경 결정: 차선 변경을 하는 것이 합리적이고 안전한지 등을 판단한다.
2. 차선 변경 실행: 차선변경을 실제로 실행한다.
3. 속도 변경: 자신의 속도를 주변상황에 맞추어 변경한다.
4. 위치 변경: 바뀐 속도에 따라서 차의 위치를 실제로 변경한다.

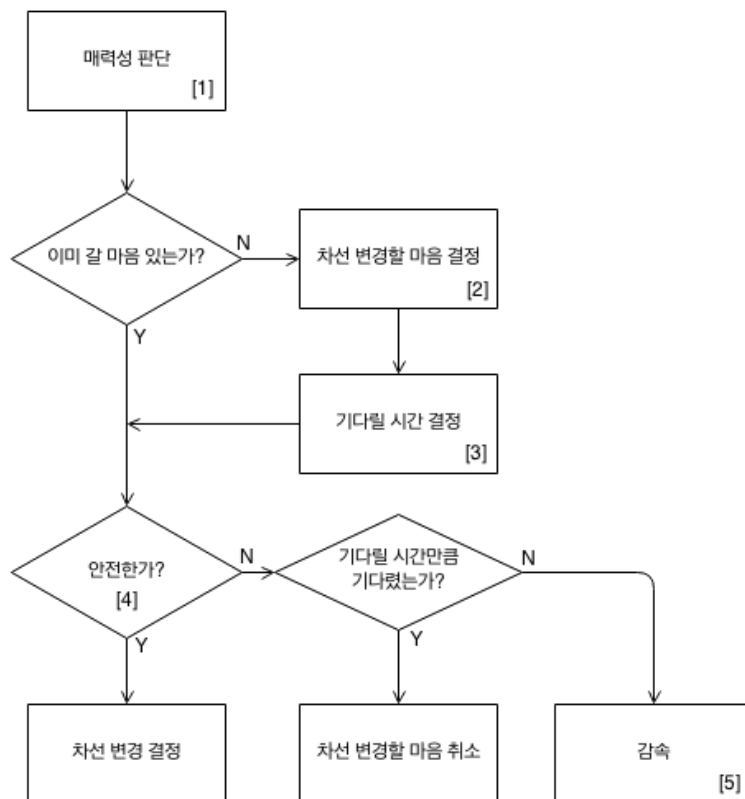
5.1. 차선 변경 결정

차선 변경을 하는 경우는 다음 두 가지로 나눌 수 있다.

경우1) 차선을 바꾸는 것이 더 빨리 갈 것처럼 보이는 경우

경우2) 좌회전 차선과 같이 다른 도로로 가기 위해 필연적으로 변경하는 경우

경우1) 차선을 바꾸는 것이 더 빨리 갈 것처럼 보이는 경우



[1] $\left(1 - \frac{v_f}{v_{\max}} \left(\frac{\Delta}{\Delta_i}\right)^h\right)^{\frac{q}{1+r(\text{truckstate})}}$ 의 값을 비교해서 가장 높은 차선이 현재차선과 오차 범위보다 큰 차이가 나면 매력적이라고 판단한다. $h=1$ 과 0 사이의 상수이다. truckstate는 바로 앞의 트럭의 유무이며 있으면 1 없으면 0이다. 또한 주체가 트럭인 경우 truckstate는 항상 0이다.

[2] 확률은 $\left(1 - \frac{v_f}{v_{\max}} \left(\frac{\Delta}{\Delta_i}\right)^h\right)^{\frac{q}{1+r(\text{truckstate})}}$ 로 결정 q 는 상수이며, 값을 늘리면 차선변경을 하는 빈도수가 줄어들며, 줄이면 빈도수가 늘어난다. r 은 상수이며, 커질수록 운전자가 앞에 트럭이 있으면 차선변경을 더 하려고 하며, 작아질 수도 트럭이 있어도 차선변경을 덜 하려고 한다. 마음이 정해진 이후에는 깜박이를 킂다.

[3] 기다리는 시간은 $(1-p)t_{\min} + pt_{\max}$

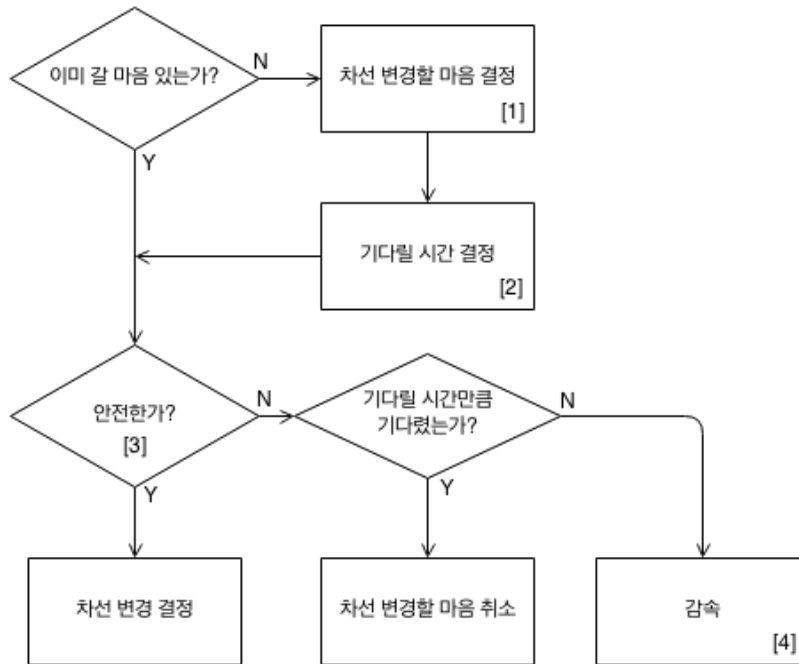
[4] 안전성은 차선을 바꿨다고 가정할 시 앞차와 뒤차와의 거리가 각각 안전거리의 $k(0 < k < 1)$ 배 이상일 때 안전하다고 판단

[5] 속도를 현재 속도의 $(1-p^k)$ 배로 결정 k 는 1보다 큰 상수(속도 변환 참고), p 는

$$\left(1 - \frac{v_f}{v_{\max}} \left(\frac{\Delta}{\Delta_i}\right)^h\right)^{\frac{q}{1+r(\text{truckstate})}}$$

따라서 차는 자신이 있는 차선과 옆 차선에 있는 앞 차량의 속도를 비교함으로써 더 빨리 가는 차선을 택하며, 비슷하면 앞차와의 거리를 비교하여 더 많은 거리가 남은 차선으로 가려고 한다. 또한 그렇게 해서 실제로 차선을 변경을 하려고 마음을 먹을 확률은 옆 차선의 앞차의 속도가 크면 클수록 앞차와의 거리가 크면 클수록 커진다. 마음을 먹은 다음에는 깜박이를 키고 감속을 서서히 하며, 차선을 변경할 기회를 엿보며 차선을 바꾸기에 안전한 상태면 차선을 변경한다. 차선을 변경할 마음을 먹을 확률은 옆 차선이 얼마나 더 매력적인지를 보여주므로 그 확률이 크면 클수록 차선 변경할 기회를 엿보며 기다리는 시간은 늘어나고 감속도 더 많이 한다. 또한 차선을 바꿀 기회를 엿보며 기다리고 있는데 지금 있는 차선의 상황이 더 좋아지면 차선변경을 할 마음을 접는다.

경우2) 좌회전 차선과 같이 다른 도로로 가기 위해 필연적으로 변경하는 경우



[1] $\left(\frac{L-l}{L}\right)^{\frac{1}{n}}$ 의 확률로 정함 L은 적당히 큰 수로 여기서는 도로의 길이로 적용시켰다. l은 좌회전 혹은 우회전을 해야 하는 교차로까지 남은 거리이다. n은 그 차선까지 가기 위해 해야 하는 차선변경의 수이다. 마음을 결정하였다면 깜박이를 켜고.

[2] 기다리는 시간은 $\frac{t_{\min}}{(1-p)^z}$

[3] 차선을 바꿨다고 가정할 시 앞차와 뒤차와의 거리가 각각 안전거리의 $k(0 < k < 1)$ 배 이상일 때 안전하다고 판단한다.

[4] 속도를 현재 속도의 $(1-p^c)$ 배로 결정하며, c는 1보다 큰 상수이다. (속도 변환 참고), p는 $\left(\frac{L-l}{L}\right)^{\frac{1}{n}}$ 로 차선 변경할 마음을 먹을 확률이다.

좌회전을 하기 위해 좌회전 차선으로 가야하는 경우로 생각하자(물론 우회전을 해야 하는 경우나 고속도로에서 다른 도로로 빠지기 위함 등 다른 도로로 가기 위해 차선을 변경해야 하는 경우는 동일하게 적용된다.). 그러면 좌회전 차선으로 바꿀 수 있는 남은 거리 즉 교차로까지의 거리가 많이 남으면 많이 남을수록

여유를 부리며 차선을 덜 바꾸려고 하고, 그 차선으로 가기위해 거쳐 가야 하는 차선의 수가 많으면 많을수록 급해져서 차선을 바꿀 확률이 커진다. 마음을 먹은 다음에는 깜박이를 키고 감속을 서서히 하며, 차선을 변경할 기회를 엿보며 차선을 바꾸기에 안전한 상태면 차선을 변경한다. 차선을 변경할 마음을 먹을 확률은 운전자가 느끼는 급박함을 보여주므로 그 확률이 크면 클수록 차선 변경할 기회를 엿보며 기다리는 시간은 늘어나고 감속도 더 많이 한다.

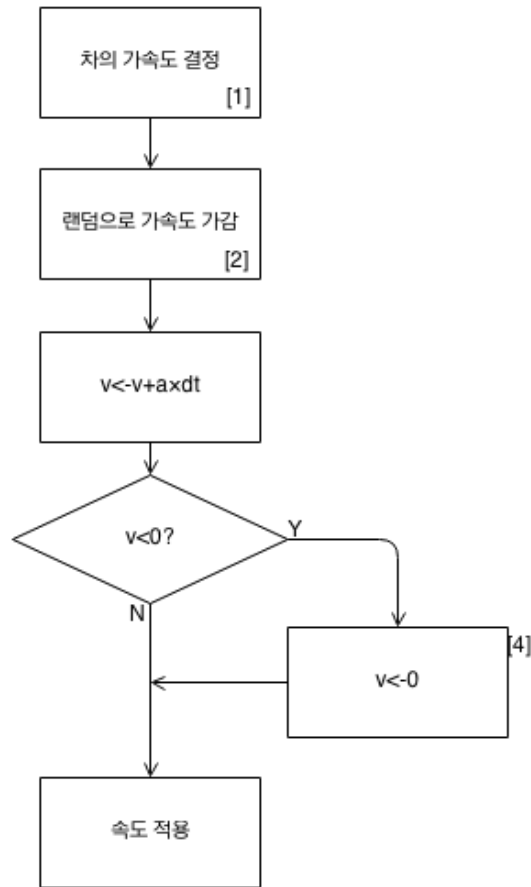
5.2. 차선 변경 실행

차를 $\sin^{-1}a\frac{w}{v}$ 만큼 회전시키며, 차가 완전히 반대편 차선에 도착하면, 다시 차를 원래대로 회전시켜 놓는다. w 는 차선의 폭이다.(a 는 $\frac{4}{11}$ 로 정하며, 필요에 따라서는 정규분포를 적용하여, 이 주변 값으로 랜덤으로 적용한다.) 그리고 차가 최대한 회전할 각을 설정하여 이를 θ_{\max} 라고 하면, 만약 $v \leq \frac{aw}{\sin\theta_{\max}}$ 인 경우 회전각을 θ_{\max} 로 설정한다.

따라서 특별한 일이 없어서, 이 속도를 유지할 경우 약 2.75초정도가 걸려서 차선변경이 완료되도록 차를 회전시킨다.

속도와 가속도는 속도변경에서 결정된 대로 따르며, 차선방향(y방향)과 차선에 수직인 방향(x방향)의 속도와 가속도는 차가 θ 만큼 원래상태보다 회전된 상태라면, $v_y = v\cos\theta$, $a_y = a\cos\theta$, $v_x = v\sin\theta$, $a_x = a\sin\theta$ 로 결정된다.

5.3. 속도 변경



$$[1] \ a = \max(\min(k(\Delta - d - sl - (v - v_f)) - v, na, -Ap^c v), a_{\min})$$

따라서 가속도가 a_{\min} 보다 작아지면, a_{\min} 을 적용시킴으로써 차가 무한정으로 감속할 수 없게 만든다.

$$[1]-1. \ k(\Delta - d - sl - (v - v_f)) - v$$

앞차와의 안전거리를 유지하기 위한 항이다.

만약 $(\Delta - d - sl - (v - v_f)) > 0$ 이면 k의 값은 0부터 1사이의 상수이고,

만약 $(\Delta - d - sl - (v - v_f)) < 0$ 이면 k의 값은 1이다.

d는 적당히 작은 수를 적용하며, 차가 심지어 서로 정지해있더라도 서로 간에 유지하려는 거리를 의미한다.

$k(\Delta - d - sl - (v - v_f)) - v$ 의 값은 사람의 공주시간을 고려하여 약 1초전의 값을 바탕으로 적용한다. (그러나 컴퓨터로 구현하는 경우 모든 1초전의 값을 계속 저장하고 있기에는 너무 많은 메모리가 소비되므로, 0.5초마다 값을 저장하여, 매 순간마다 1초전에 가까운 값으로 적용한다.)

또한 만약 $(\Delta - d - sl) < 0$ 일 때 가속도가 0보다 크게 나오면, 가속도 값은 0으로 결정한다. 따라서 안전거리까지 합하여 앞차와 유지하려는 거리보다 실제거리가 짧은 경우에는 가속을 하면 안 되는 상황이므로 가속도가 0보다 크게 나오면, 가속도 값은 0으로 결정한다.

또한 차선변경을 마음에 두고 있는 경우 가려는 옆 차선의 앞 차량과의 델타와 바로 앞 차량과의 델타 중 최솟값으로 계산한다.

따라서 이 모델에 따르면, 앞 차량이 어느 정도 가까이 있으면 앞 차량과의 안전거리보다 앞 차량이 더 멀리 있다면 서서히 가속함으로써 안전거리를 넘지 않도록 하고, 안전거리보다 앞 차량이 더 가까이 있으면 더 빨리 안전거리를 유지할 수 있도록 감속한다. 또한 옆 차선에 있는 앞쪽의 차가 깜박이를 키면 정해진 확률에 따라 그 차가 차선을 변경을 하도록 감속할지 말지를 판단하며, 감속하기로 마음 먹었다면, $k(\Delta - sl - (v - v_f)\Delta t)$ 에서 sl , Δ , v_f 등은 그 차가 앞에 있다고 계산하며, k 는 1과 0 사이의 값으로 계산함으로써 서서히 감속하도록 한다.

[1]-2. na

앞에 차가 멀리 있어서 거의 아무런 제약 없이 가속하는 경우를 나타내는 항이다.

만약 $v < v_{\max} - e$ 이면 na 는 $a_{\max} \left(1 - \frac{v}{v_{\max}}\right)^f$ 이고,

만약 $v_{\max} - e < v < v_{\max} + e$ 이면 na 는 0이고,

만약 $v_{\max} + e < v$ 이면 na 는 $a_{\max} \left(1 - \frac{v}{v_{\max}}\right)$ 이다.

여기서 e 는 작은 수로, 제한속도 부근을 결정한다.

일정확률로 $na=0$ 으로 정하며 이 확률은 낮다.

그리고 $na=0$ 이 되면 약 1초간 $na=0$ 으로 계속 유지시킨다. 그리고 정규분포를 이용하여 차들로 하여금 그 시간은 1초 내외로 랜덤하게 정한다.

따라서 앞에 차가 충분히 떨어져 있을 시에는 제한속도에 다가갈수록 조금씩 더 약하게 가속하다가 제한속도에 가까우면 가속을 하지 않으며, 제한속도 부근에 있

으면 가속이나 감속을 하지 않고, 제한속도를 넘으면 감속을 한다.

[1]-3. $-Ap^c v$

차선변경하고 싶어서 기다릴 때 하는 감속을 나타낸다.

p는 차선 변경 마음을 먹었을 시의 확률이고

A는 p가 특정 정해진 확률보다 클 경우 1 아니면 영으로 결정된다.

c는 1보다 큰 상수다.

따라서 차선변경을 하고 싶은 정도가 어느 정도 이상이면, 서서히 감속하게 된다.

또한 트럭을 적용시킬 경우, a_{\min} 과 v_{\min} 은 일반차보다 큰 값을, a_{\max} , v_{\max} 은 더 작은 값을 적용시킨다.

따라서 트럭이 가속은 더 천천히 되고, 더 작은 속도로 달리며, 조금 더 천천히 감속을 하며 앞차보다 조금 더 큰 안전거리를 유지하도록 한다.

[2] 평균이 주어진 가속도이고 표준편차는 Δ 에 비례하게 하며 일정범위이외의 값은 자른다.

그리고 가속할 때는 항상 현실성을 부가하기 위해 정규분포를 이용하여 차마다 같은 상황에서도 약간씩 다른 가속도를 부과하며, 그 오차 폭은 앞차와의 거리가 작아질수록 작아진다.

5.4. 위치 변경

차의 새로운 좌표는 (차선, $x+v_x \times dt, y+v_y \times dt$)로 저장한다.

그런데 만약 새로운 x좌표가 -(차선의 폭)보다 작거나 +(차선의 폭)보다 크면 차선 변경을 완성한 것이기 때문에 차선이 바뀌고 x좌표는 그에 맞게 변경이 된다.

또한 만약 새로운 y좌표가 (도로의 길이)를 넘어가버리면 도로를 벗어난 것이기 때문에 그 도로에 연결된 다음 장소로 넘어간다.

IV. 연구 결과 및 해석

이 모델을 이용하여 간단한 실험을 해보았다.

1. 모델을 이용한 실험의 개요

기술의 발달로 차들을 전체적으로 통제 관리함으로써 교통체증을 덜 일어나게 할 수 있는 상황이 올 수 있다. 그리고 그 중 가장 기초적인 상황은 두 차선의 교통량이 차이가 날 때 교통량이 많은 차선에서 상대적으로 적은 차선으로 차량을 분배해보는 상황을 생각해볼 수 있다. 이 상황을 직접 실험해본다.

파라미터는 다음과 같이 맞춘다.

A: 4/11 C: 2 E: 1 F: 1 H: 1 K: 0.5 Q: 1 R: 1 Z: 1 tMin: 3 tMax: 10 Max Vision: 500 Max Angle: 30 Max Accel: 4 Min Accel: -4 Init Accel: 3 Max Speed: 27 Init Speed: 20 Fast Forward: 10 Truck Ratio: 0.05 Min Lane Change Y: 300 Road Length: 1000

차선은 두 개(a와 b차선)만 있고 실험 방법을 이렇다:

1. 시뮬레이션을 실행시킨 후 처음에는 상황을 조작하지 않은 채로 차 100대가 전체도로를 통과한 후부터 상황을 변경시킨다. (따라서 일반적인 도로 상황이 만들어진 후부터 실험을 한다.)

2. 100대가 통과한 이후부터는 a차선의 차들 중 일정한 비율로 차선변경을 허용하며 허용된 차는 오로지 b차선으로만 갈 수 있으며, 안전성과 매력성이 확보된 상태면, 확률은 적용시키지 않고 무조건 차선변경을 한다. 그 비율은 1/50에서 50/50까지 총 50가지를 실험한다.

구하는 데이터는 600초 동안 전체도로를 통과하는 차 대수, 전체도로를 지나가는데 걸리는 평균시간이며 이 데이터들을

- a차선에서 차선변경을 하지 않아 a차선으로 도착한 차들
- a차선에서 b차선으로 차선변경한 차와 b차선에서 이동하던 차들을 합한 총 b차선으로 도착한 차들
- a차선에서 생성된 차들
- b차선에서 생성된 차들
- a차선에서 생성되어 차선변경을 하여 b차선으로 도착한 차들
- 전체 차들

로 나누어서 구해본다.

2. 세부적인 실험 결과

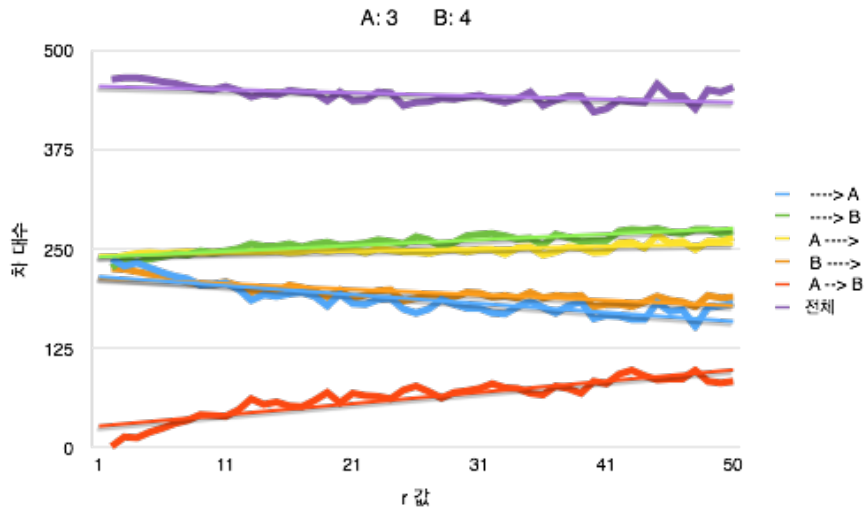
우리는 다음과 같은 3가지 실험을 할 것이다.

- 실험1) a차선에는 차들을 3초 주기로 b차선에는 4초 주기로 차들을 생성 시
- 실험2) a차선에는 차들을 2초 주기로 b차선에는 4초 주기로 차들을 생성 시
- 실험3) a차선에는 차들을 1초 주기로 b차선에는 4초 주기로 차들을 생성 시

그래프에서 가로축인 $r/50$ 은 a차선의 차들 중 b차선으로 분해하는 비율이다. 그리고 최소자승법으로 자료를 대표할 수 있는 직선을 같이 나타내었다. 이를 통해 자료의 경향성을 알 수 있게끔 하였다.

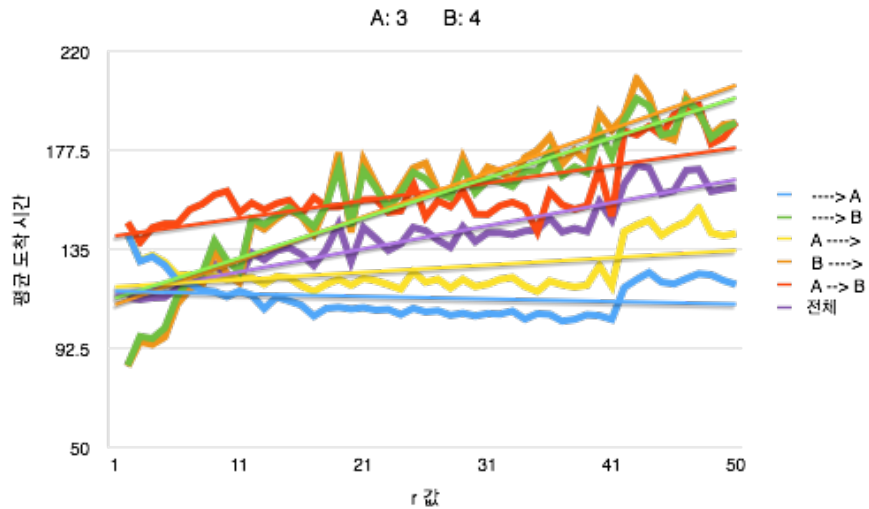
실험1) a차선에는 차들을 3초 주기로 b차선에는 4초 주기로 차들을 생성 시

차의 대수는 다음과 같다.



따라서 인위적으로 차들을 더 차선을 변경시키도록 조작할수록, 전체적으로 통과하는 차들의 대수는 적었다. 그러나 교통량이 더 많은 a차선에서 출발한 차들은 더 많이 통과하였다. 그러나 b차선에서 출발한 차들은 더 적어진 것을 보니, 인위적으로 차를 더 분배할수록 확실히 a차선에게는 좋고, b차선에는 안 좋은 것을 알 수 있다. 그러나 전체적으로는 상황을 더 안 좋게 하는 것임을 알 수 있다.

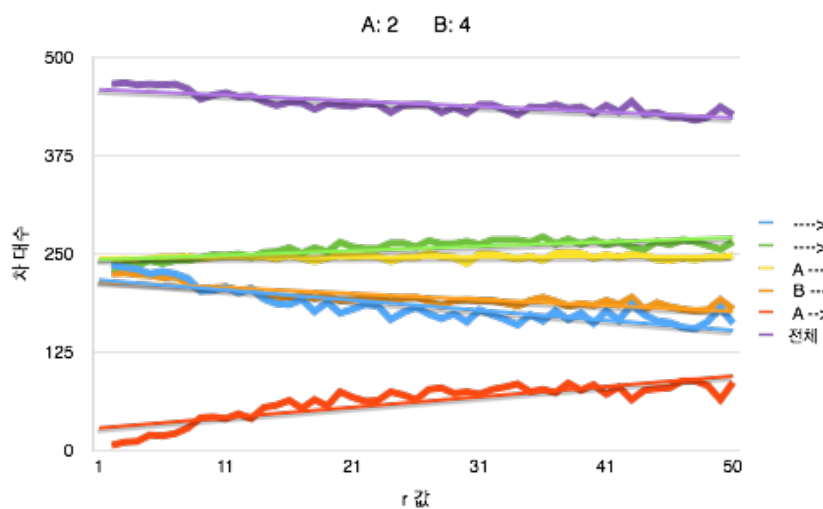
차들의 평균 이동시간은 다음과 같으며



b차선으로 차를 더 보낼수록 평균 이동시간은 전체적으로 높아지고, b차선의 차들이 시간이 더 걸리게 됐으며, a차선의 차들, a차선에서 도착하는 차들 심지어 차선 변경하는 차들도 걸리는 시간이 전반적으로 작아지다가 갑자기 커지는 것을 볼 수 있다. 따라서 적어도 어느 정도까지는 차를 b차선으로 내보는 것이 b차선의 차들에게는 손해 a차선의 차들에게는 이익이지만, 지나치게 내보내면 모두에게 손해임을 알 수 있다. 50대중에 5대 정도를 b차선으로 내보낼 때 a차선과 b차선의 교통상황이 비슷해진다.

실험2) a차선에는 차들을 2초 주기로 b차선에는 4초 주기로 차들을 생성 시

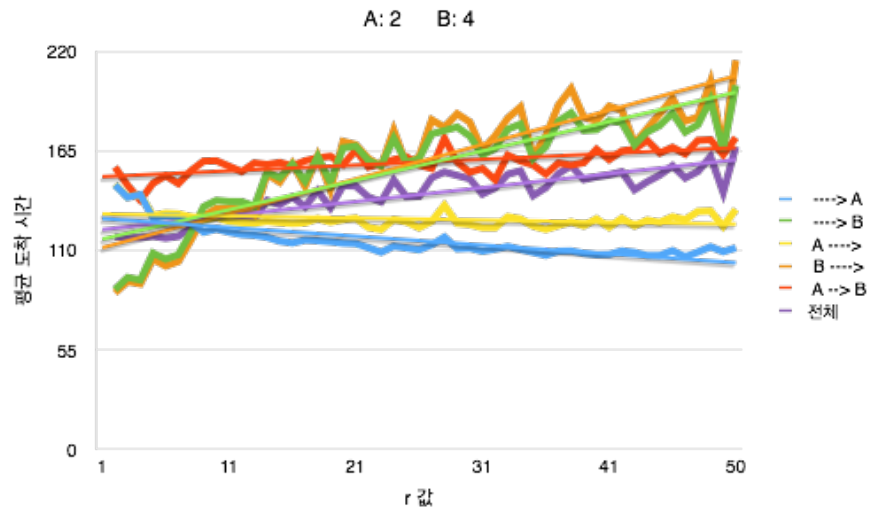
차의 대수는 다음과 같다.



따라서 인위적으로 차들을 더 차선을 변경시키도록 조작할수록, 전체적으로 통과하는 차들의 대수는 적었다. 그러나 교통량이 더 많은 a차선에서 출발한 차들은 더 많

이 통과하였다. 그러나 b차선에서 출발한 차들은 더 적어진 것을 보니, 인위적으로 차를 더 분배할수록 확실히 a차선에게는 좋고, b차선에는 안 좋은 것을 알 수 있다. 그러나 전체적으로는 상황을 더 안 좋게 하는 것임을 알 수 있다.

차들의 평균 이동시간은 다음과 같으며,



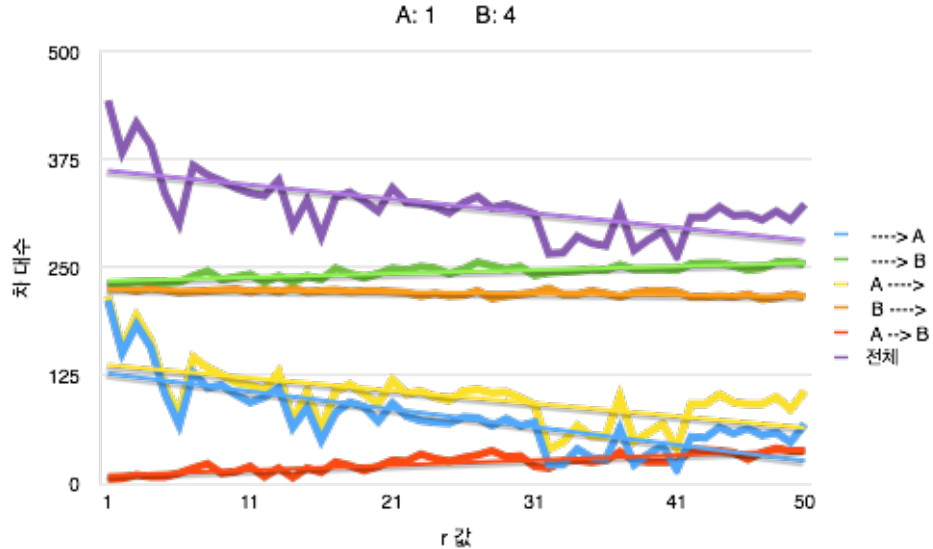
오히려 차선변경을 인위적으로 더 할수록, 전체 이동시간은 늘어나며, b차선에 있던 차들, b차선에서 도착한 차들 모두 더 큰 손해를 보았다. 차선변경을 한 차들도 전체적으로 더 손해를 보았다. 그러나 a차선에 끝까지 남아있었던 차들은 오히려 더더욱 이득을 보는 것을 알 수 있다.

결국 전체적으로 인위적으로 차들을 b차선에 더 분배시킬수록, 원래대로라면 b차선의 차들이 더 빠르게 이동하였으나, 점점 b차선의 차들은 느려지고, a차선의 차들 특히 a차선에 계속 남게 한 차들은 더 빨리지는 것을 알 수 있다.

a차선과 b차선의 차들의 속도는 a차선의 차를 약 50대중 7~8대를 b차선으로 보낼 때 비슷해지는 것을 볼 수 있다.

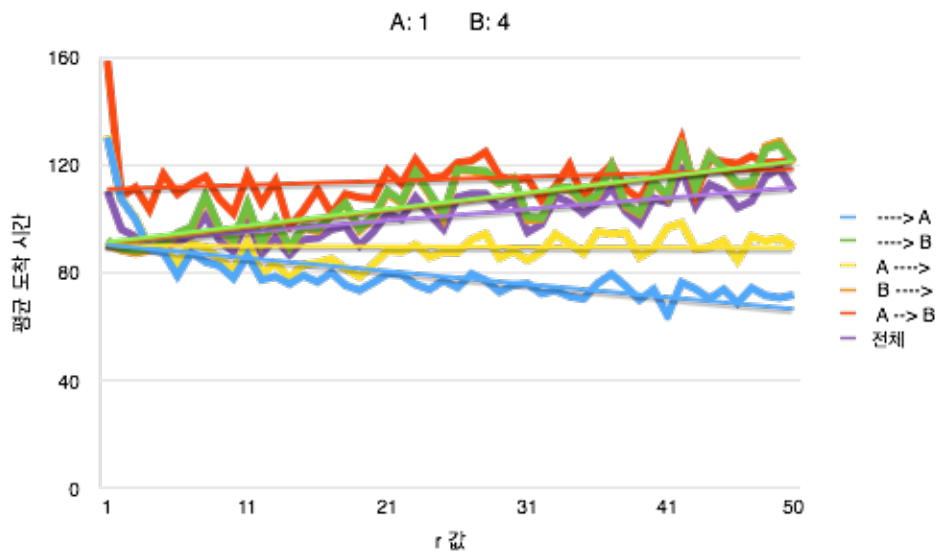
실험3) a차선에는 차들을 1초 주기로 b차선에는 4초 주기로 차들을 생성 시

차 대수는 다음과 같으며,



전체적으로는 전체도로를 통과하는 차 대수는 차선 변경하는 차들의 비율이 늘어날 수록, 점점 줄어들다가 하는 경향이 나타내는데 이는 a차선의 차들 역시 그러하다. 결국 a차선의 차들이 전체적인 상황에 큰 영향을 끼쳤고 가장 영향을 많이 받았음을 알 수 있다. 오히려 a차선의 차들을 b차선으로 보냈으나, 그 상황이 a차선에는 더 안 좋았음을 알 수 있다. 그러나 오히려 b차선의 차들은 큰 영향을 받지 않았음을 알 수 있다.

차들의 평균이동시간은 다음과 같으며,



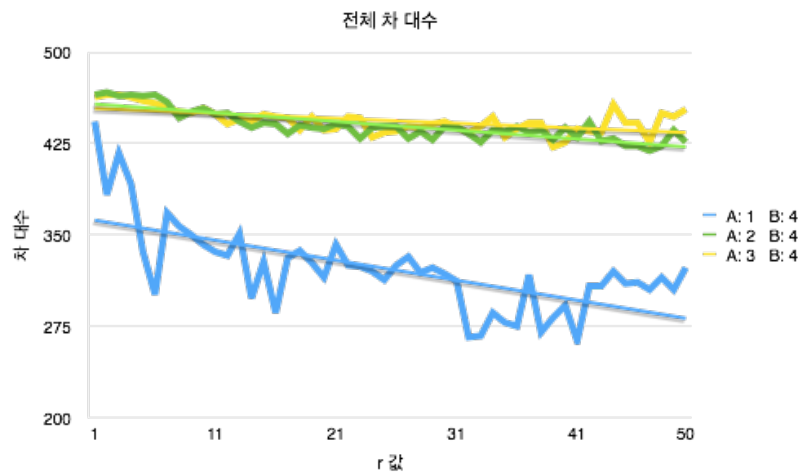
차를 조금만 분배했을 경우 그렇지 않았을 때보다 a차선의 교통상황은 훨씬 좋았다. 그러나 더 분배할수록 오히려 a차선에는 악영향을 끼쳤음을 알 수 있다. 또한 b차선

의 교통상황은 a차선의 차가 들어오면 울수록 당연히 더 안 좋아졌다. a차선과 b차선의 교통상황은 a차선의 차를 50대 중 3~4대를 b차선으로 보냈을 때 비슷하였다.

3. 종합적인 결과

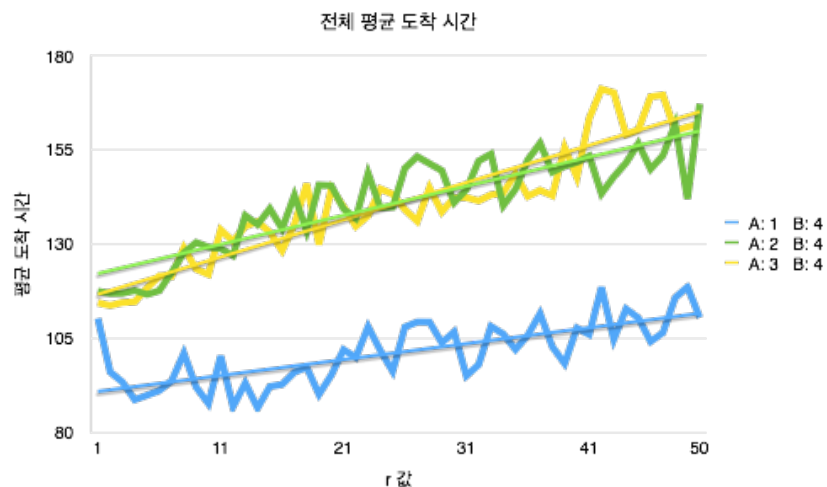
세 가지 경우를 동시에 비교해보면,

차의 대수는 다음과 같다.



이를 살펴보면, a차선의 차 생성주기가 2초인경우와 3초인 경우가 거의 비슷함을 알 수 있으나, 1초인 경우는 더 작고, 더 크게 감소하는 것으로 보아, b차선으로 보내는 차의 수를 늘리는 것에 더 민감하게 영향을 받고, 더 안 좋은 영향을 받음을 알 수 있다.

차의 평균 이동시간은 다음과 같다.



이 역시 a차선의 차 생성주기가 2초일 때와 3초일 때는 큰 차이가 없으나, 1초일 때는 도착시간이 훨씬 작다.

4. 결과 해석 및 의의

전체적인 실험을 정리하면, 두 차선이 있을 때 교통량에 차이가 있더라도 교통량이 많은 쪽에서 적은 쪽으로 차를 보낸다고 상황이 나아지기보다는 더 악화되었다. 또한 교통량의 차이가 적을 때에는 차를 보낼 때 교통량이 많았던 차선이 더 이익을 보고 적은 차선이 더 피해를 보았지만 교통량의 차이가 큰 경우에는 두 차선 모두에게 피해가 되었다.

그러나 이 실험은 단 한번밖에 안 했기 때문에 신뢰성이 떨어지고, 또한 인위적으로 분배해서 순전히 매력성이 더 높음과 안전성이 갖추어 지기만 하면 무조건 차선변경을 하기 때문에 실제로는 상황을 조금 더 고려해가면서 차선변경을 하는 데 시간이 별로 안 걸릴 것 같고, 쉬워 보일 때 할 것이며, 감속도 상황에 따라서는 덜 할 수 있다. 그리고 파라미터의 값들이 현실과는 동떨어질 수도 있다. 또한 차들을 도로 앞에서 생성되었기 때문에 생성주기가 짧은 경우 도로 앞부분에 유독 더 큰 정체가 일어나곤 했기 때문에, 실제 상황과 약간 거리가 있을 수는 있다.

V. 결론

1. 의의

위의 실험으로 알 수 있듯이 우리가 만든 기초적인 모델은 그래도 어느 정도는 현실적인 자동차들의 움직임을 재현할 수 있다고 볼 수 있다. 차선 변경하기 위해 눈치를 보는 행위, 앞차와의 안전한 간격을 유지하며 가속 또는 감속하는 행위, 앞차가 트럭이면 피하려는 성질들 미세한 것들까지 구현하려 노력을 했고 그것이 모델에도 반영되었다. 그리고 현실에서도 볼 수 있는 것처럼, 각 차는 도로 위를 움직일 때 오직 그의 주위 차들만 고려한다. 이를 바로 알 수 있는 현상으로 이 모델을 통해 시뮬레이션을 하는 경우, 최초로 생성된 차가 가속을 하는 동안 그 뒤에 새로운 차들이 생성되고, 아직 그 첫 번째 차가 완전히 가속을 못해서 뒤 차들이 조금 조금씩 막히는 지연 효과를 볼 수 있었다. 그리고 시간이 조금 지나면 여러 가지 상황으로 인해 차가 가속과 감속을 주기적으로 반복하는 현상을 볼 수 있었는데, 이 현상 때문에 차들이 밀집된 부분과 그렇지 않은 부분이 파동처럼 뒤로 점진하면서 일어났는데, 이 현상은 실제로도 자주 일어난다. 또 관찰 할 수 있었던 현상은, 어떤 차가 차선을 변경하려 할 때 가려는 차선에는 여러 다른 차들이 이미 빠르게 달리고 있어서, 어쩔 수 없이 감속을 하고 끝내 정지를 하였다. 그로인해 그의 뒤차들은 어쩔 수 없이 정지를 하거나, 그 차선으로 변경하는 현상을 볼 수 있었다.

이렇게 우리 교통 모델은 도로와 교차로로 이루어진 작은 도로망에 위치한 차들을 연구하기에도 적합할 정도로 미시적인 모델이다. 미시적인 모델은 차들의 자세한 움직임 하나하나가 실험에 큰 영향을 미치기 때문에 때때로 교통 흐름 분석자들은 도로의 방대한 데이터를 분석한다. 이 때문에 우리 실험에서 많은 요소를 고려해야 하는 점이 있었고, 덕분에 현실에 더 가까워질 수 있었다.

2. 아쉬운 점 및 미래 연구

실제로는 교통 분석을 위해 기본적인 모델을 간단히 만들자는 목적으로 이 연구를 시작하였으나, 이 모델의 창작 과정에 노력이 너무나 많이 들어가 이 연구의 방향이 조금 틀어지기도 했다. 그러면서 많은 관문들에 부딪치면서 연구하는데 시간이 많이 걸렸고, 따라서 연구를 하다가 결국 실현시키지 못한 부분이 많다. 이 중 몇 가지를 언급 하겠다.

2.1. 교차로

도로는 원래 생성기와 소멸기만을 도착점과 종점으로 할 계획이 아니었다. 현실 세계에 있듯이 3차선, 4차선과 같은 교차로까지도 포함하려 하였다. 그렇다면 너무 많은 추가 연구를 했었을 것이다. 예를 들어 신호 체제, 교차로 위에서의 비보호 우회전, 교차로 위에서의 차들의 회전 궤적의 이동 등 여러 복잡한 시스템들이 있다. 이번 연구에는 이 교차로 모델의 기본적인 구상은 간단하게 끝냈지만, 실제로 컴퓨터상의 구현까지는 시간이 부족해 끝내지 못하여 아예 논문에서는 내용을 언급하지 않았다. 만약 이 교차로를 구현 했더라면 고속도로만이 아닌 일반 도심 부근의 교통 체제까지 연구할 수 있었을 것이다.

2.2. 곡선 도로

지금의 도로는 한 지점과 또 다른 지점을 잇는 일직선 도로이다. 그러나 현실에서는 구부러진 도로도 존재한다. 그래서 이것을 표현하기 위해 우리는 원형 호로의 근사, Bezier 곡선, Catmull-rom 스플라인 곡선 등 여러 가지 곡선들을 시도해 보았다. 그러나 한 도로를 표현하기 위해서는 차선이 여러 개 있는 것을 구현해야 하므로 쉽게 단 하나의 곡선으로 나타내기가 어렵고, 그 위에서 차가 이동하는 것을 구현하기에도 여러 난관들이 있어서 연구 시간의 한계로 인해 간단하게 직선 도로만을 구현해 낼 수밖에 없었다. 곡선 도로를 구현할 수 있었다면 더더욱 제한적이지 않고 다양한 도로망을 표현할 수 있었을 것이다.

참 고 문 헌

1. Henning S. Mortveit, Christian M. Reidys(2007),『*An Introduction to Sequential Dynamical Systems*』, Springer Verlag

감사의 글

이 연구가 이루어지는 과정에는 많은 분들의 도움이 있었고, 그분들의 도움덕분에 연구를 무사히 마칠 수 있었다. 특별히 이 자리를 빌려 그러한 분들께 감사의 마음을 표현하고자 한다. 연구주제 선정 및 연구과정에서, 관심을 갖고 조언을 아끼지 않으시며, 나중에 이러한 일이 있을 때 도움을 주시겠다는 약속을 지키신 유연주 교수님, 이 연구에 관심을 갖고 조언을 해주신 박병건 선생님, 이 프로젝트를 진행하는 동안, 관심을 갖고 조언을 해주시고 책을 추천해주셔서 연구를 잘 시작할 수 있게 도와주신 kangdh2140@naver.com님, 그 외에도 이 프로젝트 중에 관심을 가져주셨거나 조언을 해주신, ckdwo0605@naver.com, dlwlgus041@naver.com, euler21@naver.com, greengb@naver.com, pilotwls@naver.com, pr_0116@naver.com, tetori@naver.com, wnqlswndnjs@naver.com과 그 외의 많은 이웃 분들께도 감사드리며, 연구과정을 지켜보시며, 끊임없는 조언과 응원을 해주신 부모님, 그리고 끝으로 이 프로젝트를 끝까지 지도해주시며, 아낌없는 관심과, 조언을 해주시며 이 프로젝트를 이끌어 주신 저희 조 담당선생님인 이희영 선생님께 감사드립니다.