Road Rage at a Red Light

New Mexico High School
AiS Challenge
Final Report Team 56
April 5, 2000

Team Members:
Kenny Lesenberg
Clario Marez

Teacher:
Mrs.Avery

Executive Summary

The purpose of this project is to model ideal traffic flow and ideal signal light timing which may prove to help decrease the amount of time cars must wait for a green light at an intersection. Decreasing the wait time may help to decrease road rage. Our first goal was to discover the optimum flow of traffic through an intersection. To do this, we wrote a C++ program in which we defined flow as the number of cars per second that pass a given point, thus defining flow in turns of time. Our second goal was to discover the correct signal light timing to avoid the backup at an intersection to reduce tailgating. We did this by using our flow rate from goal one and a Poisson process to determine the probability of a car arriving at an intersection in any one second. We brought these two problems together in a C++ program by inserting our flow rate into a program that simulates the traffic flow and the timing of the signal light. The program outputs the number of cars still waiting to go through the intersection. Our model helps you to determine what the timing of the signal light should be in order for traffic to flow at an even pace. Implementing new light timing according to our model will not eliminate tailgating and road rage completely, but it could possibly help reduce this problem significantly.

TABLE OF CONTENTS

1. Executive Summary……………………………………… 1

2. Research on Road Rage………………………………….. 2

3. Introduction ……………………………………………... 3

4. Statement of Problem …………………………………… 4

5. Method of Solution ……………………………………... 5

6. Conclusion …………………………………………….... 6

7. References ………………………………………………. 7

Appendix A – Code

Appendix B – Program Description and Listing

RESEARCH ON ROAD RAGE

Road rage is the expression of aggressive behavior towards other road users in response to an emotive or angry appraisal of the traffic context; and altering of an individual's personality while driving caused by process of dehumanization. Road rage has become a new form of violence that has developed in the last few years. Most of the time not much happens but a honk, or a flash of bright lights. In some cases people really loose control and give the finger or curse. In rare cases people fight and sometimes kill. Men are more susceptibly than woman. Men ages 18-26 cause more road rage than anybody else. In 4,400 roads rage cases the use of arms or objects have been used like firearms, knives, fist, or club and the favorite the automobile.

METHOD OF SOLUTION

Our method of solution deals with the flow of the cars during a and or after a red light. This will help to avoid or decrease tailgating and road rage. Our object is to find an appropriate speed to have traffic flow be at its safest and to avoid a traffic jam. Our math models were from information we collected from several sources. We will put all this information together to form a two congruent program to solve our problem. We will use several equations to find the appropriate speed and the right timing of a red light. We will minimize or decease the timing of the red light so that traffic flow can be at steady pace.

```
// programmers: Clario Marez, Melisa Kalas, Kenny Leseberg
,Wendy,

#include <iostream.h>
#include <stdlib.h>
#include <string.h>


#include <iomanip.h>
#include <math.h>

int main()
{
int s;          //speed (mph) of cars-assumed to be the same for each
car
int c;          //length of all cars (f)
float f;        //flow - number of cars passing an observation post per
sec
float max=0.0;  //maximum value of the flow
float save;     //the speed which results in the maximum flow
double xx;      //speed squared
int d;
for (c=6;c<60;c=c+5)
{
for(s = 5; s<=70; s=s+5)
{
xx = pow(s,2);
f = (1.467*s)/(c+s+.05*xx);
    if (f> max)
   {
    max = f;
    save = s;
   }

}           //end speed loop
```

```
cout<<"Length of vehicle: "<<setprecision(2)<< c<<endl;
cout<<"Ideal speed is: "<<save<<endl;
cout<< "Flow of the traffic is: " <<setprecision(2)<<max<<"
vehicles per second"<<endl;
cout<<endl;
system("PAUSE");
save = 0;       //set value to 0 for next vehicle length
max = 0;        //set value to 0 for next vehicle length
}               //end car length loop

cout<<endl;
return 0;
}
```

```
#include <iostream.h>
#include <stdlib.h>
#include <time.h>
#include<math.h>


void go();
void stop();
void print_line(int cars);
float get_rand();
float i;


int cars=0;              // number of cars in line
int sec,sim;             // counters
int green=8,red=5;       // period lights are green/red
int green_timer=0;       // counter for green lights
int red_timer=0;         // counter for red lights
int t=48;               //period of simulation; 48 10 second intervals
float p = 0.3;           // probability a car arrives in any second
float r;                // random number
char lights = 'r';       // lights are red at first

int main()
{
 for (sim=1;sim<t;sim++)    //run simulation for t 10 second
intervals
    {
    for (sec=1;sec<=10;sec++)
    {                          //start random number
        r=get_rand();
        if (r<p)              //start IF
        {
```

```
            cars = cars +1;

         }                    //end IF
      }                       //end random number

      if (lights =='g')
         go();
      else stop();

   cout<<endl;

   system ("pause");

   }                 //end simulation
   }                 //end main

  void go()
  {
  cout<<"inside go"<<endl;                        //begin GO; lights are
green here
  green_timer = green_timer + 1;   // advance green timer
  cars = cars - 2;                  // let 2 cars through
  cout<<cars<<"# cars"<<endl;
  if (cars<=0)
  cars=0;
  print_line(cars);

  if (green_timer == green)
     {                        //begin IF
       lights='r';
       green_timer=0;
     }                        //end IF
  }                           //end GO
```

```
float get_rand()
{
r = rand()%20+1;
r = r/10;
return r;
}

void print_line(int cars)
{
cout<< sim<<" ";
cout<<lights<<" ";
cout <<cars<<" cars - ";
for (i=0;i<cars;i++)
   {
     cout<<'*';
    }
}


void stop()
{         //Lights are red here
cout<<"inside stop"<<endl;
red_timer =  red_timer + 1;        // advance red timer
print_line(cars);
if (red_timer == red)
 {
 lights = 'g';
 red_timer = 0;
 }

}                                  //end STOP
```