Artificial Intelligence, Traffic Flow and Communication

New Mexico Adventures in
Supercomputing Challenge
Final Report
April 6, 2004

Team 88
Yucca Jr. High School

Team Members:
Cara Cope
Javier Portillo

Teacher:

Mrs. Kemp

Executive Summary

Communication during wartime situations is critical. If communication is jeopardized then the lives of our soldiers would also be at stake. Artificial intelligence could improve communication between field commanders and the U.S. leader. This would, in turn, save hundreds of lives.

We have decided to study AI to try and help to prevent the reoccurrence of past problems during wartime situations. As an example, in World War II, the U.S. and French forces fought each other, and weeks went by before we were able to communicate that we were allies on the same side.

In future wars we could communicate with our allies quicker and more effectively than in previous wars. An artificial intelligence program or robot could lessen the traffic jam of information by sorting it out and then getting it to the main commander. Thus, it would help him make clearer decisions as to the types of strategies to use or whether or not to stay and fight.

An effective communication system could help transmit clearer messages to the main commander. He would also be able to communicate directly to the soldiers delivering food and munitions to troops. We would also be able to communicate with allies on different fronts. This could in turn reduce our number of casualties.

We are trying to see whether or not AI would effect the outcome of a battle through a simulation that would basically be trying to find a difference in the outcome between the side with AI and the side without AI.

Introduction

Have you ever been close to the enemy and never know whether it is your ally or truly your enemy? Have you ever seen a commander be frustrated to give orders and his troops not follow? These are questions that people ask to the government to find out of their effective communication with their troops. Communication in an all out war is more significant than a big gun. If you do not have a clue who is out there, you might have no way to know who you are shooting at or what you are attempting to do. Using artificial intelligence will help to make us more efficient in battle.

Description

Our entire project is based on the Bat project completed in 1997, aimed at investigating the feasibility of real-time decision-theoretic algorithms for vision-guided automated intelligent vehicles. Bat stands for Bayesian Automated Taxi. Researchers at Berkley used a Machine Vision Based Traffic Surveillance system thast analyzed traffic flow at various times of the day. We connected the RoadWatch tracker with the Commander overseeing the events on the battlefield. If the commander could track the position of his field commanders, he could inform him on which direction to proceed. The core idea in this study was to have video cameras mounted on poles or other tall structures looking down at the traffic scene. Video is captured, digitized, and processed by onsite computers, and then transmitted in summary form to a Transportation Management Center (TMC) for collation and computation of multi-site statistics such as link travel times. Processing occurs in three stages:

* Segmentation of the scene into individual vehicles and tracking each individual vehicle to refine and update its position and velocity in 3D world coordinates, until it leaves the tracking zone.

* Processing the track data to compute local traffic parameters including vehicle counts per lane, average speeds, lane change frequencies, etc. These parameters, together with track information (time stamp, vehicle type, color, shape, position), are communicated to the TMC at regular intervals.

* At the TMC, local traffic parameters from each site are collated and displayed as desired, and/or used in controlling signals, message displays, and other traffic control devices. Computers at the TMC also process the track information from neighboring camera sites to compute long-distance parameters such as link times and origin--destination counts.

Using AI, we could implement similar strategies that were used in this study. We could place AI machines that could track the location of friendly forces and enemy forces. Also AI machines could track the type of terrain in enemy territory. AI could give an awareness to what the situation was like increasing the effectiveness on the field of battle. Through years of experience we have known that communication breaks down when the commander is unable to communicate with his troops and send orders. In the past we have seen allies attack allies, like in World War 2. The United States troops were attacking the French troops when we were suppose to be attacking the Germans (when Hitler was in power.) My partner and I are seeking a greater meaning in the project we are exploring.

Results

We have experienced many things on this project: 1) how input code on a computer and understand why the program works. Also we learned about C++, Java and Dream weaver, programs of a computer system that we had no idea existed. We have never had the experience of being able to make or even start a web site. Knowing of our experiences in the AIS Challenge will help in the future with understanding more of computers and their uses. Our project has helped us understand more of the technological way of solving a problem with simulations and showing them in a web site for other people's use.

What we have learned from doing this project is that AI can be useful in many different ways. It would not only help us with communication and demining, but also with many more types of research that humans cannot perform. We have also learned more about computers in more this year than in the past. We were not sure at the beginning if we would understand what we were doing, but now it all makes sense.

Our results showed that if we do not make awareness into improving communications, the United States will not be ready for another attack or another World War. The US military efficiency depends on its ability to communication effectively. If the world went to war again, there would have to be a communication bridge to know who was on your side and that an airlift would arrive there to help your troops not the enemy's. Knowing that communication must be bridged, though it has improved in the last few years, is not fully efficient to the type of quality we need communication. As a result of this project, we have that awareness.

What we tried to do was to make a program that showed what it is like in the battlefield and what we need to do. We studied and searched for answers to find that communication must always take place. It is always needed to get and accomplish something. Communication is a national issue and is not only needed in military issues but in politics and government. Communication is a big issue. We chose to studying traffic flow managed by a RoadWatch tracker. The RoadWatch tracker had the ability to locate vehicle position along the lane as a function of time, and illustrated the ability of the tracker to detect individual vehicles and follow them even over a wide range of speeds.

Mathematical Model

Constant Acceleration:

Steady acceleration that results in a linear increase in speed (up to the speed limit).

Starting Distance:

The length of empty road a stopped car must see before beginning to move

Distance Traveled (D) during a time interval of Length T is the average speed (S) times T.

D=S*T

If Acceleration (A) is a constant, the change in speed over a time interval of length T is A*T

The Program

This program was written by Jason Ethier. We used his program to attempt to simulate traffic flow—creating the effectiveness in decisionmaking on the field of battle by the Head Commander in Chief.

```
//Jason Ethier
//Traffic Simulation
//The Application File
#include <iostream>
#include <fstream>
#include <string>
#include <sstream>
#include <iomanip>
#include <cstdlib>
#include <cmath>
#include "Simulation.h"

using namespace std;

double randomDouble(double max);
double doubleRange(double min, double max);
int randomInt(int max);
double intRange(int min, int max);

int randomInt(int max){

   return floor(max*(rand()/double(RAND_MAX)) + 0.5);
}

double intRange(int min, int max){

   return randomInt((max+1)-min)+min;
}

void main(){

   //Creates simulation instance and calls member function run
   //Simulation does the rest

   Simulation* sim = new Simulation();
   sim->run(21,1);
```

```
    string s;
    cout << "\n\nType in anything to quit \n";
    cin >> s;
}
```

This is the first implementation file:

```
//Cars.cpp
#include <iostream>
#include <fstream>
#include <string>
#include <sstream>
#include <iomanip>
#include "Cars.h"

using namespace std;

double randomDouble(double max){

    return max*(rand()/double(RAND_MAX));
}

double doubleRange(double min, double max){

    return randomDouble((max+1)-min)+min;
}

//Implementations for class Car

//Function to move all cars except for one
void Car::step(int timestep){

    if (carInFront){
        actualSeparation = carInFront->position - position;
        actualSpeedDifference = carInFront->speed - speed;

        //Decreases speed of car
        if((actualSeparation < OKseparation)||(actualSpeedDifference <
OKspeedDifference)){
        speed -= 2.0;

        if (speed < 0.0){
            speed = 0.0;
```

```
        }

    }
//Increases speed of car
    else if ((actualSeparation > OKseparation)||(actualSpeedDifference >
OKspeedDifference)){
        speed=speed+1;

        if (speed > preferredSpeed){
            speed=preferredSpeed;
        }
    }
}

    position += speed * timestep;

}

//Function to move one "special" car
    void Car::specialStep(int timeStep, int elapsedTime){

        //Slows down the car
        if (elapsedTime < 13){
            speed = speed -2;
        }

        //Speeds up the car
        if (elapsedTime >= 17){
            speed=speed+1;

            if (speed > preferredSpeed){
                speed=preferredSpeed;
            }
        }
     if (speed<0.0){
        speed=0.0;
    }

    position += speed * timeStep;
}

//Constructor for class Car
Car::Car(double OKsep, double OkDiff, double p){

    OKseparation = OKsep;
    OKspeedDifference = OkDiff;
```

```
    position = p;
    preferredSpeed = doubleRange(25.0, 35.0);
    speed = preferredSpeed;
}
```

This is the second implementation file:

```
//Simulation.cpp
#include <iostream>
#include <fstream>
#include <string>
#include <sstream>
#include <iomanip>
#include "Simulation.h"

using namespace std;

//Implementations for class Simulation

//Calls stepAll function to move the cars
void Simulation::run(int numOfSteps, int stepSize) {

    int totalT=0;
    for(int cnt2=0; cnt2 < 10; cnt2++){
        //initial spacing between cars
        myCars[cnt2]->position = cnt2*60;
    }

    stepAll(numOfSteps, stepSize);

}

//Function to print out the positions of all the cars
void Simulation::report(){


    int scaledLocation;
    string empty[71];
    //Empty string is filled with blank characters
    for (int cnt3=0; cnt3<70; cnt3++){
        empty[cnt3] = " ";
    }

    for (cnt3=0; cnt3 < 10; cnt3++){
```

```
        //Scales the cars location to fit on the screen
        scaledLocation = myCars[cnt3]->position/20;
        empty[scaledLocation] = char(cnt3 + 65);
    }


    for (cnt3 = 0; cnt3 < 70; cnt3++){
        //Displays the array containing all the cars
        cout << empty[cnt3];
    }
    cout << "|\n";
}

//Constructor for class Simulation
Simulation::Simulation(){

    //Creates a car for each element in the array
    for (int i=0; i<10; i++){
        myCars[i] = new Car(50.0, 4.0, i*40.0);
}

for (int ii=0; ii<10; ii++){
    if(ii==9){
        myCars[ii]->carInFront=0;
    }
    else{
        myCars[ii]->carInFront = myCars[ii+1];
    }
}

}

//Function loops over all cars and calls either function
//step or specialStep for each car
void Simulation::stepAll(int stepsArg, int sizeArg){

    int numOfSteps = stepsArg;
    int stepSize = sizeArg;

    int totalT=0;
    for (int cnt1=1; cnt1 < numOfSteps+1; cnt1++){

        for (int cnt=1; cnt < stepSize+1; cnt++){

            //The variable for total time is incremented
            totalT = cnt1*cnt;
```

```
            for (int carNum=0; carNum < 10; carNum++){
                if (carNum==5){
                    //Called for the "special" car
                    myCars[carNum]->specialStep(stepSize, totalT);
                }
                else {
                    //Called for every car except the "special" one
                   myCars[carNum]->step(stepSize);
                }

            }

        }
    //Report function is called to display cars
    report();
    }
}
```

This is the first interface file:

```
//cars.h
class Car {

    //ADT specification for class Car
    public:

        void step(int timeStep);
        Car(double OKseparation, double OKspeedDifference, double position);
        void specialStep(int timeStep, int elapsedTime);

        double speed;
        double position;
        double preferredSpeed;
        double OKseparation;
        double OKspeedDifference;
        double actualSeparation;
        double actualSpeedDifference;
        Car* carInFront;

};
```

This is the second implementation file:

```
//simulation.h
#include "Cars.h"

//ADT specification for class Simulation
class Simulation {

    public:

        void run(int numOfSteps, int stepSize);
        void report();
        void stepAll(int numOfSteps, int stepSize);
        Simulation();

        Car* myCars[10];

};
```

Bibliography

RoadWatch
http://http.cs.berkeley.edu/~pm/RoadWatch/results.html

Program
http://www.ccs.neu.edu/ethier

CNN News
http://www.cnn.com/SPECIALS/2003/iraq/forces/coalition/deployment/army/camp.doha.html

Ai Research- Creating a new form of life- Origin
http://www.a-i.com/show_tree.asp?id=44&level=3&root=1&page=2

Brief History of Artificial Intelligence
http:www.aaai.org/Pathfinder/bbhist.html

Understanding Basic Statistics
by Brase and Brase
1997 Houghton Mifflin Company

Traffic Flow Model
aspire.cs.uah.edu/aspire/summer/manuals/ secondary_manual/trafficflow10-31

Lugar, George F. and Stubblefield, William A. *Artificial Intelligence Structures and Strategies for Complex Problem Solving.* New York: The Benjamin/Cummings Publishing Company, Inc., 1993

Champand, Alex J. "Artificial Intelligence."In the Artificial Intelligence Depot, 2001-2002

Internet on-line, http://ai-depot.com. [14 November 2003].

Ahmad, Muhammad Aurangzeb, *Artificial Intelligence: Perspectives in History*. Internet on-line. Available from
<http://www.rit.edu/~maa2454/ai-intro.htm>