

Краевой конкурс учебно-исследовательских и проектных работ учащихся
«Прикладные вопросы математики»

Математическое моделирование

Имитационное моделирование движения автомобилей по автодорогам

Волков Владислав Сергеевич
МОУ «Лицей №1» г. Перми, 11 кл.

Волегов Павел Сергеевич
к.ф.-м.н., доц. каф. ММСП

ПНИПУ

Пермь

2011

Содержание

Введение.....	4
1. Имитационное моделирование.....	7
2. Концептуальная постановка задачи.....	10
3. Правила клеточного автомата.....	11
4. Результаты исследования.....	14
Заключение.....	19
Список литературы.....	20

Введение

За последние пять лет количество автомобилей в России возросло в полтора раза, соответственно возросла возможность пробок, да и время ожидания водителя автомобиля на перекрестке существенно возросло. Для решения этой проблемы создаются скоростные автомагистрали, эстакады, однако в черте города здания и другие постройки мешают расширять дорожное полотно, поэтому проблема регулировки движения автотранспорта становится актуальной. Определение оптимального режима работы светофора методом «проб и ошибок» чревато последствиями (дорожными заторами), поэтому наиболее безопасным методом решения этой проблемы является создание математической модели движения автомобилей на перекрестке[3].

По сравнению с натурным экспериментом математическое моделирование имеет определенные преимущества:

- Экономичность (в частности, сбережение ресурсов реальной системы).
- Возможность моделирования гипотетических, т.е. не реализованных в природе объектов (прежде всего на разных этапах проектирования).
- Возможность реализации режимов, опасных или трудновоспроизводимых в натуре.
- Возможность изменения масштаба времени.
- Легкость многоаспектного анализа.
- Большая прогностическая сила вследствие возможности выявления общих закономерностей.
- Универсальность технического и программного обеспечения проводимой работы (ЭВМ, системы программирования и пакеты прикладных программ широкого назначения).

Цель работы

Разработать и исследовать математическую модель движения автомобилей на X-образном перекрестке, с учетом различной дорожной разметки, дорожных условий и режимов работы светофоров.

Для того чтобы достичь цели исследования, необходимо решить следующие **задачи исследования**:

- сформулировать требования к модели;
- выбрать метод решения задачи;
- сформулировать правила клеточного автомата;
- провести реальный эксперимент на дорогах города;

- разработать программу позволяющую подсчитать оптимальное время ожидания автомобиля на светофоре;

Требования к модели

- Модель должна позволять исследовать движение автомобилей: т.е. определять количество автомобилей, скопившихся у светофора и среднее время нахождения автомобиля в системе.
- Модель должна быть адекватной: автомобили должны передвигаться по правилам дорожного движения. В модели находится количество автомобилей заданное пользователем, пользователь также задает потоки автомобилей, появляющихся на дороге, входящих в систему, интервалы времени горения зеленого света и красного сигналов светофора. Поток автомобилей определяется интервалом времени, через который автомобили появляются.
- Модель должна случайным образом определить скорости автомобилей и направление их движения на перекрестке.

Выходные данные:

- Геометрические размеры моделируемого пространства, координаты центров перекрестков, координаты автомобилей;
- Скорости потоков автотранспорта;
- Вероятность поворота на перекрестке;

Светофор – главный инструмент управления потоками транспорта на дорогах и перекрестках, где с его помощью регулируется движение. Режим работы светофора – это повторяющийся цикл (свой для каждой группы светофоров), состоящий из 2-ух основных состояний: проезд разрешен и проезд запрещен. Так как в разных направлениях поток машин разный, то светофор должен обладать разными периодами включенного и выключенного состояния. Изменяя эти периоды можно регулировать движение автотранспорта на дорогах, вследствие чего избежать автомобильных пробок на дорогах.

В данной работе предлагается создать математическую модель автомобильной дорожной сети, в которой можно изменять режимы работы светофоров, потоки транспорта и его направления для исследования.

Модели объектов-систем, учитывающие свойства и поведение отдельных элементов, а также взаимосвязи между ними, называются *структурными*. Среди структурных динамических систем выделены в отдельный подкласс *имитационные* системы, состоящие из конечного числа элементов, каждый из которых имеет конечное число состояний. Число связей между элементами также предполагается конечным.

Моделирование взаимодействий элементов системы друг с другом осуществляется с помощью некоторого алгоритма, реализуемого обычно с использованием ЭВМ. Для моделирования на ЭВМ реального времени вводится понятие *системного времени*. В качестве моделей отдельных элементов могут быть использованы модели любого типа.

1. Имитационное моделирование

Развитие моделей, использующих имитационный подход, связано с необходимостью исследования очень сложных систем, встречающихся на практике. Аналитические и численные методы, позволяющие провести наиболее полное исследование математической модели объекта, применимы далеко не для всех систем. Для многих сложных систем при построении аналитических моделей исследователю зачастую приходится идти на серьезные упрощения, чтобы получить представление о некоторых общих свойствах моделируемой системы, например, оценить устойчивость стационарного состояния системы. Моделируемая система может быть настолько сложна, а поведение ее – так многообразно и непредсказуемо, что принятая система гипотез может приводить не только к существенным количественным, но и качественным отличиям результатов моделирования от поведения системы в реальных условиях. При этом повышение степени адекватности модели может оказаться невозможным по многим причинам: вследствие незрелости существующих аналитических и численных методов, невозможности построения аналитического описания поведения отдельных элементов системы, взаимодействия между элементами и так далее.

Ниже перечислены ситуации, когда исследователю можно рекомендовать применять модели, имитирующие поведение реального объекта[1]:

1. Если не существует законченной постановки задачи исследования и идет процесс познания объекта моделирования или отдельных его элементов.
2. Если аналитические методы имеются, но математические процедуры трудно реализуемы, сложны и трудоемки.
3. Когда кроме оценки влияния параметров сложной системы желательно осуществить наблюдение за поведением отдельных компонентов этой системы в течение определенного периода времени.
4. Когда имитационный подход оказывается единственным способом исследования сложной системы из-за невозможности наблюдения явлений в реальной обстановке.

5. Когда необходимо контролировать протекание процессов в сложной системе путем замедления или ускорения явлений в ходе имитации.

6. При подготовке специалистов и освоении новой техники, когда имитатор обеспечивает возможность приобретения необходимых навыков в эксплуатации новой техники.

7. Когда изучаются новые ситуации в сложных системах, о которых мало что известно. В этом случае имитация служит для предварительной проверки новых стратегий и правил принятия решений перед проведением экспериментов на реальной системе.

8. Когда основное значение имеет последовательность событий в проектируемой сложной системе и модель используется для предсказания узких мест в функционировании системы и других трудностей, связанных с добавлением в систему новых элементов.

Имитационный подход оправдан, если вопросы, на которые должна ответить модель, относятся не к выяснению фундаментальных законов и причин, определяющих динамику реальной системы, а к анализу поведения системы, выполняемому в сугубо практических целях.

Суть подхода, используемого при разработке имитаторов, состоит в том, что процесс функционирования сложной системы представляется в виде определенного алгоритма, реализуемого на ЭВМ. Для систем, состоящих из множества элементов, приходится строить модель не только всей системы, но и модели отдельных элементов. Как и для аналитического подхода, разработка имитатора ведется с использованием некоторой совокупности гипотез. Изменение даже одной гипотезы для одного из элементов системы может привести к необходимости пересмотра всей модели системы и поиску новых методов исследования. Имитационный подход позволяет максимально использовать всю имеющуюся в распоряжении исследователя информацию о системе.

Клеточные автоматы как средство имитационного моделирования

Очень многие физические, биологические, социальные процессы могут быть смоделированы клеточными автоматами. В самом деле, если представить сообщество людей как клетки, состояния которых будут характеризовать состояние здоровья человека (можно в виде двух состояний: 0 – здоровый, 1 – больной), а окрестностью считать тех людей, с которыми контактирует человек, то можно смоделировать процесс распространения инфекционного заболевания. Образование планет из околозвездного пылевого облака можно смоделировать клеточным автоматом, у которого состояние

клетки будет обозначать плотность материи в определенном объеме пространства. Закон изменения состояния клеток будет описывать процесс гравитационного взаимодействия близких масс газопылевого облака при образовании планет. Различные биологические процессы моделируются клеточными автоматами. Например, процесс развития зародыша из одной оплодотворенной клетки заключается в последовательном делении клеток. При этом происходит взаимодействие соседних клеток и осуществляется их дифференциация (изменение вида). Так образуются специализированные клетки многоклеточного организма (нервные, клетки кожного покрова, внутренних органов и пр.)[4].

Концептуальная постановка задачи

При *имитационном подходе* объект исследования разбивается на отдельные элементы. В этом случае система математических соотношений для объекта-системы в целом не записывается, а заменяется некоторым алгоритмом, моделирующим ее поведение и учитывающим взаимодействие друг с другом моделей отдельных элементов системы. В качестве моделей отдельных элементов могут быть использованы как аналитические, так и алгебраические модели[1].

Алгоритмические модели, использующие как численный, так и имитационный подход, не позволяют получить решения задач в аналитической форме, что затрудняет и усложняет процесс анализа результатов моделирования. Так как применение моделей данного типа возможно лишь при наличии вычислительной техники, то их эффективность зависит от мощности и быстродействия ЭВМ. Несомненным достоинством алгоритмических моделей является отсутствие принципиальных ограничений на сложность модели, что позволяет применять их для исследования систем произвольной сложности.

Использование математической модели, построенной с применением алгоритмических методов, аналогично проведению экспериментов с реальным объектом, только вместо реального эксперимента с объектом проводится *вычислительный эксперимент* с его моделью. Задаваясь конкретным набором значений исходных параметров модели, в результате вычислительного эксперимента получают конкретный набор приближенных значений искомым параметров. Для исследования поведения объекта при новом наборе исходных данных необходимо проведение нового вычислительного эксперимента.

Объектом моделирования является перекресток с находящимся на нем автотранспортом.

Разработку модели будем выполнять при следующих **предположениях (гипотезах модели)**:

- Объектом исследования является участок дорожного полотна.
- Объектом оптимизации является режим работы светофора.
- Полосы и перекресток состоят из клеток, каждую клетку может занимать только одна машина.
- Каждая клетка должна содержать номер машины, которая в нем находится.
- Движение машин происходит в одной плоскости.
- Машины двигаются по законам равномерного движения.
- Имеется один тип машин.
- Скорость всех автомобилей при движении одинакова.
- Интервал времени между автомобилями на входе в систему задается случайным образом.
- При поворотах автомобиль должен перестроиться в нужную полосу в соответствии со стороной поворота.

Правила клеточного автомата

- Дорога должна представлять совокупность маленьких ячеек (рис. 1).
- Сторона ячейки – ширина одной полосы.
- Автомобиль движется внутри ячейки.
- Машина не может въехать на занятую ячейку или на ячейку, которая будет занята.

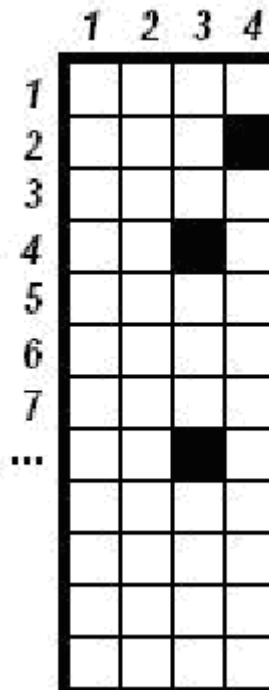


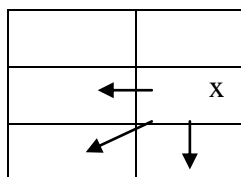
Рис. 1. Схема полос и препятствий.

Существует четырехполосная дорога с двусторонним движением. Первым шагом происходит прорисовка полосы и клеточек. Затем прорисовываются машины и препятствия.

Перечислим основные состояния ячеек:

	Ячейка свободна
	Ячейка занята
	Препятствие

В качестве окрестности каждой ячейки рассматривается следующая область:



Перечислим основные правила, по которым изменяется состояние ячеек:

1. Если ячейка красная, и впереди нее стоит белая, то красная ячейка может занять ее (рис. 2).
2. Если ячейка красная, и впереди стоит красная, то занять ее нельзя.
3. Если, впереди черная ячейка, то сзади идущая красная клеточка останавливается.
4. отслеживает справа или слева пустую белую клетку, и если таковая есть, перестраивается в другой ряд и совершает обгон препятствия (рис.3).

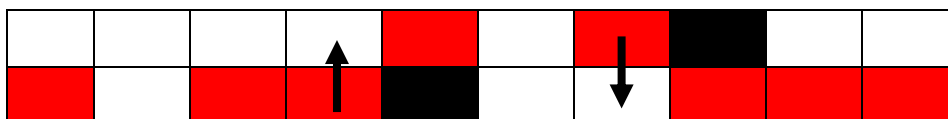


Рис.2.

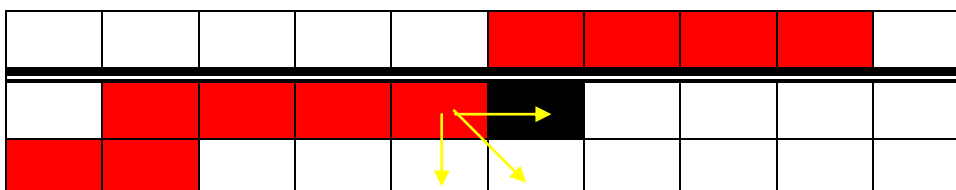


Рис.3.

В данной программе реализована возможность имитации препятствия на дороге, а так же возможность задать интенсивность потока машин на полосах. Предложенный алгоритм реализован в виде программного кода в среде Delphi (приложение 2).

Программа умеет (рис.4.):

- Осуществлять ввод параметров:
- Автотранспорта - интенсивность появления автомобилей на дороге;
- Дорог – количество полос на дороге;
- Препятствия - количество препятствий, расстояние между ними, выбор полос на которых будут располагаться препятствия.

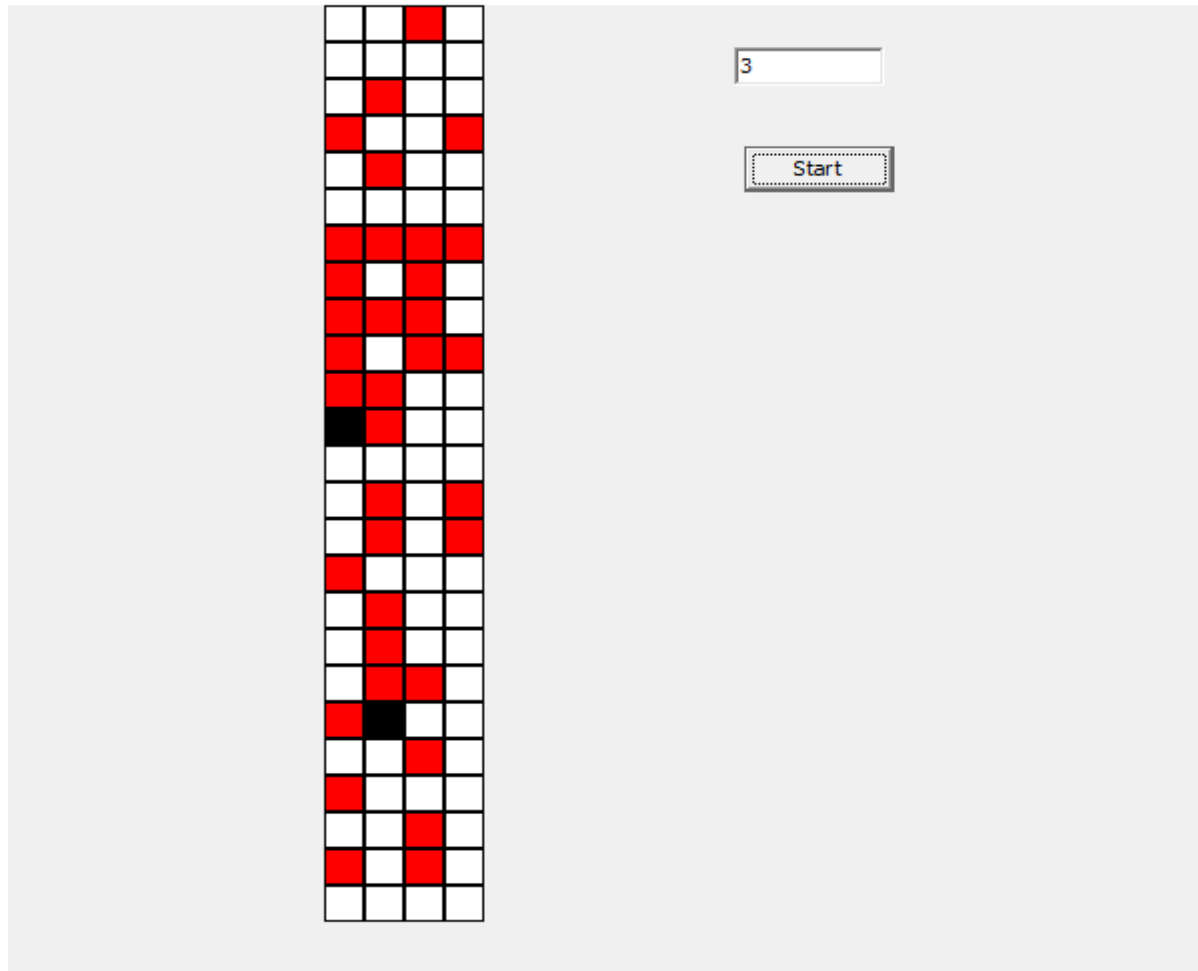


Рис.4. Скриншот работы программы.

Результаты исследования

В исследовании речь идет о качественном поведении модели, все числа на входе и выходе – модельные и не имеют отношения к реальной автодороге.

Задано расстояние между препятствиями длиной в 3 клеточки (машины).

При установке интенсивности потока машин около (расчеты приблизительны, так как в программе нет возможности точно измерить количество машин в единицу времени) 1 машины за 1 секунду и расположении препятствий на 2-ух полосах пробка не образовывается вообще. При увеличении потока машин до 3 машин в секунду образование пробки незначительно, на процесс движения машин в системе не влияют. Далее было установлена интенсивность машин примерно 5 машин в секунду, при этой интенсивности, пробки стали заметны, и ожидание машины в пробке стало 10-13 секунд (рис.5).

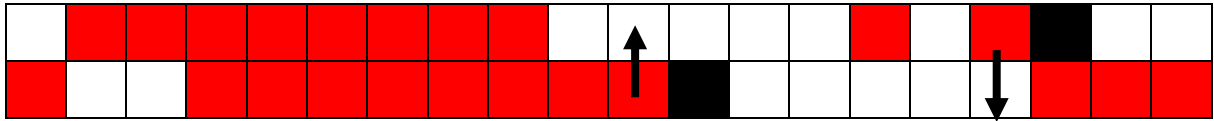
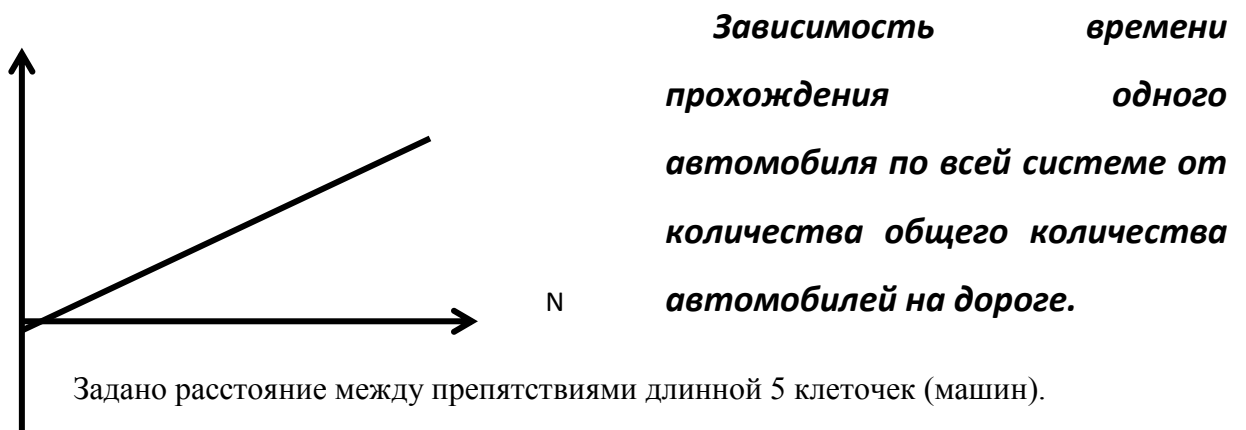


Рис.5

Из этого можно сделать логичный вывод, что при увеличении интенсивности потока машин, увеличивается размер пробки, и соответственно ожидание автомобиля в пробке.



При установке интенсивности потока машин примерно до 1 машин за 1 секунду и расположении препятствий на 2-ух полосах пробка не образовывается вообще. При увеличении потока машин до 3 машин в секунду образование пробки так же не наблюдается. При установке интенсивности потока машин до 5 машин в секунду, образование пробки прослеживается, но серьезного влияния на всю систему пробка не имеет. Однако при интенсивности потока 7 машин в секунду, пробки образуются, причем, время нахождения машины в пробке может достигать 17 секунд. (рис.6.)

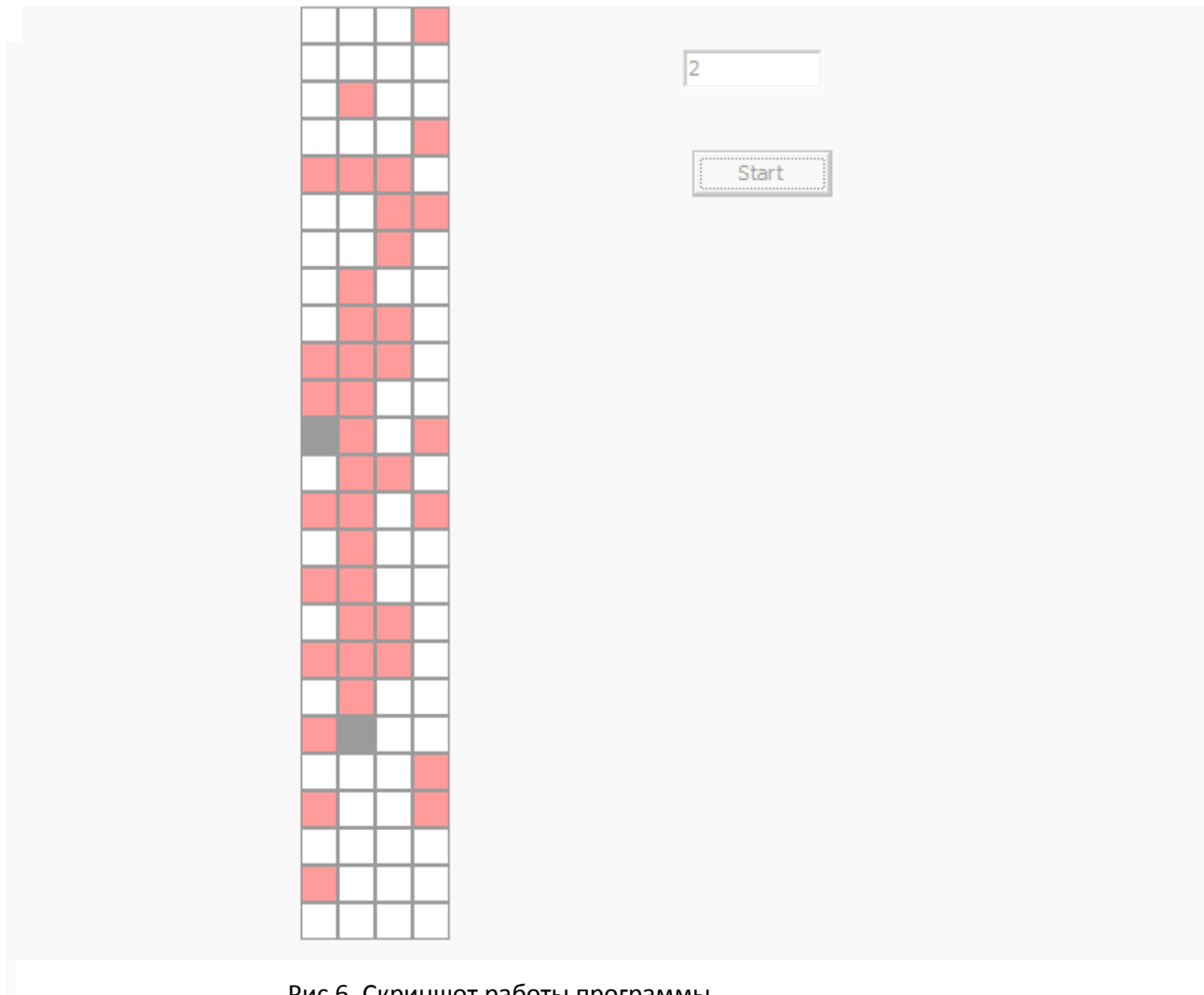
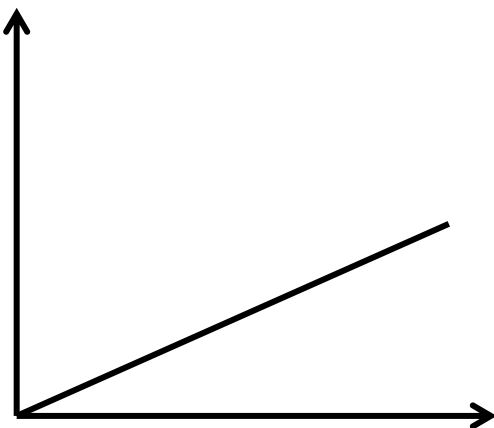


Рис.6. Скриншот работы программы.

На основе этого можно прийти к выводу, что при увеличении расстояния между препятствиями и увеличении интенсивности потока машин пробки увеличиваются, и время нахождения автомобиля в системе, так же увеличивается.

1)
т

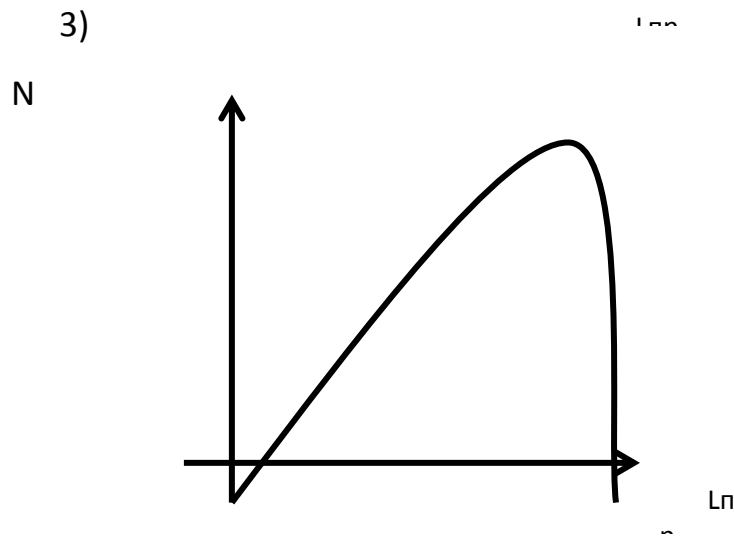
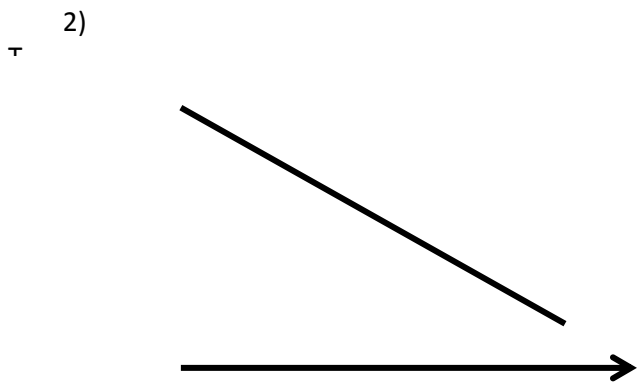


**Зависимость времени
прохождения одного
автомобиля по всей системе
от общего количества
автомобилей на дороге.**



l

**Зависимость времени
прохождения одного
автомобиля по всей системе**



Зависимость общего количества автомобилей на дороге от расстояния между препятствиями.

Заключение

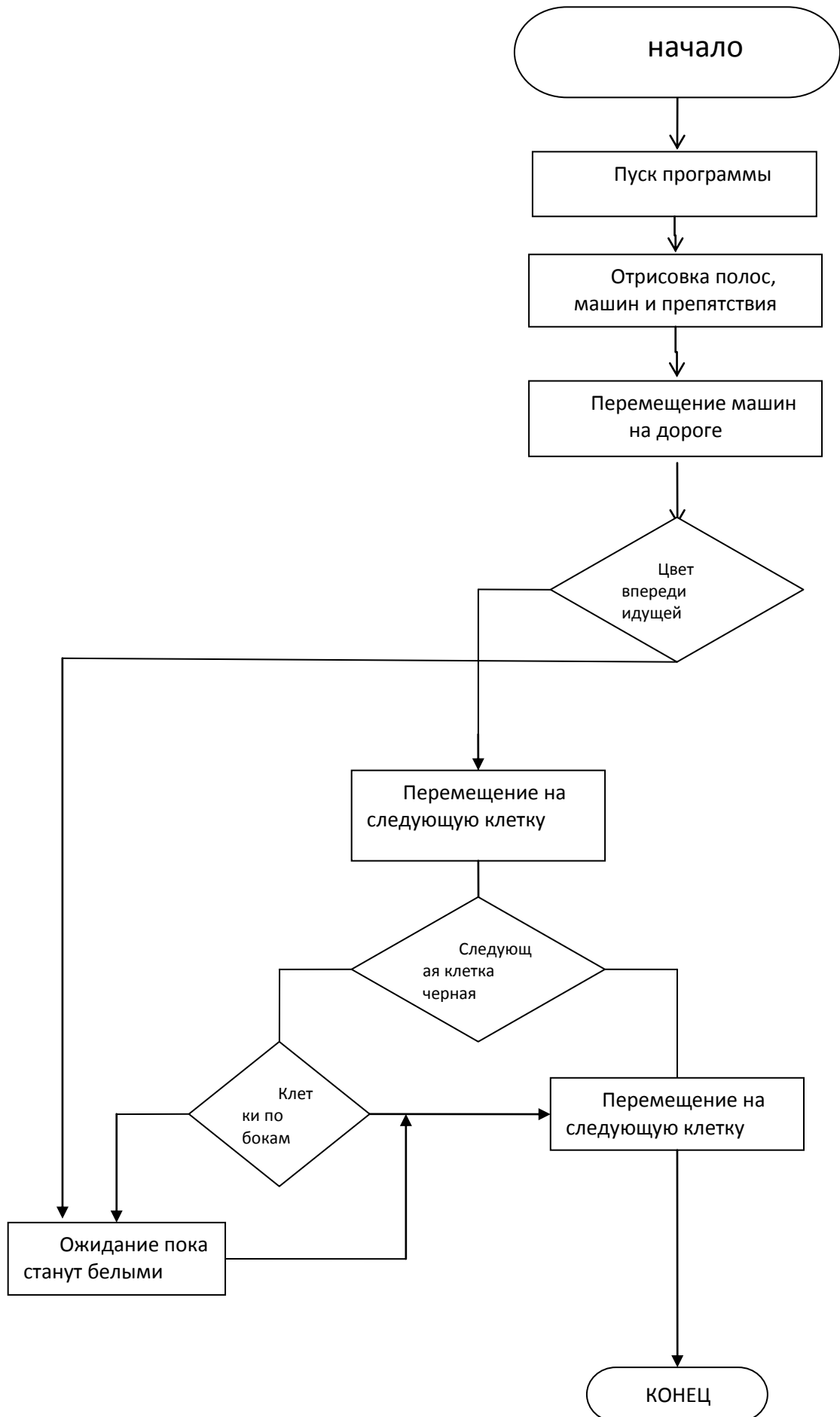
В данной работе был поставлен реальный эксперимент на дорогах города Перми. На основе изучения данных эксперимента была реализована программа, при помощи которой создана модель, охватывающая большой участок автомобильной сети. Программа позволяет с большой точностью промоделировать сложное поведение объектов моделирования, в данном случае автомобилей. Были выявлены закономерности и построены графики зависимостей движения автомобилей от расстояния между препятствиями, коэффициентом интенсивности появления автомобилей на дороге, времени прохождения автомобиля по системе. Разработанная программа может быть использована для моделирования движения автотранспорта по главным улицам городов, для создания зеленой полосы¹, а также для оперативного реагирования на закрытия дорог в связи с авариями или ремонтами.

¹ Зеленая полоса – поочередные включения светофоров в одном направлении, при которых автомобиль, едущий с определенной скоростью, всегда попадает на зеленый сигнал светофора, при этом пересекая перекресток не останавливаясь.

Список литературы

1. Введение в математическое моделирование. Учебное пособие /Под ред. П.В.Трусова.– М.: "Логос". 2003.
2. Турбо Паскаль 7.0/ Е.Р. Алексеев, О.В. Чеснокова, В.Н. Павлыш, Л.В. Славинская. – М.: ООО «Издательство АСТ»: Издательство «НТ Пресс», 2004. – 270с.
3. http://ru.wikipedia.org/wiki/Клеточный_автомат
4. http://written.ru/articles/science/life/cell_auto

Блок – схема программы



Программный код

```
unit Unit1;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls;

type
  Tcell=record
    i,J: integer;
    cl:tcolor;
  end;

  Tarray = array [1..4,1..25] of TCell;

  TForm1 = class(TForm)
    PaintBox1: TPaintBox;
    Button3: TButton;
    Edit1: TEdit;
    procedure Button3Click(Sender: TObject);
    procedure Timer1Timer(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;
  arr : Tarray ;
  x,y,dx,dy:integer;
  procedure draw;

implementation

{$R *.dfm}
procedure draw;
var i:integer;
    j:integer;
    k:integer;
begin
  for i := 1 to 4 do
    for j := 1 to 25 do
      begin
        if arr [i,j].cl = clwhite then
          begin
            Form1.PaintBox1.Canvas.Brush.Color:=clwhite;
            Form1.PaintBox1.Canvas.Rectangle(x+dx*(i-1),y+dy*(j-1),x+dx*(i),y+dy*(j))
```

```

end;
if arr [i,j].cl = clred then
begin
Form1.PaintBox1.Canvas.Brush.Color:=clred;
  Form1.PaintBox1.Canvas.Rectangle(x+dx*(i-1),y+dy*(j-1),x+dx*(i),y+dy*(j));
end;

if arr [i,j].cl = clblack then
begin
Form1.PaintBox1.Canvas.Brush.Color:=clblack;
  Form1.PaintBox1.Canvas.Rectangle(x+dx*(i-1),y+dy*(j-1),x+dx*(i),y+dy*(j));
end;
  end;
end;

procedure TForm1.Button3Click(Sender: TObject);
var
i: integer;
j:integer;
k:integer;

begin
dx:=20;
dy:=20;
x:=30;
y:=0;
Form1.PaintBox1.Canvas.Brush.Color:=clWhite;
for i := 1 to 4 do
  for j := 1 to 25 do
    Form1.PaintBox1.Canvas.Rectangle(x+dx*(i-1),y+dy*(j-1),x+dx*(i),y+dy*(j));
  {начальная раскраска дороги}
for i := 1 to 4 do
  for j := 1 to 25 do
    begin
      if random > StrToInt(Edit1.Text) then arr[i,j].cl:=clRed else arr[i,j].cl:=clWhite;
    end;
  {ставим препятствие на середине 1 полосы}
  arr[1,12].cl:=clBlack;

  {ставим препятствие на 2 полосе}
  arr[2,20].cl:=clBlack;

for k := 0 to 100 do
begin
  sleep(1000);

  {1 полоса}
  {появление машинок на дороге}
  if (random > 0.7) and (arr[1,1].cl=clWhite) then arr[1,1].cl:=clRed;
  if (arr[1,25].cl=clred) then arr[1,25].cl:=clwhite;

```

```

for j := 25 downto 1 do
begin

    {движение машинок вперед}
    if (arr[1,j].cl=clred) and (arr[1,j+1].cl=clwhite) then
    begin
        arr[1,j].cl:=clwhite;
        arr[1,j+1].cl:=clred;
    end;

    {объезд препятствия на 1й полосе}
    if (arr[1,j].cl=clred) and (arr[1,j+1].cl=clblack) and (arr[2,j].cl=clwhite) and
    ((arr[2,j+1].cl=clwhite)) then
    begin
        arr[1,j].cl:=clwhite;
        arr[2,j].cl:=clred;
    end;
    {объезд затора из 4х машин на 1й полосе}
    if (arr[1,j].cl=clred) and (arr[1,j+1].cl=clred) and (arr[1,j+2].cl=clred)
    and (arr[1,j+3].cl=clred) and (arr[2,j].cl=clwhite) and ((arr[2,j+1].cl=clwhite)) then
    begin
        arr[1,j].cl:=clwhite;
        arr[2,j].cl:=clred;
    end;

end;

{2 полоса}
if (random > 0.7) and (arr[2,1].cl=clWhite) then arr[2,1].cl:=clRed;
if (arr[2,25].cl=clred) then arr[2,25].cl:=clwhite;

for j := 25 downto 1 do
begin

    {движение машинок вперед}
    if (arr[2,j].cl=clred) and (arr[2,j+1].cl=clwhite) then
    begin
        arr[2,j].cl:=clwhite;
        arr[2,j+1].cl:=clred;
    end;

    {объезд препятствия на 1й полосе}
    if (arr[2,j].cl=clred) and (arr[2,j+1].cl=clblack) and (arr[1,j].cl=clwhite) and
    ((arr[1,j+1].cl=clwhite)) then
    begin
        arr[2,j].cl:=clwhite;
        arr[1,j].cl:=clred;
    end;

```

```
{объезд затора из 4х машин на 1й полосе}
if (arr[2,j].cl=clred) and (arr[2,j+1].cl=clred) and (arr[2,j+2].cl=clred)
and (arr[2,j+3].cl=clred) and (arr[1,j].cl=clwhite) and ((arr[1,j+1].cl=clwhite)) then
begin
  arr[2,j].cl:=clwhite;
  arr[1,j].cl:=clred;
end;
```

```
end;
```

```
{3 полоса}
if (random > 0.7) and (arr[3,25].cl=clWhite) then arr[3,25].cl:=clRed;
if (arr[3,1].cl=clred) then arr[3,1].cl:=clwhite;
```

```
for j := 1 to 25 do
begin
  if (arr[3,j].cl=clred) and (arr[3,j-1].cl=clwhite) then
  begin
    arr[3,j].cl:=clwhite;
    arr[3,j-1].cl:=clred;
  end;
end;
```

```
{4 полоса}
if (random > 0.7) and (arr[4,25].cl=clWhite) then arr[4,25].cl:=clRed;
if (arr[4,1].cl=clred) then arr[4,1].cl:=clwhite;
```

```
for j := 1 to 25 do
begin
  if (arr[4,j].cl=clred) and (arr[4,j-1].cl=clwhite) then
  begin
    arr[4,j].cl:=clwhite;
    arr[4,j-1].cl:=clred;
  end;
end;
```

```
draw;
end;
```

```
end;
```

```
procedure TForm1.Timer1Timer(Sender: TObject);
begin
  //Button3.Click;
end;
```

```
end.
```