Юные техники и изобретатели

**Численное и экспериментальное решение задачи Штейнера**

**Автор:** Панин Артём Валерьевич

МБОУ г. Мурманска Мурманский политехнический лицей, 9 класс

**Научный руководитель:** Белоушко Константин Евгеньевич

Старший преподаватель кафедры МИСиПО

Мурманского государственного технического университета

Москва

2016

**Оглавление**

1. **Аннотация…………………………………………………………………3**
2. **Введение………..………………………………………...………………..4**
3. **Основная часть………………….......…………………………………....5**
4. **Заключение…………...…………………………………………………...7**
5. **Список литературы…………………...…………………………………8**
6. **Приложения………………...…………………………………………….9**

**Аннотация**

В наше время мы все чаще встречаемся с проблемой неэкономного расхода ресурсов, например, при построении сети автомобильных и железных дорог между городами, проведение линий электропередач и связи. Якоб Штейнер в XIX веке задумался над тем, как можно соединить несколько деревень таким образом, чтобы иметь кратчайшую сеть дорог. Для трех деревень эта задача была им успешно решена и позже доказана математически. Однако, до сих пор не существует универсального математического решения задачи Штейнера для большого количества точек. Мы попробуем решить задачи для трех, четырех и пяти точек, соединенных минимальной сетью, двумя методами. Первый - экспериментальное решение задачи, основанное на свойстве мыльной плёнки минимизировать площадь своей поверхности. Для этого использовалась специальные пластиковые модели со стержнями, соответствующими точкам сети, на которую натягивалась мыльная плёнка. Второй - решение задачи численно с помощью программы, разработанной на языке Pascal. При этом вначале строилось первое приближение сети, а затем в цикле производилось случайным образом смещение точек соединения, подсчитывалась новая длина сети и в случае её минимизации на данном шаге, смещение оставалось, иначе отбрасывалось.

Было установлено, что оба метода дают с точностью до погрешностей примерно одинаковые результаты. Так были получены минимальные сети для 3, 4 и 5 вершин, причём для 4 вершин с несколькими вариантами соединения сети. Планируется усложнить численную модель с целью расчёта минимальной сети на трехмерном рельефе, тем самым приблизившись к решению реальных задач.

**Введение**

В наше время мы все чаще встречаемся с проблемой неэкономного расхода ресурсов, например, при построении сети автомобильных и железных дорог между городами, проведение линий электропередач и связи, построении подземных переходов и путей метро. Якоб Штейнер в XIX веке задумался над тем, как можно соединить несколько деревень таким образом, чтобы иметь кратчайшую сеть дорог. Для трех деревень эта задача была им успешно решена и позже доказана математически. Однако, до сих пор не существует универсального математического решения задачи Штейнера для большого количества точек. Мы попробуем решить задачи для трех, четырех и пяти точек, соединенных минимальной сетью, двумя методами: экспериментальным и численным.

**Основная часть**

Первый метод - экспериментальное решение задачи для трех, четырех и пяти точек. Для него нам больше всего подходит мыльная пленка, имеющая достаточно большой коэффициент поверхностного натяжения (рис.1 в приложении), и трехслойную структуру строения (рис.2 в приложении), то есть нам хватит времени на проведение измерений, пока связи молекул мыльной жидкости не разорвутся. Также мы используем специальные пластиковые модели со стержнями, соответствующими точкам сети. Модели опускаются в мыльный раствор, мыльная пленка натягивается между стержнями (рис.3 в приложении) и образует за счет поверхностного натяжения минимальную сеть дорог.

Второй метод - решение задачи численно, с помощью программы, разработанной на языке Pascal для трех, четырех и пяти точек. В программе задаются координаты, соединяемых точек, случайные координаты перемещающихся точек соединения, и затем в цикле происходит перемещение точек в случайном направлении, подсчет длины сети, и в случае ее минимизации, выведение новой сети на экран (рис.4 в приложении). Справа на экран выводятся длины всех отрезков сети, их сумма, координаты точек соединения и косинусы углов между отрезками сети.

Теперь проанализируем результат решения для четырех точек. Сначала изменим вид сети, то есть изменим соединения закрепленных и промежуточных точек. При подсчете новой минимальной сети, мы получаем результат, несколько меньший, чем предыдущий. И введем новое понятие - сумма маршрутов. Это сумма путей из точки 1 в точку 2, из 1 в 3, из 1 в 4 из 2 в 3, из 2 в 4 и из 3 в 4. Она также выводится на экран справа. Затем, для наглядности, построим графики изменения суммы маршрутов и изменения длины всей сети (рис.5 в приложении). Длина сети постоянно уменьшается, так как это и есть наш основной параметр программы, а сумма маршрутов сначала уменьшается, а потом увеличивается.

Также мы начали работать над созданием программы вычисляющей минимальную сеть дорог на трехмерном рельефе. Мы разработали алгоритм для программы, рассчитывающей минимальную сеть для двух точек на рельефе:

1. Экран делится на ячейки, каждая из которых имеет свою высоту и удаленность. Таким образом мы задаем рельеф функцией f(x;y;z), то есть в трехмерной системе координат.
2. На рельефе задаем две точки, которые соединяются прямой, частично проходящей сквозь землю, прямая делится на несколько равных отрезков, их количество будет равно количеству поворотов итоговой сети.
3. Этим промежуточным точкам присваивается высота, соответствующая ячейкам, на которых они находятся.
4. Происходит подсчет длины всей сети.
5. Затем в цикле промежуточные точки перемещаются в случайном направлении, и после каждого перемещения новому положению точки присваивается новая высота и происходит подсчет новой длины сети.
6. Если длина сети уменьшилась, то новая сеть выводится на экран.
7. Цикл продолжается, пока смещение точек не перестанет влиять на уменьшение длины сети.

**Заключение**

Оба метода показали, что минимальная сеть должна иметь все углы, равные 120 ̊. Так были получены минимальные сети для 3, 4 и 5 вершин, причём для 4 вершин с несколькими вариантами соединения сети.

Планируется усложнить численную модель с целью расчёта минимальной сети на трехмерном рельефе, тем самым приблизившись к решению реальных задач.

**Список литературы**

1. Берн М.У., Грэм Р.Л. Поиск кратчайших сетей // В мире науки, 1989, № 3.

2. Бойс Ч. Мыльные пузыри. - М.: 1937.

3. Лисин А.В., Файзуллин Р.Т. Эвристический алгоритм писка приближённого решения задачи Штейнера, основанный на физических аналогиях // Компьютерная оптика, 2013, том 37, № 4.

4. Гордеев Э.Н., Тарасцов О.Г. Задача Штейнер. Обзор // Дтискретная математика, 1993, том 5, № 2.

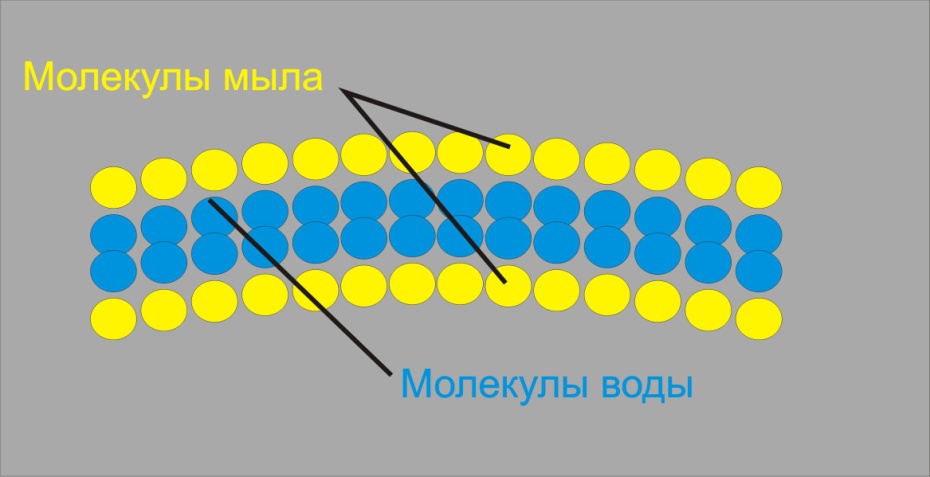
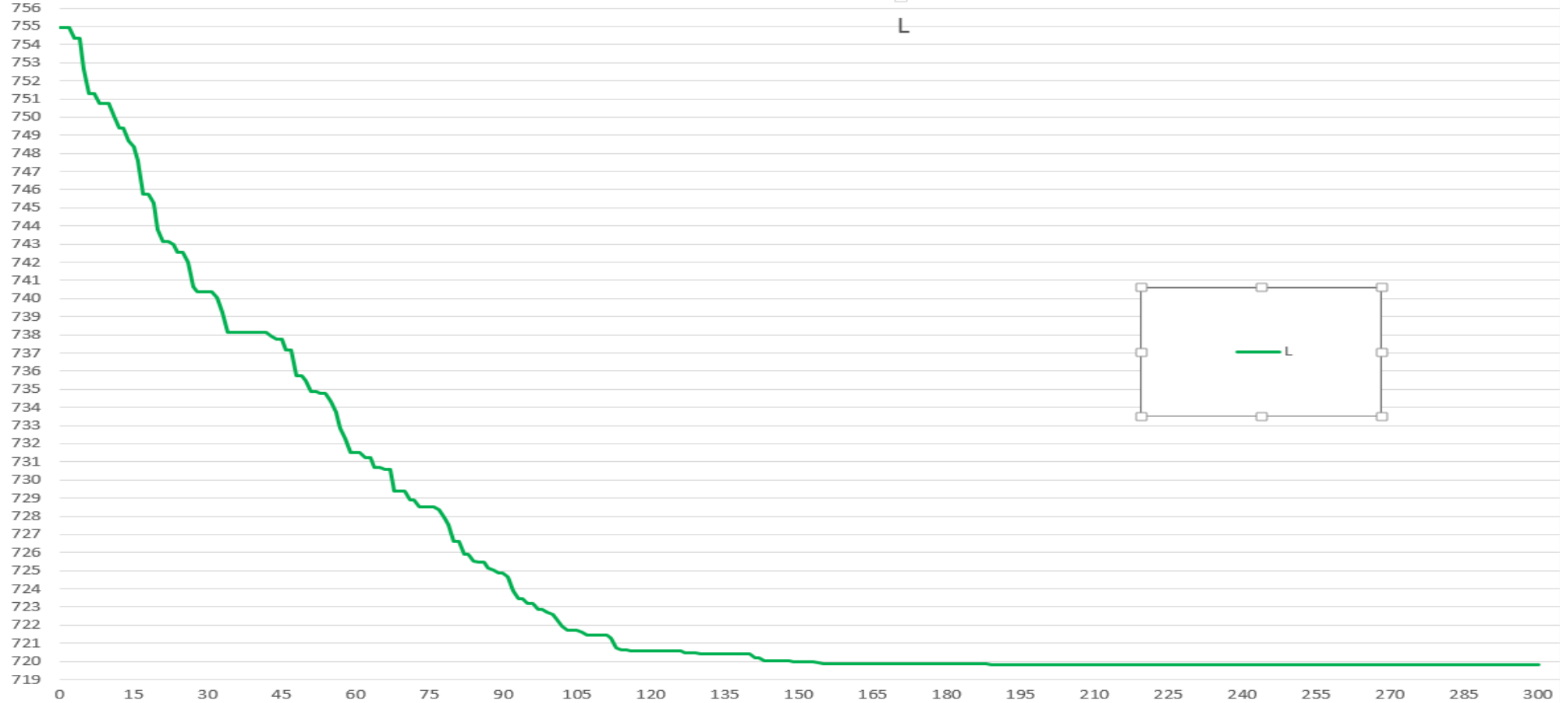
**Приложения**

Рис.1 строение мыльной пленки

|  |  |  |
| --- | --- | --- |
| Жидкость | Температура, °С | σ, Н/м |
| Вода | 20 | 0,0725 |
| Раствор мыла | 20 | 0,040 |
| Спирт | 20 | 0,022 |
| Ртуть | 20 | 0,47 |
| Золото | 1130 | 1,102 |
| Жидкий водород | -253 | 0,0021 |
| Жидкий гелий | -269 | 0,00012 |

|  |
| --- |
| Рис.2 таблица коэффициентов поверхностного натяжения жидкостей |
|  |
|  |
| Рис.3 Результаты экспериментального метода |

|  |
| --- |
|  |
|  |
|  |
| Рис.4 Результаты численной программы для трех, четырех и пяти точек |



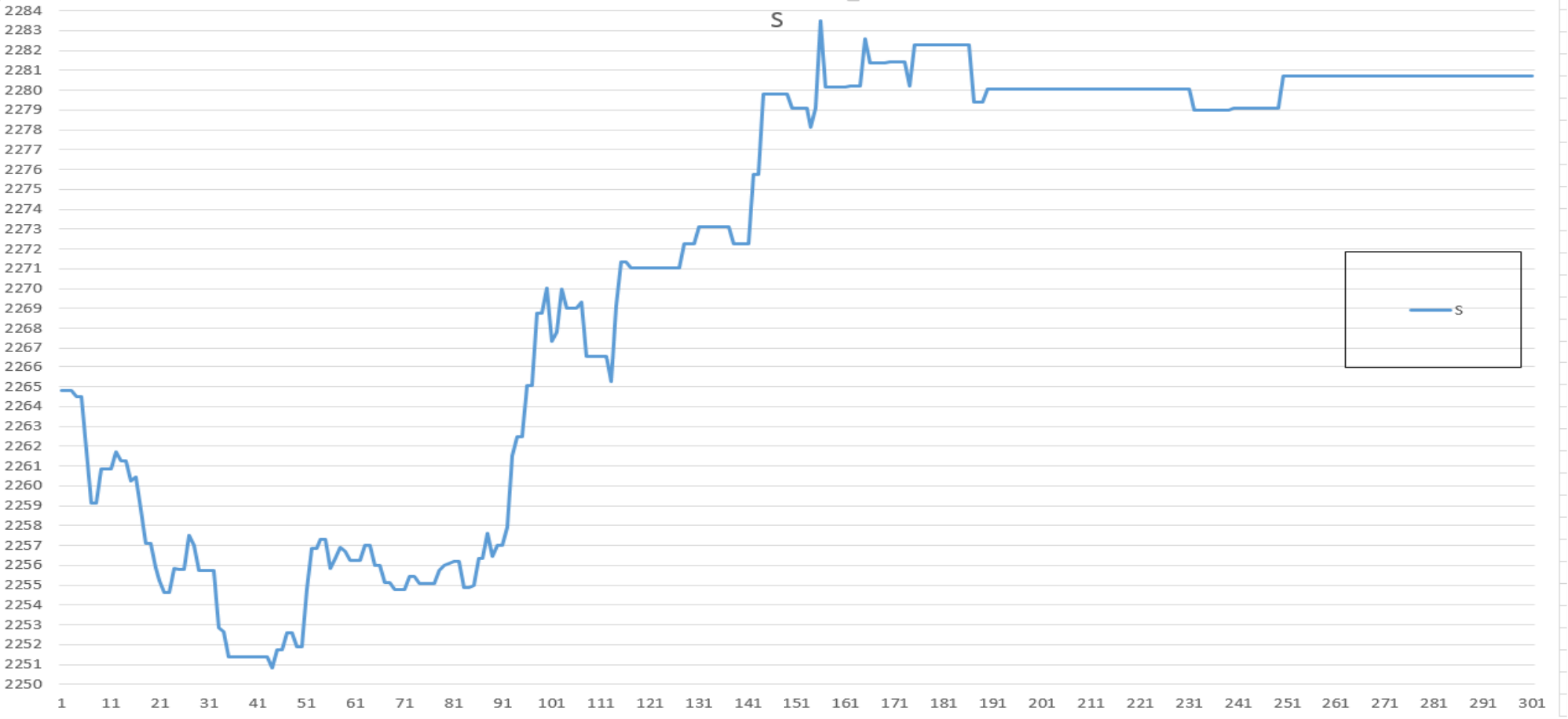


Рис.5 Графики изменения суммы маршрутов(S) и изменения длины всей сети(L)

**Листинг программы**

// Программа для вычисления наикратчайшей "сети дорог"

program Minimize;

uses

crt, graphABC;

const

x1 = 100;

y1 = 100; // Координаты первой точки

x2 = 400;

y2 = 100; // Координаты второй точки

x3 = 200;

y3 = 300; // Координаты третьей точки

var

x0, y0: real; // координаты перемещаемой точки

x00, y00: real;

L, L0: real; // суммарная длина "дорог"

L1, L2, L3: real; // длины отдельных дорог

col: integer;

s: string;

cs12, cs23, cs13: real;// косинусы углов между "дорогами"

begin

clearwindow(clGray);

setpencolor(clYellow);

setbrushcolor(clBlue);

// Первое приближение для "блуждающей точки" - среднее арифметическое координат

x0 := (x1 + x2 + x3) div 3;

y0 := (y1 + y2 + y3) div 3;

col := 0;

setbrushcolor(clGray);

setfontcolor(clWhite);

textout(x1, y1, '1');

textout(x2, y2, '2');

textout(x3, y3, '3');

repeat

clearwindow(clgray);

setfontcolor(clWhite);

textout(x1,y1,'1');

textout(x2,y2,'2');

textout(x3,y3,'3');

// Изобразим "сеть дорог"

line(round(x0), round(y0), x1, y1);

line(round(x0), round(y0), x2, y2);

line(round(x0), round(y0), x3, y3);

setpencolor(clYellow);

setbrushcolor(clBlue);

// Изобразим три точки

circle(x1, y1, 4);

circle(x2, y2, 4);

circle(x3, y3, 4);

// Изобразим "блуждающую точку"

circle(round(x0), round(y0), 4);

// Вычислим суммарную длину дорог

L1 := sqrt(sqr(x0 - x1) + sqr(y0 - y1));

L2 := sqrt(sqr(x0 - x2) + sqr(y0 - y2));

L3 := sqrt(sqr(x0 - x3) + sqr(y0 - y3));

L := L1 + L2 + L3;

// Вычислим косинус угла между "дорогами"

cs12 := ((x1 - x0) \* (x2 - x0) + (y1 - y0) \* (y2 - y0)) / (L1 \* L2);

cs23 := ((x2 - x0) \* (x3 - x0) + (y2 - y0) \* (y3 - y0)) / (L2 \* L3);

cs13 := ((x1 - x0) \* (x3 - x0) + (y1 - y0) \* (y3 - y0)) / (L1 \* L3);

setbrushcolor(clGray);

setfontcolor(clLime);

str(L1:6:3, s);

textout(500, 10, 'L1 = ' + s);

str(L2:6:3, s);

textout(500, 30, 'L2 = ' + s);

str(L3:6:3, s);

textout(500, 50, 'L3 = ' + s);

str(L:6:3, s);

textout(500, 70, 'L = ' + s);

str(x0:6:3, s);

textout(500, 90, 'x0 = ' + s);

str(y0:6:3, s);

textout(500, 110, 'y0 = ' + s);

str(cs12:6:3, s);

textout(500, 130, 'cos(1,2) = ' + s);

str(cs23:6:3, s);

textout(500, 150, 'cos(2,3) = ' + s);

str(cs13:6:3, s);

textout(500, 170, 'cos(1,3) = ' + s);

// Задаём виртуальное смещение для "блуждающей точки"

x00 := x0 + random \* (5 - random(10));

y00 := y0 + random \* (5 - random(10));

// Вычисляем длину виртуальной сети дорог

L1 := sqrt(sqr(x00 - x1) + sqr(y00 - y1));

L2 := sqrt(sqr(x00 - x2) + sqr(y00 - y2));

L3 := sqrt(sqr(x00 - x3) + sqr(y00 - y3));

L0 := L1 + L2 + L3;

// если виртуальная короче - сделаем её реальной!

if L0 <= L then

begin

x0 := x00;

y0 := y00;

end;

delay(100)

until keypressed;

end.