

Impact Factor:

ISRA (India) = 1.344	SIS (USA) = 0.912	ICV (Poland) = 6.630
ISI (Dubai, UAE) = 0.829	ПИИИ (Russia) = 0.234	PIF (India) = 1.940
GIF (Australia) = 0.564	ESJI (KZ) = 3.860	IBI (India) = 4.260
JIF = 1.500	SJIF (Morocco) = 2.031	

SOI: [1.1/TAS](#) DOI: [10.15863/TAS](#)

International Scientific Journal Theoretical & Applied Science

p-ISSN: 2308-4944 (print) e-ISSN: 2409-0085 (online)

Year: 2017 Issue: 05 Volume: 49

Published: 5.05.2017 <http://T-Science.org>

Ular Abenov

Master student

Taraz State University after M.Kh.Dulaty

ular.abenov@mail.ru

Alexandr Shevtsov

candidate of technical sciences, member of PILA (USA),
member of European Academy of Natural History (UK),

member of Federation of Robotics Kazakhstan,

Department of «Mathematics», Deputy Director on

Science of faculty of information technologies,

automation and telecommunications,

Taraz state University named after M.Kh. Dulaty

Shev_AlexXXXX@mail.ru

SECTION 2. Applied mathematics. Mathematical modeling.

THE APPLICATION OF SYSTEM MAPLE AND DELPHI TO SOLVE PROBLEMS OF DISCRETE MATHEMATICS

Abstract: A modern approach to the solution of problems of discrete mathematics due to the necessity of introducing and using known methods of solving problems. Existing programs and algorithms are not implemented to the extent necessary at the present time. The article is devoted to modeling and programming for Delphi and Maple for solving problems in discrete mathematics. The basis of research are well known and tested algorithms for the tasks of kommivojazhera – Greedy algorithm allows to find the optimal route.

Key words: greedy algorithm, Delphi, Maple.

Language: Russian

Citation: Abenov U, Shevtsov A (2017) THE APPLICATION OF SYSTEM MAPLE AND DELPHI TO SOLVE PROBLEMS OF DISCRETE MATHEMATICS. ISJ Theoretical & Applied Science, 05 (49): 101-119.

Soi: <http://s-o-i.org/1.1/TAS-05-49-17> **Doi:**  <https://dx.doi.org/10.15863/TAS.2017.05.49.17>

ПРИМЕНЕНИЕ СИСТЕМ MAPLE И DELPHI К МОДЕЛИРОВАНИЮ ЗАДАЧ ДИСКРЕТНОЙ МАТЕМАТИКИ

Аннотация: Современный подход к решению задач дискретной математики обусловлен необходимостью внедрения и использования известных методов решения задач. Существующие программы и алгоритмы не реализованы в необходимой мере в настоящее время. Статья посвящена моделированию и разработке программ для Delphi и Maple для решения задач дискретной математики. Основу исследований составляют хорошо известные и апробированные алгоритмы задачи коммивояжера – «Жадный алгоритм» позволяющие находить оптимальный маршрут движения.

Ключевые слова: жадный алгоритм, Delphi, Maple.

Introduction

Стремительное развитие уровня всех компьютерных математических приложений привело к парадоксальной ситуации, не оцененной пока должным образом.

С одной стороны, аналитические решения многих задач дискретной математики уже нельзя считать рациональными, так как компьютерные решения проводятся быстрее, то есть часто решать аналитически все равно, что ехать из Тараза в Шымкент через Астану.

С другой стороны, компьютерные решения не принимаются как полноценные! Однако несомненно, что такое положение временное и в недалеком будущем компьютерный способ решения станет если не основным, то равноправным с аналитическим. В Maple имеется мощная справочная система Help с пояснениями

и примерами. Рекомендуется использовать ее как можно чаще, тем более что Maple достаточно жесткая система, не допускающая никаких отклонений от установленных в ней правил. До последней версии Maple в пакете student, содержащем встроенные функции основных математических вычислений, необходимых в учебном процессе, была возможность получения пошаговых решений. В последней версии Maple пошаговые функции пакета student переданы пакету Student [Calculus], хотя сам пакет student остался. Однако основное назначение Maple состоит в оптимизации математических вычислений, включающей в себя получение результата минимумом нажатий на клавиши.

Применение Maple к проблемам дискретной математики довольно обширно:



Impact Factor:

ISRA (India) = 1.344	SIS (USA) = 0.912	ICV (Poland) = 6.630
ISI (Dubai, UAE) = 0.829	РИНЦ (Russia) = 0.234	PIF (India) = 1.940
GIF (Australia) = 0.564	ESJI (KZ) = 3.860	IBI (India) = 4.260
JIF = 1.500	SJIF (Morocco) = 2.031	

Горюшкин А. рассматривает особенности применения пакета символьных математических вычислений Maple при исследовании комбинаторных конфигураций, объектов математической логики и алгебры графов [1].

Сдвижков О.А. рассматривает основные правила работы в среде, методы и способы решения задач по элементарной и высшей математике, геометрическим построениям, теории вероятностей и математической статистике. Отдельная глава посвящена математическим моделям в экономике. Приведено много примеров решения задач: теории вероятности, математической статистики, алгебры логики, линейного программирования, матричные игры, транспортная задача, балансовые модели, потоки в сетях, сетевое планирование, целочисленное программирование, задача Эрланга. [2]

Тарасевич Ю.Ю. в книге «Элементы дискретной математики для программистов» рассмотрел следующие разделы: теория графов, теоремы Понтрягина-Куратовского и Эйлера, Эйлеровы и гамильтоновы графы, рёберные и дуальные графы, применение пакета Maple для решения задач теории графов, комбинаторика, матрица перестановок, рекуррентные соотношения, производящие функции, числа Фибоначчи, числа Каталана, неоднородные рекуррентные соотношения, применение пакета Maple для решения комбинаторных задач, алгоритмы обхода двоичного дерева, задача о коммивояжере, алгоритм Хошена-Копельмана, алгоритм поиска в глубину, алгоритм поиска в ширину [3].

Применимость жадных алгоритмов

Общего критерия оценки применимости жадного алгоритма для решения конкретной задачи не существует, однако, для задач, решаемых жадными алгоритмами характерны две особенности: в о-первых, к ним применим *Принцип жадного выбора*, а во-вторых, они обладают свойством *Оптимальности для подзадач*.

Принцип жадного выбора

Говорят, что к оптимизационной задаче применим **принцип жадного выбора**, если последовательность локально оптимальных выборов даёт глобально оптимальное решение. В типичном случае доказательство оптимальности следует такой схеме:

- Сначала доказывается, что жадный выбор на первом шаге не закрывает пути к оптимальному решению: для всякого решения есть другое, согласованное с жадным выбором и не хуже первого.
- Затем показывается, что подзадача, возникающая после жадного выбора на первом шаге, аналогична исходной, и рассуждение завершается по индукции.

Materials and Methods

Теоретическая постановка задачи

Задача коммивояжёра — это одна из самых известных задач, заключающаяся в отыскании самого выгодного маршрута, проходящего через указанные города хотя бы по одному разу с последующим возвратом в исходный город. В условиях задачи также указываются критерий выгодности маршрута:

- кратчайший,
- самый дешёвый,
- совокупный критерий

и т. п.

Маршрут должен проходить через каждый город только один раз (выбор осуществляется среди **гамильтоновых** циклов).

Поскольку коммивояжер в каждом из городов встает перед выбором следующего города из тех, что он ещё не посетил, существует

$$\frac{(n-1)!}{2}$$

маршрутов для симметричной задачи коммивояжера. Таким образом, для случаев

№	n	Число маршрутов
1	10	$\frac{9!}{2} = 181440$
2	100	$\frac{99!}{2} \approx 4 \cdot 10^{155}$
3	300	$\frac{299!}{2} \approx 5 \cdot 10^{611}$

Impact Factor:

ISRA (India) = 1.344	SIS (USA) = 0.912	ICV (Poland) = 6.630
ISI (Dubai, UAE) = 0.829	ПИИЦ (Russia) = 0.234	PIF (India) = 1.940
GIF (Australia) = 0.564	ESJI (KZ) = 3.860	IBI (India) = 4.260
JIF = 1.500	SJIF (Morocco) = 2.031	

Совершенно очевидно, что решить задачу методом прямого перебора (или «грубой силы») не сможет даже самый мощный компьютер!

Не случайно в условии сделан акцент на приближённый алгоритм.

«Жадный алгоритм», или «метод ближайшего соседа» — это один из простейших методов решения задачи коммивояжёра.

Его можно сформулировать следующим образом:

- Города последовательно включаются в маршрут,
- при этом каждый очередной включаемый город должен быть ближайшим к последнему выбранному городу среди всех остальных, ещё не включенных в состав маршрута.

2.3 Разработка компьютерного алгоритма и блок схемы программы

Составим математический алгоритм.

Пользователь задаёт количество городов – константа CITY_COUNT.

Расстояния между городами хранятся в квадратном массиве Distances[1..CITY_COUNT][1..CITY_COUNT].

А оптимальный путь, представляющий собой оптимальную последовательность индексов городов, хранится в линейном массиве Path[1..CITY_COUNT].

1. Происходит первоначальная инициализация карты городов. Для этого используем случайный алгоритм (выполняя требование исходной задачи «Города определить случайным образом»).

2. Ищется путь коммивояжёра – процедура CalculationPath.

1. В ней рассчитывается матрица взаимных расстояний между городами Distance. По диагонали в матрице хранятся -1, верхний треугольник матрицы рассчитывается и копируется в нижний, т.к. матрица симметрична относительно главной диагонали.

2. Далее «пробегаем» по всем городам (переменная iCurrent), начиная с начального (iStart), и для каждого ищем ближайший город (до которого расстояние минимально), запоминаем его в переменной iMedian и добавляем в путь Path. При поиске ближайшего города игнорируем те города, в которые уже заходили (дистанция до которых =-1). Попутно ищем общую протяжённость пути (Length);

3. После включения очередного города в путь, вычёркиваем его из рассмотрения (ставим в матрицу расстояний -1 в соответствующие этому городу столбец и строку).

Анализ найденных решений, особенно для трёхсот городов (длинная дорога, по которой коммивояжер возвращается в родной город из последнего пункта назначения), подтверждает тезис о том, что «жадный алгоритм» может давать результат, не более чем в два раза превышающий действительно оптимальный маршрут. Одним из эвристических критериев оценки решения является правило: если путь, пройденный на последних шагах алгоритма, сравнимым с путём, пройденным на начальных шагах, то можно условно считать найденный маршрут приемлемым, иначе, вероятно, существуют более оптимальные решения.

Разработка программы для Maple

Impact Factor:

ISRA (India) = 1.344	SIS (USA) = 0.912	ICV (Poland) = 6.630
ISI (Dubai, UAE) = 0.829	PIHHI (Russia) = 0.234	PIF (India) = 1.940
GIF (Australia) = 0.564	ESJI (KZ) = 3.860	IBI (India) = 4.260
JIF = 1.500	SJIF (Morocco) = 2.031	

```
restart;
CITY_COUNT := 5 :
s := 1 : ds := 0 : im := 1 :
for i from 1 to CITY_COUNT do
  g_x[i] := rand() mod 750;
  g_y[i] := rand() mod 500;
od;

for i from 1 to CITY_COUNT do
  Distances[i, i] := -1;
od;

for i from 1 to CITY_COUNT - 1 do
  for j from i + 1 to CITY_COUNT do
    x := g_x[i] - g_x[j];
    y := g_y[i] - g_y[j];
    d :=  $\frac{\text{trunc}(\text{sqrt}(x \cdot x + y \cdot y) \cdot 100)}{100}$ ;

    Distances[i, j] := d;
    Distances[j, i] := d;
  od;
od;

im0 := 1 :
im1 := 1 :

for k from 1 to CITY_COUNT - 1 do
  dmin := 10000000;
  for i from 1 to CITY_COUNT do
    d := Distances[i, im];
    if (0 < d) and (d < dmin) then
      it := i;
      dmin := d;
    fi;
  od;
  im := it;

for i from 1 to CITY_COUNT do
  Distances[i, im0] := -1;
  Distances[im0, i] := -1;
od;
s := s, im;
ds := ds + dmin;
im0 := im :
od;
```

284
415
715
459
119



Impact Factor:	ISRA (India) = 1.344	SIS (USA) = 0.912	ICV (Poland) = 6.630
	ISI (Dubai, UAE) = 0.829	ПИИИ (Russia) = 0.234	PIF (India) = 1.940
	GIF (Australia) = 0.564	ESJI (KZ) = 3.860	IBI (India) = 4.260
	JIF = 1.500	SJIF (Morocco) = 2.031	

442
590
180
450
265
-1
-1
-1
-1
-1
10000000
3
1, 3
16719
100
3
10000000
5
1, 3, 5
27127
50
5
10000000
4
1, 3, 5, 4
17658
25
4
10000000
2
1, 3, 5, 4, 2
25301
25
2

Таблица 1

Проведем ряд расчетов для различных параметров

Количество городов	Оптимальный путь	Длина оптимального пути
10	1, 6, 9, 3, 5, 10, 4, 7, 8, 2	<u>166053</u> 100
100	1, 49, 19, 28, 69, 92, 55, 87, 6, 11, 18, 53, 96, 76, 41, 58, 56, 3, 45, 90, 9, 25, 40, 39, 81, 80, 91, 84, 67, 65, 62, 42, 24, 63, 14, 13, 75, 8, 59, 88, 99, 29, 2, 47, 12, 74, 51, 10, 15, 16, 22, 4, 60, 98, 20, 34, 5, 26, 38, 72, 61, 68, 33, 35, 48, 100, 46, 54, 31, 7, 52, 97, 66, 82, 17, 93, 85, 94, 70, 30, 57, 64, 43, 95, 32, 83, 37, 27, 89, 78, 21, 73, 44, 50, 71, 86, 23, 79, 36, 77	<u>550317</u> 100



Impact Factor:

ISRA (India) = 1.344	SIS (USA) = 0.912	ICV (Poland) = 6.630
ISI (Dubai, UAE) = 0.829	ПИИИ (Russia) = 0.234	PIF (India) = 1.940
GIF (Australia) = 0.564	ESJI (KZ) = 3.860	IBI (India) = 4.260
JIF = 1.500	SJIF (Morocco) = 2.031	

300	<p>1, 191, 268, 49, 284, 179, 200, 146, 13, 14, 235, 219, 294, 230, 75, 233, 208, 259, 8, 159, 59, 155, 266, 51, 209, 136, 187, 281, 111, 271, 150, 85, 93, 166, 189, 17, 213, 260, 117, 31, 103, 210, 161, 156, 272, 239, 115, 289, 100, 48, 129, 107, 192, 206, 142, 26, 262, 34, 288, 188, 267, 261, 251, 274, 256, 215, 278, 250, 290, 291, 264, 214, 196, 182, 88, 123, 275, 228, 242, 186, 74, 157, 125, 12, 175, 220, 47, 286, 2, 180, 163, 245, 29, 183, 254, 99, 164, 207, 168, 176, 160, 133, 199, 237, 217, 293, 92, 69, 55, 153, 87, 6, 131, 244, 132, 165, 11, 18, 53, 248, 280, 128, 283, 96, 76, 41, 151, 114, 102, 154, 184, 120, 232, 40, 148, 25, 39, 134, 224, 172, 121, 201, 138, 81, 285, 226, 143, 45, 108, 3, 56, 243, 90, 229, 135, 9, 300, 299, 58, 101, 221, 106, 231, 185, 227, 152, 212, 23, 296, 86, 71, 126, 170, 141, 295, 44, 223, 89, 177, 263, 78, 190, 21, 277, 292, 110, 36, 137, 37, 252, 276, 32, 43, 95, 236, 234, 105, 270, 72, 61, 253, 147, 109, 140, 57, 64, 255, 77, 83, 124, 149, 171, 198, 27, 205, 50, 193, 282, 116, 241, 112, 197, 73, 297, 119, 222, 269, 79, 91, 80, 204, 238, 84, 67, 65, 127, 249, 194, 225, 240, 42, 130, 62, 139, 195, 104, 38, 265, 145, 178, 24, 63, 5, 20, 98, 246, 144, 60, 4, 22, 16, 181, 257, 15, 10, 211, 30, 70, 298, 167, 202, 158, 66, 82, 97, 203, 118, 122, 174, 52, 7, 54, 162, 46, 35, 33, 216, 113, 68, 287, 279, 94, 218, 247, 19, 28, 258, 173, 273, 169</p>	<p><u>986499</u> 100</p>
-----	---	------------------------------

Разработка программы для Delphi

Для разработки программы используем Delphi XE8. Разместим на форме необходимые компоненты, разделив выходящие данные на 4 вкладки.

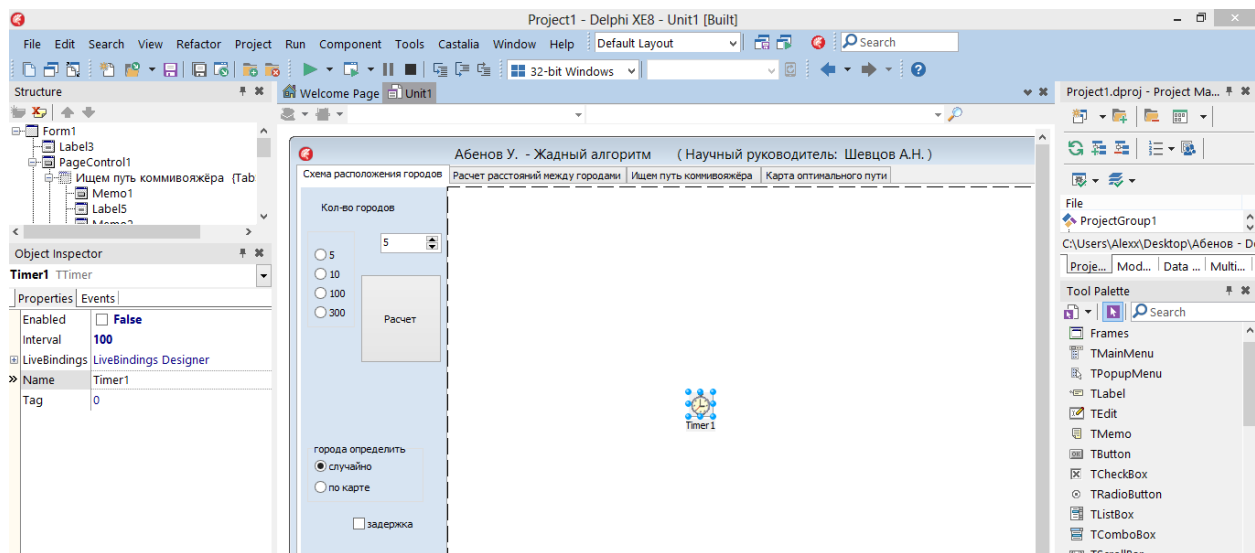


Рисунок 1 – Окно формы программы. Вкладка 1.

Impact Factor:

ISRA (India)	= 1.344	SIS (USA)	= 0.912	ICV (Poland)	= 6.630
ISI (Dubai, UAE)	= 0.829	ПИИЦ (Russia)	= 0.234	PIF (India)	= 1.940
GIF (Australia)	= 0.564	ESJI (KZ)	= 3.860	IBI (India)	= 4.260
JIF	= 1.500	SJIF (Morocco)	= 2.031		

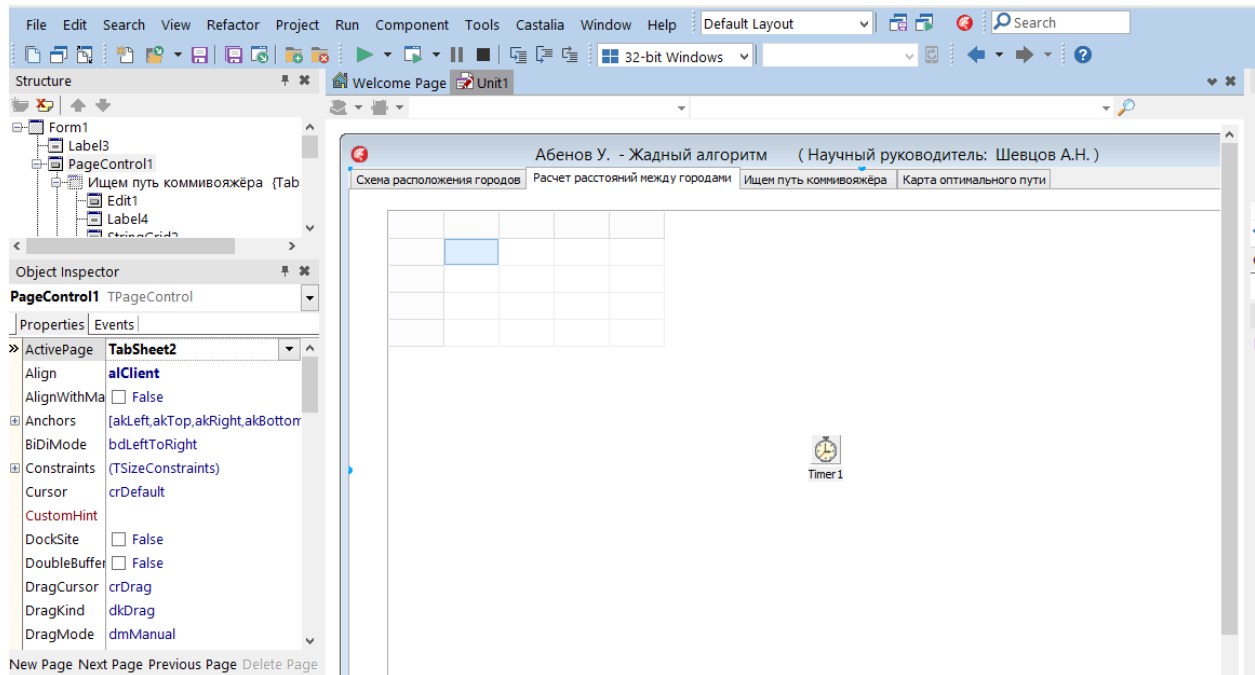


Рисунок 2 – Окно формы программы. Вкладка 2.

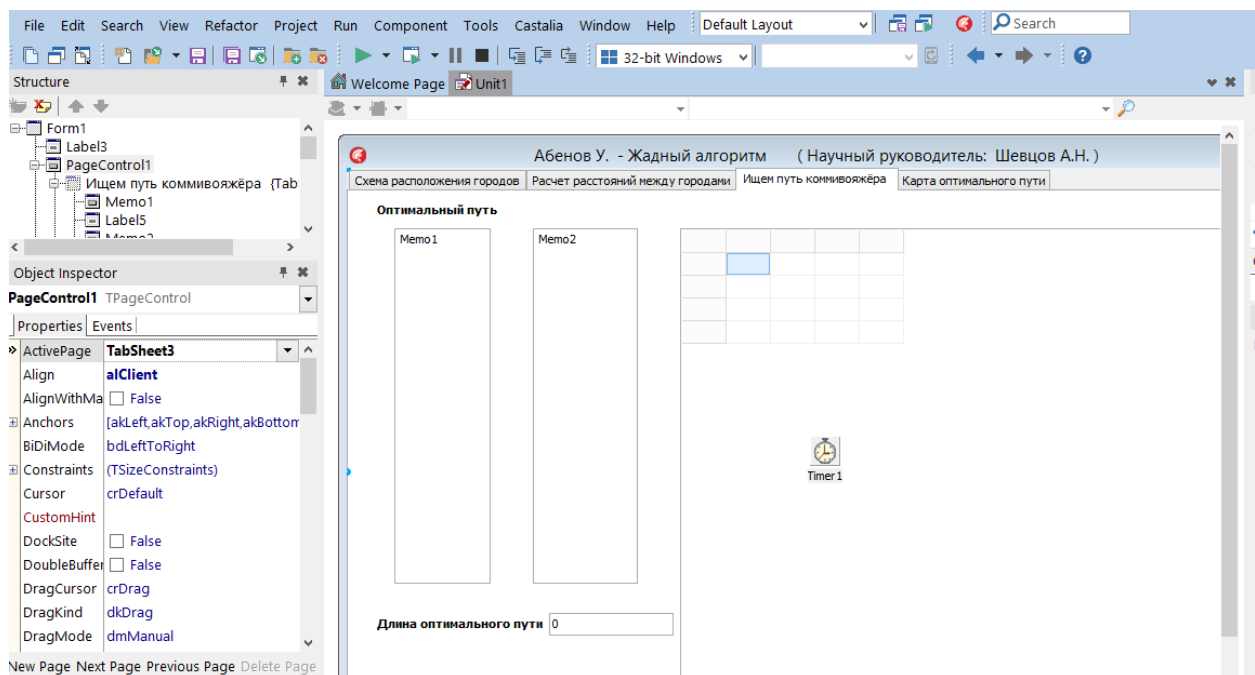


Рисунок 3 – Окно формы программы. Вкладка 3.

Impact Factor:

ISRA (India) = 1.344	SIS (USA) = 0.912	ICV (Poland) = 6.630
ISI (Dubai, UAE) = 0.829	ПИИЦ (Russia) = 0.234	PIF (India) = 1.940
GIF (Australia) = 0.564	ESJI (KZ) = 3.860	IBI (India) = 4.260
JIF = 1.500	SJIF (Morocco) = 2.031	

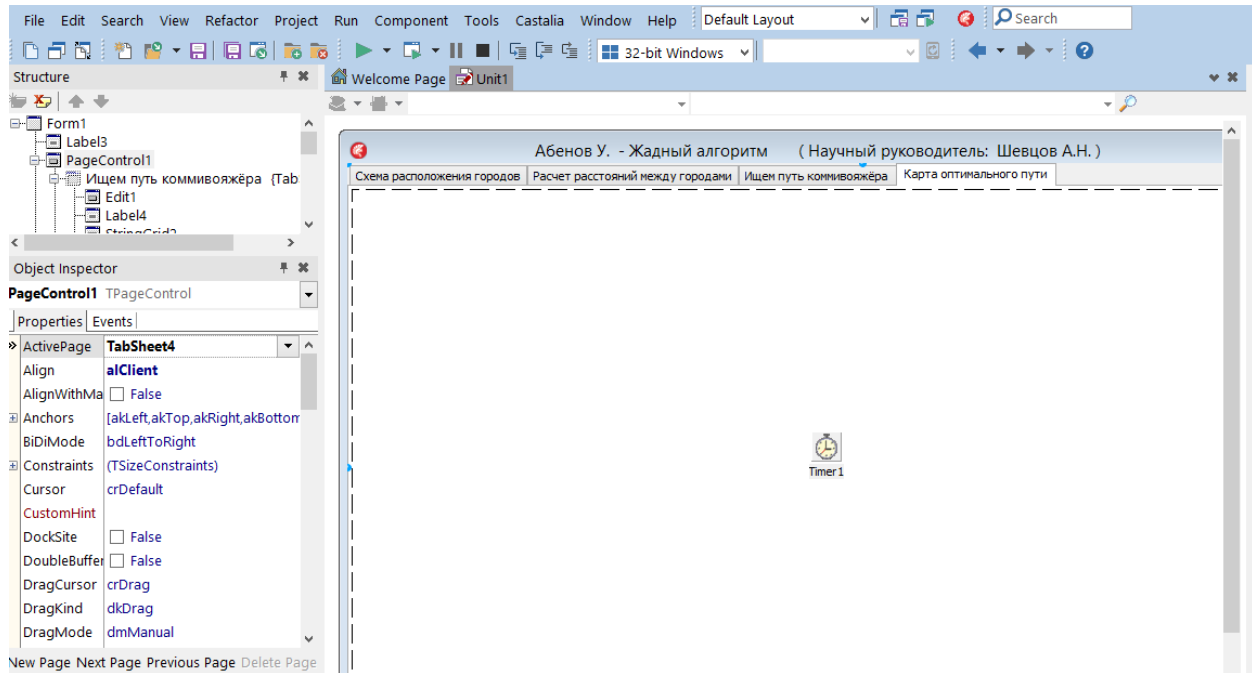


Рисунок 4 – Окно формы программы. Вкладка 4.

Описание алгоритмов «Жадного алгоритма» удобно привести в виде таблицы

Таблица 2

Реализация алгоритма на Delphi

Код Delphi	Подробное описание
<pre>procedure TForm1.Button1Click(Sender: TObject); begin image1.Canvas.Brush.Color:=clWhite; Image1.Canvas.FillRect(rect(0,0,1000,1000)); CITY_COUNT:=SpinEdit1.Value;</pre>	<p>Очистка карты городов</p> <p>Задание кол-ва городов</p>
<pre>case RadioGroup2.ItemIndex of 0:begin for I := 1 to CITY_COUNT do begin g[i].X:=random(750); g[i].Y:=random(500); image1.Canvas.Brush.Color:=clred; image1.Canvas.Ellipse(rect(g[i].X,g[i].y,g[i].X+5,g[i].y+5)); image1.Canvas.Brush.Color:=clWhite; image1.Canvas.TextOut(g[i].X+10,g[i].y-10,inttostr(i)); image2.Canvas.Brush.Color:=clred; image2.Canvas.Ellipse(rect(g[i].X,g[i].y,g[i].X+5,g[i].y+5)); image2.Canvas.Brush.Color:=clWhite; image2.Canvas.TextOut(g[i].X+10,g[i].y-10,inttostr(i)); end; end; 1:begin end ; end;</pre>	<p>Прорисовка городов на карте</p> <p>Произвольный выбор города</p> <p>Нумерация города</p>

Impact Factor:

ISRA (India) = 1.344	SIS (USA) = 0.912	ICV (Poland) = 6.630
ISI (Dubai, UAE) = 0.829	ПИИЦ (Russia) = 0.234	PIF (India) = 1.940
GIF (Australia) = 0.564	ESJI (KZ) = 3.860	IBI (India) = 4.260
JIF = 1.500	SJIF (Morocco) = 2.031	

<pre>stringgrid1.ColCount:=CITY_COUNT+1; stringgrid1.RowCount:=CITY_COUNT+1;</pre>	<p>Задание размеров таблицы расстояний между городами Причем расстояние между одинаковыми городами считаем равным -1</p>
<pre>for I := 1 to CITY_COUNT do begin stringgrid1.Cells[0,i]:=inttostr(i); stringgrid1.Cells[i,0]:=inttostr(i); stringgrid1.Cells[i,i]:=inttostr(-1); Distances[i,i]:=-1; end;</pre>	<p>Заполнение колонтитулов таблицы и значений между одинаковыми городами</p>
<pre>for I := 1 to CITY_COUNT-1 do for j := i+1 to CITY_COUNT do begin x:=g[i].X-g[j].X; y:=g[i].Y-g[j].Y; d:=int(sqrt(x*x+y*y)*100)/100; stringgrid1.Cells[j,i]:=floattostr(d); stringgrid1.Cells[i,j]:=floattostr(d); Distances[i,j]:=d; Distances[j,i]:=d; end;</pre>	<p>Расчет расстояний между всеми городами и симметричное заполнение таблицы относительно главной диагонали</p> <p>Заполнение массива с данными</p>
<pre>//ищем путь коммивояжера memo1.Clear; memo2.Clear; im0:=1; im:=1; memo1.Lines.Add(inttostr(im));</pre>	<p>Очистка окон вывода данных</p> <p>Задание начальных параметров</p>
<pre>for k := 1 to CITY_COUNT-1 do begin dmin:=100000000; for i := 1 to CITY_COUNT do begin d:=Distances[i,im]; if (0<d)and(d<dmin) then begin it:=i; dmin:=d; end; end;</pre>	<p>Расчет оптимальной траектории по Жадному алгоритму</p>
<pre>im:=it; for I := 1 to CITY_COUNT do begin Distances[i,im0]:=-1; Distances[im0,i]:=-1; end;</pre>	<p>Вычеркивание пройденного города из списка городов путем вычеркивания расстояний до других городов</p>
<pre>memo1.Lines.Add(inttostr(im));</pre>	<p>Вывод очередного города в</p>

Impact Factor:

ISRA (India) = 1.344	SIS (USA) = 0.912	ICV (Poland) = 6.630
ISI (Dubai, UAE) = 0.829	ПИИЦ (Russia) = 0.234	PIF (India) = 1.940
GIF (Australia) = 0.564	ESJI (KZ) = 3.860	IBI (India) = 4.260
JIF = 1.500	SJIF (Morocco) = 2.031	

memo2.Lines.Add(floattostr(dmin));	оптимальном пути и расстояния до него
<pre> im0:=im; stringgrid2.ColCount:=CITY_COUNT+1; stringgrid2.RowCount:=CITY_COUNT+1; for I := 1 to CITY_COUNT do for j := 1 to CITY_COUNT do stringgrid2.Cells[i,j]:=floattostr(Distances[i,j]); Application.ProcessMessages; if checkbox1.Checked then sleep(trunc(10000/CITY_COUNT)); end; ii:=1; timer1.Enabled:=true; </pre>	<p>Обновление таблицы с данными на 3 вкладке</p> <p>Анимация данных</p>
<pre> s:=0; for i:=0 to memo2.Lines.Count-1 do s:=s+strtofloat(memo2.Lines.Strings[i]); edit1.Text:=floattostr(s); end; </pre>	Расчет общей длины оптимального пути

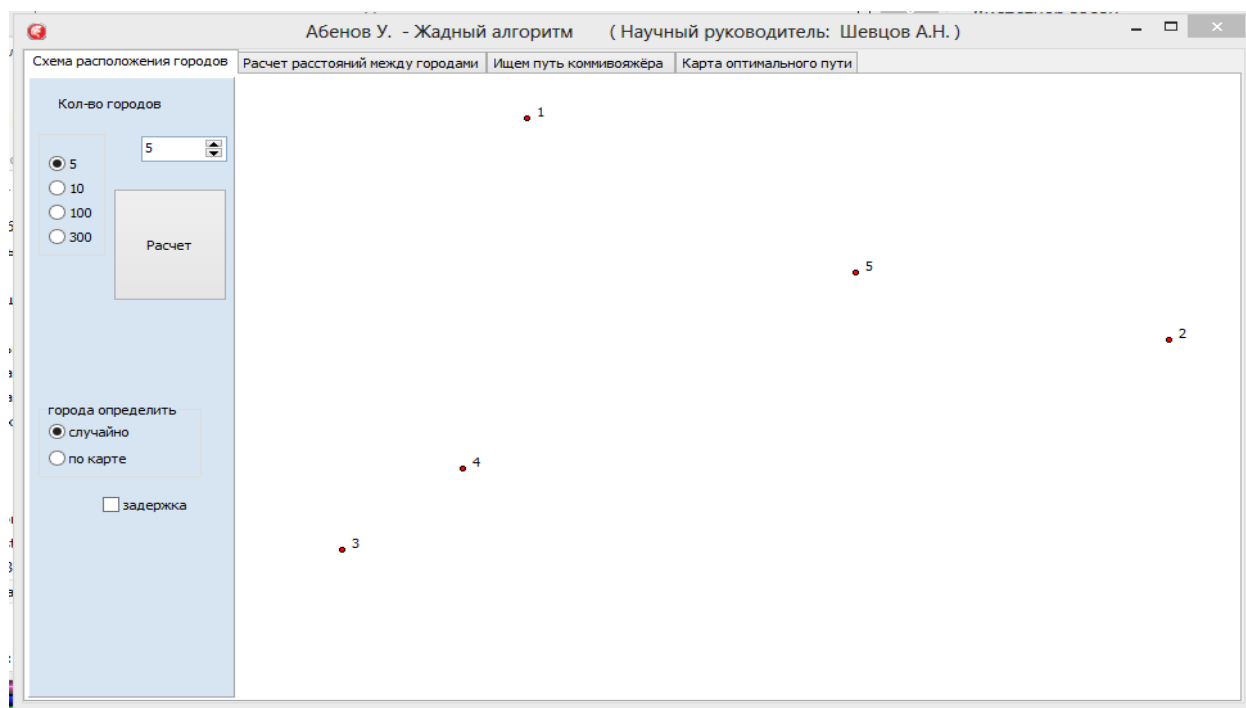


Рисунок 5 – Моделирование 5 городов.

Impact Factor:

ISRA (India)	= 1.344	SIS (USA)	= 0.912	ICV (Poland)	= 6.630
ISI (Dubai, UAE)	= 0.829	ПИИЦ (Russia)	= 0.234	PIF (India)	= 1.940
GIF (Australia)	= 0.564	ESJI (KZ)	= 3.860	IBI (India)	= 4.260
JIF	= 1.500	SJIF (Morocco)	= 2.031		

	1	2	3	4	5
1	-1	535,23	398,89	306,11	288,48
2	535,23	-1	668,95	561,08	250,79
3	398,89	668,95	-1	117,2	465,96
4	306,11	561,08	117,2	-1	349,56
5	288,48	250,79	465,96	349,56	-1

Рисунок 6 – Моделирование 5 городов. Расчет расстояний.

Оптимальный путь

1	288,48
5	250,79
2	561,08
4	117,2
3	

	-1	-1	-1	-1	-1
	-1	-1	-1	-1	-1
	-1	-1	-1	-1	-1
	-1	-1	-1	-1	-1
	-1	-1	-1	-1	-1

Длина оптимального пути

Рисунок 7 – Моделирование 5 городов. Расчет оптимального пути.

Impact Factor:

ISRA (India) = 1.344	SIS (USA) = 0.912	ICV (Poland) = 6.630
ISI (Dubai, UAE) = 0.829	ПИИЦ (Russia) = 0.234	PIF (India) = 1.940
GIF (Australia) = 0.564	ESJI (KZ) = 3.860	IBI (India) = 4.260
JIF = 1.500	SJIF (Morocco) = 2.031	

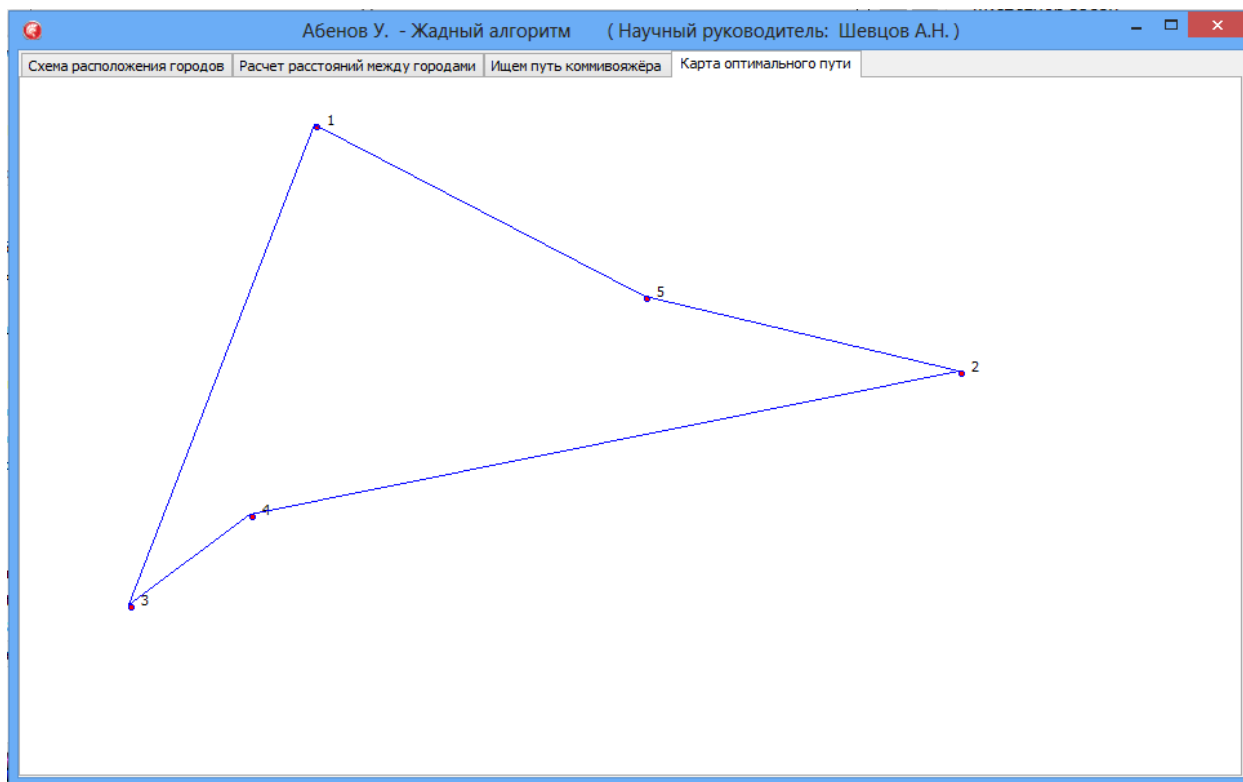


Рисунок 8 – Моделирование 5 городов. Траектория оптимального пути.

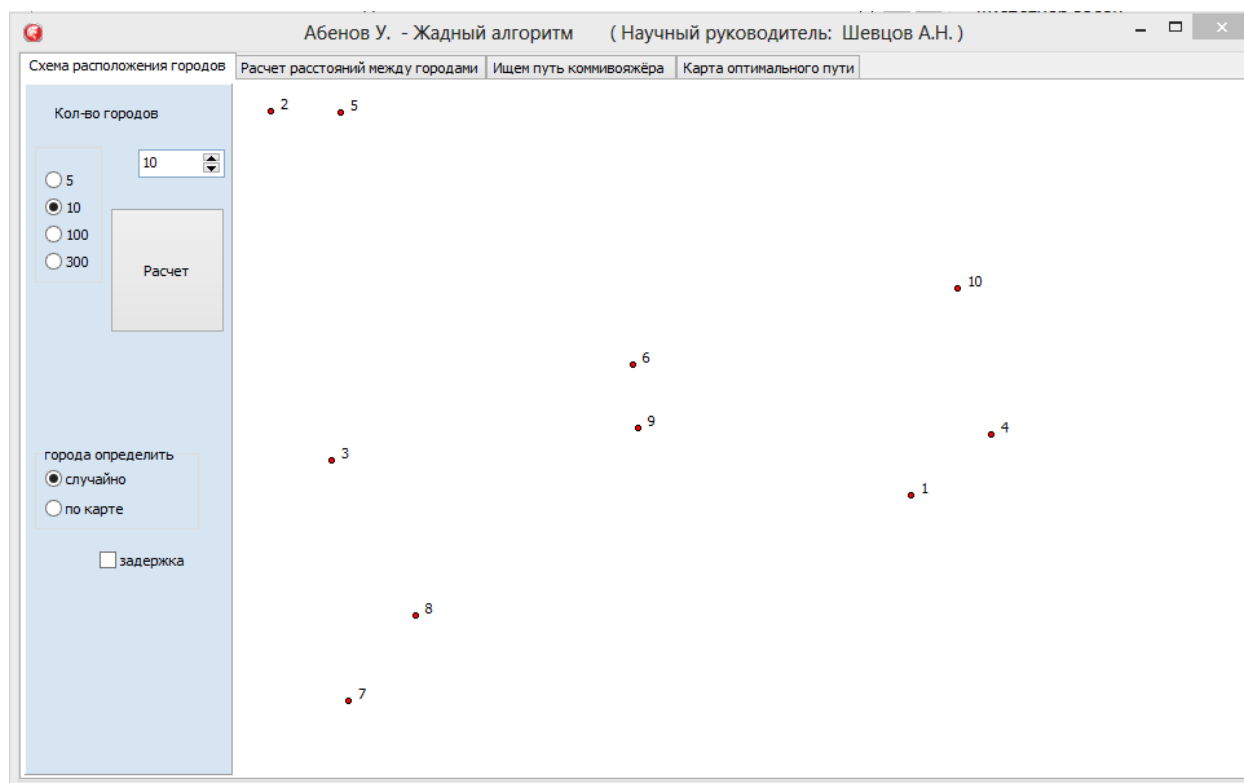


Рисунок 9 – Моделирование 10 городов. Расположение городов.

Impact Factor:

ISRA (India) = 1.344	SIS (USA) = 0.912	ICV (Poland) = 6.630
ISI (Dubai, UAE) = 0.829	ПИИЦ (Russia) = 0.234	PIF (India) = 1.940
GIF (Australia) = 0.564	ESJI (KZ) = 3.860	IBI (India) = 4.260
JIF = 1.500	SJIF (Morocco) = 2.031	

Абенов У. - Жадный алгоритм (Научный руководитель: Шевцов А.Н.)

Схема расположения городов | Расчет расстояний между городами | Ищем путь коммивояжера | Карта оптимального пути

	1	2	3	4	5	6	7	8	9	10
1	-1	577,26	448,81	77,8	531,12	237,54	463,14	394,12	217,31	164
2	577,26	-1	274,06	610,53	54	341,78	459,93	405,76	375,07	548,38
3	448,81	274,06	-1	510,39	269,09	244,46	186,45	136,47	238,31	501,94
4	77,8	610,53	510,39	-1	561,25	282,21	538	466,5	273,04	115,95
5	531,12	54	269,09	561,25	-1	298,49	455,03	393,3	335,31	496
6	237,54	341,78	244,46	282,21	298,49	-1	340,58	256,63	49,16	257,84
7	463,14	459,93	186,45	538	455,03	340,58	-1	84,02	307,72	568,86
8	394,12	405,76	136,47	466,5	393,3	256,63	84,02	-1	224,96	489,45
9	217,31	375,07	238,31	273,04	335,31	49,16	307,72	224,96	-1	269,57
10	164	548,38	501,94	115,95	496	257,84	568,86	489,45	269,57	-1

Рисунок 10 – Моделирование 5 городов. Расчет расстояний между городами.

Абенов У. - Жадный алгоритм (Научный руководитель: Шевцов А.Н.)

Схема расположения городов | Расчет расстояний между городами | Ищем путь коммивояжера | Карта оптимального пути

Оптимальный путь

1	77,8
4	115,95
10	257,84
6	49,16
9	224,96
8	84,02
7	186,45
3	269,09
5	54
2	

-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1

Длина оптимального пути: 1319,27

Рисунок 11 – Моделирование 10 городов. Расчет оптимального пути и его длины.

Impact Factor:

ISRA (India) = 1.344	SIS (USA) = 0.912	ICV (Poland) = 6.630
ISI (Dubai, UAE) = 0.829	ПИИЦ (Russia) = 0.234	PIF (India) = 1.940
GIF (Australia) = 0.564	ESJI (KZ) = 3.860	IBI (India) = 4.260
JIF = 1.500	SJIF (Morocco) = 2.031	

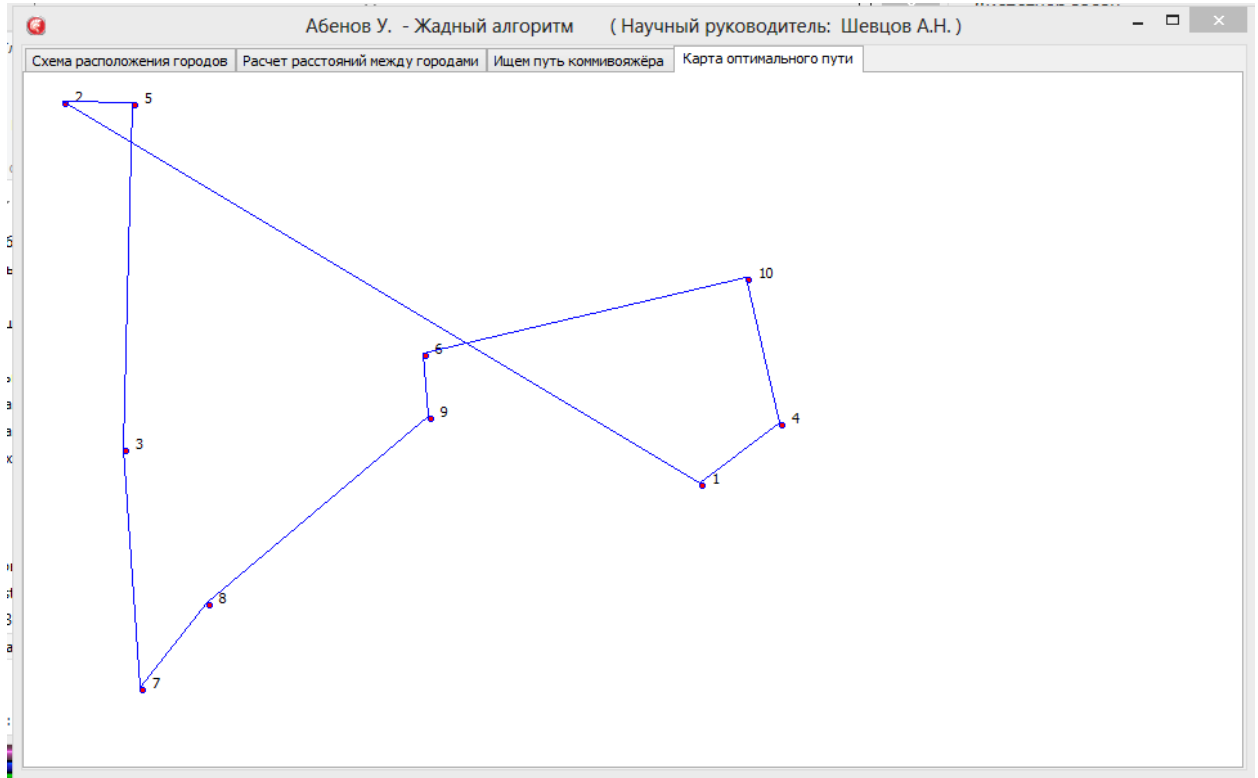


Рисунок 12 – Моделирование 10 городов. Траектория оптимального пути.

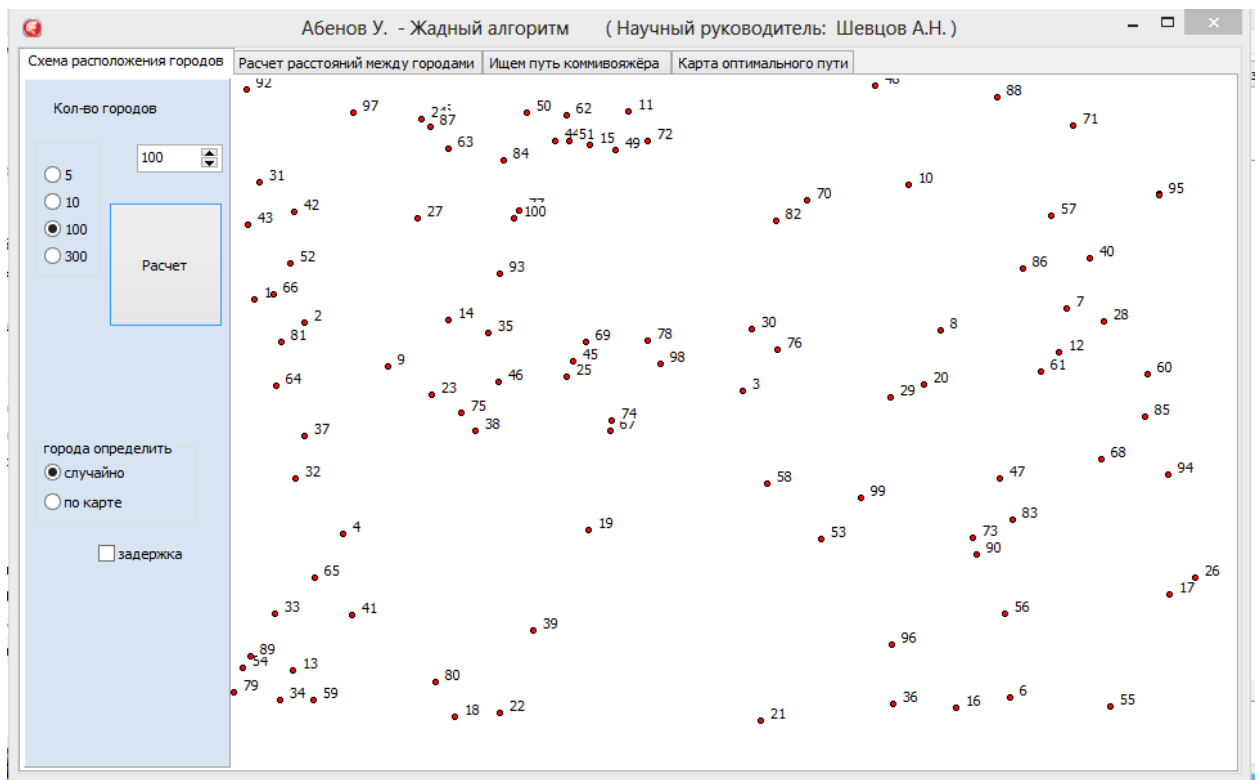


Рисунок 13 – Моделирование 100 городов. Положение на карте.

Impact Factor:

ISRA (India) = 1.344	SIS (USA) = 0.912	ICV (Poland) = 6.630
ISI (Dubai, UAE) = 0.829	ПИИЦ (Russia) = 0.234	PIF (India) = 1.940
GIF (Australia) = 0.564	ESJI (KZ) = 3.860	IBI (India) = 4.260
JIF = 1.500	SJIF (Morocco) = 2.031	

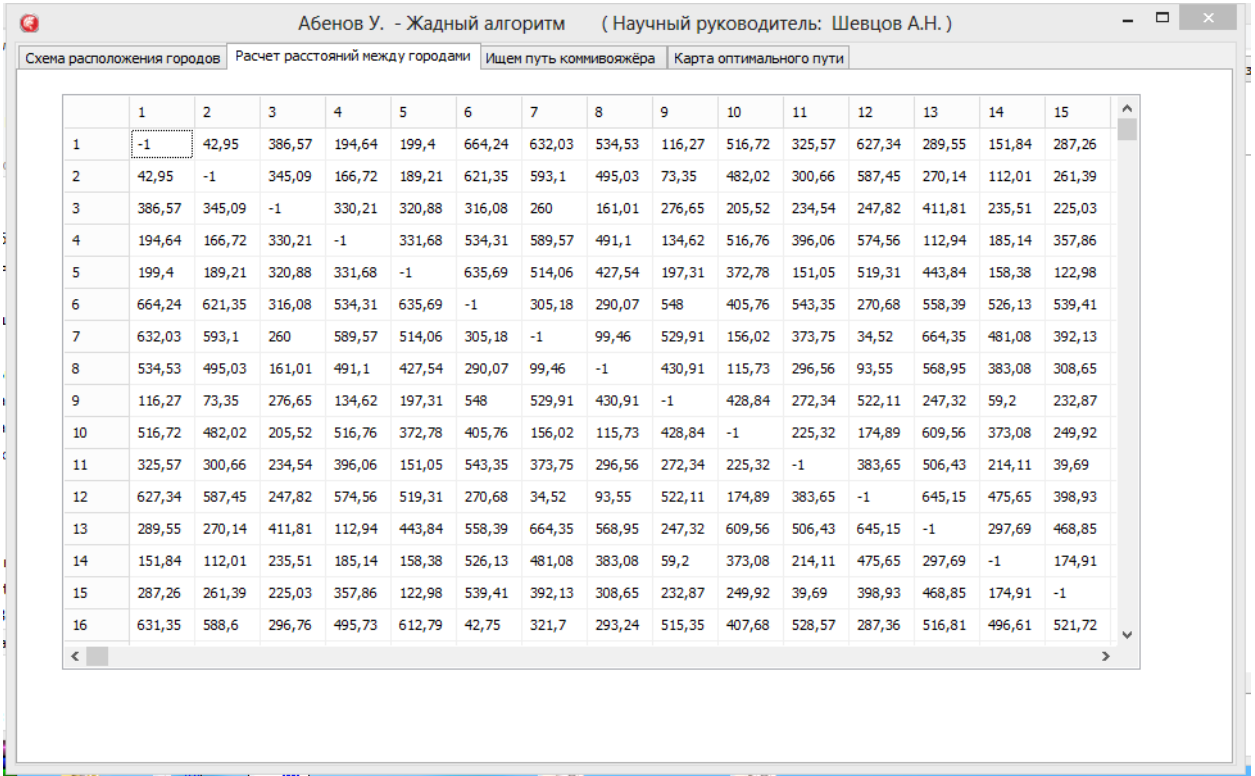


Рисунок 14 – Моделирование 100 городов. Таблица расстояний между городами.

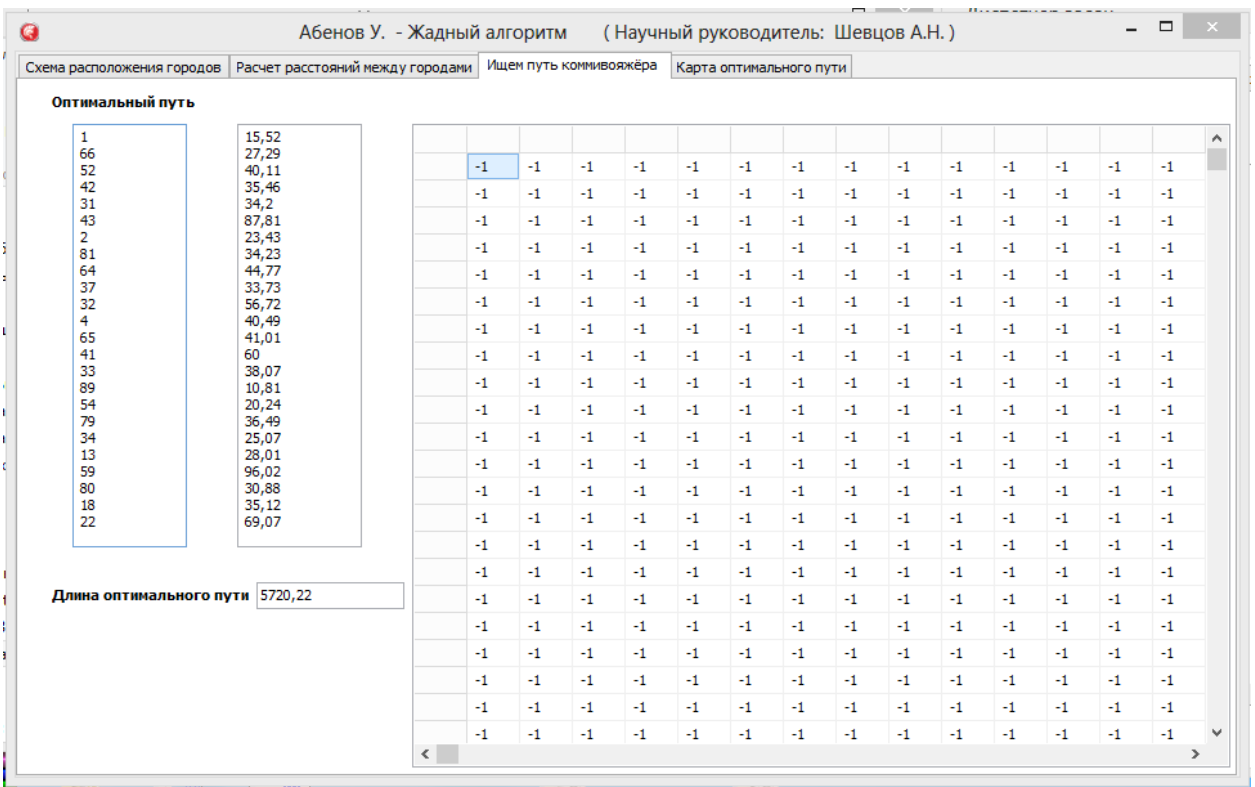


Рисунок 15 – Моделирование 100 городов. Расчет длины оптимального пути.

Impact Factor:

ISRA (India)	= 1.344	SIS (USA)	= 0.912	ICV (Poland)	= 6.630
ISI (Dubai, UAE)	= 0.829	ПИИЦ (Russia)	= 0.234	PIF (India)	= 1.940
GIF (Australia)	= 0.564	ESJI (KZ)	= 3.860	IBI (India)	= 4.260
JIF	= 1.500	SJIF (Morocco)	= 2.031		

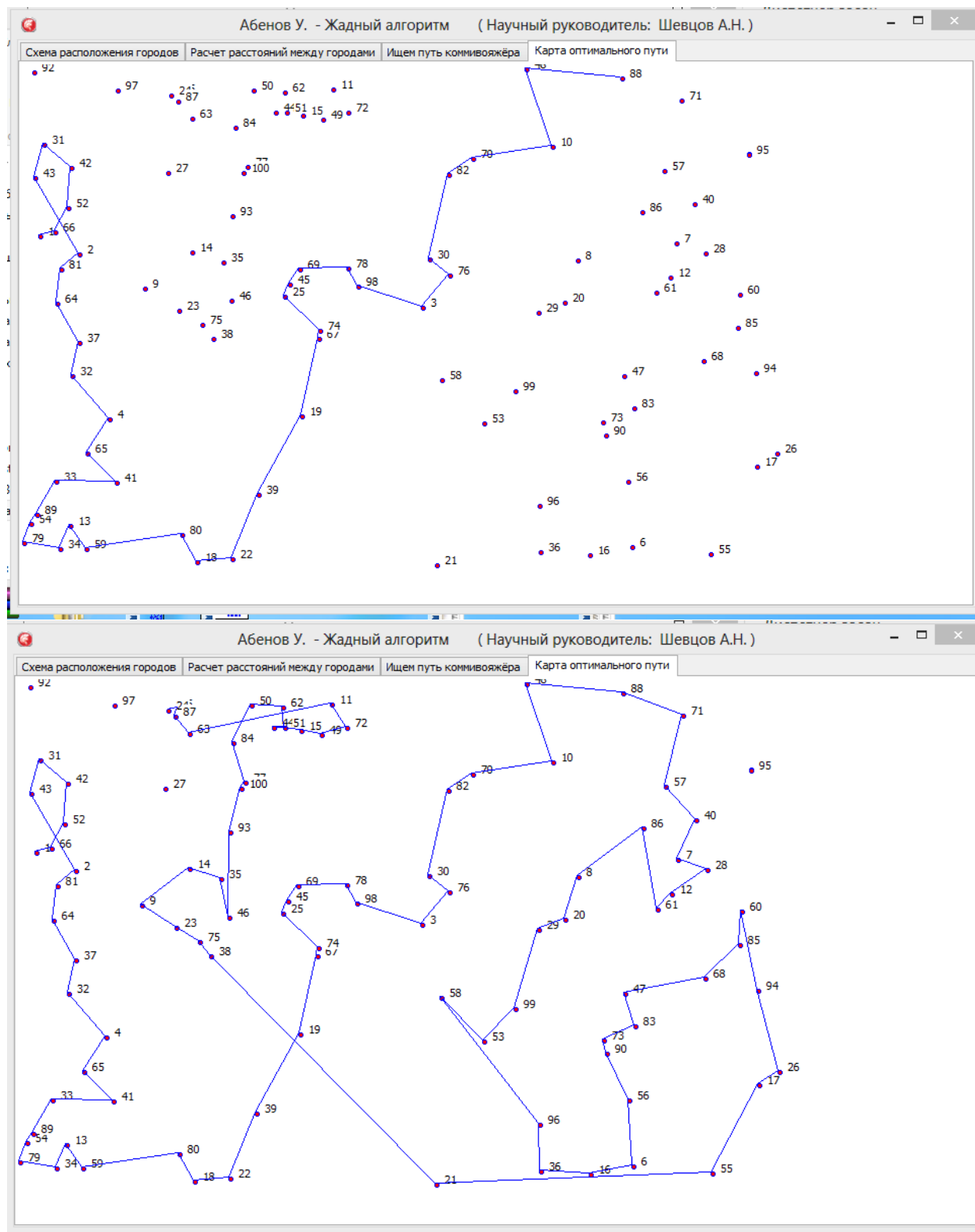


Рисунок 16 – Моделирование 100 городов.

Impact Factor:

ISRA (India) = 1.344
ISI (Dubai, UAE) = 0.829
GIF (Australia) = 0.564
JIF = 1.500

SIS (USA) = 0.912
ПИИЦ (Russia) = 0.234
ESJI (KZ) = 3.860
SJIF (Morocco) = 2.031

ICV (Poland) = 6.630
PIF (India) = 1.940
IBI (India) = 4.260

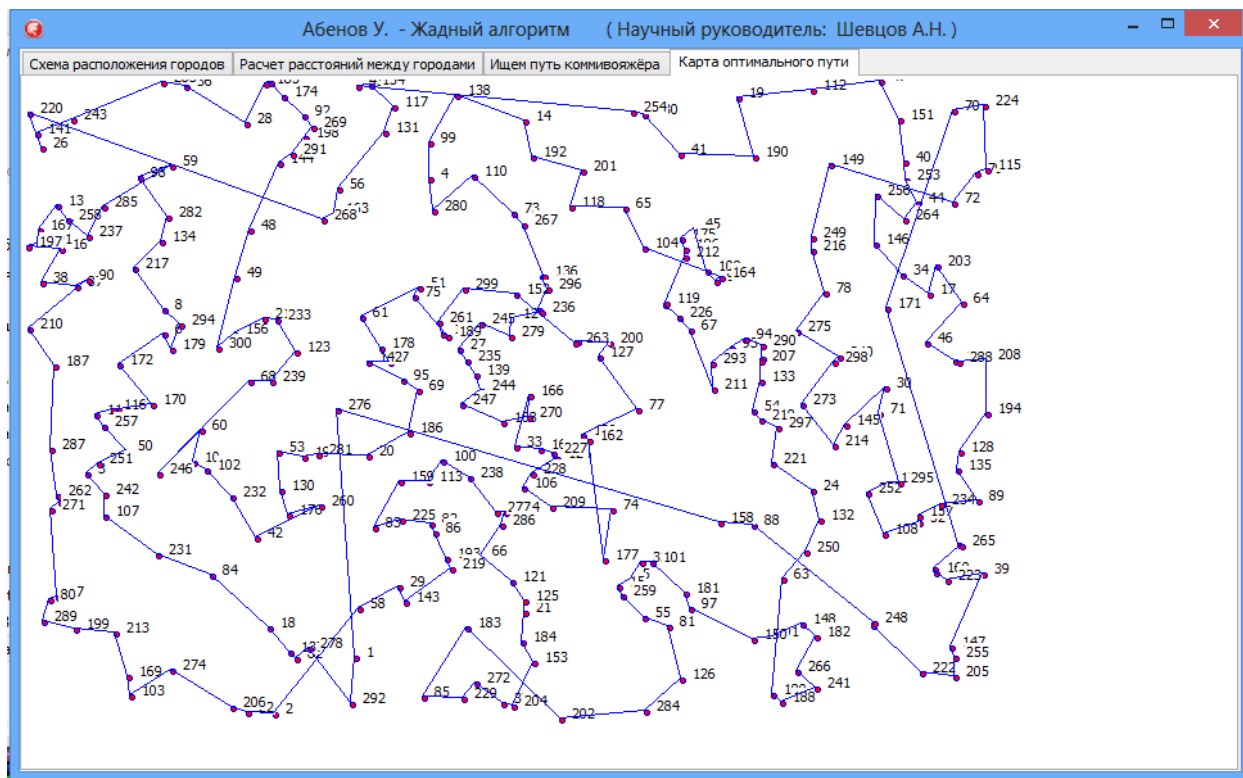


Рисунок 19 – Моделирование 300 городов. Траектория оптимального пути.

Conclusion

В результате проведенного исследования были получены следующие результаты:

- Разработаны алгоритмы и программа для решения задачи поиска оптимального пути на Maple,
- Разработаны алгоритмы и программа для решения задачи поиска оптимального пути на Delphi,
- Разработанные алгоритмы реализованы и апробированы для различных начальных условий.

References:

- Gorjushkin A (2016) Diskretnaja matematika v Maple. 2016. LAP LAMBERT Academic Publishing. 436 p.
- Sdvizhkov OA (2003) Matematika na komp'yutere: Maple 8. — M.: SOLON-Press, 2003. —176 p.
- Tarasevich JJ (2002) Jelementy diskretnoj matematiki dlja programmistov.- Jelektronnoe uchebnoe posobie.- Astrahan': Astrahanskij gosudarstvennyj pedagogicheskij universitet.
- Savotchenko SE, Kuz'micheva TG (2001) Metody reshenija matematicheskikh zadach v Maple: Uchebnoe posobie – Belgorod: Izd. Belaudit, 2001. – 116 p.
- Aronovich AB, Afanas'ev MJ, Suvorov BP (1997) Sbornik zadach po issledovaniju operacij. M: Izd-vo MGU. 256 p.
- Gmurman VE (1998) Rukovodstvo k resheniju zadach po teorii verojatnostej i matematicheskoj statistike: Ucheb. Posobie dlja studentov vtuzov. Izd. 4-e, M.: VSh. 400 p.
- D'jakonov V (2002) Maple 7: uchebnyj kurs. — SPb: Piter, 2002. 672 s, il.4.
- Efimov AV, Demidovich BP, et al. (1993) Sbornik zadach po matematike dlja vtuzov. Linejnaja algebra i osnovy matematicheskogo analiza. — M.: Nauka, 1993. 480 p.
- Efimov AV, Demidovich BP, et al. (1986) Sbornik zadach po matematike dlja vtuzov. Special'nye razdely matematicheskogo analiza. — M.: Nauka, 1986. 368 p.
- Kalihman IL (1975) Sbornik zadach po matematicheskomu programirovaniju. Izd. 2-e, dop. i pererab. M., «Vysshaja shkola».
- Kulanin ED, et al. (1999) 3000 konkursnyh zadach po matematike. 3-e izd., ispr.i dop. — M.: Rol'f, 1999. 624 p.

Impact Factor:

ISRA (India) = 1.344	SIS (USA) = 0.912	ICV (Poland) = 6.630
ISI (Dubai, UAE) = 0.829	PIHII (Russia) = 0.234	PIF (India) = 1.940
GIF (Australia) = 0.564	ESJI (KZ) = 3.860	IBI (India) = 4.260
JIF = 1.500	SJIF (Morocco) = 2.031	

- Ryzhikov JI (2000) Reshenie nauchno-tehnicheskikh zadach na personal'nom komp'yutere. — SPb.: KORONA print. 272 p.
- Rychkov V, D'jakonov V, Novikov J (2001) Komp'yuter dlja studenta. Samouchitel' — SPb.: Piter. 592 p.
- Samojlenko AM, Krivosheja SA, Perestjuk NA (1989) Differencial'nye uravnenija: primery i zadachi. Ucheb. posobie. 2-e izd., pererab. — M.: Vyssh. shk.. 383 p.
- Skanavi MI, Egerev VK, Kordemskij BA, Zajcev VV, et al. (1998) Sbornik zadach po matematike dlja postupajushhih vo vtuzy: Ucheb. posobie. Izd. 5-e, pererab. i dop. — M.: VSh. 431 p.
- Shelobaev SI (2000) Matematicheskie metody i modeli v jekonomike, finansah, biznese: Uchebnoe posobie dlja vuzov. — M.: JuNITI DANA. 376 p.

