

## Разработка пчелиного алгоритма для переезда магазина

*Скандаленко Александр Валерьевич*

*Приамурский государственный университет им. Шолом-Алейхема*

*студент*

*Научный руководитель:*

*Баженов Руслан Иванович*

*Приамурский государственный университет им. Шолом-Алейхема*

*к.п.н., доцент, зав. Кафедрой информационных систем, математики и правовой информатики*

### Аннотация

В данной статье была рассмотрена проблема создания программ для поиска оптимальной позиции на карте. В качестве исследования был рассмотрен принцип работы настоящего пчелиного улья, а также изучены существующие реализации пчелиного алгоритма. В результате была разработана программа по нахождению оптимальной позиции на карте с учётом расстояния, использующая пчелиный алгоритм.

**Ключевые слова:** пчелиный алгоритм, алгоритмы поиска, роевые алгоритмы

## Development of bee algorithm for moving shop

*Skandalenko Aleksandr Valerievich*

*Sholom-Aleichem Priamursky State University*

*student*

*Bazhenov Ruslan Ivanovich*

*Sholom-Aleichem Priamursky State University*

*Ph. D., associate Professor, head. Department of information systems, mathematics and legal informatics*

### Abstract

This article deals with the problem of creating programs to find the optimal position on the map. As a research the principle of work of the present bee hive was considered, and also existing realizations of bee algorithm are studied. As a result, a program was developed to find the optimal position on the map, taking into account the distance, using the bee algorithm.

**Keywords:** bee algorithm, search algorithms, swarm algorithms

Роевые алгоритмы – это алгоритмы, построенные на взаимосвязях между отдельными сущностями для достижения общей цели. К таковым

относят алгоритм серых волков, алгоритм роя частиц, алгоритм косяка рыб, муравьиный алгоритм и алгоритм пчелиного роя.

Для нахождения оптимальной позиции на карте, пчелиный алгоритм является наиболее подходящим вариантом среди всех роевых алгоритмов. Данный алгоритм был разработан благодаря наблюдениям за поведением пчёл, выполнявшим поиск цветов на территории, окружающей улей.

Целью данной работы является создание программы, осуществляющей поиск наилучшей позиции на карте с учётом расстояния на языке Object Pascal в среде Delphi.

Объектом исследования является алгоритм «пчелиного роя».

Предметом исследования является разработка программы, осуществляющей поиск наилучшей позиции на карте с учётом расстояния на языке Object Pascal в среде Delphi.

Задачи исследования:

- Изучить научно-техническую литературу по теме исследования.
- Разработать программу, находящей наилучшую позицию на карте с помощью алгоритма «пчелиного роя».

Методы исследования:

- Изучение научно-технической литературы по теме исследования.
- Компьютерное моделирование алгоритма «пчелиного роя».

Практическая значимость данной работы состоит в применении алгоритма пчелиного улья для решения задач, связанных с поиском оптимальной позиции на карте с учётом расстояния до начальной точки.

В данной работе будет рассмотрена реализация программы, находящей оптимальную позицию на математической карте с помощью алгоритма пчелиного роя.

Пчелиный алгоритм выглядит следующим образом:

1. Каждая пчела из роя вылетает на случайные позиции в поле. У каждой пчелы эта случайная позиция становится конечной.

2. Все эти пчелы исследуют всё на своём пути, но только в областях на некотором расстоянии от пути. Таким образом пчела находит свою наиболее оптимальную область для сбора нектара.

3. Все пчелы возвращаются в улей. В улье пчелы обмениваются информацией и, таким образом, находят самую оптимальную (наилучшую) область для сбора нектара из всех предложенных.

4. Каждая пчела (кроме той, чья область была определена как наилучшая) меняет свою конечную позицию на позицию, расположенную между первоначальной и наилучшей. Однако, у всех пчёл есть вероятность случайного изменения конечной позиции, независимо от «привлекательности» наилучшей позиции. Благодаря такому изменению, пчелы могут найти другие наилучшие позиции, некоторые из которых будут более «привлекательны», чем первоначальная наилучшая область.

Данный алгоритм повторяется столько раз, сколько потребуется улью.

Возможности пчелиного алгоритма были рассмотрены в статье В.С. Решетовой [1]. А.В. Леонов и Г.А. Литвинов провели эксперимент, используя пчелиный алгоритм для навигации беспилотных летательных аппаратов в сети FANET [2]. А.В. Никофорова, А.Н. Полетайкин и В.С. Канев в своей работе методом пчелиного роя решили задачу оптимизации при продвижении услуг связи на региональный рынок [3]. М.А.С. Муслим и Г. Радхамани использовали пчелиный алгоритм для оптимизации нагрузки задач на виртуальных машинах [4]. В.М. Курейчук и А.А. Кажаров нашли применение пчелиному алгоритму для раскраски графов [5]. Т.В. Бутько и Г.О. Прохорченко посвятили свою статью оптимизации графика движения поездов с помощью пчелиного алгоритма [6]. А.Ю.А. Иванов рассмотрел задачу оптимизации вычислительного процесса на многоядерной платформе с использованием алгоритма пчелиного улья [7].

Для данной работы была взята задача, связанная с «переездом предприятия»:

1. «Предприятие» должно переехать на новое место в связи с малой прибыльностью старой позиции.

2. «Предприятие» может практически без затрат переехать в примыкающий район, однако из-за переезда в дальние районы прибыль с новой позиции падает. Это связано с затратами на переезд, имеющие прямую зависимость от расстояния. Прибыль с учётом переезда высчитывается по формуле:

$Y = X * (100\% - I * (N - 1))$ , где  $Y$  – новая прибыль области,  $X$  – старая прибыль области,  $I$  – понижающий коэффициент, а  $N$  – расстояние между старой позицией «предприятия» и областью. Благодаря понижающему коэффициенту устанавливается максимальный радиус переезда «предприятия», ведь, даже в реальной жизни, предприятия, если они не могут себе это позволить, не переезжают и не открывают новые точки слишком далеко от старых.

3. У «предприятия» есть математическая карта «города», на которой указана прибыль для каждого его района.

Необходимо найти наиболее оптимальную позицию для переезда предприятия.

Для данной задачи можно применить и пчелиную аналогию: пчёлы, как живые существа, не обладая бесконечной выносливостью, вынуждены выбирать между большой поляной с цветами, расположенной далеко от улья, и небольшим кустиком, расположенным практически вплотную к улью. Именно данная ситуация послужила идейным примером будущей программы.

Таким образом, задача сводится к поиску оптимального решения с максимальным балансом между прибылью и ценой затрат на достижение данной области. Данную задачу можно решить и вручную, однако это потребует больших умственных затрат даже для небольшой области поиска. Поэтому была разработана программа для решения данных задач.

В качестве среды программирования был выбран Delphi, а в качестве языка программирования – Object Pascal.

Фрагменты кода:

Выдача массиву пчел первичных координат:

```
randomize;
for i:= 0 to bees-1 do begin
xbee[i]:=random(xmax);
ybee[i]:=random(ymax);
end;
boo1:= true;
while boo1 do begin
boo1:=false;
for i:= 0 to bees-1 do
if (xbee[i]=x) and (ybee[i]=y) then begin
boo1:=true;
xbee[i]:=random(xmax);
ybee[i]:=random(ymax);
end;
end;
```

На этом участке кода  $xbee[i]$  является  $x$ -координатой ячейки, в которую полетит пчела.  $ybee[i]$ , в свою очередь, является  $y$ -координатой целевой ячейки пчелы. Далее есть фрагмент, проверяющий пчёл «на ленивость»: если координаты целевой точки совпадают с координатами улья, то для этих пчёл нужно задать точки заново. Таким образом будет достигнута максимальная эффективность алгоритма.

Поиск клеток, через которые пролетит пчела:

```
for p := 0 to att do begin
for d := 0 to bees-1 do
for i := Min(x,xbee[d]) to Max(x,xbee[d])+1 do
for j := Min(y,ybee[d]) to Max(y,ybee[d])+1 do
if (i<>x) or (j<>y) then
begin
if (x=xbee[d]) then
begin
beem[d]:=max(beem[d],city[x,ybee[d]]*(1-cd*(abs(y-ybee[d])-1)));
xbeemax[d]:=xbee[d];
ybeemax[d]:=ybee[d];
end else if (y=ybee[d]) then
begin
beem[d]:=max(beem[d],city[xbee[d],y]*(1-cd*(abs(x-xbee[d])-1)));
xbeemax[d]:=xbee[d];
ybeemax[d]:=ybee[d];
end else
begin
k:=(ybee[d]-y)/(xbee[d]-x);
```

```

    b:=(y*xbee[d]-ybee[d]*x)/(xbee[d]-x);
    if (not (k*(i-0.5)+b<j-0.5) and not(k*(i+0.5)+b<j-0.5)) or (not (k*(i-
0.5)+b>j+0.5) and not(k*(i+0.5)+b>j+0.5)) then
    begin
        beem[d]:=max(beem[d],(((city[xbee[d],ybee[d]])*(1-cd*(sqrt((x-
xbee[d])*(x-xbee[d])+(y-ybee[d])*(y-ybee[d]))-1)))));
        xbeemax[d]:=xbee[d];
        ybeemax[d]:=ybee[d];
    end;
end;
end;
end;

```

В данном фрагменте идёт поиск всех областей, через которые пролетит пчела. Среди всех этих областей будет выбрана одна с максимальной выгодой. Для оптимизации, поиск ведётся не на всей карте, а только на прямоугольном участке, где находится траектория полёта пчелы. Данная оптимизация позволит уменьшить число ненужных проверок.

Поиск наилучшей позиции среди предложенных:

```

for i := 0 to bees-1 do
if massloc<beem[i] then
begin
    massloc:=beem[i];
    xloc:=xbeemax[i];
    yloc:=ybeemax[i];
end;
end;

```

На данном участке происходит отбор наилучшей позиции среди всех оптимальных позиций, найденных пчёлами. Эта позиция и будет наилучшей для первого «вылета», т.е. первого цикла поиска наилучшей области.

Изменение координат:

```

randomize;
for i := 0 to bees-1 do
if (random(5)>0) and (massloc>0) then
begin
    xbee[i]:=Round(xbee[i]+((massloc-beem[i])/(2*massloc))*(xloc-
xbee[i]));
    ybee[i]:=Round(ybee[i]+((massloc-beem[i])/(2*massloc))*(yloc-
ybee[i]));
end else
begin
    xbee[i]:=random(xmax);
    ybee[i]:=random(ymax);
end;
bool:= true;
while bool do begin
bool:=false;

```

```

for i:= 0 to bees-1 do
  if (xbee[i]=x) and (ybee[i]=y) then begin
    bool:=true;
    xbee[i]:=random(xmax);
    ybee[i]:=random(ymax);
  end;
end;

```

На этом фрагменте происходит следующее:

У каждой пчелы есть вероятность изменить свою целевую точку с учётом наилучшей области. В остальных случаях, целевая точка меняется на любую случайную. Это случайное изменение позволяет находить другие оптимальные области, которые могут быть лучше первоначально найденной. Изменение с учётом наилучшей области происходит следующим образом:

Сравниваются «силы» наилучшей области и оптимальной области пчелы.

Пчела выбирает некую точку, расположенную между этими областями. Расположение этой точки зависит от разницы между «силами» - чем больше разница, тем ближе к середине отрезка между этими областями будет расположена точка, чем меньше разница, тем меньше изменятся новые координаты целевой области пчелы. Также, в данном фрагменте вероятность случайного изменения примерно равна 25%. На самом деле она немного больше - это обусловлено повторной проверкой пчелы «на ленивость».

Пример интерфейса программы изображён на рисунке №1.

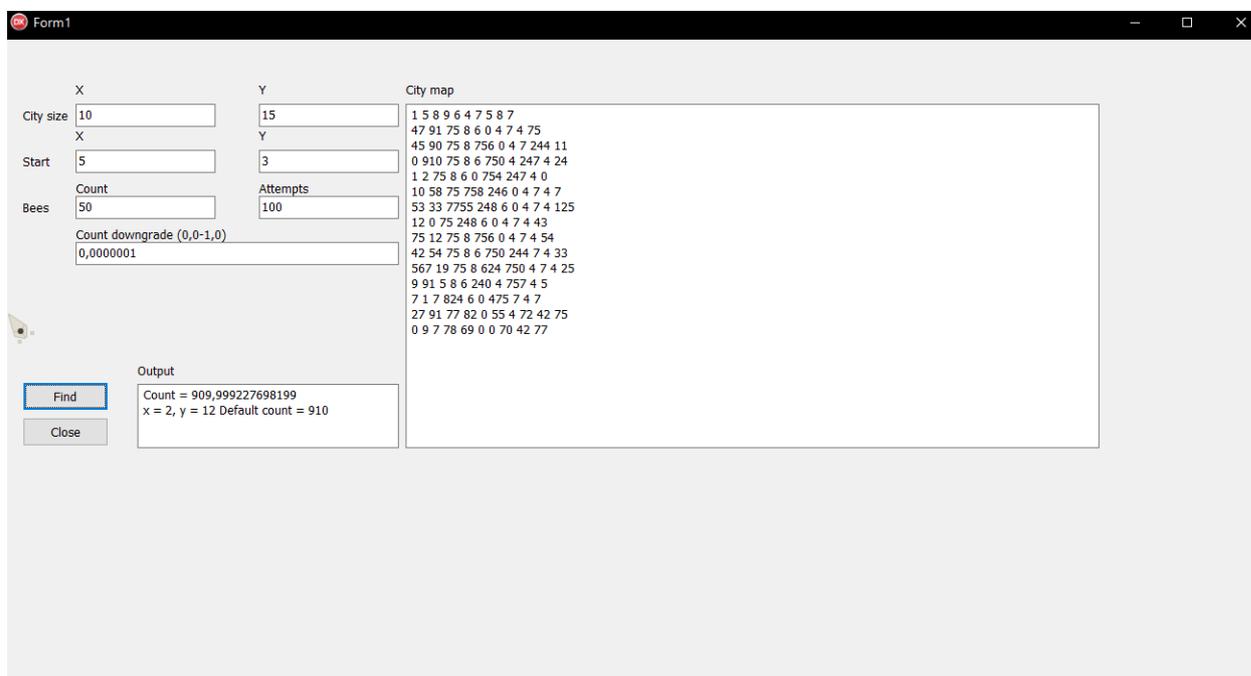


Рисунок 1. Интерфейс программы

На рисунке №1 можно подробно рассмотреть все его элементы:

1. Размерность города (City size). В целочисленных координатах X и Y указываются количество ячеек по ширине и длине карты соответственно.
  2. Начальная позиция (Start). Указывается в целых числах и показывает, из какой точки начинается отсчёт.
  3. Количество пчёл и попыток-«вылетов» (Bees Count & Bees Attempts). Эти целочисленные значения показывают, сколько пчёл будет вылетать за один вылет и сколько таких вылетов стоит делать. На небольших картах достаточно небольших значений, в то время как большие карты потребуют большего количества пчёл и вылетов для обеспечения должной точности. Каждая дополнительная пчела и каждая дополнительная попытка уменьшают неточность параболически: 2-я пчела уменьшает неточность в 2 раза, 3-я в 1.5, 4-я – 1.25 и так далее.
  4. Коэффициент уменьшения прибыли (Count downgrade). Записывается в долях от 0 до 1. Данная величина характеризует, насколько упадёт прибыль в области при переезде. При этом, диагонально соединённые имеют между собой расстояние не в 1, а в  $2^{0,5}$ , что примерно равно 1,4, а значит, между ним уже произойдёт уменьшение прибыли. При коэффициенте равном нулю, программа не будет учитывать расстояние и выдаст участок с наибольшей прибылью.
  5. Карта города (City map). В данной карте каждой ячейке соответствует своё целое число. Данное число может быть как положительным, так и быть равным 0. Это число характеризует прибыль на каждой клетке. Естественно, если прибыль в области меньше нуля, то в эту область предприятие не переедет ни под каким предлогом, а значит на карте данным ячейкам будет соответствовать значение 0. Координаты ячейки такие же, как и в стандартной математической системе координат, где начало отсчёта находится в левом нижнем углу.
  6. Вывод решения (Output). Решение выводится в определённом формате:  
Итоговая прибыль с области (Count)  
Координаты области  
Первоначальная прибыль с области (Default count).
  7. Кнопки поиска (Find) и закрытия программы (Close).
- Примеры использования программы показаны на рисунках №2, №3 и №4.

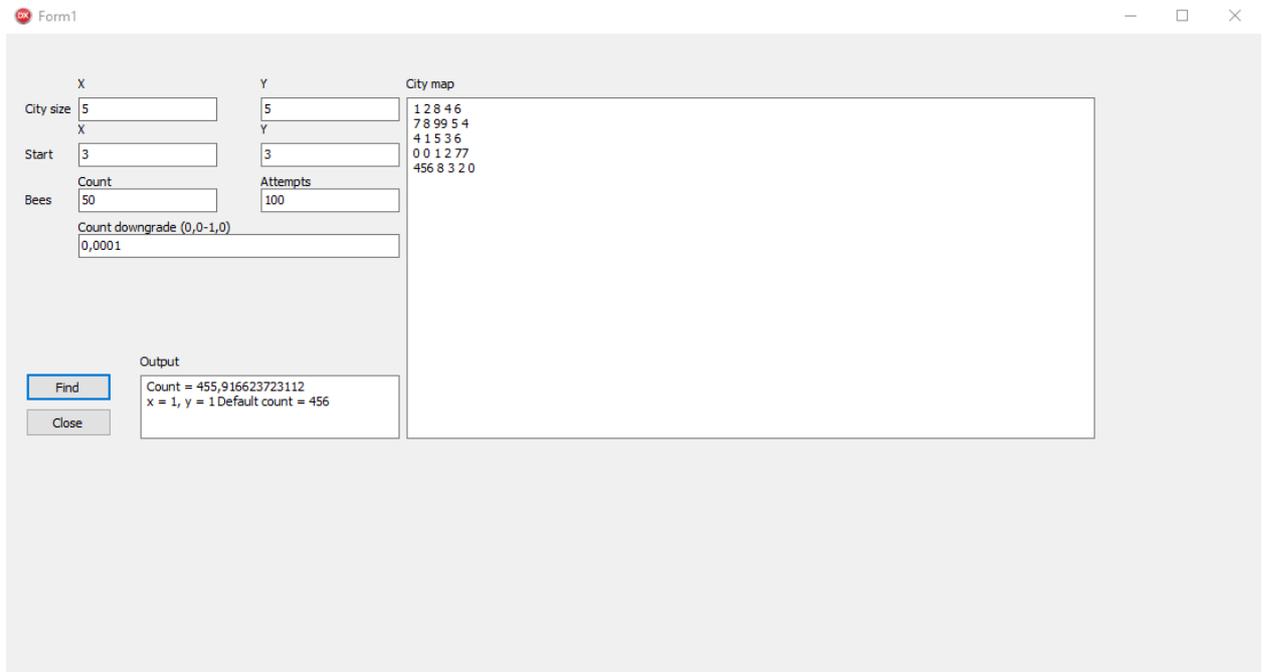


Рисунок 2. Пример использования программы №1

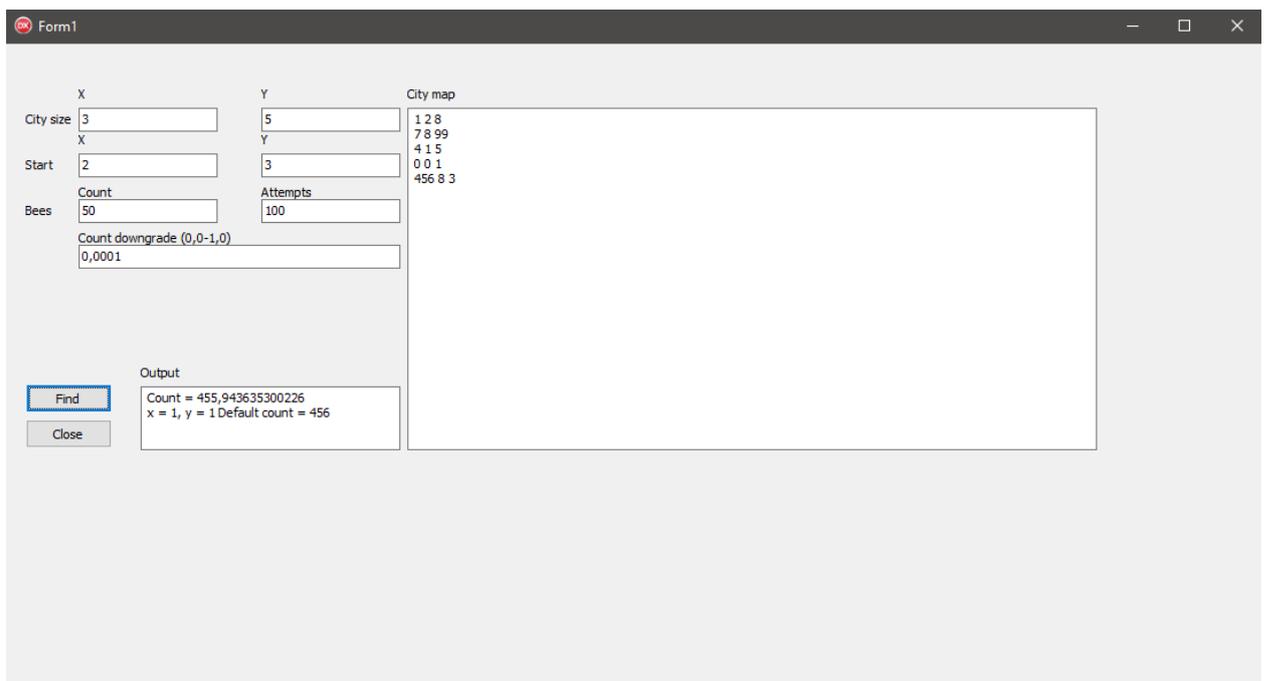


Рисунок 3. Пример использования программы №2

Рисунок 4. Пример использования программы №3

На рисунке 2 данные трактуются следующим образом:

Из начальной области с координатами  $x=3$ ,  $y=3$  на данной карте наиболее оптимальной областью для переезда предприятия будет область с координатами  $x=1$ ,  $y=1$ . Начальная прибыль в этой области составит 456 у.е., однако, с учётом падения прибыли из-за расстояния между начальной областью и оптимальной и коэффициентом падения прибыли, равным 0,0001, итоговая прибыль составит  $\sim 455,917$  у.е.

На рисунке 3 данные трактуются следующим образом:

Из начальной области с координатами  $x=2$ ,  $y=3$  на данной карте наиболее оптимальной областью для переезда предприятия будет область с координатами  $x=1$ ,  $y=1$ . Начальная прибыль в этой области составит 456 у.е., однако, с учётом падения прибыли из-за расстояния между начальной областью и оптимальной и коэффициентом падения прибыли, равным 0,0001, итоговая прибыль составит  $\sim 455,944$  у.е.

На рисунке 4 данные трактуются следующим образом:

Из начальной области с координатами  $x=1$ ,  $y=1$  на данной карте наиболее оптимальной областью для переезда предприятия будет область с координатами  $x=3$ ,  $y=3$ . Начальная прибыль в этой области составит 99 у.е., однако, с учётом падения прибыли из-за расстояния между начальной областью и оптимальной и коэффициентом падения прибыли, равным 0,0001, итоговая прибыль составит  $\sim 98,982$  у.е.

Роевые алгоритмы в основном предназначены для поиска путей в графе, нахождению оптимальной области или оптимизации процессов. Но даже так они являются довольно перспективной веткой развития алгоритмизации решений задач оптимизации.

В данной курсовой работе была рассмотрено решение задачи поиска наилучшей области с учётом расстояния с помощью алгоритма пчелиного улья, используя Delphi как среду программирования на языке Object Pascal.

Во время работы над курсовым проектом была изучена научно-техническая литература и разработан алгоритм, позволяющий находить максимальное значение в заданной области.

Разработанный алгоритм, основанный на поведении пчелиного роя, будет иметь большой потенциал для улучшения и развития, основным недостатком которого является узкая направленность.

### Библиографический список

1. Решетова В.С. Изследване на възможностите на алгоритма “пчелен рой” // Парадигма. 2016. Т. 1. № 1. С. 120-126.
2. Леонов А.В., Литвинов Г.А. Применение алгоритма пчелиной колонии ВЕЕАДНОС для маршрутизации в FANET // Вестник СибГУТИ. 2018. № 1 (41). С. 85-95.
3. Никифорова А.В., Канев В.С., Полетайкин А.Н. Методика оптимального планирования продвижения услуг связи на региональный рынок при помощи пчелиного алгоритма // В сборнике: Проблемы оптимизации сложных систем Труды 12-й Международной Азиатской школы-семинара. Под редакцией С.И. Кабанихина, А.В. Кельманова, А.С. Родионова. 2016. С. 442-446.
4. Mosleh M.A.S., Radhamani G. A novel fuzzy QoS based improved honey bee behavior algorithm for efficient Load balancing in cloud // Труды СПИИРАН. 2018. № 2 (57). С. 26-44.
5. Курейчик В.М., Кажаров А.А. Применение пчелиного алгоритма для раскраски графов // Известия ЮФУ. Технические науки. 2010. № 12 (113). С. 30-36.
6. Бутько Т.В., Прохорченко Г.О. Формування процедури автоматизації розробки графіку руху поїздів на основі алгоритму штучних бджолиних колоній // Транспортні системи та технології перевезень. 2015. № 9. С. 10-15.
7. Иванов Ю.А. Оптимизация вычислительного процесса на многоядерной системе с использованием алгоритма искусственной пчелиной колонии // Наукові праці Донецького національного технічного університету. Серія: Інформатика, кібернетика та обчислювальна техніка. 2013. № 2 (18). С. 18-25.
8. Кузнецова И.А. Практикум по Delphi для решения прикладных задач. // Нижегородский государственный университет им. Н.И. Лобачевского. URL: [http://e-learning.fpm.chnu.edu.ua/pluginfile.php/13727/mod\\_resource/content/1/praktikum\\_Delphi\\_laboratorni.pdf](http://e-learning.fpm.chnu.edu.ua/pluginfile.php/13727/mod_resource/content/1/praktikum_Delphi_laboratorni.pdf)