

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**  
(ФГБОУ ВО «КубГУ»)

**Кафедра математических и компьютерных методов**

**КУРСОВАЯ РАБОТА**

**МЕТОДЫ И ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА  
МОДЕЛИРОВАНИЯ БИЗНЕС–ПРОЦЕССОВ ПРОИЗВОДСТВЕННОГО  
ПРЕДПРИЯТИЯ**

Работу выполнил \_\_\_\_\_ \_А.М.Воронков  
(подпись, дата)

Факультет экономический курс 1

Направление подготовки 27.03.03. системный анализ и управление

Научный руководитель  
доцент кафедры МКМ,  
канд.эконом.наук,  
доцент \_\_\_\_\_ Г.Н. Библия  
(подпись, дата)

Нормоконтролер  
ст. лаборант \_\_\_\_\_ Ю.Д. Кравченко  
(подпись, дата)

Краснодар 2018

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**  
**(ФГБОУ ВО «КубГУ»)**  
**Кафедра математических и компьютерных методов**

**ЗАДАНИЕ**  
на курсовую работу

Студенту \_\_\_\_\_ группы 113  
направления подготовки 27.03.03 Системный анализ

**Тема курсовой работы: «Алгоритмы поведения децентрализованных самоорганизующихся систем»**

**Цель:** Изучить предметную область. Рассмотреть возможности применения методики интеллектуального анализа и оптимизации данных для предмета исследования.

**Основные вопросы, подлежащие разработке (исследованию):**

- 1) Теоретический обзор современных алгоритмов исследования проблемы;
- 2) Анализ предметной области, обоснование *спецификации* исследуемой системы;
- 3) Проектирование и реализация разработки.

**Основная литература:**

- 1) Фрейдина, Е.В. Исследование систем управления организации. Учебное пособие [Электронный ресурс]: учебное пособие / Е.В Фрейдина — Электрон. текстовые дан. — М.: Омега-Л, 2013. 368 с.
- 2) Бочкарев, А. А. Логистика городских транспортных систем: учебное пособие для бакалавриата и магистратуры / А. А. Бочкарев, П. А. Бочкарев. — 2-е изд., перераб. и доп. — М. : Издательство Юрайт, 2018. — 150 с.
- 3) Васильева, Н. В. Основы землепользования и землеустройства: учебник и практикум для СПО / Н. В. Васильева. — М.: Издательство Юрайт, 2018. — 376 с.
- 4) Дубина, И. Н. Основы математического моделирования социально-экономических процессов: учебник и практикум для бакалавриата и магистратуры / И. Н. Дубина. — М.: Издательство Юрайт, 2018. — 349 с

Срок представления законченной работы 29 мая 2017 г.

Дата выдачи задания 05 февраля 2018 г.

Руководитель \_\_\_\_\_ /Г.Н. Библя /

Задание получил 05 февраля 2018 г.

Студент \_\_\_\_\_ / \_\_\_\_\_ /

## РЕФЕРАТ

Курсовая работа 45 с., 9 рис, 2 табл., 10 источников.

### АЛГОРИТМЫ ПОВЕДЕНИЯ ДЕЦЕНТРАЛИЗОВАННЫХ САМООРГАНИЗУЮЩИХСЯ СИСТЕМ, АЛГОРИТМЫ ОПТИМИЗАЦИИ ДАННЫХ, МУРАВЬИНЫЙ АЛГОРИТМ, ЗАДАЧА КОММИВОЯЖЕРА

Объект исследования – ОАО «ТАНДЕР»

Предмет исследования – алгоритмы поведения децентрализованных самоорганизующихся систем.

Целью курсовой работы является выявление алгоритмов и их применение для оптимизации процессов и совершенствование структуры управления ОАО «ТАНДЕР», а также создание моделей бизнес-процессов организации на основе проведения системного анализа деятельности компании и подробного изучения алгоритмов оптимизации децентрализованных самоорганизующихся систем. Рассмотрение возможности применения алгоритмов оптимизации децентрализованных самоорганизующихся систем к предмету исследования.

Метод исследования – статистические и аналитические методы, методы системного анализа, методы функционально-структурного анализа предметной области

Для выполнения поставленных задач была использована документация, предоставленная ОАО «ТАНДЕР».

Актуальность и практическая значимость предлагаемой работы заключаются в решении проблем, связанных с выявлением наиболее оптимальных алгоритмов оптимизации децентрализованных самоорганизующихся систем.

## СОДЕРЖАНИЕ

Введение .....	3
1. Виды алгоритмов поведения децентрализованных самоорганизующихся систем .....	4
1.1. Самоорганизующиеся системы .....	4
1.2. Естественные алгоритмы .....	4
1.2.1. Метод роя частиц .....	8
1.2.1.1. Подбор параметров .....	10
1.2.2. Генетический алгоритм .....	11
1.2.2.1. Описание алгоритма .....	11
1.2.3. Алгоритм имитации отжига .....	13
1.2.3.1. Выбор начального решения .....	14
1.2.3.2. Оценка решения .....	15
1.2.3.3. Основной шаг алгоритма .....	15
1.2.3.4. Случайное изменение решения .....	15
1.2.3.5. Критерий допуска .....	15
1.2.3.6. Уменьшение температуры .....	16
1.2.3.7. Выбор начальной и пороговой температуры .....	16
1.2.4. Оптимизация передвижением бактерий .....	17
1.2.5. Алгоритм летучих мышей .....	23
1.2.5.1. Общая характеристика алгоритма .....	23
1.2.5.2. Описание алгоритма .....	24
1.2.5.3. Преимущества и недостатки алгоритма летучей мыши .....	24
2. Алгоритм муравьиной колонии для задачи размещения распределительных центров розничной торговой сети (для задачи о р-медиане) .....	27

3. Решение задачи коммивояжера .....	31
Заключение.....	41
Список использованных источников.....	42

## ВВЕДЕНИЕ

В теории системного анализа одну из важнейших ролей занимает поведение самоорганизующихся децентрализованных систем. На данный момент это одно из актуальнейших направлений изучения в науке. Ведь далеко не все системы (как и экономические так и любые другие) централизованы и регулируются со стороны человека. Поэтому изучение этих систем занимает одну из лидирующих позиций. Целью курсовой работы является выявление алгоритмов поведения этих самых систем.

Задачи в данной работе таковы:

- 1) Анализировать объект исследования
- 2) Провести эксперимент
- 3) Выявить закономерности
- 4) Сформулировать алгоритмы
- 5) Сделать вывод.

Объектом исследования в курсовой работе являются децентрализованные самоорганизующиеся системы. А предметом исследования являются закономерности и алгоритмы поведения децентрализованных самоорганизующихся систем.

Для изучения этой проблемы используют статистические и аналитические методы. Обоснованием необходимости исследования будет необходимость в нем. Нам необходимо понимать поведение подобных систем для того чтобы предугадать что будет дальше и суметь удачно подстроиться под их поведение.

Было проведено множество исследований на эту тему. В том числе и в России. К примеру исследование, проведенное в Московском Государственном университете информационных технологий, радиотехники и электроники. В.В. Баранюк и О.С. Смирнова изучали роевой интеллект как одну из частей онтологической модели бионических технологий. В ходе исследования выведены алгоритмы поведения децентрализованных самоорганизующихся

систем, основанные на поведении насекомых (муравьев, пчел и т.д.) и побужденные неживой природой и бактериями.

## 2 Виды алгоритмов поведения децентрализованных самоорганизующихся систем

Децентрализованная система - это система, в которой нет главного элемента. Важнейшие подсистемы в такой системе имеют приблизительно одинаковую ценность и построены не вокруг центральной подсистемы, а соединены между собой последовательно или параллельно.

### 2.1 Самоорганизующиеся системы.

Это системы, которые черпают информацию из окружающей среды и подстраиваются под нее посредством использования активных элементов (агентов), делающих систему целенаправленной. Особенностью таких систем является непостоянность, уникальность и непредсказуемость систем в конкретных условиях и централизация (децентрализация) системы. Недостатками таких систем являются низкая скорость приспособления(адаптации) и повышенная сложность управления.

### 2.2 Естественные алгоритмы (Роевой интеллект).

Одним из наименее глубоко (подробно) представленных методов оптимизации децентрализованных самоорганизующихся систем является использование так называемого интеллекта роя. Роевой интеллект (англ. Swarm intelligence) — это способ оптимизации, применяемый в теории ИИ и изображающий коллективное поведение этих самых систем. Существует значительное количество способов оптимизации, базирующихся на роевом интеллекте — метод роя частиц, пчелиный алгоритм, алгоритм имитации отжига, оптимизация передвижением бактерий, светлячковый алгоритм, алгоритм капель воды и т.д. Стоит заметить, что роевой интеллект относится к эволюционным вычислительным методам. Суть подхода состоит в том, что

необходимо провести анализ и использовать модели поведения муравьёв, разыскивающих пути от колонии к источнику питания и представляет собой метаэвристическую оптимизацию.

Роевой интеллект подражает процессу самоорганизации роевых насекомых. Рой обычно трансформируется в пространственно–временные структуры при помощи самоорганизации, происходящей за счет текущей динамики роевых структур, а не за счет какого–то управления из центра. Муравьи (агенты) взаимодействуют, друг с другом на протяжении долгого промежутка времени, дополняя окружающую среду за счет стигмергии (stigmergy) — механизма непрямого и временного взаимодействия агентов. Принцип стигмергии заключается в том, что след, оставленный в окружающем мире посредством действия, побуждает возникновение следующего действия того же или другого агента. И тогда последующие действия принимаются усиливать друг друга и сооружаться за счет друг друга, приводя к самопроизвольному появлению очевидной систематической активности. Например, записка, оставленная на холодильнике — это стигмергия, не прямое взаимодействие между людьми, повлияющее на их поступки спустя несколько часов. В цифровой интерпретации виртуальный рой визуализируется на теоретическом уровне — он представляет собой набор локальных правил и взаимодействий между цифровыми единицами.

В биологическом смысле, муравьи перемещаются в случайном порядке, не замечая и не влияя друг на друга, и не считая себя частью чего–то организованного, пока не находят источник еды. После этого они возвращаются в колонию, оставляя за собой феромонные следы, которые замечают остальные муравьи колонии и, с большой вероятностью, следуют именно им. Вместо отслеживания цепи, остальные муравьи укрепляют феромонный след в случае нахождения продовольствия. Спустя некоторое время феромонный след начинает испаряться, тем самым понижая свою привлекательность. Чем больше время, необходимое для прохождения до цели и обратно, тем сильнее испарение феромонного следа. На коротком пути,

например, прохождение будет более быстрым и, исходя из этого, концентрация феромонов останется высокой. Испарение феромонов также имеет свойство избегания стремления к единственно–оптимальному решению. Если бы феромоны не испарялись, то путь, выбранный первым, был бы самым привлекательным. В этом случае исследования пространственных возможных случаев были бы ограниченными.

На рисунке ниже (Рисунок 1.1) изображено графическое представление роевых правил. Частица  $I$  в пространстве  $S$  в настоящий момент находится в координате  $X$  и передвигается со скоростью  $v$ . Она притягивается к частице притяж.  $j$  и отдалается от точки  $k$ . Остальные частицы остаются за гранью доступности частицы  $i$ .

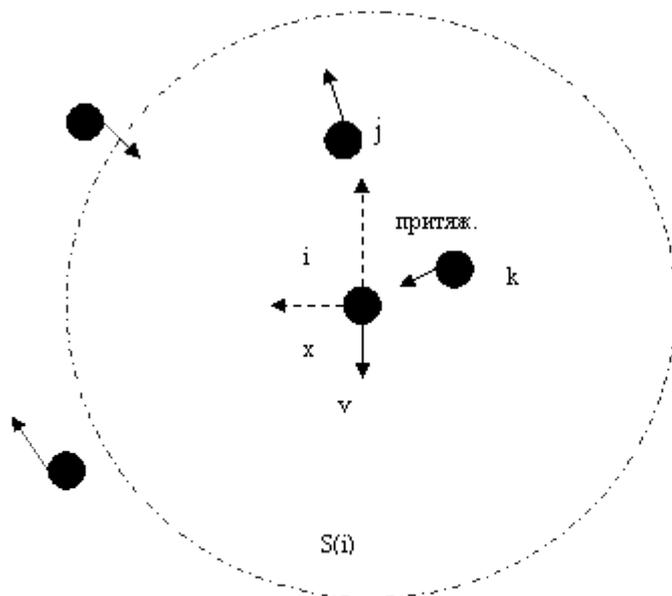


Рисунок 1.1- графическое представление роевых правил

### Роевые правила

Иными словами, каждая частица имеет две зоны восприятия — наружную и внутреннюю. Частица не видит никаких других частиц, которые находятся за гранью внешней зоны. Частицы отталкиваются от центра масс частиц

внутренней зоны и притягиваются навстречу к центру масс частиц во внешней зоне.

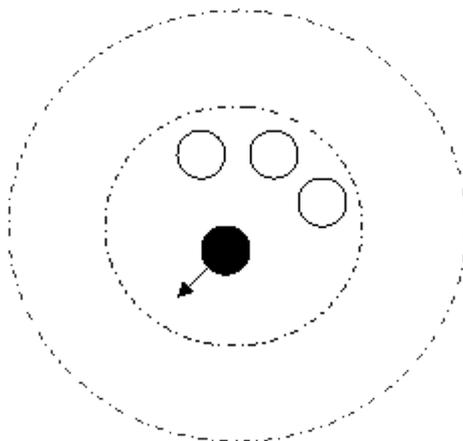


Рисунок 1.2- Частица отталкивается от центра масс внутренней зоны.

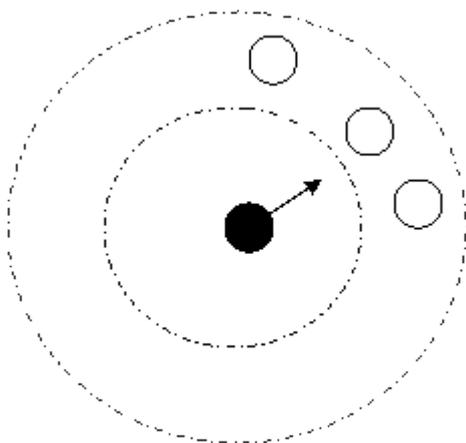


Рисунок 1.3- Частица притягивается к центру масс внешней зоны.

отталкивание частицы от трех соседних во внутренней зоне (Рисунок 1.2) и притяжение частицы к трем соседним в наружной зоне (Рисунок 1.3).

Подробно алгоритм выглядит следующим образом:

- Если внутренняя зона частицы в  $x$  занята, найти соседний центр масс  $y_{CM}$ , где  $y$  – это положение каждого соседа. Ускорение частицы при этом  $a = x - y_{CM}$ . Центр масс набора точек  $y_1, y_2 \dots y_n$  —  $y_{CM} = \sum_{i=1}^n y_i$ .
- В противном случае, если внешняя зона частицы в  $x$  занята, найти соседний центр масс  $y_{CM}$ , где  $y$  – это положение каждого соседа. Ускорение частицы при этом  $a = y_{CM} - x$ .
- В противном случае,  $a=0$

Полное обновление положения частиц происходит по следующим формулам:

$$a = \begin{cases} x - y_{CM} \\ y_{CM} - x \\ 0 \end{cases}$$

$$v(t+1) = v(t) + a(t)$$

$$v(t+1) \leftarrow v_0 \frac{v_{t+1}}{|v_{t+1}|}$$

$$x(t+1) = x(t) + v(t)$$

### 2.2.1 Метод роя частиц

Метод роя частиц (МРЧ) - способ численной оптимизации, для использования которого нет необходимости знать точного градиента оптимизируемой функции.

МРЧ был доказан Кеннеди, Эберхартом и Ши Иблы необходим для имитации социального поведения. Алгоритм был сделан более легким для понимания, и было замечено, что он может быть использован для выполнения оптимизации. Книга Кеннеди и Эберхарта описывает много философских аспектов МРЧ и так называемого роевого интеллекта. Обширное исследование приложений МРЧ сделал Поли. МРЧ оптимизирует функцию, сохраняя популяцию возможных решений, называемых частицами, и перемещая эти частицы в пространстве ответов (решений) согласно простой формуле. Перемещения подчиняются принципу оптимальнейшего найденного в этом пространстве

положения, которое постоянно трансформируется при нахождении частиц более выгодных положений.

### Алгоритм

Пусть  $f: \mathbb{R}^n \rightarrow \mathbb{R}$  — целевая функция, которую требуется минимизировать,  $S$  — количество частиц в рое, для каждой из которых сопоставлена координата  $x_i \in \mathbb{R}^n$  в пространстве ответов (решений) и скорость  $v_i \in \mathbb{R}^n$ . Пусть также  $p_i$  — лучшее из известных положений частицы  $i$ , а  $g$  — наилучшее известное состояние роя в целом. Тогда общий вид метода роя частиц таков.

1. Для каждой частицы  $i = 1, \dots, S$  сделать:
2. Создать наачал  
льное  
ное положение частицы с помощью случайного вектора  $x_i \sim U(blo, bu$   
 $p)$ , имеющего распределение  
многомерное равное распределение.  $blo$  и  $bup$  —  
нижняя и верхняя границы пространства решений соответственно.
3. Присвоить наилучшему известному положению частицы его начальное значение:  $p_i \leftarrow x_i$ .
4. Если  $(f(p_i) < f(g))$ , то обновить наилучшее известное состояние роя:  $g \leftarrow p_i$ .
5. Присвоить значение скорости частицы:  $v_i \sim U(-(bup-blo), (bup-blo))$ .
6. По **Ошибка!** значения целевой функции), повторять:
7. Для каждой частицы  $i = 1, \dots, S$  сделать:
8. Сгенерировать случайные векторы  $g_r, r_g \sim U(0,1)$ .
9. Обновить скорость частицы:  $v_i \leftarrow \omega v_i + \varphi_r g_r \times (p_i - x_i) + \varphi_g r_g \times (g - x_i)$ , где операция  $\times$  означает покомпонентное умножение.
10. Обновить положение частицы переносом  $x_i$  на вектор скорости:  $x_i \leftarrow x_i + v_i$ . Заметим, что этот шаг выполняется  
не зависимо от улучшения значения целевой функции.
11. Если  $(f(x_i) < f(p_i))$ , то делать:
12. Обновить лучшее известное положение частицы:  $p_i \leftarrow x_i$ .

13. Если  $f(p_i)$

$< f(g)$ , то обновить лучшее известное состояние роя в целом:  $g \leftarrow p_i$ .

14. Теперь  $g$  содержит лучшее из найденных решений.

15. Параметры  $\omega$ ,  $\varphi_r$ , и  $\varphi_g$  выбираются вычислителем и определяют поведение и эффективность метода в целом. Эти параметры составляют предмет многих исследований.

### 2.2.1.1 Подбор параметров

Выбор оптимальных параметров метода роя частиц — тема большого количества исследовательских

ра

бот, см., например, работы Ши и Эберхарта, Карлисла и Дозера, ван ден Берга, Клерка и Кеннеди, Трелеа, Браттона и Блеквэлла, а также Эверса. Простой и эффектив

**Ошибка!**

ами. Они же провели эксперименты с различными оптимизационными проблемами и параметрами. Техника выбора этих параметров названа мета-оптимизацией, так как другой оптимизационный алгоритм использует

**Ошибка!**

оказались противоречащими основным принципам, описанным в литературе, и часто дают удовлетворительные результаты оптимизации для простых случаев МРЧ. Реализацию их можно найти в открытой библиотеке SwarmOps.

Варианты алгоритма

Постоян

**Ошибка!**

ости метода. Существует несколько направлений в этих исследованиях, одна из которых предлагает создать гибридный оптимизацио

нный метод, использующий МРЧ в комбинации с иными алгоритмами. Другая тенденция предлагает каким-либо образом увеличить скорость работы метода, например, отходом назад или переменной порядка движения частиц. Также есть попытки адаптировать поведенческие параметры МРЧ в процессе оптимизации.

## 2.2.2 Генетический алгоритм

Генетический алгоритм — это эвристический поиск, используемый для решения задач оптимизации и моделирования путём случайного подбора, комбинирования и вариации искомым параметров с использованием механизмов, напоминающих эволюцию в биологии. Является одной из разновидностей эволюционных вычислений, с помощью которых решаются оптимизационные задачи с использованием

**Ошибка!**

овер. Главной особенностью генетического алгоритма является ориентированность на использование оператора

«скрещива

**Ошибка!** роли скрещивания в живой природе.

### 2.2.2.1 Описание алгоритма

Задача формализуется таким образом, чтобы её решение могло быть закодировано в виде вектора («генотипа») генов, где каждый ген может быть битом, числом или неким другим объектом. В классических реализациях ГА предполагается, что генотип имеет фиксированную длину. Однако существуют вариации ГА, свободные от этого ограничения.

Некоторым, обычно случайным, образом создаётся множество генотипов первичной популяции. Они оцениваются с использованием «функции приспособленности», в результате чего с каждым генотипом ассоциируется определённое значение («приспособленность»), которое определяет насколько хорошо фенотип, описываемый им, решает поставленную задачу.

При выборе «функции приспособленности»

(

**Ошибка!».**

Из полученного множества решений («поколения») с учётом значения «приспособленности» отбираются решения (обычно лучшие особи имеют большую вероятность быть выбранными), к которым применяются «генетические операторы» (в большинстве случаев «скрещивание» — crossover и «мутация» — mutation), результатом чего является получение новых решений. Для них также вычисляется значение приспособленности

**Ошибка!**

Этот набор действий повторяется итеративно, так моделируется «эволюционный процесс», продолжающийся некоторое количество жизней

**Ошибка!** критерием может быть:

1. нахождение глобального, либо субоптимального решения;
2. исчерпание числа поколений, отпущенных на эволюцию;
3. исчерпание времени, отпущенного на эволюцию.

Генетиче

**Ошибка!**

Таким образом, можно выделить следующие этапы генетического алгоритма:

1. Задать целевую функцию (приспособленности) для особей популяц

ии

2. Создать начальную популяцию

(Начало цикла)

1. Размножение (скрещивание)

2. Мутирование

3. Вычислить значение целевой функции для всех особей

4. Формирование нового поколения (селекция)

5. Если выполняются условия остановки, то (конец цикла), иначе (начало цикла).

### 2.2.3 Алгоритм имитации отжига

Алгоритм имитации отжига - алгоритм решения различных оптимизационных задач. Он базируется на моделировании реального физического процесса, который происходит при кристаллизации вещества из жидкого состояния в твёрдое, в том числе при отжиге металлов.

Целью алгоритма является максимальная минимизация некоторого функционала. В процессе работы алгоритма изображенного на рисунке 1.4 хранится текущее решение, которое является промежуточным результатом. А после работы алгоритма оно и будет ответом.

1. Основные шаги алгоритма. Выбор начального решения и начальной температуры

2. Оценка начального решения

3. Основной шаг алгоритма

3.1. Случайное изменение текущего решения

3.2. Оценка измененного решения

3.3. Критерий допуска

4. Уменьшение температуры и, если температура больше некоторого порога, то переход к основному шагу



Рисунок 1.4-Основные шаги алгоритма.

### 2.2.3.1 Выбор начального решения

Хорошей стратегией может быть случайный выбор начального решения. Также в качестве первичного решения можно предложить решение полученное другими методами. Это предоставляет алгоритму базу, на основании которой он будет строить более оптимальное решение.

#### 2.2.3.2 Оценка решения

Этот этап полностью зависит от особенностей задачи. Единственным требованием является выведение в качестве оценки одного вещественного числа, которое будет описывать оптимальность предлагаемого решения. Это число в алгоритме имитации отжига принято называть энергией. Если выбор такого числа является сложным, то, возможно, стоит отказаться от использования предлагаемого метода.

#### 2.2.3.3 Основной шаг алгоритма

Основной шаг при некоторой температуре повторяется некоторое количество раз. Возможно, что один раз. Также возможен случай с зависимостью числа повторов от температуры.

#### 2.2.3.4 Случайное изменение решения

Этот этап сильно зависит от особенностей задачи. Однако, изменение стоит производить локальные. Например, для задачи коммивояжера, хорошей стратегией будет обмен, в текущем порядке следования городов, двух случайных городов местами. В результате изменения у нас будет два решения: текущее и измененное.

#### 2.2.3.5 Критерий допуска

Для определенности будем считать, что оптимизация заключается в минимизации энергии. В большинстве случаев этот подход справедлив. Критерий допуска заключается в проверке и возможной замене текущего решения измененным.

Если измененное решение имеет меньшую энергию, то оно принимается за текущее. Если же измененное решение имеет большую энергию, то оно принимается с вероятностью  $P = \exp(-\delta E/T)$ , где:

$P$  — вероятность принять измененное решение,

$\delta E$  — модуль разности между энергией оптимального решения и энергией измененного решения,

$T$  — текущая температура.

### 2.2.3.6 Уменьшение температуры

Важной частью алгоритма является уменьшение температуры. При большой температуре вероятность выбора менее оптимального решения высока. Однако, в процессе работы алгоритма температура снижается, и вероятность выбора менее оптимального решения снижается.

Выбор способа уменьшения температуры может быть различным и выбирается экспериментально. Главное, чтобы температура монотонно убывала к нулю. Хорошей стратегией является умножение на каждом шаге температуры на некоторый коэффициент немного меньший единицы.

### 2.2.3.7 Выбор начальной и пороговой температуры

Этот выбор тоже следует производить экспериментально. Естественными рекомендациями могут служить выбор пороговой температуры близкой к нулю, а начальной достаточно высокой.

#### Области применения

1. Создание пути
2. Реконструкция изображения
3. Назначение задач и планирование
4. Размещение сети
5. Глобальная маршрутизация

6. Обнаружение и распознавание визуальных объектов
7. Разработка специальных цифровых фильтров

#### 2.2.4 Алгоритм интеллектуальной оптимизации «Оптимизация передвижением бактерий»

Биологические основы. Одной из самых изученных бактерий является бактерия E.Coli. Живёт в кишечнике большинства млекопитающих, самовоспроизводится за 20 мин, способность двигаться обеспечивается за счёт жгутиков мотора.

Бактерия E.Coli чередует свои действия: то перемещается прямолинейно, то кувырывается (изменение направления).

Хемотаксис – это двигательная реакция бактерии в ответ на появление в среде аттрактанта (аттрактант – вещество, привлекающее бактерии) или репеллента (репеллент – вещество, отпугивающее бактерий).

Примерная схема движения бактерии выглядит следующим образом: когда мотор работает в одном направлении, все жгутики у бактерии складываются и крутятся вместе, и бактерия движется прямолинейно. В конце пробега бактерия останавливается, мотор переключается и начинает работать в другом направлении. Бактерия при этом найдет другой ориентир в пространстве случайным образом. После этого, мотор опять переключается и начинает работать в ту сторону, в которую жгутики работают вместе, и возникает следующий отрезок прямолинейного движения.

Таким образом, можно выделить следующие хемотаксические действия бактерии E.Coli:

- если бактерия находится в нейтральной среде, то чередуются кувырки и передвижения, за счёт чего выполняется поиск.
- если бактерия перемещается по градиенту аттрактанта, то перемещение продолжается в этом же направлении. Таким образом, обеспечивается поиск более благоприятной окружающей среды.

– если происходит перемещение в направлении противоположном градиенту репеллента, то обеспечивается избегание неблагоприятной окружающей среды. Бактерии часто умирают или растворяются, и это должно учитываться при моделировании их деятельности. Мутации у *E. coli* происходят в нормальных условиях приблизительно  $10^{-7}$  в ген и в поколение.

*E. coli* могут формировать сложные крепкие пространственно-временные структуры в некоторых полутвердых полезных средах. Бактерии могут поглощать питательные вещества по их радиусу, начиная от внешней границы, заканчивая серединой, даже в случае если бактерии были изначально размещены в центре питательных веществ. Кроме того, при определённых условиях, они могут скрывать притягивающие сигналы от клетки к клетке, за счёт чего они группируются и защищают друг друга. Таким образом, эти бактерии могут группироваться в колонии.

Метод оптимизации на базе моделирования перемещения бактерий (Bacterial Foraging Optimization, BFO)

Данный метод предназначен для нахождения минимума функции  $J(X)$ ,  $X \in R^p$  при неизвестном градиенте  $\nabla J(X)$ , где  $X$  – позиция бактерии в пространстве поиска  $R^p$ , а с помощью  $J(\theta)$  моделируются благоприятные и неблагоприятные свойства среды, т.е.  $J(X)$  характеризует, где находятся аттрактанты и репелленты. Таким образом,  $J < 0$ ,  $J = 0$ ,  $J > 0$  означает, что бактерия находится в полезной, нейтральной и вредной среде, соответственно.

Пусть  $P(j, k, l) = \{X_i(j, k, l), i=1, 2, \dots, S\}$  описывает позицию каждого члена популяции  $S$  бактерий на  $j$ -ом хемотаксическом шаге,  $k$ -ом шаге воспроизведения и на  $l$ -ом событии исключения-рассеивания.

Тогда работу метода оптимизации с помощью моделирования фуражировки бактерий можно представить в виде последовательности выполнения следующих шагов изображенных на рисунке 1.5 .

Шаг 1. Инициализация. Задаются параметры, влияющие на работу метода:  $S$  – количество бактерий,  $N_{re}$  – количество шагов воспроизведения,  $N_s$  – количество шагов-повторений на одном хемотаксическом шаге,  $N_c$  –

количество хемотаксических шагов,  $N_{ed}$  – количество событий исключения-рассеивания;  $P_{ed}$  – вероятность рассеивания. Случайным образом распределить начальные значения  $X_i$ ,  $i = 1, 2, \dots, S$  по пространству поиска. Рассчитываются начальные значения целевой функции для каждой бактерии  $J_i$ .

Шаг 2. Установить:  $l = l + 1$ . Шаг 3. Установить:  $k = k + 1$ . Шаг 4. Установить:  $j = j + 1$ .

Шаг 5. Для каждой бактерии моделируется хемотаксис: кувыркание, перемещение и скольжение.

Шаг 5.1. Установить:  $i = i + 1$ .

Шаг 5.2. Кувыркание. Моделирование кувыркания достигается за счёт генерации вектора случайных чисел  $\varphi(j) \in R^p$ : где – вектор случайных чисел в интервале  $[-1; 1]$ .

Вектор  $\varphi$  представляет собой множество длин для соответствующих измерений.

Шаг 5.3. Перемещение. Рассчитывается новое положение  $i$ -ой бактерии по формуле:

$$X_{i(j+1, k, l)} = X_{i(j, k, l)} + C(i)\varphi(j),$$

где  $C(i) > 0$  – размер шага в определённом направлении, позволяющий моделировать процесс кувыркания.

Для новой позиции  $X_{i(j+1, k, l)}$  рассчитывается соответствующее значение целевой функции  $J(i, j+1, k, l)$ .

Шаг 5.4. Скольжение. Если в позиции  $X_{i(j+1, k, l)}$  значение  $J(i, j+1, k, l)$  лучше, чем в позиции  $X_{i(j, k, l)}$ , то есть выполняется условие:  $J(i, j+1, k, l) < J(i, j, k, l)$ , тогда производится следующий хемотаксический шаг с тем же вектором  $\varphi$  и в том же направлении (переход к шагу 5.3), и такое повторение может повторяться  $N_s$  раз. Если условие не выполняется, то переход к шагу 5.5.

Шаг 5.5 Если  $i < S$ , то переход к шагу 5.1, в противном случае – переход к шагу 6.

Шаг 6. Если  $j < N_c$ , то переход к шагу 4, в противном случае – переход к шагу 7.

Шаг 7. Воспроизведение. Больные бактерии умирают, а остальные, более здоровые, разделяются на две бактерии, при этом новые бактерии размещаются

в ту же самую точку пространства поиска. За счёт такого подхода обеспечивается неизменность общего количества бактерий. Для этого все агенты сортируются в соответствии с полученными значениями целевой функции, после чего худшая половина (больные бактерии) отбрасывается, а лучшая (более здоровые бактерии) – дублируется.

Шаг 8. Если  $k < N_{re}$ , тогда выполняется переход к шагу 3, в противном случае – переход к шагу 9.

Шаг 9. Исключение и рассеивание. Жизнь популяции бактерий в окружающей среде может меняться либо равномерно (например, путём потребления полезных веществ), либо резко в связи с некоторым другим воздействием. Может произойти так, что все бактерии в области погибнут, или колония бактерий будет рассеяна в другую часть окружающей среды. Данный эффект может помешать возможному хемотаксическому прогрессу, но в то же время этот эффект и помогает, поскольку, в случае рассеивания, бактерии могут разместиться около хороших источников с полезными веществами. Исключение и рассеивание помогают понизить вероятность стагнации, то есть заикливания в локальном оптимуме, что часто наблюдается в традиционных методах оптимизации (например, метод Коши, Ньютона, Левенберга-Марквардта и т.п.).

В соответствии с данным подходом каждая бактерия с вероятностью  $P_{ed}$  размещается в случайно выбранной точке пространства поиска.

Таким образом, проверяется условие:  $U_i < P_{ed}$ , где  $U_i$  – случайное число в интервале  $[0; 1]$  для  $i$ -ой бактерии.

Если данное условие выполняется, то бактерия помещается в позицию  $X_i(j, k, l)$ , полученную случайным образом.

Шаг 10. Если  $l < N_{ed}$ , тогда выполняется переход к шагу 2, в противном случае – к шагу 11.

Шаг 11. Выбирается и сохраняется лучшее решение  $J_{best}$  и соответствующая позиция  $X_{best}$ , в которой достигается лучшее решение  $J_{best}$ .

Шаг 12. Проверка на окончание поиска. Если были выполнены все циклы для всех агентов, то выполняется переход к шагу 14, в противном случае выполняется перезапуск – переход к шагу 13.

Шаг 13. Перезапуск агентов: выбираются новые случайные позиции для каждого агента  $X_i$ ,  $i = 1, 2, \dots, S$  и рассчитываются соответствующие значения целевой функции  $J_i$ ,  $i = 1, 2, \dots, S$ . Счётчики циклов сбрасываются в 0:  $j = 0$ ;  $k = 0$ ;  $l = 0$ .

Шаг 14. Остановка.

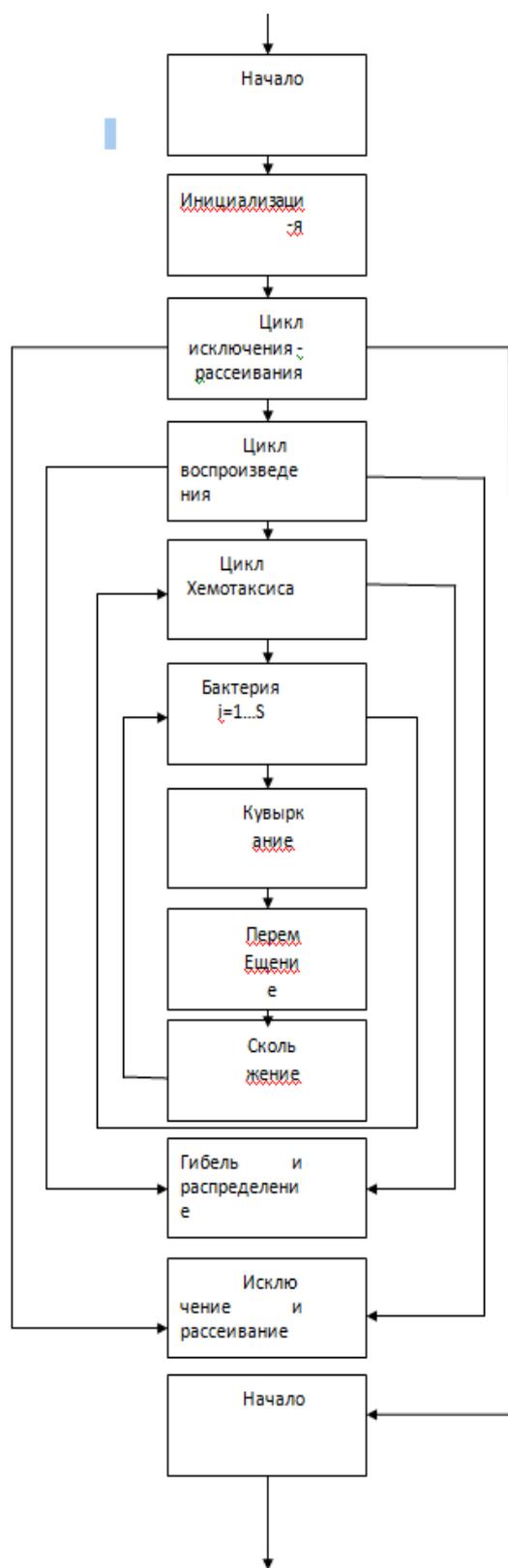


Рисунок 1.5-Цикл бактерий.

Общая схема работы метода оптимизации на основе моделирования перемещения бактерий представлена на (Рис.1.5)

Рассмотренный метод оптимизация на основе моделирования перемещения бактерий не учитывал одну из важных особенностей поведения бактерий: бактерии E.Coli, как указывалось ранее, способны, группируясь, создавать пространственно-временные структуры. В связи с этим было предложено несколько подходов, позволяющих учитывать в процессе моделирования перемещения бактерий и данную особенность их поведения, основными из которых являются: моделирование сигналов между клетками и применение PSO-оператора (Particle Swarm Optimization, PSO).

## 2.2.5 Алгоритм интеллектуальной оптимизации «Алгоритм летучей мыши»

### 2.2.5.1 Общая характеристика алгоритма

Алгоритмы роевого интеллекта – это алгоритмы коллективного поведения децентрализованных, самоорганизующихся естественных или искусственных систем. Данные алгоритмы получили свое начало в 1989 году, когда ХерардоБени и Ван Цзин ввели данный термин в контексте системы клеточных роботов.

Алгоритм летучих мышей (BatAlgoithm) – это алгоритм оптимизации, разработанный Янгом (X.-Sh.Yang) в 2010 году. Одним из основных преимуществ алгоритма является скорость его выполнения. Этот алгоритм потенциально более мощный, чем алгоритм роя частиц и генетический алгоритм, а так же гармонический поиск. Более того, гармонический поиск и алгоритм роя частиц являются особыми случаями алгоритма летучих мышей при соответствующих упрощениях.Алгоритм может показаться немного сложнее, чем большинство других алгоритмов роевого интеллекта, а также эволюционных алгоритмов, однако он может быть достаточно эффективно

применен к проблемам оптимизации и давать хорошие результаты, затрачивая меньшее количество времени.

#### 2.2.5.2 Описание алгоритма

Большинство видов летучих мышей обладает совершенными средствами эхолокации, которые используются ими для обнаружения добычи и препятствий, а также для обеспечения возможности разместиться в темноте на насесте.

Алгоритм летучих мышей подчиняется следующим правилам:

- Все летучие мыши используют эхолокацию, чтобы анализировать расстояние, а также иметь различие между едой (добычей) и природными препятствиями;
- Летучие мыши перемещаются случайным образом со скоростью  $V_i$  в позиции  $x_i$  с фиксированной частотой  $f_{min}$ , изменяемой длиной волны  $\lambda$  и громкостью  $A_0$ , чтобы найти добычу. Они могут автоматически регулировать длину волны (или частоту, т.к. частота =  $1/\text{длина волны}$ ) испускаемого импульса и скорость импульса  $\hat{r}_i \in [0,1]$ , зависящих от близости цели.
- Громкость изменяется от большего (положительного)  $A_0$  к меньшему постоянному значению  $A_{min}$ .

Схема алгоритма может быть представлена в виде последовательности шагов:

Шаг 1. Определение объективной функции  $f(x)$ ,  $x=(x_1 \dots x_d)T$ .

Шаг 2. Инициализация исходной популяции летучих мышей  $x_i$  ( $i=1 \dots n$ ), скорости летучих мышей  $v_i$ , громкости  $A_i$ , скорости распространения импульса  $r_i$ .

На данном шаге задается начальное положение летучей мыши, т.е. случайный вектор, который и является текущим решением. Также указываются начальные значения громкости, скорости распространения импульса и числа итераций.

Шаг 3. Определение частоты испускаемого импульса  $f_i$  в  $x_i$ .

Шаг 4. Определение оптимума, т.е. текущего лучшего решения.

Шаг 5. Генерация нового решения регулированием частоты и обновлением скорости и положения:

$$f_i = f_{min} + (f_{max} - f_{min}) * \beta, \beta \in [0,1] \quad (1)$$

$$v_i^t = v_i^t - 1 + (x_i^t - \bar{x}) * f_i \quad (2)$$

где  $\bar{x}$  - текущее лучшее положение.

$$x_i^t = x_i^{t-1} + v_i^t \quad (3)$$

Шаг 6. Генерация локального решения при условии, что случайно заданное число rand в интервале (0,1) больше скорости распространения импульса (rand > ri):

$$x_{new} = x_{old} + \beta_{|x|} * A^t, \beta_{|x|} \in [-1,1] \quad (4)$$

где  $A^t$  – среднее значение громкости.

Шаг 7. Если локальное решение меньше, чем текущее лучшее решение, и случайное заданное число rand в интервале (0,1) меньше, чем громкость (rand > Ai), принимается новое решение, а также понижается громкость, и увеличивается скорость распространения импульса:

$$A_i^{t+1} = \alpha * A_i^t, \alpha \in [0,1] \quad (5)$$

$$r_i^{t+1} = r_0^t * [1 - \exp(-\gamma * t)], \gamma > 0$$

Шаг 8. В случае нахождения оптимума целевой функции прекращается работа алгоритма, иначе повторяются шаги 5 – 7, пока не закончены итерации.

Шаг 9. Вывод результатов и визуализация.

### 2.2.5.3 Преимущества и недостатки алгоритма летучей мыши

Преимуществами алгоритма являются:

- масштабируемость, возможность решать задачи независимо от их размерности;
- гибкость, системы не имеют жесткой структуры, могут без изменений использоваться в совершенно различных областях;
- удобство в применении к задачам с динамически изменяющимися условиями;

- эффективность решения оптимизационных задач.

Недостатки алгоритма заключаются в следующем:

- обычно необходимо применение дополнительных методов таких, как локальный поиск;
- сильно зависит от настройки параметров, которые подбираются только исходя из экспериментов.

### Применение алгоритма летучей мыши

Рассмотренный алгоритм может быть применен для решения задач оптимизации, проектирования, планирования, классификации, решения проблемы эргономики рабочего места и др.

Алгоритм может быть применен в следующих областях:

1. производственные системы;
2. разработка программного обеспечения;
3. системы массового обслуживания;
4. все сферы применения управления проектами.

### 3 Алгоритм муравьиной колонии для задачи размещения распределительных центров розничной торговой сети (для задачи о р-медиане)

Объектом для исследования выбрано АО «Тандер», которое является учредителем и управляющей компанией, владеющей сетью магазинов «Магнит» - лидирующей розничной сетью по торговле продуктами питания в России.

Сеть «Магнит» является лидером по количеству продовольственных магазинов и территории их размещения. На 30 сентября 2014 г. сеть компании включала 9 020 магазинов, из них: 7 891 магазин в формате «магазин у дома», 175 гипермаркетов, 68 магазинов «Магнит Семейный» и 886 магазинов «Магнит Косметик».

Для данной торговой сети актуальной является задача размещения распределительных центров, которая формулируется следующим образом.

Задача о р-медиане - это задача о размещении заданного числа (скажем, р) пунктов обслуживания, при которых сумма кратчайших расстояний от вершин графа до ближайших пунктов принимает минимально возможное значение.

Дано:

- множество пунктов возможного размещения;
- список розничных торговых точек компании (расположение магазинов);
- затраты на транспортировку (стоимость прохождения 1 км пути);
- необходимое количество распределительных центров для открытия;
- расположение ближайшего распределительного центра компании.

Цель: открыть склад и прикрепить к нему магазины так, чтобы затраты были минимальными.

Задача целочисленного линейного программирования:

$$F(z, X) = \sum_{t \in I} Tt + \sum_{t \in I} \sum_{j \in J} t(tj)x(tj) \rightarrow mln \quad (1)$$

$$\begin{cases} \sum_{t \in I} X_{tj} = 1, j \in J \\ \sum_{t \in I} Z_t = p \\ Z_t \geq X_{tj}, t \in I, j \in J \\ Z_t \in \{0|1\}; X_{tj} \geq 0, t \in I, j \in J \end{cases} \quad (2)$$

Обозначения:

$m$  – число пунктов возможного размещения распределительного центра;

$i$  – номер пункта возможного размещения распределительного центра;

$n$  – число торговых точек;

$j$  – номер торговой точки;

$t_{tj}$  – затраты на удовлетворение спроса  $j$  магазина  $i$  распределительного центра (транспортные затраты);

$x_{tj}$  – затраты на удовлетворение спроса  $j$  клиента;

$$Z_t = \begin{cases} 1, \text{ если предприятие открыто в } t, \\ 0 \text{ в ином случае} \end{cases} \quad (3)$$

$T_t$  – транспортные затраты на доставку товаров в распределительный центр с ближайшего распределительного центра;

$p$  – количество распределительных центров, которые нужно открыть.

В крупной розничной торговой компании, такой как АО «Тандер», товар доставляется из одного распределительного центра в другой распределительный центр, и доставка в конечный пункт будет осуществляться с ближайшего распределительного центра. Необходимо учитывать транспортные затраты на доставку товара на распределительный центр. Для описания алгоритма воспользуемся комбинаторной постановкой задачи, которая имеет следующий вид [1].

Дана не отрицательная  $m \times n$  матрица  $T = (t(tj))$  с множеством индексов строк  $I$  и множеством индексов столбцов  $J$ , а также натуральное число  $p \leq m$ . Для всякого непустого подмножества  $I' \subseteq I$  положим

$$F(I') = \sum_{j \in J} \min_{t \in I'} t(tj)$$

Задача состоит в отыскании такого множества  $I'$  мощности  $p$ , что значение  $F(I')$  минимально, то есть

$$F(I') = \sum_{j \in J} \min_{t \in I'} t(tj)$$

Решение  $s$  задачи будем называть булев вектор  $z$  размерности  $m$  такой, что  $z_t = 0$ , если  $t \in I$  и  $0$  – в противном случае, где  $I$  – множество открытых в решении  $s$  распределительных центров.

Введем вектор  $a^k = (a_t^k)$ ,  $t \in I$ , с положительными координатами. Величину  $a_t^k$  будем называть уровнем феромона для  $i$ -го распределительного центра на итерации  $k$  алгоритма муравьиной колонии,  $t \in I$ .

Обозначим  $\widehat{s}^k$  лучшее найденное решение для начала итерации  $k$ , тогда  $\widehat{z}^k$  и  $\widehat{f}^k$  – булев вектор и значение целевой функции, соответствующие рекордному решению  $\widehat{s}^k$ . Пусть  $s^k$  – лучшее решение по значению целевой функции, найденное на итерации  $k$ , тогда  $z^k$  и  $f^k$  – булев вектор и значение целевой функции (рекорд на итерации  $k$ ), соответственно. Параметр  $a_{\min}$  – вещественное положительное число, задающее минимальное возможное значение уровня феромона  $a_t^k$  для всех  $t \in I$ .

0. Определяем начальный вектор феромона  $a^1$ ;  $f = \infty$  итерация  $k$ ,  $k \geq 1$

1. Строим  $L$  допустимых решений одним из алгоритмов искусственного муравья (алгоритм искусственного муравья представляет собой вероятностную модификацию жадного алгоритма, описан ниже). Алгоритм искусственного муравья определяет привлекательные возможные расположения распределительных центров.

2. Среди этих решений выбираем  $l$  лучших по целевой функции с помощью локального поиска. Выбирается наилучшее расположение распределительного центра из возможных привлекательных расположений.

3. Находим значения  $a_t^{k+1}$ ,  $t \in I$ .

Определяем уровень феромона

4. Если  $f^k < \widehat{f}^k$ ;  $\widehat{f}^{k+1} = f^k$ ;  $\widehat{s}^{k+1} = s^k$  для ненулевых компонент  $z^k$  полагаем  $a_t^{k+1} = a_{\min}$ . Иначе  $\widehat{f}^{k+1} = f^k$ ;  $\widehat{s}^{k+1} = s^k$ . Проверяем затраты у найденного расположения распределительного центра.

5. Если выполнен критерий остановки, то работа завершается. Переходим на следующую итерацию,  $k:=k+1$ . При запуске  $k$  итераций будет определено  $k$  мест размещения и одно или несколько из них будет с наименьшими производственно-транспортными затратами. Данное решение будет искомым местом размещения распределительного центра.

Компоненты вектора  $a_t^{k+1}$  вычисляются следующим образом:

$$a_t^{k+1} = \frac{a_{\min} + q * \gamma_t^k * (a_t^k - a_{\min})}{\beta_k}, l \in I \quad (4)$$

где  $\beta_k \in (0,1)$  – коэффициент затухания (испарение феромона) на итерации  $k$ ;  $\gamma_t^k \in [0,1]$  – частота появления распределительного центра  $i$  в  $l$  лучших решениях выбираемых на шаге 2 итерации  $k$ ; параметр  $q \in (0,1)$ . Таким образом, при данных значениях параметров  $\beta_k$  и  $q$ , чем чаще распределительный центр  $i$  попадает в  $l$  лучших решений по значению целевой функции, тем меньше становится соответствующее значение  $a_t$ ,  $t \in I$ .

Таким образом, муравьиный алгоритм позволяет компании оптимально расположить центры своего распространения и минимизировать издержки, вызванные транспортировкой или же порчей продукции и подобное.

Муравьиный алгоритм в современном мире используется для успешного развития производства и сферы услуг. В следствие, чего необходимо решать задачи оптимального размещения распределительных центров (складов, филиалов, офисов и т.п.). В управлении крупными предприятиями большое значение стали придавать физическому логистике. Причиной возросшего интереса связаны с повышением стоимости транспортных услуг.

В следствие, этого главной задачей транспортной системы является определение: оптимального количества предприятий; мест их расположения; характеристик каждого предприятия; оптимальной системы перевозок товаров

### 3. Применение муравьиных алгоритмов для задачи коммивояжёра.

Задача формулируется как задача поиска минимального по стоимости замкнутого маршрута по всем вершинам без повторений на полном взвешенном графе с  $n$  вершинами. Содержательно вершины графа являются городами, которые должен посетить коммивояжёр, а веса рёбер отражают расстояния (длины) или стоимости проезда. Эта задача является NP-трудной, и точный переборный алгоритм её решения имеет факториальную сложность.

Моделирование поведения муравьёв связано с распределением феромона на тропе – ребре графа в задаче коммивояжёра. При этом вероятность включения ребра в маршрут отдельного муравья пропорциональна количеству феромона на этом ребре, а количество откладываемого феромона пропорционально длине маршрута. Чем короче маршрут, тем больше феромона будет отложено на его рёбрах, следовательно, большее количество муравьёв будет включать его в синтез собственных маршрутов. Моделирование такого подхода, использующего только положительную обратную связь, приводит к преждевременной сходимости – большинство муравьёв двигается по локально оптимальному маршруту. Избежать этого можно, моделируя отрицательную обратную связь в виде испарения феромона. При этом если феромон испаряется быстро, то это приводит к потере памяти колонии и забыванию хороших решений, с другой стороны, большое время испарения может привести к получению устойчивого локального оптимального решения.

Теперь с учётом особенностей задачи коммивояжёра, мы можем описать локальные правила поведения муравьёв при выборе пути.

1. Муравьи имеют собственную «память». Поскольку каждый город может быть посещён только один раз, то у каждого муравья есть список уже посещённых городов – список запретов. Обозначим через  $J_{i,k}$  список городов, которые необходимо посетить муравью  $k$ , находящемуся в городе  $i$ .

2. Муравьи обладают «зрением» – видимость есть эвристическое желание посетить город  $j$ , если муравей находится в городе  $i$ . Будем считать, что

$$\eta_{ij} = \frac{1}{D_{ij}}$$

видимость обратно пропорциональна расстоянию между городами

3. Муравьи обладают «обонянием» – они могут улавливать след феромона, подтверждающий желание посетить город  $j$  из города  $i$  на основании опыта других муравьёв. Количество феромона на ребре  $(i,j)$  в момент времени  $t$  обозначим через  $\tau_{ij}(t)$

На этом основании мы можем сформулировать вероятностно-пропорциональное правило, определяющее вероятность перехода  $k$ -ого муравья из города  $i$  в город  $j$ :

$$\begin{cases} P_{ij,k}(t) = \frac{[\tau_{ij}(t)]^\alpha * [\eta_{ij}]^\beta}{\sum_{l \in J_{i,k}} [\tau_{il}(t)]^\alpha * [\eta_{il}]^\beta}, j \in J_{i,k} \\ P_{ij,k}(t) = 0, j \notin J_{i,k}; \end{cases} \quad (1)$$

Где  $\alpha, \beta$  – параметры, задающие веса следа феромона. При  $\alpha = 0$  алгоритм вырождается до жадного алгоритма (будет выбран ближайший город). Заметим, что выбор города является вероятностным, правило (1) лишь определяет ширину зоны города  $j$ ; в общую зону всех городов бросается случайное число, которое и определяет выбор муравья. Правило (1) не изменяется в ходе алгоритма, но у двух разных муравьёв значение вероятности перехода будут отличаться, т.к. они имеют разный список разрешённых городов.

5. Пройдя ребро  $(i,j)$ , муравей откладывает на нём некоторое количество феромона, которое должно быть связано с оптимальностью сделанного выбора. Пусть  $Tk(t)$  есть маршрут, пройденный муравьём  $k$  к моменту времени  $t$ ,  $Lk(t)$  – длина этого маршрута, а  $Q$  – параметр, имеющий значение порядка длины оптимального пути. Тогда откладываемое количество феромона может быть задано в виде

$$\Delta \tau_{ij,k}(t) = \begin{cases} \frac{Q}{L_k(t)}, (i, j) \in T_k(t); \\ 0, (i, j) \notin T_k(t); \end{cases}$$

Правила внешней среды определяют, в первую очередь, испарение феромона. Пусть  $\rho \in [0,1]$  есть коэффициент испарения, тогда правило испарения имеет вид

$$\tau_{ij}(t+1) = (1-\rho) * \tau_{ij}(t) + \Delta \tau_{ij}(t); \Delta \tau_{ij}(t) = \sum_{k=1}^m \Delta \tau_{ij,k}(t), \quad (2)$$

где  $m$  – количество муравьёв в колонии.

В начале алгоритма количества феромона на рёбрах принимается равным небольшому положительному числу. Общее количество муравьёв остаётся постоянным и равным количеству городов, каждый муравей начинает маршрут из своего города.

Дополнительная модификация алгоритма может состоять в ведении так называемых «элитных» муравьёв, которые усиливают рёбра наилучшего маршрута, найденного с начала работы алгоритма. Обозначим через  $T^*$ наилучший текущий маршрут, через  $L^*$  – его длину. Тогда если в колонии есть  $e$  элитных муравьёв, то рёбра маршрута получают дополнительное

$$\Delta \tau_e = e * \frac{Q}{L^*} \quad (3)$$

количество феромона

Реализация муравьиного алгоритма на примере задачи коммивояжера.

*Постановка задачи.* Задан взвешенный граф  $G1$  (рис.1). Найти длину пути муравья в задаче коммивояжера. Начальная вершина 1. Дана последовательность  $P = 65,61,35$  случайных чисел, выпавших при выборе очередной вершины. Расстояния  $L_{k,j}$ , между вершинами  $k,j$  и интенсивность феромона  $\tau_{k,j}$  на ребре  $[k,j]$  заданы таблицы

Таблица 1- Значения k и j на ребрах

Ребро	k	j
1-2	38	3
1-3	74	2
1-4	59	2
1-5	45	2
2-3	46	1
2-4	61	1
2-5	72	1
3-4	49	2
3-5	85	2
4-5	42	1

**Решение :** Секторы вероятности перехода сортировать по возрастанию номеров вершин . использовать формулу вероятности перехода из вершины  $k$  в  $j$ .

$$P_{k,j} = 100 * \frac{[\eta_{kj}]^\alpha * [\tau_{kij}]^\beta}{\sum [\eta_{kj}]^\alpha * [\tau_{kij}]^\beta},$$

где  $\alpha=1, \beta=1, \eta_{kj} = 1/L_{k,j}$

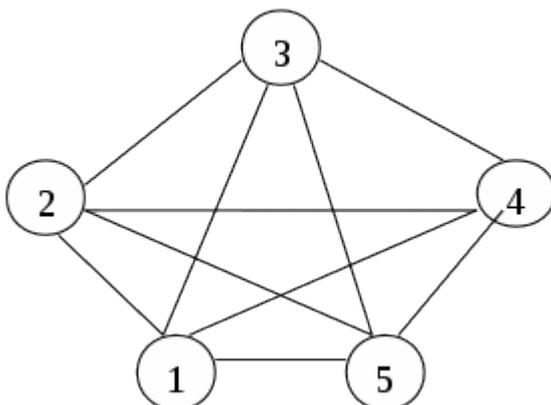


Рисунок 1.6 – схематичное изображение тропинок, по которым движутся муравьи

Представим решение в виде:

1. В начале движения из вершины 1 муравей имеет четыре возможных пути: в вершину 2, 3, 4 или 5. Вычислим вероятности перехода в эти вершины

$$P_{1,2} = 100 * \frac{3/38}{3/38 + 2/74 + 2/59 + 2/45} = \frac{7,9}{0,18} = 42,83$$

$$P_{1,3} = 100 * \frac{2/74}{0,18} = \frac{2,7}{0,18} = 14,66$$

$$P_{1,4} = 100 * \frac{2/59}{0,18} = \frac{3,39}{0,18} = 18,4$$

$$P_{1,5} = 100 * \frac{2/45}{0,18} = \frac{4,44}{0,18} = 24,11$$

Вычисляем границы четырех секторов  $[\rho'_j, \rho''_j]$ ,  $j=2,3,4,5$  вероятностей:

$$\rho'_2 = 0, \rho''_2 = P_{1,2} = 42,83$$

$$\rho'_3 = \rho''_2 + P_{1,3} = 57,5$$

$$\rho'_4 = \rho''_3 + P_{1,4} = 75,89$$

$$\rho'_5 = \rho''_4 + P_{1,5} = 100$$

Таким образом, отрезок  $[0,100]$  разбился на четыре участка

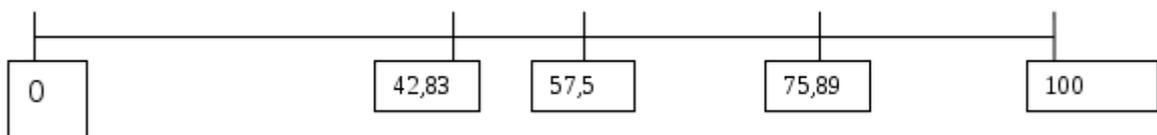


Рисунок 1.7 – Значения длины пути на отрезке

Остается запустить генератор случайных чисел и узнать, куда попадает случайное число. В нашем случае генератор дает  $=65$ , что указывает на третий участок  $57,5 < 65 < 75,89$ . Следовательно, муравей должен направиться к вершине 4.

2. Из вершины 4 только три возможные пути: 4-2, 4-3, 4-5. Пройденная вершина 1 попадает в список запрещенных вершин.

Вероятности перехода в эти вершины

$$P_{4,2} = 100 * \frac{1/61}{1/61 + 2/49 + 1/42} = \frac{1,64}{0,0081} = 20,23,$$

$$P_{4,3} = 100 * \frac{2/49}{0,18} = \frac{4,08}{0,081} = 50,38,$$

$$P_{4,5} = 100 * \frac{1/42}{0,18} = \frac{2,38}{0,081} = 23,39$$

Вычисляем границы трех секторов  $[\rho'_j, \rho''_j]$ ,  $j=2,3,5$  вероятностей:

$$\rho'_3 = \rho''_2 + P_{4,3} = 70,71$$

$$\rho'_5 = \rho''_3 + P_{4,5} = 100$$

Таким образом, отрезок  $[0,100]$  разбился на три участка

“2”

“3”

“4”



Рисунок 1.8 – Значения длины пути на отрезке (2)

Случайное число =61, полученное генератором случайных чисел попадает на второй участок. Этот участок указывает на вершину 3. Далее муравей будет выбирать маршрут из этой вершины.

3. При выходе из вершины 3 имеется только 2 возможности - направиться в вершину 2 или 5. Остальные вершины попадают в список запрещенных вершин. Оценим возможности перехода:

$$P_{3,2} = 100 * \frac{1/42}{1/46 + 2/85} = \frac{2,17}{0,045} = 48,02,$$

$$P_{3,5} = 100 * \frac{2,85}{0,18} = \frac{2,35}{0,045} = 51,98,$$

Таким образом, отрезок [0,100] разбился на два участка

“2”

“5”

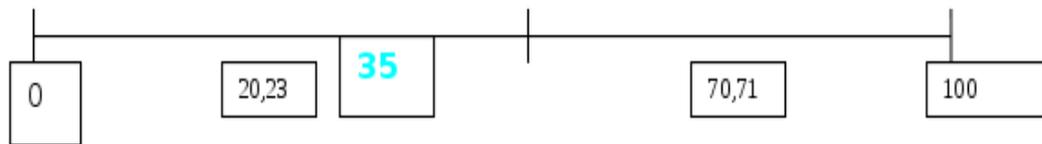


Рисунок 1.9 – Значения длины пути на отрезке (3)

Случайное число  $P_3=35$ , полученное генератором случайных чисел указывает на вершину 2.

4. В вершине 2 выбор делать не приходится. Все вершины, кроме 5 попали в список запрещенных вершин, поэтому дальнейший путь муравья очевиден. Сначала он идет в вершину 5, а затем завершает маршрут в 1, там, откуда он и вышел. Общая длина маршрута 1-4-3-2-5-1 равна

$$L_{1,4} + L_{4,3} + L_{3,2} + L_{2,5} + L_{5,1} = 59 + 49 + 46 + 72 + 45 = 271$$

## 1.2 Муравьиный алгоритм для задачи коммивояжера в псевдокоде

1. Ввод матрицы расстояний  $D$

2. Инициализация параметров алгоритма –  $\alpha, \beta, e, Q$

3. Инициализация рёбер – присвоение видимости  $\eta_{ij}$  и начальной концентрации феромона

4. Размещение муравьёв в случайно выбранные города без совпадений

5. Выбор начального кратчайшего маршрута и определение  $L^*$

//Основной цикл

6. Цикл по времени жизни колонии  $t=1, t_{max}$
7. Цикл по всем муравьям  $k=1, m$
8. Построить маршрут  $T_k(t)$  по правилу (1) и рассчитать длину  $L_k(t)$
9. конец цикла по муравьям
10. Проверка всех  $L_k(t)$  на лучшее решение по сравнению с  $L^*$
11. В случае если решение  $L_k(t)$  лучше, обновить  $L^*$  и  $T^*$
12. Цикл по всем рёбрам графа
13. Обновить следы феромона на ребре по правилам (2) и (3)
14. конец цикла по рёбрам
15. конец цикла по времени
16. Вывести кратчайший маршрут  $T^*$  и его длину  $L^*$

Сложность данного алгоритма, как несложно заметить, зависит от времени жизни колонии ( $t_{max}$ ), количества городов ( $n$ ) и количества муравьёв в колонии ( $m$ ).

Для исследования работы муравьиного алгоритма проведен вычислительный эксперимент. Предложенным алгоритмом решена задача коммивояжера. Количество городов и количество муравьев совпадают. Программа составлена в среде Delphi (Приложение 1).

Зададим начальные значения: количество городов, интенсивность феромона и стоимость маршрута.

После подсчета программа выводит кратчайший путь 1-2-3-4-5, и длину пути равную 220. Интенсивность феромона на путях после прохождения по ним муравьев увеличилась.

В ходе вычислительного эксперимента были сравнены решения данной задачи с помощью муравьиного и генетического алгоритмов. В результате пришли к выводу что, что вычисление муравьиным алгоритмом происходит быстрее.

Таблица – 2 значения полученные в ходе задачи.

	Опыт 1	Опыт 2	Опыт 3
Муравьиный алгоритм	1,600мс	1,550мс	1,750мс
Генетический алгоритм	4,300мс	3,100мс	4,700мс

## ЗАКЛЮЧЕНИЕ

Таким образом, в ходе выполнения курсовой работы мной был сделан вывод, что данные алгоритмы отлично подходят для решения задач по оптимизации различных процессов и расположения территориально необходимых ресурсов. Это очень важно ведь никто не хочет тратить ресурсы впустую, осознавая тот факт, что как ресурсы каждого отдельного предприятия, так и всей планеты не безграничны. Поэтому эти алгоритмы необходимы каждому, кто хочет организовать деятельность как физического предприятия, так и во всемирной сети.

У всех описанных выше алгоритмов есть свои преимущества и недостатки. На данный момент проводится разработка новых методов для более успешной оптимизации бизнес процессов.

Был приведен пример задачи коммивояжера, которая описывает Гамильтонов цикл обхода всех вершин графа сети на базе алгоритма случайных чисел и результат был достигнут за 3 шага алгоритма случайных чисел.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Кориков, А. М. Теория систем и системный анализ [Текст]: учебное пособие для студентов вузов, обучающихся по специальности 09.03.03 "Прикладная информатика" (квалификация (степень) "бакалавр") и другим экономическим специальностям / А. М. Кориков, С. Н. Павлов. - Москва: ИНФРА-М, 2017. - 287 с.
2. Голубков, Е. П. Методы принятия управленческих решений в 2 ч. Часть 1: учебник и практикум для академического бакалавриата / Е. П. Голубков. — 3-е изд., испр. и доп. — М. : Издательство Юрайт, 2017. — 196 с.
3. Алексеева, М. Б. Теория систем и системный анализ: учебник и практикум для академического бакалавриата / М. Б. Алексеева, П. П. Ветренко. — М.: Издательство Юрайт, 2017. — 304 с.
4. Фрейдина, Е.В. Исследование систем управления организации. Учебное пособие [Электронный ресурс]: учебное пособие / Е.В Фрейдина — Электрон. текстовые дан. — М.: Омега-Л, 2013. 368 с.
5. Бочкарев, А. А. Логистика городских транспортных систем: учебное пособие для бакалавриата и магистратуры / А. А. Бочкарев, П. А. Бочкарев. — 2-е изд., перераб. и доп. — М. : Издательство Юрайт, 2018. — 150 с.
6. Васильева, Н. В. Основы землепользования и землеустройства: учебник и практикум для СПО / Н. В. Васильева. — М.: Издательство Юрайт, 2018. — 376 с.
7. Дубина, И. Н. Основы математического моделирования социально-экономических процессов: учебник и практикум для бакалавриата и магистратуры / И. Н. Дубина. — М.: Издательство Юрайт, 2018. — 349 с

8. Кожевникова, И. А. Стохастическое моделирование процессов : учебное пособие для вузов / И. А. Кожевникова, И. Г. Журбенко. — 2-е изд., перераб. и доп. — М. : Издательство Юрайт, 2018. — 148 с. — (Серия : Авторский учебник). — ISBN 978-5-534-06254-0.
9. Малугин, В. А. Теория вероятностей и математическая статистика : учебник и практикум для бакалавриата и магистратуры / В. А. Малугин. — М. : Издательство Юрайт, 2018. — 470 с. — (Серия : Бакалавр и магистр. Академический курс). — ISBN 978-5-534-05470-5.
10. Левкин, Г. Г. Коммерческая логистика : учебное пособие для вузов / Г. Г. Левкин. — 2-е изд., испр. и доп. — М. : Издательство Юрайт, 2018. — 375 с. — (Серия : Университеты России). — ISBN 978-5-534-01642-0.