

Субботин С.А.

Олейник Ан.А.

Олейник Ал.А.

ФРАГМЕНТ РАБОЧИХ МАТЕРИАЛОВ МОНОГРАФИИ

**ЧАСТЬ III**  
**ИНТЕЛЛЕКТУАЛЬНЫЕ МУЛЬТИАГЕНТНЫЕ МЕТОДЫ**  
**(SWARM INTELLIGENCE)**

## СОДЕРЖАНИЕ

<b><u>ЧАСТЬ III ИНТЕЛЛЕКТУАЛЬНЫЕ МЕТОДЫ МУЛЬТИАГЕНТНОЙ ОПТИМИЗАЦИИ (SWARM INTELLIGENCE)</u></b> .....	<b>4</b>
<b><u>Розділ 10 Основы Swarm Intelligence</u></b> .....	<b>5</b>
<u>10.1 Мультiaгентні системи</u> .....	6
<u>10.2 Основні принципи колективного інтелекту</u> .....	7
<b><u>Розділ 11 Метод муравьиных колоний</u></b> .....	<b>11</b>
<u>11.1 Биологические основы</u> .....	11
<u>11.2 Метод муравьиных колоний</u> .....	14
<u>11.3 Разновидности метода муравьиных колоний</u> .....	18
<u>11.4 Применение метода муравьиных колоний к решению задачи отбора информативных признаков</u> .....	22
<u>11.4.1 Модификация на основе представления пунктов назначения признаками</u> .....	23
<u>11.4.2 Модификация на основе представления пунктов назначения в виде информативности признаков</u> .....	25
<u>11.4.3 Модификация с использованием операций над чёткими множествами</u> .....	28
<u>11.4.4 Модификация с использованием операций над нечёткими множествами</u> .....	30
<u>11.4.5 Сравнение модификаций метода муравьиных колоний, применяемых к решению задачи отбора признаков</u> .....	31
<u>11.5 Кластер-анализ на основе метода муравьиных колоний</u> .....	32
<u>11.6 Преимущества и недостатки метода муравьиных колоний</u> .....	34
<b><u>Розділ 12 Метод пчелиной колонии</u></b> .....	<b>35</b>
<u>12.1 Биологические основы метода пчелиной колонии</u> .....	35
<u>12.2 Формализация поведения пчёл в процессе фуражировки</u> .....	38
<u>12.3 Методы на основе моделирования пчелиной колонии для решения задач дискретной оптимизации</u> .....	41
<u>12.4 Многомерная оптимизация на основе метода пчелиной колонии</u> ...	45
<u>12.5 Отбор информативных признаков на основе метода пчелиной колонии</u> .....	49
<u>12.6 Особенности метода пчелиной колонии</u> .....	52
<b><u>Розділ 13 PSO–метод</u></b> .....	<b>55</b>
<u>13.1 Фундаментальные принципы PSO-метода</u> .....	55
<u>13.2 Подход gbest PSO</u> .....	56

<u>13.3 Подход lbest PSO</u> .....	58
<u>13.4 Компоненты скорости частицы в PSO-методе</u> .....	60
<u>13.5 Инициализация и окончание PSO-метода</u> .....	62
<u>13.6 Выбор параметров PSO-метода</u> .....	63
<u>13.7 Многокритериальный PSO-метод</u> .....	64
<u>13.8 PSO-метод с управлением изменением скорости частиц</u> .....	65
<u>13.9 Сравнение разновидностей PSO-методов</u> .....	68
<u>13.10 Приложения PSO-метода</u> .....	69
<b><u>Розділ 14 Метод оптимизации на основе моделирования</u></b>	
<b><u>перемещения бактерий</u> .....</b>	<b>70</b>
<u>14.1 Биологические основы</u> .....	70
<u>14.2 Метод оптимизации на основе моделирования перемещения</u> <u>бактерий (Bacterial Foraging Optimization, BFO)</u> .....	71
<u>14.3 Метод оптимизации на основе моделирования перемещения</u> <u>бактерий с группировкой за счёт связи между клетками</u> .....	75
<u>14.4 Метод оптимизации на основе моделирования перемещения</u> <u>бактерий с группировкой за счёт использования PSO-оператора</u> .....	78
<u>14.5 Приложения и соответствующие модификации метода</u> <u>оптимизации на основе моделирования перемещения бактерий</u> .....	80
<u>14.6 Метод оптимизации на основе моделирования перемещения</u> <u>бактерий для отбора информативных признаков</u> .....	83
<u>14.7 Преимущества и недостатки BFO</u> .....	85
<u>Выводы по части III</u> .....	87
<b><u>Литература к части III</u>.....</b>	<b>88</b>

### **ЧАСТЬ III**

## **ИНТЕЛЛЕКТУАЛЬНЫЕ МЕТОДЫ МУЛЬТИАГЕНТНОЙ ОПТИМИЗАЦИИ (SWARM INTELLIGENCE)**

У першому розділі представлено огляд колективного інтелекту (Swarm Intelligence), його основних принципів і біологічних основ. Важливим для розуміння колективного інтелекту є уявлення про основні принципи й особливості такого інтелекту, тому ці принципи й особливості описані в першому розділі.

У другому розділі детально розглядається метод мурашиних колоній. Для його кращого розуміння приводяться біологічні основи методу, тобто особливості поведінки мурах при фуражировці. Далі представлено базовий метод мурашиних колоній, після чого аналізуються різні його модифікації, які були розроблені згодом для поліпшення його роботи в аспекті рішення різних задач оптимізації. Далі описані запропоновані авторами модифікації методу мурашиних колоній для рішення задачі відбору інформативних ознак і для виконання кластерного аналізу.

У третьому розділі розглянуто метод бджолоїної колонії. Так само, як і в другому розділі, так і в третьому, спочатку описуються біологічні основи методу бджолоїної колонії – особливості поведінки бджіл при пошуку джерел нектару. Після чого розглядаються запропоновані авторами методи оптимізації для відбору інформативних ознак і для виконання багатомірної оптимізації, засновані на моделюванні поведінки бджіл. Далі представлена формалізація поведінки бджіл, особливості методу бджолоїної колонії й різних методів, призначені для рішення певних завдань оптимізації.

У четвертому розділі приводиться опис методу оптимізації з використанням груп часток. У даному розділі розглядаються основні принципи даного методу, різні PSO-методи, аналізуються основні етапи роботи даного методу, описується процес вибору параметрів для кращої роботи методу, а також приводяться основні напрямки використання PSO-методу.

## РОЗДІЛ 10

### ОСНОВИ SWARM INTELLIGENCE

Новим напрямком розвитку методів штучного інтелекту є мульти-агентні методи інтелектуальної оптимізації, що моделюють колективний інтелект суспільних тварин, комах і інших живих істот, – методи Swarm Intelligence (SI). Даний напрямок штучного інтелекту є молодим і ще мало вивченим, однак мультиагентні методи показують гарні результати при рішенні різних задач оптимізації, що говорить про перспективність розвитку даного напрямку.

Головною особливістю мультиагентних методів SI є їх біонічна природа. При цьому слід зазначити, що дані методи моделюють поведінку груп різних соціальних тварин, комах і інших живих істот, що дозволяє цим групам вирішувати різні важкі практичні задачі в природі, що свідчить про ефективність їхнього поведінки, а, отже, і про ефективність роботи даних методів. Тому актуальним є вивчення методів, можливих областей їхнього застосування, а також розробка нових математичних моделей, заснованих на поведінці колективних тварин, в аспекті рішення оптимізаційних задач, до яких дані методи ще не застосовувалися.

При розв'язку оптимізаційних задач досить перспективним є використання мультиагентних методів [1, 2]. При реалізації даних методів використовується парадигма агентно-орієнтованого програмування, що сприяє підвищенню продуктивності обчислень.

До інтелектуальних методів мультиагентної оптимізації, заснованих на моделюванні суспільного інтелекту, відносяться: метод мурашиних колоній (Ant Colony Optimization, ACO) [3–5], метод бджолоїної колонії (Bee Colony Optimization, BCO) [6, 7], оптимізація за допомогою груп часток (Particle Swarm Optimization, PSO) [8] і інші методи [9]. Дані методи є новими, і багато в чому ще недостатньо вивчені, однак деякі з них уже ефективно застосовуються для рішення різних задач оптимізації. Наприклад, ACO застосовується для розв'язку задачі комівояжера [10, 11], задачі календарного планування [12], відбору інформативних ознак [13–15], кластеризації [16] і ін. [17–21], BCO – для розв'язку задачі календарного планування [22], розв'язку задачі комівояжера [23, 24], розв'язку транспортного завдання [25] і ін. [26, 27].

## 10.1 Мультиагентні системи

Мультиагентну систему (МАС) в загальному випадку можна подати у вигляді множини з трьох елементів [29, 30]: Агенти, Середовище, Зв'язки між Середовищем та Агентами:

$$МАС = \langle \text{Агенти}, \text{Середовище}, \text{Зв'язки} \rangle$$

Кожен  $\text{Агент}_i$  описується за допомогою множини чотирьох елементів:  $\text{Стан}_i$ ,  $\text{Вхід}_i$ ,  $\text{Вихід}_i$ ,  $\text{Процес}_i$ :

$$\text{Агент}_i = \langle \text{Стан}_i, \text{Вхід}_i, \text{Вихід}_i, \text{Процес}_i \rangle,$$

де  $\text{Стан}_i$  – це множина змінних, що повністю визначають агента;  $\text{Вхід}_i$  та  $\text{Вихід}_i$  – підмножина  $\text{Стан}_i$ , елементи яких пов'язані із середовищем.

$\text{Процес}_i$  – автономний метод, що виконує відповідні зміни над  $\text{Станом}$ . “Автономні методи” означають, що даний метод викликається без будь-якого зовнішнього впливу.

$\text{Середовищем}$  являється множина з двох елементів:

$$\text{Середовище} = \langle \text{Стан}_c, \text{Процес}_c \rangle$$

Важливою особливістю такого подання  $\text{Середовища}$  є те, що  $\text{Середовище}$  є саме по собі активним, оскільки воно містить свій власний  $\text{Процес}_c$ , котрий може змінювати  $\text{Стан}_c$  незалежно від агентів, що входять в це  $\text{Середовище}$ .

$\text{Вхід}_i$  та  $\text{Вихід}_i$  різноманітних агентів пов'язані з елементами  $\text{Стану}_c$ , але середовище не відрізняє, які з елементів  $\text{Стану}_c$  знаходяться в залежності з ними. Відсутність  $\text{Входу}$  та  $\text{Виходу}$  у  $\text{Середовища}$  означає, що воно, на відміну від агентів, є необмеженим. Якби в  $\text{Середовище}$  були дані  $\text{Вхід}$  та  $\text{Вихід}$ , то це означало б, що середовище обмежене та є фактично агентом високого рівня. Таким чином, може моделюватися взаємодія агентів високого та низького рівнів.

У відповідності з наведеною концепцією подання мультиагентних систем можуть створюватися їхні різноманітні типи, при цьому така концепція організації систем може бути використана не тільки в програмних (штучних) системах, а й в штучних системах. Така класифікація мультиагентних систем [28] подана на рис. 10.1.



Рисунок 10.1 – Класифікація мультиагентних систем

У даній роботі розглядається розподілений штучний інтелект, що базується на колективному інтелекті [31]. Робочі групи у людей описані в [32, 33], соціолобологія у тварин – в [34, 35], мультироботні системи – в [36], штучне життя – в [37, 38].

## 10.2 Основні принципи колективного інтелекту

До мультиагентних розподілених систем штучного інтелекту відносяться мультиагентні методи інтелектуальної оптимізації (методи колективного інтелекту, *Swarm Intelligence*). Дані методи мають біонічну природу, тобто вони засновані на моделюванні поведінки комах, птахів, тварин і т.п., поведінки яких носить колективний характер, за рахунок чого досягається, так званий, колективний інтелект.

Тому для визначення основних принципів колективного інтелекту досить розглянути принципи поведінки різних колективних комах. До колективних комах відносяться комахи, які живуть колоніями, наприклад, мурахи, бджоли, терміти, деякі види ос і т.п. В основі поведінки колоній таких комах лежить самоорганізація [39, 40].

Самоорганізація – множина динамічних механізмів, відповідно до яких система регулюється на глобальному рівні за рахунок взаємодії її компонентів на нижньому рівні без прямої взаємодії між цими компонентами. Самоорганізація базується на чотирьох складових.

1. Позитивний зворотний зв'язок: досягається за рахунок виконання простих поведінкових емпіричних прийомів, які забезпечують створення рішень. Наприклад, при фуражировці мурах збільшення феромонів на шляху до джерела їжі – це різновид позитивного зворотного зв'язку, оскі-

льки в такий спосіб створюється мережа з можливих шляхів до джерела їжі. У бджіл позитивний зворотний зв'язок полягає в тому, що бджоли, ґрунтуючись на отриманій інформації від інших бджіл, починають летіти до зазначеного джерела ресурсів.

2. Негативний зворотний зв'язок: урівноважує позитивний зворотний зв'язок, що веде до стабілізації колективного поведіння. При фуражировці мурах негативний зворотний зв'язок може бути викликаний наступними факторами: обмежена кількість фуражирів, виснаження джерела їжі, випаровування феромону. У бджіл негативний зворотний зв'язок полягає в тому, що бджола, ґрунтуючись на інформації, отриманій від інших бджіл, може вирішити, що знайдену нею джерело гірше.

3. Нестійкість позитивного зворотного зв'язку: поведіння соціальних комах засновано на стохастичних правилах. Випадкові відхилення в рішеннях – основа для нових рішень. Крім того, хаотичність може бути корисною, оскільки це дозволяє колонії виявити нові рішення. Наприклад, фуражир, що збився з курсу, може знайти нові, неопрацьовані джерела харчування, і потім привести за собою інших робочих особин для розробки цього джерела їжі.

4. Вимога множинної взаємодії між особинами, за рахунок чого досягається поява нових гарних і витривалих рішень. Множинність взаємодії в бджіл полягає в тому, що інформація про джерело ресурсів, знайдене однією бджолою, доступна для всіх інших у вулику за допомогою виконання, так званого, виляючого танцю.

Крім основних складових, самоорганізація характеризується певними властивостями.

1. Динамічність: формування рішень вимагає постійної взаємодії між особинами колонії й навколишнім середовищем. Ця взаємодія забезпечує позитивний зворотний зв'язок, за рахунок якого створюються колективні групи й можливості для їхнього проживання, у той час як негативний зворотний зв'язок знижує вплив прямого зворотного зв'язку.

2. Емерджентність: властивості систем, що самоорганізуються, є більш складними, ніж властивості окремих агентів, що входять у неї. Такі властивості системи є результатом нелінійної комбінації взаємодій між агентами.

3. Нелінійні взаємодії між агентами ведуть також до розгалуження в поведінці систем, що самоорганізуються. Розгалуження (біфуркація) – поява нових стійких рішень внаслідок зміни певних параметрів системи. За рахунок цього досягається якісне поліпшення в колективній поведінці.

4. Мультистійкість – для заданих встановлених параметрів система може досягти різних стійких станів в залежності від початкового стану й випадкових відхилень.



Для розуміння колективного інтелекту необхідно визначити функції, які виконують соціальні комахи в процесі рішення різних задач. Можна виділити чотири функції такого виду: координація, кооперація, колективне прийняття рішень, спеціалізація. Вони не є взаємно виключними, а навпаки – спільно вносять вклад у досягнення поставлених перед колонією цілей.

Координація – відповідна організація задач окремих індивідів у часі й просторі, що дозволяє вирішити виниклу перед колонією проблему. Дана функція приводить до певного просторово-тимчасового розподілу особин, їхніх дій і/або результатів їхніх дій для досягнення поставленої мети.

Наприклад, координація виконується при організації переміщення гнізда бджіл або рою сарани [41, 42]. У цьому випадку взаємодії між окремими особинами приводять до синхронного (тимчасова організація) і спрямованого (просторова організація) руху особин до певної мети.

Координація також проявляється при фуражировці у мурах шляхом застосування феромона, що залишають мурахи при своєму переміщенні. За рахунок виділення феромону створюються мережі, які приводять до просторової організації у фуражировці [43, 44].

В якості останнього прикладу координації можна привести будівельну діяльність у колоніях комах. У процесі будівництва гнізда в деяких різновидах соціальних ос [45-48] або термітів [49, 50] за рахунок процесів стигмержі (stigmergy) особини поліпшують отримане гніздо (просторова організація), попередньо досягнуте (тимчасова організація) іншими особинами.

Кооперація досягається за рахунок того, що особини виконують разом загальну задачу, що не могла б бути вирішена окремим індивідом. Індивіди повинні об'єднувати свої зусилля, щоб успішно вирішити загальну задачу, що перебуває поза межами можливостей окремих індивідів.

Кооперація спостерігається в процесі видобутку їжі, коли окремого індивіда недостатньо, щоб доставити їжу. Відомо багато прикладів спільного транспортування їжі різними видами мурах: *Oecophylla longinoda* [51], *Eciton burchelli* [52], *Formica* [53, 54]. Таке транспортування їжі може бути дуже ефективним при доставці їжі в гніздо. Наприклад, мурахи *Pheidologeton diversus* спільно можуть транспортувати в десять разів більше ваги, ніж якби робили це окремо одне від одного [55].

Колективне прийняття рішень відноситься до механізмів, які спрацьовують, коли колонія зіштовхується із проблемою вибору. Цей механізм завершується колективним вибором одного з можливих рішень.

Наприклад, медоносні бджоли вибирають більш продуктивні ділянки із квітками шляхом вербування незайнятих бджіл за допомогою ви-

ляючого танцю, виконуваним фуражиром, що повернувся від джерела нектару [56].

Спеціалізація полягає в тому, що різні дії виконуються окремими спеціалізованими групами індивідів. Наприклад, фуражировка або догляд за потомством [57, 58, 35]. Така спеціалізація може здійснюватися як за рахунок поведінкової диференціації, так і залежно від віку індивіда.

У співтоваристві медоносних бджіл праця особин чітко розподілена. Одні з них добувають і приносять у вулик їжу, інші – забирають її й складають у стільники, треті – чистять комірки, четверті – крильми, як вентилятором, провітрюють вулик, п'яті – несуть варту службу, чергуючи при вході, і не пропускають ворогів. У родині існують розвідники для пошуку їжі, няньки для догляду за малятами, бджоли-прибиральники, гробарі, які виносять із вулика мертвих. Бджоли-солдати організовані в спеціальні військові підрозділи, які несуть службу по захисту вулика від викрадачів меду.

Таким чином, організація колективного поведіння у соціальних комах може бути розглянута як сукупність чотирьох складових: координація, кооперація, колективне прийняття рішень і спеціалізація. Кожна із цих складових проявляється на колективному рівні за рахунок непрямої взаємодії між особинами. Така взаємодія забезпечує можливість оцінювання зовнішньої інформації відповідно до двох основних напрямків.

1. Координація й спеціалізація забезпечують формування просторових, тимчасових і соціальних зв'язків, що виникають у процесі роботи колонії. Координація регулює просторово-тимчасове розміщення індивідів, у той час як за рахунок спеціалізації досягається розподіл їхніх дій.

2. Кооперація й колективне прийняття рішень є інструменти, за допомогою яких колонія може реагувати на зміни в зовнішньому середовищі. Колективне прийняття рішень надає механізми, які забезпечують рішення колонії, у той час як кооперація надає механізми, які дозволяють вийти за рамки можливості окремих індивідів.

Разом чотири складові в поведінці соціальних комах формують уявлення, що колонія в цілому планує свою роботу для досягнення поставлених цілей.

## РОЗДІЛ 11

### МЕТОД МУРАВЬИНЫХ КОЛОНИЙ

#### 11.1 Биологические основы

Известно, что насекомые, включая муравьев различных видов, термитов и некоторые подразновидности пчел и ос, живут в колониях, составленных из большого количества взаимодействующих индивидуумов. Колонии насекомых способны к решению различных задач оптимизации, которые ни одно насекомое не было бы способно решить отдельно (например, нахождение кратчайших путей в процессе кормодобывания, решения задачи при назначении рабочей силы и кластеризация при организации гнезд).

Для роя насекомых необходима некоторая форма связи, чтобы сотрудничать при решении задачи. Эта связь между индивидуумами колонии может быть более или менее прямой, в зависимости от определённых разновидностей. Когда пчела находит источник продовольствия, она сообщает направление и расстояние до местоположения, где она нашла продовольствие другим пчелам, выполняя характерный танец. Это пример прямой связи, поскольку другие пчелы должны чувствовать танец, который выполняет одна пчела, чтобы определить местонахождение источника продовольствия. Другие формы прямой связи: возбуждение физическим контактом или обменом продовольствием или жидкостью.

Косвенная связь между индивидуумами колонии является более тонкой и требует, чтобы один индивидуум изменил окружающую среду таким способом, чтобы это изменило поведение индивидуумов, проходящих через эту измененную окружающую среду в будущем.

Один вариант, где существует этот тип экологически вызванного действия в природе – это, когда термиты строят гнездо, которое имеет очень сложную структуру и показывает свойства подобные управлению климата. Всякий раз, когда стадия строительства заканчивается, среда рабочего меняется, и допускается следующая стадия работы, которая в свою очередь заканчивается новой средой, и т.д.

Другой пример косвенной связи – наложение следов феромона, выполняемое некоторыми разновидностями муравьев. Муравей при кормодобывании отмечает дорожку, оставляя определённое количество феромона в след за собой, за счёт чего побуждает следовать по его пути другого муравья, который также занимается кормодобыванием.

Принцип изменения окружающей среды как средство связи для изменении поведения называют *stigmergy*. Он является основным в органи-

зации в муравьиных колониях. Хотя муравьи имеют королеву, она – специализированный муравей, который является ответственным только за то, чтобы класть яйца и не имеет никакой управляющей функции. Вместо этого, муравьиные колонии самоорганизуются.

Термин самоорганизация (СО) используется, чтобы описать сложное поведение, которое возникает при взаимодействии сравнительно простых агентов. С помощью СО муравьи способны решить сложные задачи, с которыми они сталкиваются ежедневно. Выгоды СО при решении задачи особенно очевидны в её распределенном и здоровом характере. Муравьиная колония может эффективно поддерживать осмысленное поведение, даже если большое количество муравьев неспособно к содействию.

Чтобы лучше понять механизм и способность муравьиных колоний сходиться к хорошим решениям при поиске кратчайшего пути от гнезда к источнику продовольствия были проведены некоторые эксперименты [17, 18]. При проведении экспериментов колонии муравьёв Аргентины *Linepithema humile* давали два пути идентичной длины, и после того, как прошло некоторое время, было замечено, что муравьи сходились к одной из дорожек, после чего фактически была исключена альтернатива. Чтобы проверить, сходился ли бы этот вид муравьёв к наиболее короткому из множества путей, был проведен двойной эксперимент моста, где муравьи должны были выбирать дважды между коротким и длинным путями (рис. 11.1).

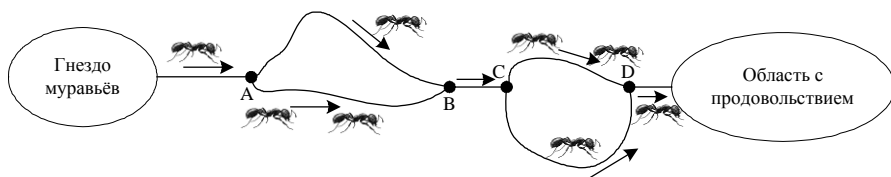


Рисунок 11.1 – Двойной эксперимент моста

Муравей Аргентины является фактически слепым, поэтому он не имеет никаких непосредственных средств идентификации короткого пути. Однако, несмотря на этот недостаток, как показали результаты эксперимента, колония муравьев способна к обнаружению кратчайшего пути, соединяющего гнездо с областью, содержащей продовольствие.

Первоначально, все муравьи расположены на участке гнезда. Множество муравьев отправляется от гнезда в поиске продовольствия, каждый муравей оставляет феромон на своём пути, и достигает первой вилки в точке А. Так как муравьи не имеют никакой информации о том, какой

путь выбрать, то есть никакой другой муравей не проходил тут ранее и не оставлял за собой след феромона, каждый муравей будет выбирать идти или вправо или влево с равной вероятностью.

Впоследствии приблизительно одна половина муравьев выберет более короткий путь, остальные – более длинный маршрут к пересечению с В. Муравьи, которые выберут более короткий путь, достигнут этого пересечения быстрее, и должны будут решить, какой способ выбрать, чтобы вернуться.

Снова нет никакой информации для муравьев, чтобы использовать её как ориентир, поэтому половина муравьев, достигших пересечения В, возвратится к гнезду, в то время как остальные продолжают перемещение к области, содержащей продовольствие.

Муравьи на более длинном участке между пересечениями А и В, незатронутом другими муравьями, которых они встретили ранее на прямом участке, достигают пересечения В и также разделятся; однако, так как интенсивность следа феромона, который находится на пути назад к гнезду приблизительно вдвое больше, чем следа феромона, достигающего области с продовольствием, большинство возвратится к гнезду, прибывая туда в то же время как и другие муравьи, которые выбрали длинный путь. Интересно, что так как больше муравьев теперь шло по короткому участку между пересечениями А и В по сравнению с длинным, следующие муравьи, покидающие гнездо теперь уже будут более склонны выбрать короткий путь, который является первым удачным выбором при поиске самого короткого пути.

Поведение муравьев на втором мосту в пересечениях между С и D фактически идентично поведению, показанному прежде на первом мосту между пересечениями А и В. В конечном счете, множество муравьев достигнет продовольствия и соберет некоторое количество продовольствия, чтобы принести назад к гнезду.

Достигая пересечения D, муравьи предпочтут короткий участок по тем же самым причинам, что и муравьи, начинающие перемещение из гнезда, и то же самое происходит снова в пересечение В.

Так как количество феромона в пересечениях А и С на пути назад к гнезду примерно равно сумме количества феромона на этих двух участках от гнезда, то также наиболее вероятен самый короткий полный путь от области с продовольствием назад к гнезду при выборе муравьев при возвращении.

Поскольку муравьи непрерывно распределяют феромон во время своего перемещения короткий путь непрерывно усиливается все большим количеством муравьев, пока количество феромона, помещенного в дальнейшем, относительно других альтернативных маршрутов настолько не

высоко, что фактически все муравьи будут использовать самый короткий путь, то есть система сойдётся к самому короткому пути через самоукрепление.

Заметим, что феромон, используемый муравьями, медленно испаряется через какое-то время, что не оставляет сомнений при объяснении двойного эксперимента моста. Действительно, длины путей, которые не были выбраны в течение некоторого времени, неизменно большие, и эти пути не будут содержать почти никаких следов феромона из-за их испарения после определённого промежутка времени, далее увеличивается вероятность выбора муравьями коротких путей. Следует обратить внимание, что количество феромона на самом коротком пути имеет максимальную ценность, которая достигнута большим количеством феромона, оставленным муравьями.

## 11.2 Метод муравьиных колоний

Метод муравьиных колоний основан на моделировании взаимодействия нескольких искусственных аналогов муравьёв, программно представляемых в виде интеллектуальных агентов, являющихся членами большой колонии. Моделируемые агенты, перемещаясь по графу решений, совместно решают проблему и помогают другим агентам в дальнейшей оптимизации решения.

Базовая идея метода муравьиных колоний состоит в решении оптимизационной задачи путем применения не прямой связи между автономными агентами.

Настоящий муравей во время перемещения по пути оставляет за собой некоторое количество феромона. Феромоны – химические вещества, вырабатываемые экзокринными железами (или специальными клетками) животных; выделяясь во внешнюю среду одними особями, феромоны оказывают влияние на поведение, а иногда на рост и развитие других особей того же вида. Особенно важную роль феромоны играют в жизни насекомых. У общественных насекомых регулируют состав колонии и специфическую деятельность ее членов. В методе муравьиных колоний агент оставляет феромон на гранях сети.

Обобщённая схема работы метода муравьиных колоний представлена на рис. 11.2.

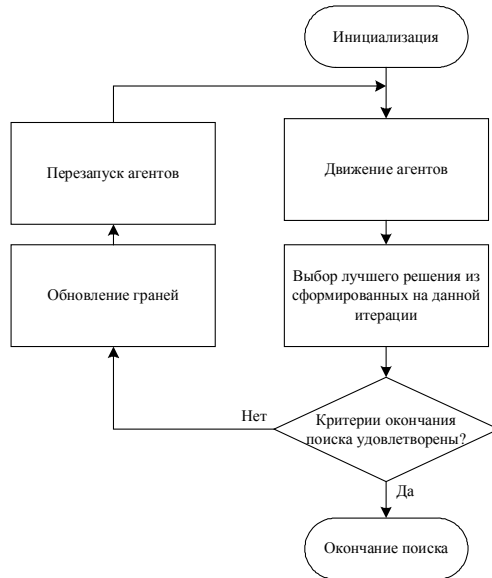


Рисунок 11.2 – Укрупнённая схема работы метода муравьиных колоний

Первой задачей, к которой был применён метод муравьиных колоний, была задача коммивояжёра (Traveling Salesman Problem, TSP). Основной причиной, по которой была выбрана данная задача, является то, что в ней необходимо находить кратчайший путь, поэтому аналогия метода муравьиных колоний легко приспосабливается для решения данной задачи.

Для решения данной задачи было разработано несколько различных методов, основанных на оптимизации с помощью муравьиных колоний.

Первым методом был метод муравьиных систем (Ant System – AS) [59, 60]. В дальнейшем этот метод послужил основой для многих других методов, работающих на принципе муравьиных колоний.

В методе муравьиных систем агент формирует своё решение в процессе перемещения от одного узла к другому на графе решений. Метод работает до выполнения  $t_{\max}$  итераций. На каждой итерации агенты составляют свои решения за  $n$  шагов, на каждом из которых применяется правило выбора следующего узла – правило выбора агентом, находящимся в узле  $r$ , следующего узла для перемещения в него.

В [4, 59–62] было предложено три метода муравьиных систем, различных между собой способом обновления путей – рёбер. Эти методы

назывались: плотностный (ant-density), количественный (ant-quantity) и циклический (ant-cycle) методы муравьиных систем. В плотностном и количественном методах агенты оставляли феромоны в процессе составления решения, в то время как в циклическом методе агенты оставляли феромоны после окончания перемещения, т.е. после составления решения.

Проведенные эксперименты по решению тестовых задач [4, 59, 60] показали, что циклический метод имел значительно лучшие результаты по сравнению с другими двумя. В связи с этим, два худших метода были отброшены. Поэтому, в дальнейшем, под методом муравьиных колоний понимается именно циклический метод муравьиных систем.

Количество феромонов  $\tau_{ru}(t)$ , оставляемого агентом, соответствует дуге  $(r, u)$  – это количество характеризует преимущество выбора данной грани по сравнению с другими при перемещении. Информация о феромонах граней изменяется в процессе составления решений. При этом количество феромонов, оставляемое агентами, пропорционально качеству решения, составленного соответствующим агентом: чем меньше путь, тем больше будет оставлено феромона, и наоборот, чем длиннее путь, тем меньше будет оставлено феромона на соответствующих рёбрах. Такой подход позволяет обеспечить непосредственный поиск в направлении нахождения лучшего решения.

Память об узлах, которые были посещены агентом, обеспечивается путём введения, так называемого списка *табу*  $tList$  – в нём хранится битовый массив, с помощью которого определяются посещённые и не посещённые узлы. Таким образом, агент должен проходить через каждый узел только один раз. Узлы в списке “текущего путешествия” *Path* располагаются в том порядке, в котором агент посещал их. Позже список используется для определения протяженности пути между узлами.

Метод муравьиных колоний включает следующие основные шаги.

Шаг 1. Задать параметры метода:  $\alpha$  – коэффициент, определяющий относительную значимость пути;  $\beta$  – параметр, показывающий значимость расстояния;  $\rho$  – коэффициент количества феромона, которое агент оставляет на пути, где  $(1-\rho)$  показывает коэффициент испарения феромона на пути после его завершения;  $Q$  – константу, относящуюся к количеству феромона, которое было оставлено на пути; *startPheromone* – начальное значение феромона, которое находится на путях до начала моделирования.

Шаг 2. Инициализация метода. Создание популяции агентов. После создания популяция агентов поровну распределяется по узлам сети. Необходимо равное разделение агентов между узлами, чтобы все узлы имели одинаковые шансы стать отправной точкой. Если все агенты начнут дви-



жение из одной точки, это будет означать, что данная точка считается оптимальной для старта, а на самом деле она таковой может не являться. Но при этом, если количество агентов не кратно количеству узлов, то количество агентов в узлах будет разным, но эта разница не должна превосходить 1.

Шаг 3. Движение агентов. Если агент еще не закончил путь, то есть не посетил все узлы сети, для определения следующей грани пути рассчитывается вероятность перехода в  $u$ -ый узел, когда агент находится в  $r$ -ом узле, по формуле (11.1):

$$P_{ru} = \frac{\tau_{ru}(t)^\alpha \cdot \eta_{ru}(t)^\beta}{\sum_{k \in J} \tau_{rk}(t)^\alpha \cdot \eta_{rk}(t)^\beta} < rand(1), \quad (11.1)$$

где  $P_{ru}$  – вероятность того, что агент переместиться в  $u$ -ый узел из  $r$ -го узла;  $rand(1)$  – случайное число в интервале (0; 1);  $J$  – множество узлов, ещё не посещённых агентом;  $\tau_{ru}(t)$  – интенсивность феромона на грани между узлами  $r$  и  $u$  в момент времени  $t$ ;  $\eta_{ru}(t)$  – функция, которая представляет измерение обратного расстояния для грани.

Агент перемещается только по тем узлам, которые еще не были посещены (как указано списком табу  $tList$ ). Поэтому вероятность рассчитывается только для граней, которые ведут к еще не посещенным узлам.

Шаг 3 повторяется до тех пор, пока каждый агент не завершит путь. Циклы запрещены, поскольку в метод включен список табу  $tList$ .

Шаг 4. После завершения перемещений агентов может быть подсчитана длина пути. Она равна сумме всех граней, по которым путешествовал агент. Количество феромона, которое было оставлено на каждой грани пути  $i$ -го агента, определяется по формуле (11.2):

$$\Delta \tau^i(t) = \frac{Q}{L^i(t)}, \quad (11.2)$$

где  $\Delta \tau^i(t)$  – количество феромона, оставляемое  $i$ -ым агентом;  $L^i(t)$  – длина пути  $i$ -го агента.

Результат является средством измерения пути: короткий путь характеризуется высокой концентрацией феромона, а более длинный путь – более низкой. Затем полученный результат используется для увеличения количества феромона вдоль каждой грани пройденного  $i$ -ым агентом пути по формуле (11.3):

$$\tau_{ru}(t) = \tau_{ru}(t-1) + \rho \cdot \sum_{i=1}^{N_{ru}} \Delta \tau^i(t), \quad (11.3)$$

где  $r, u$  – узлы, образующие грани, которые посетил  $i$ -ый агент;  $N_{ru}$  – общее количество агентов, посетивших грань  $ru$ .

Данная формула применяется ко всему пути, при этом каждая грань помечается феромоном пропорционально длине пути. Поэтому следует дождаться, пока агент закончит путешествие и только потом обновить уровни феромона, в противном случае истинная длина пути останется неизвестной. Константа  $\rho$  принимает значение между 0 и 1.

В начале пути у каждой грани есть шанс быть выбранной. Чтобы постепенно удалить грани, которые входят в худшие пути в сети, ко всем граням применяется процедура испарения феромона. Используя константу  $\rho$  из предыдущего выражения, можно составить формулу (11.4):

$$\tau_{ru}(t) = \tau_{ru}(t) \cdot (1 - \rho). \quad (11.4)$$

Шаг 5. Проверка на достижение оптимального результата. Проверка может выполняться в соответствии с ограничением на максимальное количество итераций или проверка может считаться успешной, когда на протяжении нескольких итераций не было отмечено изменений в выборе наилучшего пути. Если проверка дала положительный результат, то происходит окончание работы метода (переход к шагу 7), в противном случае – переход к шагу 6.

Шаг 6. Повторный запуск. После того как путь агента завершен, грани обновлены в соответствии с длиной пути и произошло испарение феромона на всех гранях, метод выполняется повторно. Список табу очищается, и длина пути обнуляется. Переход к шагу 3.

Шаг 7. Останов. Определяется лучший путь, который и является решением.

### 11.3 Разновидности метода муравьиных колоний

В связи с возможностью различного математического описания поведения муравьёв в [4, 10, 11, 59–66] были разработаны расширения метода муравьиных систем. К ним относятся: метод муравьиных систем, основанный на элитной стратегии [3, 59]; метод муравьиных систем, основанный на ранжировании ( $AS_{rank}$ ) [4]; метод системы муравьиных колоний [10, 11, 63]; макси-минный метод муравьиных систем (MAX-MIN AS – MMAS) [64–66].

Первым расширением метода муравьиных систем была элитная стратегия, предложенная в [3, 59]. Данный подход основывается на дополнительном увеличении количества феромонов для лучшего глобального пути в данный момент времени  $t$ . Таким образом, процедура добавления феромона для дуг, которые входят лучший на данный момент времени путь, выполняется повторно, при этом количество добавляемого феромона рассчитывается в соответствии с длиной лучшего пути.

Далее был предложен метод муравьиных систем, основанный на ранжировании ( $AS_{\text{rank}}$ ) [4]. Данный метод по своей сути является расширением элитной стратегии и заключается в следующем: агенты сортируются по длине составленных ими путей, после чего на глобально лучшем пути феромоны увеличиваются с весом  $w$ , а также увеличение феромонов производится также только для дуг, вошедших в пути  $(w-1)$  лучших агентов; при этом  $k$ -ый лучший агент будет добавлять феромон с весом  $(w-k)$  в соответствии с (11.5):

$$\tau_{ru}(t+1) = \rho \cdot \tau_{ru}(t) + w \cdot \Delta \tau_{ru}^{gb}(t) + \sum_{k=1}^{w-1} (k-w) \cdot \Delta \tau_{ru}^k(t), \quad (11.5)$$

где  $\Delta \tau_{ru}^{gb}(t) = 1/L^{gb}(t)$ ,  $L^{gb}(t)$  – длина лучшего глобального пути.

Метод системы муравьиных колоний (Ant Colony System – ACS) [15–17] улучшает метод муравьиных систем путём использования информации, полученной предыдущими агентами, для изучения пространства поиска. Это достигается с помощью двух механизмов. Во-первых, используется строгая элитная стратегия при обновлении феромонов на гранях. Во-вторых, агенты выбирают следующий город для перемещения, используя, так называемое, псевдослучайное пропорциональное правило [63]: с вероятностью  $q_0$  агент перемещается в пункт  $u$ , для которого произведение количества феромонов и эвристической информации является максимальным:  $u = \arg \max_{u \in J'} \{\tau_{ru} \cdot \eta_{ru}(t)\}^\beta$ , в то время как с вероятностью  $1-q_0$  будет применён базовый подход при определении следующего пункта для перехода, описанный в методе муравьиных систем. Значение  $q_0$  является константой. При этом если  $q_0$  стремится к 1, то используется только псевдослучайное пропорциональное правило, когда же  $q_0 = 0$ , тогда метод муравьиных колоний работает по принципу метода муравьиных систем.

При обновлении путей, как было сказано выше, применяется строгая элитная стратегия, в соответствии с которой только агент, составивший лучшее решение, вырабатывает феромон на пути своего перемещения. Тогда количество феромонов на гранях изменяется в соответствии с формулой (11.6):

$$\tau_{ru}(t+1) = \rho \cdot \tau_{ru}(t) + (1-\rho) \cdot \Delta \tau_{ru}^{best}(t). \quad (11.6)$$

В качестве лучшего агента может использоваться агент, получивший лучшее решение на данной итерации, или глобально лучший агент, получивший лучшее решение на всех итерациях от начала работы метода.

Последним отличием метода системы муравьиных колоний является то, что агенты обновляют количество феромонов в процессе составления решения (подобно плотностному и количественному методам муравьиных систем). Такой подход приводит к уменьшению вероятности

выбора одинаковых путей всеми агентами. За счёт этого понижается вероятность заикливания в локальном оптимуме.

Макси-минный метод муравьиных систем (MAX-MIN AS – MMAS) [64–66] вводит нижнюю и верхнюю границу для возможных значений феромонов на грани, а также данный метод отличается подходом к определению их значения при инициализации. Практически в MMAS используется интервал значений феромонов, ограниченный  $\tau_{\min}$  и  $\tau_{\max}$ ,  $\forall \tau_{\min}$ : т.е.  $\tau_{\min} \leq \tau_{\min} \leq \tau_{\max}$ . Количество феромонов граней при инициализации задаётся равным нижней границе интервала, что обеспечивает лучшее исследование пространства решений. В MMAS, также как и в ACS, только лучший агент (глобально лучший или локально) выполняет добавление феромонов после каждой итерации метода. Результаты вычислений [17] показали, что лучшие результаты получаются, когда обновление феромонов выполняется с использованием глобально лучшего решения. В MMAS также часто применяется локальный поиск для улучшения его свойств.

В общем виде различия между разновидностями метода муравьиных колоний можно отобразить в табл. 11.1.

Таблица 11.1 – Различия между разновидностями метода муравьиных колоний

Критерий	AS	ASrank	ACS	MMAS
Добавление феромона	После получения решения	После получения решения	В процессе получения решения	В процессе получения решения
Динамическое изменение коэффициента Q	Отсутствует	Отсутствует	Отсутствует	Отсутствует
Правило выбора следующего пункта	Традиционный подход	Псевдослучайное пропорциональное правило	Традиционный подход	Традиционный подход
Применение элитной стратегии	Все агенты участвуют в обновлении путей	Обновление выполняют (w-1) локально лучших агентов и глобально лучший агент	Обновление выполняет только лучший (глобально или локально) агент	

Использование ограничений для различных параметров	Отсутствуют	Ограничение на количество агентов, участвующих в обновлении путей	Отсутствует	Используется интервал значений феромонов
Применение локальной оптимизации	Отсутствует	Отсутствует	Используются традиционные методы локальной оптимизации	Отсутствует
Решаемые задачи	Задача коммивояжёра, квадратичная задача о назначениях, задача календарного планирования, транспортная задача	Задача коммивояжёра	Задача коммивояжёра, задача календарного планирования	Задача коммивояжёра, квадратичная задача о назначениях
Влияние количества агентов на нахождение результата	Сильное	Среднее	Слабое	Слабое

В дальнейшем описанные модели метода муравьиных колоний применялись для решения других оптимизационных задач. Квадратичная задача о назначениях решалась (Quadratic Assignment Problem, QAP) с помощью метода муравьиных систем [17, 67, 68], а также MMAS [69]. Применение данных методов заключалось в использовании соответствующей эвристической информации данной задачи. Также решались следующие задачи: задача календарного планирования (Job-shop Scheduling Problem, JSP) [3, 12], транспортная задача (Vehicle Routing Problem, VRP) [18, 70–72], задача выбора кратчайшей общей надпоследовательности (Shortest Common Supersequence Problem, SCSP) [19, 73], задача раскраски графа [20], задача последовательного упорядочивания [74] и другие задачи [21, 75, 76]. При решении тестовых задач методы муравьиных колоний показали хорошие результаты по сравнению с традиционными методами оптимизации, предназначенными для решения данных задач [12, 18–21, 70–76].

### 11.4 Применение метода муравьиных колоний к решению задачи отбора информативных признаков

В общем случае метод муравьиных колоний может быть применен к любой комбинаторной задаче, которая может быть согласована со следующими требованиями:

- соответствующее представление задачи – задача должна быть описана в виде графа с набором узлов и граней между узлами;
- эвристическая пригодность граней – возможность применения эвристической меры адекватности путей от одного узла к каждому соседнему узлу в графе;
- составление альтернативных решений, посредством чего можно рационально определять допустимые решения;
- правило обновления феромонов – правило, которое определяет вероятность перемещения агента из одного узла графа к другому.

Задачей отбора признаков является нахождение из полного набора признаков сокращённого набора наиболее информативных признаков с количеством элементов  $outCF$ , которое задаётся. При этом вывод об информативности делается на основании построенной определённым образом модели на основании анализируемого набора признаков. Таким образом, задачей модификации метода муравьиных колоний является нахождение такого набора признаков  $H_o$ , при котором будет достигаться заданная ошибка модели, строящейся на основании полученного набора.

Задача отбора информативных признаков может быть представлена в виде, соответствующем методу муравьиных колоний. Метод муравьиных колоний требует, чтобы задача была представлена в виде графа, узлы которого характеризуют признаки, а грани между ними – выбор следующего признака. В этом и заключаются соответствующие модификации метода муравьиных колоний для решения задачи отбора информативных признаков.

Для решения задачи отбора информативных признаков предлагается использовать такие модификации метода муравьиных колоний: модификации на основе представления пунктов назначения признаками [77, 78, 80–87] и в виде информативности признаков [80–84, 86, 87]; с использованием операций над чёткими множествами [79, 80, 83, 85]; с использованием операций над нечёткими множествами [79, 80, 83, 85].

### 11.4.1 Модификация на основе представления пунктов назначения признаками

Основная идея данной модификации метода муравьиных колоний в аспекте решения задачи отбора признаков заключается в следующем: предполагается, что агент должен совершить путь по заданному количеству пунктов назначения  $n$ , при этом каждому пункту ставится в соответствие признак  $x_i$ ,  $i=\overline{1, n}$ ; тогда путь, пройденный агентом, составляет набор признаков  $N$ , который предлагается использовать для построения модели. При этом данные хранятся не в битовом виде, а в виде номеров признаков  $j=\overline{1, n}$ , которые следует оставить. При получении итогового результата это позволяет получать оценку степени влияния каждого признака.

Шаг 1. Задать параметры метода:  $initPh$  – начальное количество феромона;  $\rho$  – коэффициент количества феромона, которое агент оставляет на пути, где  $(1-\rho)$  показывает коэффициент испарения феромона на пути после его завершения;  $Q$  – константу, относящуюся к количеству феромона, которое было оставлено на пути;  $inCF$  – количество признаков в исходном наборе;  $outCF$  – количество признаков, которое следует оставить в сокращённом наборе.

Шаг 2. Инициализация. Создание популяции агентов. После создания популяция агентов поровну распределяется по пунктам назначения.

Шаг 3. Движение агентов. Если агент еще не закончил путь, то есть не посетил пункты в количестве равном  $outCF$ , для определения следующего пункта в пути агента используется формула (11.7):

$$P_k = \frac{\tau_k(t)}{\sum_{i=1}^{n^j} \tau_i^j(t) + \tau_k(t)} < rand(1), \quad (11.7)$$

где  $P_k$  – вероятность того, что  $j$ -ый агент, который ещё не посетил  $n^j$  пунктов, для продолжения пути выберет пункт  $k$ ;  $\tau_i(t)$  – количество феромона в  $i$ -ом пункте в момент времени  $t$ ;  $rand(1)$  – случайное число в интервале  $(0;1)$ .

Агент путешествует только в пункты, которые еще не были посещены (как указано списком  $tabu\ tList$ ). Поэтому вероятность рассчитывается только для пунктов, которые ещё не были посещены агентом.

Шаг 3 продолжается до тех пор, пока каждый агент не закончит свой путь. Путь считается пройденным до конца, когда агент посетит пункты в количестве, равном  $outCF$ . Циклы запрещены, поскольку в метод включен список  $tabu\ tList$ .

Шаг 4. На основании пунктов, посещённых агентом, строится модель и для неё определяется ошибка

Шаг 4.1. Последовательность номеров узлов, посещённых агентом, переводится в битовую строку  $H^j$  по формуле (11.8):

$$a_i = \begin{cases} 0, & \text{если } i \notin L^j; \\ 1, & \text{если } i \in L^j, \end{cases} \quad (11.8)$$

где  $a_i$  –  $i$ -ый бит в битовой строке;  $i$  – номер узла;  $L^j$  – путь  $j$ -го агента.

Шаг 4.2. На основании полученной битовой строки  $H^j$  и экспериментальных данных строится модель (например, на основе регрессии или нейросети).

Шаг 4.3. Рассчитывается ошибка  $\varepsilon_j$  для полученной модели по формуле (2.9):

$$\varepsilon_j = \frac{1}{2} \sum_{i=1}^N (y_i - y_{i \text{ расч}})^2, \quad (11.9)$$

где  $y_i$  – исходные значения выходной переменной;  $y_{i \text{ расч}}$  – расчётное значение выходной переменной по построенной модели;  $N$  – количество экспериментов.

Шаг 5. Определяется количество феромона, которое было оставлено в каждом пункте пути для  $j$ -го агента по формуле (11.10):

$$\Delta \tau^j(t) = \frac{Q}{\varepsilon_j}, \quad (11.10)$$

где  $\Delta \tau^j(t)$  – количество феромонов, которое надо добавить каждому пункту, входящему в путь  $j$ -го агента в момент времени  $t$ ;  $Q$  – параметр, понижающий степень влияния ошибки модели  $\varepsilon_j$ , построенной на основании признаков, входящих в путь  $j$ -го агента.

Результат формулы (11.10) является средством измерения качества полученной модели: чем меньше ошибка построенной на основании выбранных признаков модели, тем больше концентрация феромонов в пунктах, вошедших в путь данного агента. Затем  $\Delta \tau^j(t)$  используется в формуле (11.11), чтобы увеличить количество феромона каждого пункта, в котором побывал агент:

$$\tau_i(t) = \tau_i(t-1) + \rho \cdot \sum_{j=1}^N \Delta \tau_i^j(t), \quad (11.11)$$

где  $N$  – количество агентов, принявший  $i$ -ый признак.

Данная формула применяется ко всему пути, при этом каждый пункт помечается феромоном пропорционально ошибке полученной модели. Поэтому следует дождаться, пока агент закончит путешествие и только потом



обновить уровни феромона, в противном случае истинная длина пути останется неизвестной. Константа  $\rho$  принимает значение между 0 и 1.

В начале пути у каждого пункта есть шанс быть выбранным. Чтобы постепенно удалить пункты, которые входят в худшие комбинации, ко всем путям применяется процедура испарения феромона. Используя константу  $\rho$  из (11.11), можно составить формулу (11.12):

$$\tau_i(t) = \tau_i(t) \cdot (1 - \rho). \quad (11.12)$$

Шаг 6. Проверка на достижение оптимального результата. Проверка может выполняться в соответствии с ограничением на максимальное количество итераций или проверка считается успешной, когда на протяжении нескольких итераций не было отмечено изменений в выборе наилучшего решения. Если проверка дала положительный результат, то происходит окончание работы метода (переход к этапу 7), в противном случае – переход к этапу 6.

Шаг 7. Повторный запуск. После того как путь агента завершен, грани обновлены в соответствии с длиной пути и произошло испарение феромона на всех гранях, метод выполняется повторно. Список табу очищается, и длина пути обнуляется. Агентам разрешается перемещаться по сети, основывая выбор грани на формуле (11.7). Переход к этапу 3.

Шаг 8. Останов. Определяется лучший путь, который является решением. Лучший путь выбирается на основании ошибок моделей, которые рассчитываются по принципу, описанному на шаге 4.

Шаг 9. После останова и выбора наилучшего пути строится финальная модель, что позволяет обеспечить требуемую точность при меньшем количестве признаков, чем в исходном наборе.

#### 11.4.2 Модификация на основе представления пунктов назначения в виде информативности признаков

Особенностью данной модификации является то, что пунктам назначения  $(\overline{1}, n)$ , где  $n$  – количество входных признаков *inCF*, ставится в соответствие двоичный набор  $B = \{b_i \mid b_i = \langle 0; 1 \rangle, i = \overline{1, n}\}$ , формируемый случайно, и при этом количество единичных элементов должно быть равным количеству признаков *outCF*, которые необходимо оставить. Каждый агент должен совершить путь по всем пунктам назначения, после чего по полученному пути строится модель в соответствии с двоичным набором.

Основные шаги данной модификации приведены ниже.

Шаг 1. Задать параметры метода:  $\alpha$  – коэффициент, определяющий относительную значимость пути (количество феромона на пути);  $initPh$  – начальное количество феромона;  $\rho$  – коэффициент количества феромона, которое агент оставляет на пути, где  $(1-\rho)$  показывает коэффициент испарения феромона на пути после его завершения;  $Q$  – константу, относящуюся к количеству феромона, которое было оставлено на пути;  $inCF$  – количество признаков в исходном наборе;  $outCF$  – количество признаков, которое следует оставить в сокращённом наборе,  $maxTime$  – максимальное время моделирования. Установить времени моделирования  $curTime$  в 0.

Шаг 2. Инициализация. Создание двоичного набора  $B$  и популяции агентов. Двоичный набор  $B$  следует создавать случайным образом. Предлагается получить  $outCF$  целых случайных чисел  $rNum_j$  ( $j=\overline{1, outCF}$ ) в интервале  $[1; inCF]$  и каждому  $b_{rNum_j}$  присвоить 1, остальные элементы  $b_k$

принять равными 0. Количество агентов в популяции предлагается выбирать не меньшим количества признаков. После создания популяция агентов поровну распределяется по пунктам назначения.

Шаг 3. Изменение количества времени моделирования:  $curTime = curTime + 1$ .

Шаг 4. Движение агентов. Каждый агент, пока не закончит весь путь, выбирает следующий пункт назначения. При этом при выборе следующего пункта учитывается количество феромона, уже оставленного в этом пункте, и количество феромона, оставленного в других пунктах, ещё не посещённых агентом. Добавление пункта  $k$  в путь  $j$ -го агента происходит при выполнении следующего условия:

$$\frac{\tau_k(t)}{\sum_{i=1}^{n^j} \tau_i^j(t) + \tau_k^\alpha(t)} < rand(1), \quad (11.13)$$

где  $rand(1)$  – случайное число в интервале  $(0; 1)$ ;  $\tau_i^j(t)$  – количество феромона в  $i$ -ом пункте  $j$ -го агента, который ещё не посетил  $n^j$  пунктов.

Передвижение происходит до тех пор, пока каждый агент не посетит все пункты.

Шаг 5. Если каждый агент посетил все пункты, то происходит обновление путей (шаг 6) и перезапуск агентов (шаг 7). В противном случае – переход на шаг 3.

Шаг 6. Обновление путей. Данный этап необходим для того, чтобы понизить вероятность в дальнейшем выбора таких пунктов назначения, включение которых даёт повышение ошибки построенной модели относительно других моделей, построенных без этого пункта.

Шаг 6.1. Для каждого агента строится модель на основании пунктов, посещённых им. При этом происходит декодирование из последовательности номеров пунктов в двоичный код для определения тех признаков, которые следует оставить для построения модели. Декодирование происходит на основании двоичного набора  $B$ . Каждому пункту  $x_i^j$  ( $i=\overline{1, \text{outCF}}$ )  $j$ -го агента ставится в соответствии число  $a_i$  по следующему принципу:

$$a_i = \begin{cases} 1, & \text{если } b_{x_i^j} = 1; \\ 0, & \text{если } b_{x_i^j} = 0. \end{cases} \quad (11.14)$$

Шаг 6.2. На основании полученной битовой строки  $H^j = \{a_i, i=\overline{1, \text{outCF}}\}$  и экспериментальных данных строится модель (например, на основе регрессии или нейросети).

Шаг 6.3. Рассчитывается ошибка  $\varepsilon_j$ :

$$\varepsilon_j = \frac{1}{2} \sum_{i=1}^N (y_i - y_{i \text{ расч}})^2, \quad (11.15)$$

где  $y_i$  – исходные значения выходной переменной;  $y_{i \text{ расч}}$  – расчётное значение выходной переменной по построенной модели;  $N$  – количество экспериментов.

Шаг 6.4. В соответствии с рассчитанной ошибкой количество феромонов каждого пункта, который посетил  $j$ -ый агент увеличивается на  $\Delta\tau(j)$ :

$$\tau_i^j(t) = \Delta\tau^j(t) + (\tau_i^j(t) \cdot \rho), \quad (2.16)$$

где  $\tau_i^j(t)$  – количество феромона в  $i$ -ом пункте  $j$ -го агента;

$\Delta\tau(t)$  определяется исходя из рассуждений, что больше феромона должно добавляться тем пунктам, использование которых даёт меньшую ошибку, при этом следует влияние значения самой ошибки на количество добавляемого феромона сделать не слишком существенным, поскольку может возникнуть закливание на одном пути:

$$\Delta\tau^j(t) = \frac{Q}{\varepsilon_j}, \quad (11.17)$$

где  $Q$  – параметр, понижающий степень влияния ошибки модели  $\varepsilon_j$ , построенной на основании признаков, входящих в путь  $j$ -го агента.

Шаг 6.5. В начале пути у каждого пункта есть шанс быть выбранным. Чтобы постепенно удалить пункты, которые входят в худшие комбинации, ко всем путям применяется процедура испарения феромона:

$$\tau_i(t) = \tau_i(t) \cdot (1 - \rho). \quad (11.18)$$

Шаг 7. Перезапуск агентов. На данном шаге производятся действия аналогичные шагу 2 – инициализация. Только в данном случае – уже имеющаяся популяция распределяется по начальным пунктам, обнуляется список табу  $tList$ . Также на этом шаге выбирается лучший из путей, которые были получены при данном запуске агентов и сохраняется в  $bestPath$ .

Шаг 8. Проверка на останов. Осуществляется либо на основании времени моделирования  $curTime$  (останов происходит при достижении предела  $maxTime$ ), либо на основании ошибки лучшей из полученных моделей ( $\epsilon_{bestPath} < \epsilon$ ). Если проверка на останов дала положительный результат – переход к шагу 9, в противном случае – переход к шагу 3.

Шаг 9. Останов. Определяется лучший путь, который является решением. Лучший путь выбирается на основании ошибок моделей, которые рассчитываются по принципу, описанному на шаге 4.

Шаг 10. После останова и выбора наилучшего пути строится финальная модель, что позволяет обеспечить требуемую точность при меньшем количестве признаков, чем в исходном наборе.

#### 11.4.3 Модификация с использованием операций над чёткими множествами

Основная идея данной модификации представлена на рис. 11.3.

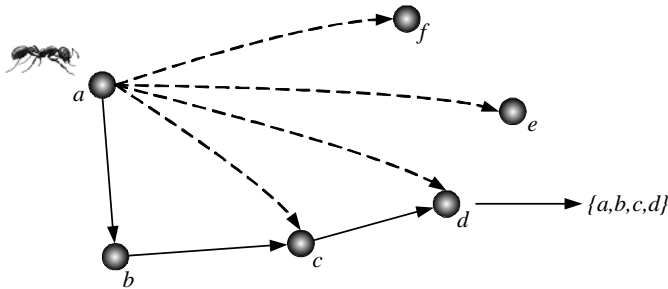


Рисунок 11.3 – Графическая интерпретация метода муравьиных колоний для отбора признаков

Каждый агент перемещается по признакам до тех пор, пока не будет выполняться определённое условие окончания перемещения. По окончании перемещения агент получается подмножество, которое рассматривается как кандидат для сокращения данных.

Основные шаги данной модификации приведены ниже.

Шаг 1. Работа метода начинается с генерации  $k$  агентов, каждый из которых размещается случайным образом по графу (т.е. каждый агент начинает обход с одного случайного признака). Количество агентов, размещающихся по графу, может быть равно количеству входных признаков; при этом каждый агент начинает формирование пути с различного признака.

Шаг 2. С начальных позиций, агенты обходят грани на основании вероятностного правила перехода, которое обозначает вероятность того, что  $k$ -ый агент переместится из признака  $i$  в признак  $j$  в момент времени  $t$ :

$$p_{ij}^k(t) = \frac{\tau_{ij}(t)^\alpha \cdot \eta_{ij}^\beta}{\sum_{l \in J_i^k} \tau_{il}(t)^\alpha \cdot \eta_{il}^\beta}, \quad (11.19)$$

где  $J_i^k$  – множество признаков, не посещённых  $k$ -ым агентом;  $\eta_{ij}$  – эвристическая пригодность выбора  $j$ -го признака при нахождении агента в  $i$ -ом узле;  $\tau_{ij}(t)$  – количество феромона на грани  $(i, j)$ .

Выбор параметров  $\alpha$  и  $\beta$  определяется экспериментально. Несколько значений параметра выбираются в интервале  $[0, 1]$  и оцениваются экспериментально.

Шаг 3. Минимальность подмножества и “качество” – два ключевых фактора работы данного метода, так что обновление феромона должно быть пропорционально “качеству” и обратно пропорционально размеру подмножества.

Для применения этого механизма при поиске сокращённого набора данных, необходимо использовать функцию зависимости  $\gamma'$  как критерий останова (11.20):

$$\gamma'_p(Q) = \frac{|POS_p(Q)|}{|U|}, \quad (11.20)$$

где  $U$  – универсальное конечное множество решений;  $POS_p(Q)$  – положительная область, выражающая отношение эквивалентности по  $U$ :

$$POS_p(Q) = \bigcup_{x \in U/Q} \underline{PX}, \quad (11.21)$$

где  $\underline{PX}$  –  $P$ -нижнее приближение множества  $P$  к множеству  $X$ :  
 $\underline{PX} = \{x | [x]_p \subseteq X\};$

Это означает, что агент закончит формирование подмножества признаков, когда зависимость подмножества достигнет максимума с входным множеством.

Шаг 4. Собираются все результирующие подмножества агентов и оцениваются определённым методом, который может выбираться в зависимости от решаемой задачи.

Шаг 5. Проверка на останов. Если оптимальное подмножество найдено или алгоритм был выполнен определённое количество раз, то процесс останавливается и происходит переход на шаг 7.

Шаг 6. Если проверка на останов не была успешной, то феромоны обновляются. Феромон на каждой грани обновляется в соответствии с формулой (11.22):

$$\tau' = (\rho \cdot \tau + \Delta\tau) \cdot (1 - \rho), \quad (11.22)$$

где  $\tau'$  – новое значение феромона;  $\tau$  – старое значение;  $\Delta\tau$  – разница, на которую должен быть увеличен феромон;  $\rho$  – значение, которое является постоянной величиной, используемой для симуляции испарения феромонов.

Рассчитывается  $\Delta\tau$ , если грань  $(i, j)$  была посещена, по формуле (11.23):

$$\Delta\tau_{ij}(t) = \sum_{k=1}^n (\gamma'(S^k) / |S^k|), \quad (11.23)$$

в противном случае:  $\Delta\tau_{ij}(t)=0$ , где  $S^k$  – подмножество признаков, выбранных  $k$ -ым агентом;  $\gamma'$  – функция зависимости.

Для ясности все агенты обновляют феромоны. Альтернативные стратегии могут использовать для этого, например, только агентов с лучшими подмножествами, позволяя им пропорционально увеличивать феромон.

После обновления феромонов происходит переход на шаг 1.

Шаг 7. Останов. Выбирается лучшее подмножество.

#### **11.4.4 Модификация с использованием операций над нечёткими множествами**

Работа модификации с использованием операций над нечёткими множествами аналогична работе модификации с использованием операций над чёткими множествами, описанной выше, и состоит из тех же шагов.

Отличиями являются лишь данные, с которыми работает метод – в данной модификации обрабатываются нечёткие множества данных, а соответственно, формула для функции зависимости  $\gamma'$ :

$$\gamma'_p(Q) = \frac{|\mu_{pos_p(Q)}(x)|}{|U|} = \frac{\sum_{x \in U} \mu_{pos_p(Q)}(x)}{|U|}, \quad (11.24)$$

где  $\mu_A(x)$  – функция принадлежности нечёткого подмножества  $A$ :

$$\begin{aligned} \mu_{pos_p(Q)}(x) &= \sup_{x \in U/Q} \mu_{\underline{p}x}(x), \\ \mu_{\underline{p}x}(x) &= \sup_{F \in U/P} \min(\mu_F(x), \inf_{y \in U} \max(1 - \mu_F(y), \mu_x(y))) \end{aligned} \quad (11.25)$$

#### 11.4.5 Сравнение модификаций метода муравьиных колоний, применяемых к решению задачи отбора признаков

Рассмотрены четыре модификации метода муравьиных колоний для отбора признаков:

- 1) с использованием операций над чёткими множествами;
- 2) с использованием операций над нечёткими множествами;
- 3) модификация на основе представления пунктов назначения признаками;
- 4) модификация на основе представления пунктов назначения в виде информативности признаков.

Основными различиями между модификациями являются:

1. Отличие видов данных, с которыми работает метод.

Модификации 1, 3 и 4 работают с чёткими множествами данных, а модификация 2 – с нечёткими множествами данных. Вследствие этого меняются некоторые этапы работы модификаций.

2. Отличие аналогий узлов.

При приведении задачи отбора признаков к виду задачи метода муравьиных колоний узлам графа ставятся в соответствие различные аналогии: в модификациях 1, 2 и 3 им соответствуют непосредственно признаки, а в 4-ой модификации узлы содержат логические числа (т.е. 0 или 1), указывающие на информативность, при этом информативность каждого узла определяется не номером узла, а порядком прохождения агентом узлов.

3. Различная принадлежность феромона. Количество феромона в модификациях 1 и 2 относится к грани и характеризует её, то есть отображает возможность перехода от одного узла к другому, а в модификациях 3 и 4 характеризует узел, то есть возможность включения данного узла в результирующий набор.

4. Способ выбора следующего узла при перемещении агента.

Выбор следующего узла осуществляется по правилу перехода.

Для модификаций 1 и 2 правило перехода обозначает вероятность того, что  $k$ -ый агент переместится из признака  $i$  в признак  $j$  в момент времени  $t$ .

Правило перехода для модификаций 3 и 4 обозначает вероятность того, что пункт  $i$  будет добавлен в путь агента

#### 5. Процедура изменения количества феромонов.

Сама процедура изменения количества феромонов одинакова для всех модификаций и является такой же, как в базовом методе муравьиных колоний, и заключается в изменении феромонов для каждой грани (в модификациях 1 и 2) или для каждого узла (в модификациях 3 и 4). Определением разницы, на которую должен быть увеличен феромон различаются рассмотренные модификации:

#### 6. Критерий останова работы метода.

Для модификаций 1 и 2 критерием останова является  $\gamma'$  – метод остановится, когда зависимость подмножества достигнет максимума с входным множеством ( $\gamma'=1$  для непротиворечивых множеств).

Для модификаций 3 и 4 останов осуществляется либо на основании времени моделирования (останов происходит при достижении предела), либо на основании ошибки лучшей из полученных моделей ( $\epsilon_{\text{bestPath}} < \epsilon$ ).

## 11.5 Кластер-анализ на основе метода муравьиных колоний

Для решения задачи кластеризации с помощью метода муравьиных колоний предлагается его модификация, в которой выполняется не моделирование феромонов, а моделирование трёх основных действий, выполняемых муравьями при выполнении сортировки пищи: подъём пищи, перенос пищи и оставление данной пищи.

Модификация метода муравьиных колоний для выполнения кластеризации может быть представлена в виде следующих шагов:

Шаг 1. Инициализация. Задаются начальные параметры метода: количество агентов, шаг перемещения  $\text{stepS}$ , максимальное количество итераций  $\text{maxIter}$ , количество кластеров, на сколько надо разбить заданную выборку, размерность сетки.

Шаг 2. Устанавливается счётчик итераций:  $\text{curIter} = 1$ .

Шаг 3. Все элементы данных размещаются случайным образом по сети. Все агенты случайно поднимают по одному элементу данных, и также случайным образом размещаются в сети.

Шаг 4. Выбирается следующий агент для перемещения. Агенты выбираются по порядку. Выбранный агент перемещается по сети с шагом  $\text{stepS}$ . Далее агент принимает решение, где оставить элемент данных. Ве-



роятность того, что агент оставит элемент данных в текущей позиции, куда он переместился, рассчитывается по формуле (11.26):

$$P_l(i) = \begin{cases} 1, & \text{если } f(i) \geq 1; \\ f(i)^{n_l}, & \text{в противном случае,} \end{cases} \quad (11.26)$$

где  $i$  – выбранный элемент данных;  $n_l$  – коэффициент, определяющий степень влияния соседних узлов при принятии решения об оставлении элемента данных. Коэффициент  $n_l$  выбирается экспериментально.

$f(i)$  – модифицированная функция Люмера–Файета [16]:

$$f(i) = \begin{cases} B \sum_{j=1}^{\sigma} (1 - \frac{d(i,j)}{\alpha}), & \text{если } f(i) > 0 \text{ и } \forall j (1 - \frac{d(i,j)}{\alpha}) > 0; \\ 0, & \text{в противном случае,} \end{cases} \quad (11.27)$$

где  $B$  – коэффициент, определяющий влияние пустых узлов сети,

$B = \frac{1}{\sigma^2}$ ;  $\sigma$  – количество соседних узлов, в которых находятся данные;

$d(i,j)$  – отличие между данными  $i$  и  $j$ ;  $\alpha$  – константа, определяемая экспериментально.

Представленная формула  $f(i)$  объединяет два важных свойства. Во-первых, использование коэффициента  $B$  обеспечивает отсечение пустых узлов сети, таким образом, обеспечивая усиленное объединение в кластеры (а не только свободную сортировку). Во-вторых, дополнительное ограничение  $\forall j (1 - \frac{d(i,j)}{\alpha}) > 0$  обеспечивает сильное отсечение больших

отличий, что значительно улучшает пространственное разделение между кластерами.

Если принимается решение, что элемент данных не следует оставлять в текущей позиции, то агент оставляет его в соседнем узле (выбирается свободный узел на основе случайного поиска).

Шаг 5. Далее агент ищет следующий элемент данных, чтобы взять его для перемещения. Случайно выбирается свободный элемент данных, для него рассчитывается его функция  $f(i)$  и в соответствии с ней принимается решение, брать для перемещения этот элемент данных или нет, в соответствии с формулой:

$$P_p(i) = \begin{cases} 1, & \text{если } f(i) \leq 1, \\ \frac{1}{f(i)^{n_p}}, & \text{в противном случае,} \end{cases} \quad (11.28)$$

где  $n_p$  – показывает степень влияния соседних узлов сети, выбирается экспериментально, предлагается  $n_p = 2$ .

Шаг 6. После выбора элемента данных текущим агентом для перемещения выполняются шаги 4 и 5 для всех остальных агентов. Если все агенты выполнили шаги 4 и 5 – переход на шаг 7.

Шаг 7. Рассчитывается количество кластеров, на которое была разбита выборка. Если рассчитанное количество кластеров совпадает с количеством, которое надо получить, то происходит переход к шагу 9, в противном случае – к шагу 8.

Шаг 8. Увеличивается счётчик итераций:  $\text{curIter} = \text{curIter} + 1$ . Если количество  $\text{curIter}$  превосходит максимальное количество итераций:  $\text{curIter} > \text{maxIter}$ , тогда выполняется переход к шагу 9, в противном случае – переход к шагу 4.

Шаг 9. Останов.

## 11.6 Преимущества и недостатки метода муравьиных колоний

Основываясь на рассмотренные основные принципы метода муравьиных колоний, его разновидности и области применения можно выделить преимущества и недостатки метода муравьиных колоний.

К преимуществам метода муравьиных колоний можно отнести:

- может использоваться в динамических приложениях (адаптируются к изменениям окружающей среды);
- использует память всей колонии, что достигается за счёт моделирования выделения феромонов;
- сходимость к оптимальному решению гарантируется;
- стохастичность, то есть случайность поиска, за счёт чего исключается возможность закливания в локальном оптимуме;
- мультиагентность;
- скорость нахождения оптимального решения выше, чем у традиционных методов;
- применяются к множеству различных задач оптимизации.

Можно выделить следующие недостатки метода муравьиных колоний:

- теоретический анализ затруднён, поскольку итоговое решение формируется в результате последовательности случайных решений; распределение вероятностей меняется при итерациях; исследования являются больше экспериментальными, чем теоретическими;
- сходимость гарантируется, но время сходимости не определено;
- высокая итеративность;
- результат работы метода достаточно сильно зависит от начальных параметров поиска, которые подбираются экспериментально.

## РОЗДІЛ 12

### МЕТОД ПЧЕЛИНОЙ КОЛОНИИ

#### 12.1 Биологические основы метода пчелиной колонии

Для описания поведения пчёл в природе используются три основных понятия: источник нектара (цветок), занятые фуражиры, незанятые фуражиры.

Источник нектара характеризуется значимостью, определяемой различными факторами, такими как: удалённость от улья, концентрация нектара, удобство добычи нектара.

Занятые фуражиры закреплены за отдельным источником, на котором они добывают нектар, то есть они “заняты” им. Занятые фуражиры владеют такой информацией о данном источнике нектара, как: расстояние и направление от улья, полезность источника.

Незанятые фуражиры продолжают искать источники нектара для их использования. Существует два типа незанятых фуражиров: разведчики, которые ищут новые источники нектара, и наблюдатели, которые ждут в улье и могут выполнять другие действия в улье.

Среднее количество разведчиков в рое составляет 5–10%.

Несмотря на коллективный характер поведения общественных насекомых и на разные схемы этого поведения, отдельное насекомое также способно выполнять различные сложные действия [6], примером чего могут служить сбор и обработка нектара пчёлами, выполнение которых хорошо организовано.

Каждая незанятая пчела может полететь к источнику нектара, следуя за пчелой-разведчиком, которая нашла путь к цветку. Это достигается за счёт того, что каждый улей имеет так называемую закрытую площадку для танца, на которой пчёлы, обнаружившие источники нектара, выполняют виляющий танец, тем самым, пытаясь привлечь других незанятых пчёл последовать за ними. Если пчела решает оставить улей, чтобы получить нектар, она следует за одной из пчел-разведчиков к области с нектаром. Таким образом, незанятая пчела становится занятой.

По достижению области с нектаром занятый фуражир добывает нектар и возвращается в улей, оставляя нектар там. После того, как пчела оставляет нектар, она может выполнить одно из следующих трех действий:

- оставить источник нектара и снова стать незанятым фуражиром;

- продолжить фуражировку к тому же источнику нектара, не вербуя других особей своего улья;

- выполнить танец и таким образом выполнить вербовку.

Пчела выбирает одну из вышеупомянутых альтернатив с некоторой вероятностью. В пределах области танца, пчелы "рекламируют" различные области нектара. Механизмы, в соответствии с которыми пчела решает следовать за другой пчелой, исследованы недостаточно хорошо, но предполагается, что вербовка среди пчел с математической точки зрения всегда является функцией качества источника нектара [6]. Также отмечено, что не все пчелы начинают фуражировку одновременно.

Таким образом, выполняется разделение функций между занятыми фуражирками и разведчиками на улучшенное изучение найденных областей с нектаром и на нахождение новых областей с нектаром, соответственно. За счёт такого разделения труда достигается эффективная работа всего роя пчёл.

Самоорганизация пчелиной колонии основывается на четырёх основных свойствах.

1. Положительная обратная связь – заключается в том, что пчёлы, основываясь на полученной информации от другой пчелы, начинают летать к указанному источнику нектара.

2. Отрицательная обратная связь – заключается в том, что пчела, основываясь на информации, полученной от других пчёл, может решить, что найденный ею источник значительно хуже по сравнению с другими найденными источниками.

3. Неустойчивость – пчёлы-разведчики выполняют случайный поиск новых источников ресурсов.

4. Множественность взаимодействия – информация об источнике ресурсов, найденного одной пчелой, доступна для всех других в улье посредством выполнения, так называемого, виляющего танца.

Основные особенности поведения пчёл можно объяснить с помощью рис. 12.1.

На рис. 12.1 изображён случай, когда уже найдено два источника нектара А и В, а также изображена пчела – потенциальный фуражир. В самом начале данная пчела может принять одно из двух решений:

- стать разведчиком ("С") и после этого направиться в поисках нектара в любую область;

- направиться в улей и, в дальнейшем, участвовать в вербовке ("В").

При этом если пчела направляется в улей на площадку для выполнения танца, то она может стать либо занятым фуражиром ("ЗФ1"), либо,

в случае, если пчела не была подвергнута вербовке, тоже стать разведчиком.

Когда пчела становится занятым фуражиром, она может быть направлена как в источник А, так и в источник В. Это зависит от выполненного влияющего танца. После того, как пчела, будучи занятым фуражиром, набрала нектар у источника, она возвращается в улей и оставляет там собранный нектар. При этом, когда пчела-фуражир возвращается, она уже может оставаться как занятым фуражиром, если нектар ещё остался, так и стать незанятым фуражиром (“НФ”). После того, как пчела оставила нектар, она может направиться на площадку для выполнения танца либо сразу отправиться обратно к источнику с нектаром (“ЗФ2”). После чего, данный цикл может повторяться до тех пор, пока не будет окончательно полностью использован исследуемый источник нектара.

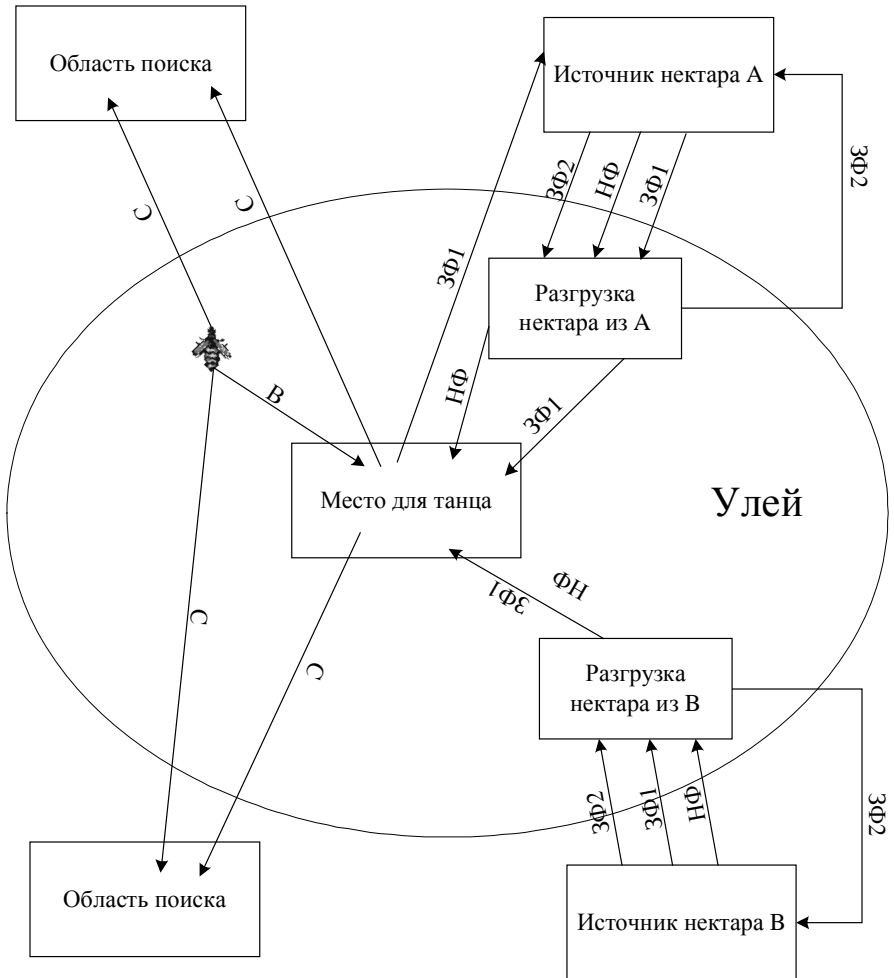


Рисунок 12.1 – Основные особенности поведения пчёл

## 12.2 Формализация поведения пчёл в процессе фуражировки

В [88] предложена поведенческая модель самоорганизации колонии пчёл, в которой пчёлы-фуражиры, которые перемещаются к цветкам, возвращаются в улей, а также обладают информацией о полезности соответ-

ствующего источника. Собранный нектар обеспечивает обратную связь с текущим состоянием потока нектара в улье. Полезность – функция качества нектара, количества нектара и расстояния от улья. Обратная связь устанавливает ответный порог для сигнала вербовки, также известный как виляющий танец, длина которого зависит и от порога, и от полезности. Виляющий танец выполняется на специальной площадке для танцев, где отдельные фуражиры могут наблюдать за их выполнением. Фуражиры могут случайным образом выбирать танец для наблюдения и в соответствии с танцем они могут определить место расположения участка цветка и покинуть улей для фуражировки. Модель самоорганизации обеспечивает пропорциональную обратную связь с качеством источника нектара.

В [7] описан подход, представленный в [88], в терминах исчисления смежных систем (Calculus of Communicating Systems, CCS).

CCS – математическая модель (формальный язык), предназначенная для описания процессов, обычно применяемая при изучении параллелизма. CCS содержит набор термов, операторов и аксиом, которые используются для описания и управления составленными выражениями. Выражения характеризуют элементы параллельной системы, а управление этими выражениями показывает, как ведёт себя система. Центральным элементом в CCS является уникально именованный агент, который обладает специфическим поведением. Поведение агента определяется множеством событий и действий, которые может выполнять агент. Множество событий, выполняемых агентом, описываются с помощью оператора “действие”, который обозначается как “.”. Другим важным оператором в CCS является оператор “+”, который является оператором выбора. Этот оператор используется в случае, если описывается участие агента в одном из нескольких альтернативных действий.

CCS в [89] работает с агентами, которые описывают состояние пчелы или состояние группы пчёл, и с действиями, которые представляют собой возможность перехода из одного состояния в другое. Таким образом, колония может быть представлена в виде связанного графа, вершинами которого являются агенты, а рёбра – действиями. Например, агент может быть представлен в следующем виде:

Разведчик<sub>*b*</sub> = хороший<sub>*b*</sub>(*s*).Поиск<sub>*b*</sub>(*s*) + плохой<sub>*b*</sub>.Незанятый<sub>*b*</sub>.

Представленный агент *b* является разведчиком, который ищет источник нектара. В случае, если он будет *плохо* искать нектар, то он станет *незанятым* фуражиром, если же он будет *хорошо* искать, то он продолжит Поиск<sub>*b*</sub>(*s*) некоторого источника *s*.

Тогда моделируемое поведение пчёл в терминах CCS можно представить на рис. 12.2.

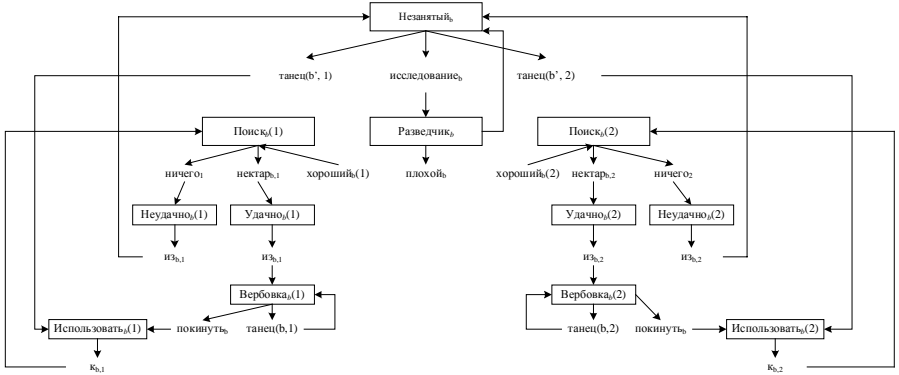


Рисунок 12.2 – Представление моделирования поведения пчёл с помощью CCS

Процесс фуражировки можно формализовать с помощью CCS в следующем виде:

$\text{Использовать}_b(s) = \kappa_{b,s}.\text{Поиск}_b(s),$

$\text{Поиск}_b(s) = \text{нектар}_{b,s}.\text{Удачно}_b(s) + \text{ничего}_s.\text{Неудачно}_b(s),$

$\text{Удачно}_b(s) = \text{из}_{b,s}.\text{Вербовка}_b(s),$

$\text{Неудачно}_b(s) = \text{из}_{b,s}.\text{Незанятый}_b,$

$\text{Вербовка}_b(s) = \overline{\text{д\`а\`и\`а\`о}}(b, s) . \text{Вербовка}_b(s) + \text{покинуть}_b.\text{Использовать}_b(s),$

$\text{Незанятый}_b = \text{танец}(b', s).\text{Использовать}_b(s) + \text{исследовать}_b.\text{Разведчик}_b,$

$\text{Разведчик}_b = \text{хороший}_b(s).\text{Поиск}_b(s) + \text{плохой}_b.\text{Незанятый}_b.$

В данном описании  $b$  – уникальный идентификатор одного агента,  $b'$  – идентификатор другого агента,  $s$  – источник нектара.

Таким образом, с помощью предложенной формализации моделируется поведение пчёл. Незанятый агент находится в улье до тех пор, пока не получит сигнал от занятого фуражира посредством виляющего танца; если такой сигнал получен, то агент перемещается к соответствующему источнику нектара; в противном случае – он может сам стать разведчиком через некоторое время; если в предполагаемом источнике нектара есть нектар, то агент приносит его в улей и моделирует выполнение виляющего танца.



### 12.3 Методы на основе моделирования пчелиной колонии для решения задач дискретной оптимизации

На основе предложенного в [88] подхода в [22] был разработан метод пчелиной колонии для решения задачи календарного планирования (BCO-JSSP).

Задача календарного планирования может характеризоваться множеством работ, каждая из которых состоит из одной или больше операций. Операции выполняются на специфической последовательности специальных машин. Целью планирования является составление расписания работ, которое минимизирует (максимизирует) меру выполнения.

Задача календарного планирования относится к NP-сложным. Мера выполнения включает: загрузку оборудования (коэффициент использования оборудования), время производственного цикла, производительность (расход, пропускную способность) и уровень запасов.

В общем случае задача календарного планирования представляется с помощью дизъюнктивного графа. Граф состоит из узлов, представляющих собой операции. Также имеются два дополнительных узла, которые представляют ресурсы и издержки. Множество ориентированных дуг используется для описания преимущества каждой работы.

Поскольку главными особенностями метода пчелиной колонии являются виляющий танец и процесс фуражировки, то предложенная модификация [22] для решения задачи календарного планирования отличается именно этими этапами работы метода пчелиной колонии по сравнению с предложенным ранее методом [88]. Аналогией источника нектара в данной модификации является путь, который может рассматриваться как решение задачи календарного планирования.

После возвращения в улей, агент выполняет виляющий танец с вероятностью  $p$ . Продолжительность  $D_i$  виляющего танца  $i$ -го агента рассчитывается по формуле:

$$D_i = d_i \cdot A,$$

где  $A$  – масштабирующий коэффициент,  $d_i$  – относительная полезность найденного источника нектара  $i$ -го агента.

Абсолютная полезность источника нектара  $i$ -го агента  $Pf_i$  для задачи календарного планирования рассчитывается по формуле:

$$Pf_i = \frac{1}{C_i},$$

где  $C_i$  – целевая функция для пути  $i$ -го агента. В данном случае она представляет собой продолжительность выполнения всех операций всех работ для пути.

Тогда, рассчитав абсолютную полезность каждого агента, можно получить среднюю полезность всей колонии  $Pf_{colony}$ :

$$Pf_{colony} = \frac{1}{n} \sum_{j=1}^n Pf_j,$$

где  $n$  – количество виляющих танцев, исполняемых в момент времени  $t$ .

Таким образом, можно рассчитать относительную полезность  $d_i$   $i$ -го фуражира:

$$d_i = \frac{Pf_i}{Pf_{colony}}.$$

Вероятность  $p_i$  того, что за  $i$ -ым агентом, после выполнения им танца, последуют другие незанятые фуражиры, определяется по формуле [26]:

$$p_i = \begin{cases} 0,60, & \text{если } Pf_i < 0,9 \cdot Pf_{colony}; \\ 0,20, & \text{если } Pf_{colony} \leq Pf_i < 0,95 \cdot Pf_{colony}; \\ 0,02, & \text{если } 0,95 \cdot Pf_{colony} \leq Pf_i < 0,95 \cdot Pf_{colony}; \\ 0,00, & \text{если } 1,15 \leq Pf_i. \end{cases}$$

Поскольку в процессе кормодобывания агенты составляют решения путём перемещения из узла к узлу на графе, который описывает возможные работы, надо рассчитать вероятность добавления в путь агента заданного узла.

Вероятность  $P_{ij}$  того, что агент выберет следующим  $j$ -ый узел, находясь в  $i$ -ом узле, рассчитывается по формуле:

$$P_{ij} = \frac{\rho_{ij}^{\alpha} \cdot d_{ij}^{-\beta}}{\sum_{j \in J^k} \rho_{ij}^{\alpha} \cdot d_{ij}^{-\beta}},$$

где  $\rho_{ij}$  – стоимость дуги между  $j$ -ым и  $i$ -ым узлами;  $d_{ij}$  – эвристическое расстояние между  $j$ -ым и  $i$ -ым узлами;  $\alpha, \beta \in [0; 1]$  – коэффициенты, выбираемые экспериментально;  $J^k$  – множество узлов, в которые можно переместиться из  $i$ -го узла.

Оценка  $\rho_{ij}$  определяется с помощью формулы:

$$\rho_{ij} = \frac{1 - m\alpha}{k - m},$$

где  $k$  – количество узлов, в которые можно переместиться из  $i$ -го узла;  $m$  – число предпочтения пути, которое может быть равно 1 или 0. Предпочтительным считается путь, который на какой-либо итерации считался пригодным для выполнения танца. При этом количество таких, так называе-

мых, элитных путей ограничено. Таким образом, на начальной итерации все рёбра имеют число  $m=0$ , что делает равными шансы выбора любого ребра.

Данный метод сравнивался с методом муравьиных колоний и с поиском с табу. Эксперименты [22] показали, что результаты, полученные с помощью метода пчелиной колонии, почти не отличаются от результатов, полученных с помощью метода муравьиных колоний, и незначительно хуже результатов, полученных с помощью поиска с табу.

Lučić и Teodorović [23, 90] первыми использовали основные принципы коллективного интеллекта пчёл для решения задач комбинаторной оптимизации. Они разработали так называемый метод системы пчел (Bee System, BS) и проверили его при решении задачи коммивояжёра. На основе BS в [24] был предложен метаэвристический метод пчелиной колонии (Bee Colony Optimization Metaheuristic, BCO) и метод нечёткой пчелиной системы (Fuzzy Bee System, FBS).

В методе BCO в начале процесса поиска все агенты расположены в улье. В течение процесса поиска агенты связываются друг с другом косвенно. Каждый агент делает ряд локальных перемещений, и таким образом постепенно составляет решение задачи. Процесс поиска состоит из итераций. Первая итерация считается законченной, когда агенты создадут хотя бы одно допустимое решение. Лучшее решение сохраняется, а затем происходит переход к следующей итерации. Далее процесс составления решений повторяется. Общее количество итераций ограничивается исходя из задачи оптимизации.

При перемещении в пространстве поиска агенты могут следовать в прямом направлении или в обратном направлении. При перемещении в прямом направлении агенты формируют различные частные решения, что достигается за счёт индивидуального исследования пространства поиска и за счёт коллективного опыта, полученного на предыдущих итерациях.

После создания частного решения агенты перемещаются в обратном направлении, то есть они возвращаются в улей, где могут участвовать в процессе вербовки путём выполнения танца, тем самым обмениваясь информацией о различных созданных частных решениях. После посещения улья агенты опять следуют в прямом направлении и продолжают создавать частные решения. Итерация заканчивается тогда, когда создаётся хотя бы одно допустимое решение. Таким образом, BCO, как и методы динамического программирования, решает комбинаторные задачи оптимизации поэтапно.

Исходя из вышесказанного, можно отметить, что в BCO, в отличие от описанного ранее метода пчелиной колонии, нет разделения в ролях агентов (занятые фуражиры, незанятые фуражиры и разведчики), и реше-

ние составляется поэтапно, по мере нахождения частных решений, пока не будет получено полное допустимое решение.

Метод FBS предназначен для решения задач, характеризующихся неопределённостью, агенты при решении задачи используют правила нечёткой логики [91, 92] для организации связи между агентами и их действий.

В соответствии с FBS при добавлении компонента решения к частному решению агент может рассматривать компонент решения как: “менее полезный”, “полезный” или “более полезный”. Также агенты способны различать дополнительные свойства: “короткий”, “средний” или “длинный”, “неценный”, “средний” или “ценный”.

При выборе следующего компонента решения для определения полезности компонента используется правило следующего вида:

ЕСЛИ свойства компонента решения – “очень хорошие”, ТО рассматриваемый компонент решения – “очень полезный”.

Вероятность  $P_j$  того, что  $j$ -ый компонент будет добавлен к частному решению, рассчитывается по формуле:

$$P_j = \frac{f_j}{\sum_{k \in J^k} f_k},$$

где  $f_j$  – полезность  $j$ -го компонента решения;  $J^k$  – множество компонентов решения, которые могут быть добавлены.

В FBS для сравнения частных решений агентов используется концепция негодности частных решений [24]. Негодность частного решения определяется следующим образом:

$$L_k = \frac{L^{(k)} - L_{\min}}{L_{\max} - L_{\min}},$$

где  $L_k$  – негодность частного решения, полученного  $k$ -ым агентом;  $L^{(k)}$  – целевая функция, полученная с помощью частного решения  $k$ -ого агента;  $L_{\min}$  – целевая функция, полученная на основании лучшего частного решения, найденного с начала поиска;  $L_{\max}$  – целевая функция, полученная на основании худшего частного решения, найденного с начала поиска.

Определение пригодности частного решения основывается на приблизительных рассуждениях:

ЕСЛИ полученное частное решение – “плохое”, ТО его пригодность – “низкая”.

Таким образом, агенты используют приблизительные рассуждения для сравнения полученных частных решений с лучшим и с худшим частными решениями, найденными с начала процесса поиска. За счёт такого

подхода обеспечивается влияние полученных решений на направления поиска на будущих итерациях.

Поскольку пчёлы относятся к общественным насекомым, вероятность  $P^*$  того, что агент продолжит перемещение по тому же частному пути без выполнения вливающего танца, очень низка ( $P^* \ll 1$ ). Соответственно, вероятность того, что агент будет выполнять процесс вербовки, равна  $(1 - P^*)$ .

Для определения количества агентов, которые покинут найденные ими пути и присоединятся к другим агентам, также существуют определённые правила. Каждое частное решение, которое предлагается в области для танца, обладает двумя характеристиками: целевая функция и количество агентов, предлагающих это частное решение.

Для определения полезности предлагаемого частного решения используются правила следующего типа:

ЕСЛИ предлагаемый путь – “короткий” и количество предлагающих агентов “маленькое”, ТО полезность предлагаемого пути – “средняя”.

Таким образом, полезность предлагаемого пути может находиться в интервале  $[0, 1]$ .

Для определения количества агентов, меняющих свой путь, используются правила:

ЕСЛИ пригодность частного решения  $i$ -го агента – “низкая” и полезность частного решения  $j$ -го агента – “высокая”, ТО количество агентов, меняющих свой путь на путь  $j$ -го агента – “велико”.

Таким образом, используя коллективные знания и обмен информацией, агенты сосредотачивают поиск на наиболее перспективных направлениях поиска.

Метод FBS был применён к решению задачи о составлении пар [24]. Проведенные эксперименты [24] показали, что применение представленного метода характеризуется достаточной точностью и быстрой сходимостью.

## **12.4 Многомерная оптимизация на основе метода пчелиной колонии**

Одной из часто возникающих задач в процессе моделирования сложных объектов и систем является нахождение глобального оптимума многомерной функции. Для решения этой задачи применяется ряд методов (например, метод Коши, Ньютона, Левенберга-Марквардта и т.п.), которые, однако, требуют непрерывности, дифференцируемости и унимодаль-

ности целевых функций. Поэтому предлагается применять метод пчелиной колонии для решения задачи оптимизации многомерной функции.

Предложенный метод можно представить в виде следующих шагов.

Шаг 1. Задаются начальные параметры работы метода: начальное количество агентов-разведчиков  $B_s$ , максимальная энергия, коэффициент  $\alpha$ , начальная температура  $T_{init}$ , конечная температура  $T_{final}$ , максимальное количество итераций  $iter_{max}$ . Также задаются параметры, связанные с оптимизируемой функцией: количество переменных  $argCnt$ ; ссылка на саму функцию; минимальный  $Range^{min}$  и максимальный  $Range^{max}$  пределы для каждой из переменных, т.е. область, в которой можно искать возможные решения; направление оптимума  $optOrient$  (определяет максимизацию или минимизацию функции); в случае использования локальной оптимизации задаётся один из традиционных методов локальной оптимизации, который будет использоваться, и задаётся класс агентов, для решения которых будет применяться локальная оптимизация – агент с лучшим решением или агенты, выполнившие моделирование виляющего танца.

Шаг 2. Создаются начальные агенты-разведчики.

Шаг 2.1. Для каждого начального агента-разведчика создаётся случайное решение:

$$bee_i.x_j = rand \cdot (Range_j^{max} - Range_j^{min}) + Range_j^{min},$$

$$\forall i = \overline{1, B_s}, j = \overline{1, argCnt},$$

где  $bee_i$  –  $i$ -ый агент-разведчик;  $rand$  – случайное число в интервале  $[0; 1]$ .

Шаг 2.2. Для полученных случайных решений рассчитывается полезность данного источника нектара как значение оптимизируемой функции:

$$bee_i.profitability = optOrient \cdot f(bee_i.x_1, \dots, bee_i.x_{argCnt}), \forall i = \overline{1, B_s},$$

где  $OptOrient = 1$ , если выполняется максимизация, в противном случае –  $OptOrient = -1$ .

Текущее количество итераций устанавливается в 1:  $iter=1$ ; количество агентов-разведчиков  $B$  устанавливается в  $B_s$ :  $B=B_s$ ; текущая температура  $T$  устанавливается в  $T_{init}$ :  $T=T_{init}$ .

Шаг 3. Выбираются рабочие агенты, т.е. такие агенты, на базе которых будут создаваться новые агенты с помощью процедуры скрещивания.

Шаг 3.1. Определяется агент  $best$  с наибольшей полезностью.

Шаг 3.2. Процедура имитации отжига [29]. Агент относится к рабочим агентам  $workBee$ , если выполняется условие:

$$\exp\left(-\frac{|bee_i.profitability - best.profitability|}{T}\right) > rand, \forall i = \overline{1, B}.$$

Шаг 4. Скрещивание. Поскольку реальные пчёлы-разведчики при выборе источника нектара пользуются также генетическим материалом (в биологии ещё не изучено, каким именно образом разведчики выбирают одни цветки и пропускают другие, то есть предполагается, что разведчики основываются на генетическом опыте), то с помощью процедуры скрещивания моделируется именно этот момент поведения пчёл. Для скрещивания используются ранее отобранные с помощью процедуры имитации отжига рабочие агенты *workBee* и лучший агент за все итерации *best*. Новые агенты создаются в два этапа: на базе решений рабочих агентов и на базе решения лучшего агента.

Шаг 4.1. Создание новых агентов на базе рабочих агентов *workBee*:

$$newWorkBee_i.x_j = workBee_i.x_j \pm rand \cdot (workBee_i.x_j - best.x_j), \forall i = \overline{1, B_w},$$

где  $B_w$  – количество рабочих агентов *workBee*; знак преобразования “+” или “–” выбирается случайным образом.

Шаг 4.2. Создание новых агентов на базе лучшего агента *best*:

$$newWorkBee_i.x_j = best.x_j \pm rand \cdot (workBee_i.x_j - best.x_j).$$

Шаг 4.3. Для всех новых агентов производится корректировка полученных решений, поскольку полученные таким образом значения переменных могут выходить за пределы  $[Range^{min}; Range^{max}]$ .

Шаг 4.4. Рассчитывается полезность полученных решений:

$$newWorkBee_i.profitability = f(newWorkBee_i.x_1, \dots, newWorkBee_i.x_{argCnt}), \forall i = \overline{1, B_n}$$

где  $B_n$  – количество созданных при скрещивании агентов *newWorkBee*.

Шаг 4.5. Выбирается новый лучший агент *best*.

Шаг 5. Моделирование выполнения влияющего танца. К возможному выполнению танца допускаются рабочие агенты *workBee*, агенты, созданные путём скрещивания, *newWorkBee*, лучший агент за все итерации *best*. Моделирование выполнения влияющего танца происходит в несколько этапов. В результате данного моделирования выбираются те агенты, которые за счёт выполнения танца выполнят вербовку других агентов для исследования найденного ими решения.

Шаг 5.1. Выполняется нормирование полезностей агентов, допущенных к возможности выполнения танца. При этом нормирование учитывает направление оптимизации *optOrient*.

Шаг 5.2. Добавление шумов к полученным нормированным полезностям и их корректировка:

$$np_i = \begin{cases} 1, & \text{если } np_i + w_i > 1; \\ np_i + w_i, & \text{если } e_n < np_i + w_i < 1; \\ 0, & \text{если } 0 < np_i + w_i < e_n, \end{cases}$$

где  $np_i$  – нормированная полезность  $i$ -го агента;  $w_i$  – шум в полезности. Шум равномерно распределён в пределах  $(-w; +w)$ . Значение  $w$  выбирается экспериментально (предлагается  $w = 0,1$ ),  $e_n$  – минимальный порог полезности. Минимальный порог выбирается экспериментально (предлагается  $e_n = 0,1$ ).

Шаг 5.3. Определение преимущества танца каждого агента.

$$L_i = \max\{np_i - \eta \cdot \overline{np}, 0\},$$

где  $L_i$  – преимущество танца  $i$ -го агента;  $\eta$  – коэффициент, управляющий влиянием величины  $\overline{np}$  на  $L_i$ ;  $\overline{np}$  – среднее значение нормированной полезности всех агентов, которые допущены к моделированию выполнения танца:

$$\overline{np} = \frac{1}{B_c} \sum_{i=1}^{B_c} np_i,$$

где  $B_c$  – количество агентов, допущенных к выполнению танца.

Шаг 5.4. Выбор тех агентов, которые за счёт выполнения танца выполняют вербовку других агентов для исследования найденного ими решения. Агент считается выполнившим вербовку, если

$$\frac{L_i}{\beta} > \gamma \cdot \overline{np}, \forall i = \overline{1, B_c},$$

где  $\beta > 0$  – коэффициент, понижающий влияние достоинства танца на вероятность выполнения вербовки;  $\gamma \in [0; 1]$  – пороговой коэффициент, определяющий, насколько выше должно быть преимущество танца данного  $i$ -го агента относительно средней полезности решений всех  $B_c$  агентов.

Шаг 6. Если требуется проводить локальную оптимизацию для найденных лучших решений, тогда выполняется локальная оптимизация с помощью указанного метода локальной оптимизации (например, метод Нелдера-Мида, Хука-Дживса, Пауэлла и т.п.[28]). В зависимости от установленных параметров локальная оптимизация выполняется либо для решений тех агентов, которые произвели вербовку, либо только для решения лучшего агента.

Шаг 7. Выбирается агент с лучшим решением *best*.

Шаг 8. Перезапуск агентов. Создаются агенты, которые будут рассматриваться как агенты-разведчики для следующей итерации.

К новым агентам-разведчикам будут относиться:

- агенты, выполнившие посредством танца вербовку, лучший агент;

- агенты, которые стали занятыми фуражирами вследствие вербовки. Поскольку такие агенты должны выполнять улучшенное изучение уже существующего источника с нектаром, то при создании решений для данных агентов должны учитываться решения завербовавших агентов. В свя-



зи с этим для завербованных агентов решение создаётся следующим образом:

$$x_j = dancedBee.x_j + range \cdot rand - \frac{range}{2}, \forall j = \overline{1, argCnt},$$

где  $range$  – это предел, в которой величина переменной может отклоняться от значения данной  $j$ -ой переменной в решении агента, моделировавшего танец,  $dancedBee$ .

– агенты, решение которых создаётся случайным образом:

$$x_j = rand \cdot (Range_j^{\max} - Range_j^{\min}) + Range_j^{\min}, j = \overline{1, argCnt}.$$

Также для всех созданных агентов рассчитывается полезность выбранного решения.

Шаг 9. Обновление динамических параметров метода:

- увеличивается количество итераций:  $iter = iter + 1$ ;
- изменяется текущая температура:  $T = \alpha T$ ;
- изменяется предел  $range$ :

$$range = range \cdot \frac{iter_{\max} - iter}{iter_{\max}}.$$

Шаг 10. Проверка на останов. Проверка на останов считается успешной, если выполняется хотя бы одно из двух условий:

- достигнуто максимальное количество итераций:  $iter = iter_{\max}$ ;
- текущая температура равна конечной температуре:  $T = T_{final}$ .

Если проверка на останов дала успешный результат, то выполняется переход к шагу 11, в противном случае – к шагу 3.

Шаг 11. Останов.

## 12.5 Отбор информативных признаков на основе метода пчелиной колонии

В [25] описано применение метода пчелиной колонии для решения задач, основанных на распределении ресурсов (например, транспортной задачи). В соответствии, с предложенными в [25] математическими моделями поведения пчёл, разработана модификация метода пчелиной колонии для отбора информативных признаков при построении классифицирующих моделей в задачах диагностики и распознавания образов по признакам, состоящая из следующих шагов.

Шаг 1. Инициализация. Задаются основные параметры метода пчелиной колонии: количество агентов  $B$ , максимальное количество итераций

$T_{max}$ , начальное количество агентов-разведчиков  $Ex_{start}$ , ограничение максимального количества агентов-разведчиков  $Ex_{max}$ , пороговое значение полезности  $s_{min}$ . Также задаётся общее количество признаков  $M$  и количество признаков  $N$ , которое следует оставить. После чего создаётся пространство поиска порядка  $N \times M$ .

Шаг 2. Запуск разведчиков. Разведчики случайным образом размещаются в пространстве поиска. При этом выбранное расположение (точка в пространстве поиска) должно быть таким, чтобы не было одинаковых значений координат.

Шаг 3. Отправка занятых фуражиров. Занятые фуражиры прикреплены к определённым источникам ресурса. Начальное значение занятых фуражиров  $B_e = 0$ , поскольку в начале работы метода ещё нет источников ресурсов, за которым могут быть закреплены занятые фуражиры.

Полезность пребывания агента в источнике  $h$  на итерации  $t$ , при условии, что в этом источнике находится  $x_h$  агентов, рассчитывается по формуле:

$$s_h(t) = \frac{a_h}{x_h(t)}, h = \overline{1, N \times M},$$

где  $a_h$  – количество полезного вещества, вырабатываемое источником, в единицу времени. Количество полезного вещества  $a_h$  определяется после составления модели на основе положения соответствующего источника. В аспекте задачи отбора признаков количество полезного вещества  $a_h$  предлагается рассчитывать как обратное значение ошибки модели  $\varepsilon_h$ :

$$a_h = \frac{E}{\varepsilon_h},$$

где  $E$  – коэффициент, понижающий степень влияния ошибки  $\varepsilon_h$ .

Если полезность пребывания  $s_h(t)$  достигает порогового значения ( $s_h < s_{min}$ ), то агент помещается в близлежащую точку от точки  $h$  пространства поиска. Новое положение определяется путём изменения значения одной из координат текущего положения агента:

$$z_r = z_r + k \cdot \Delta z,$$

где  $z_r$  – координата, которая меняется;  $r$  – случайным образом выбранный номер координаты для изменения;  $k$  – коэффициент, определяющий направление изменения значения координаты, может быть равен  $+1$  или  $-1$ ;  $\Delta z$  – предел, в котором может изменяться переменная.

За счёт этого выполняется исследование области, в которой находится ранее выявленный источник ресурсов.

Шаг 4. Расчёт полезности полученного ресурса. Суммарная полезность фуражировки занятого фуражира или разведчика  $i$  рассчитывается по формуле:

$$F^i(t) = \begin{cases} 1, & \text{если } J_f(h^i(t)) + w_f^i(t) > 1; \\ J_f(h^i(t)) + w_f^i(t), & \text{если } e_n < J_f(h^i(t)) + w_f^i(t) < 1; \\ 0, & \text{если } 0 < J_f(h^i(t)) + w_f^i(t) < e_n, \end{cases}$$

где  $F^i(t)$  – полезность фуражировки  $i$ -го агента,  $w_f^i(t)$  – шум в суммарной полезности. Шум равномерно распределён в пределах  $(-w_f; +w_f)$ . Значение  $w_f$  выбирается экспериментально (предлагается  $w_f = 0,1$ ),  $e_n$  – минимальный порог полезности. Минимальный порог выбирается экспериментально (предлагается  $e_n = 0,1$ );  $J_f(h^i(t))$  – полезность источника  $h^i$ , в котором побывал  $i$ -ый агент на итерации  $t$ . Полезность источника  $h$  предлагается рассчитывать по формуле:

$$J_f(h) = \frac{\varepsilon^*}{\varepsilon_h},$$

где  $\varepsilon^*$  – заданная (требуемая) точность решения.

Полезность незанятых фуражиров и отдыхающих полагается 0:  $F^i(t)=0$ .

Шаг 5. Выбор лучшего результата и проверка, достигается ли заданная точность  $\varepsilon^*$ . Если точность достигается, то выполняется переход к шагу 9, в противном случае – переход к шагу 6.

Шаг 6. Моделирование выполнения танца, за счёт чего достигается обмен информацией. Каждый агент принимает решение выполнять или не выполнять танец. При этом вероятность выполнения виляющего танца  $i$ -ым агентом на итерации  $t$  рассчитывается по формуле:

$$p(i,t) = \frac{1}{\beta} L_f^i(t),$$

где  $\beta > 0$  – коэффициент, понижающий влияние преимущества пути на вероятность выполнения танца;  $L_f^i(t)$  – достоинство танца  $i$ -го агента на итерации  $t$ .  $L_f^i(t)$  рассчитывается по формуле:

$$L_f^i(t) = \max\{(F^i(t) - \alpha \bar{F}(t)), 0\},$$

где  $\bar{F}(t)$  – среднее значение полезности всех источников;  $\alpha$  – коэффициент, управляющий влиянием величины  $\bar{F}(t)$  на  $L_f^i(t)$ .

Шаг 7. Выделение новых разведчиков и вербовка. Каждый незанятый фуражир может стать разведчиком или последовать за другим агентом.

Вероятность того, что незанятый фуражир станет разведчиком, рассчитывается по формуле:

$$p_e(t) = \exp\left(-\frac{1}{2} \frac{L_t^2(t)}{\sigma^2}\right),$$

где  $\sigma$  – коэффициент, который необходим для моделирования поведения фуражировки;  $L_t(t)$  – сумма преимущества танцев разных агентов:

$$L_t(t) = \sum_{i=1}^B L_f^i(t).$$

Кроме того, незанятый фуражир может быть подвергнут вербовке, т.е. последовать за  $i$ -ым агентом. Вероятность того, что незанятый фуражир последует за  $i$ -ым агентом, предлагается рассчитывать по формуле:

$$p_i(t) = \frac{L_f^i(t)}{\sum_{j=1, j \neq i}^B L_f^j(t)}.$$

Шаг 8. Увеличивается счётчик итераций:  $t = t + 1$ . Если  $t < T_{max}$ , то выполнить переход к шагу 2, в противном случае – переход к шагу 9.

Шаг 9. Останов.

## 12.6 Особенности метода пчелиной колонии

Особенности рассмотренных модификаций метода пчелиной колонии и различия между ними представлены в таблице 12.1.

Таблица 12.1 – Различия и особенности модификаций метода пчелиной колонии

Критерий сравнения	BCO-JSSP	BCO	FBS
Процедура выполнения виляющего танца	Моделируется продолжительность выполнения танца	Выполняется в зависимости от качества составленного решения	Как таковой танца нет. Метод виляющего танца не применяется путём виляния вербовки
Выбор агентов-разведчиков	Начальное количество агентов-разведчиков равно общему количеству агентов. После вербовки количество разведчиков уменьшается	Разведчиками являются все незанятые фуражиры	Все незанятые фуражиры являются разведчиками

Особенности выбора решений разведчиком	Разведчики находят решения случайным образом	Разведчики создают случайные решения	Используется формация, оная предыдущими агентами
Использование методов локальной оптимизации	Не используются	Не используются	Не используются
Исследование решений за счёт других агентов	Осуществляется за счёт того, что вербовка осуществляется фактически в каждом узле	Завербованные агенты просто следуют по выбранному решению	Не происходит фуражируют по выбранному решению
Класс решаемых задач	Задача календарного планирования	Задачи комбинаторной оптимизации	Задача о сопоставлении
Вербовка	Осуществляется при выборе агентом следующего пункта в графе решений	Осуществляется после создания частного решения	Вербовка происходит исходя из решения

Исходя из различных применений метода пчелиной колонии можно выделить следующие преимущества метода:

- не склонен к заиканию в локальных оптимумах, поскольку основан на случайном поиске;
- поиск лучшего решения основывается на решениях агентов всей колонии пчёл;
- может применяться в динамических приложениях, поскольку способен адаптироваться к изменениям окружающей среды;
- мультиагентность реализации;
- может использоваться для решения как дискретных, так и непрерывных задач оптимизации.

К недостаткам метода пчелиной колонии можно отнести:

- достаточно высокую итеративность;
- трудности теоретического анализа процесса получения решений, обусловленные тем, что поиск решения имеет стохастическую природу;
- априорная неопределённость времени сходимости, хотя сходимость гарантируется;
- зависимость метода от настройки параметров, подбираемых экспериментально.

Метод пчелиной колонии обладает следующими особенностями:

1. Все агенты делятся на различные типы в соответствии с действиями, которые они выполняют в процессе решения задачи.

1.1 Занятые фуражиры обеспечивают использование уже найденных источников нектара, то есть незначительно изменяют уже найденные ранее решения задачи.

1.2 Незанятые фуражиры обеспечивают продолжение поиска новых источников нектара, то есть агенты такого типа выполняют поиск новых допустимых решений задачи. Незанятые фуражиры в свою очередь бывают двух типов.

1.2.1 Наблюдатели – ожидают в улье других агентов. Они не выполняют никаких действий, они фактически ожидают момента, когда им нужно будет также начать поиск решений;

1.2.2 Разведчики – обеспечивают поиск новых источников нектара. При этом поиск осуществляется случайным образом, то есть они случайно выбирают в пространстве поиска возможное решение.

2. Связь между решениями агентов осуществляется путём моделирования выполнения пчёлами виляющего танца. При этом выполнение виляющего танца обеспечивает образование двух типов обратной связи: позитивная обратная связь и негативная обратная связь.

Позитивная обратная связь заключается в том, что агенты, основываясь на информации о решениях других агентов, могут начать исследовать решение, полученное другим агентом.

Негативная обратная связь состоит в том, что агенты, получив информацию о найденных решениях другими агентами, могут принять решение о прекращении рассмотрения своего решения в связи с худшими характеристиками по сравнению с другими полученными решениями.

3. Процесс поиска решения обеспечивается двумя процедурами.

3.1. Поиск новых источников нектара во всём пространстве поиска, который достигается с помощью агентов-разведчиков. Таким образом, обеспечивается исследование всего пространства поиска.

3.2. Углублённое использование областей, в которых находятся уже найденные источники нектара (достигается с помощью занятых фуражиров). То есть находятся решения, находящиеся в пространстве поиска вблизи от рассматриваемого решения.

## РОЗДІЛ 13

### PSO–МЕТОД

Оптимизация с использованием роя частиц (Particle Swarm Optimization, PSO) – это метод поиска, который базируется на понятии популяции, и моделирует поведение птиц в стае и косяков рыб. Первоначальной целью использования концепции роя частиц была графическая имитация красивого и непредсказуемого движения птиц или рыб в стае, с целью обнаружения базовых принципов, благодаря которым птицы летают (рыбы плавают) синхронно и умеют менять направление движения с перегруппировкой в оптимальные формации. С тех пор концепция развилась в простой и перспективный оптимизационный метод [93].

#### 13.1 Фундаментальные принципы PSO-метода

В PSO-методе, особи, называемые частицами, перемещаются в многомерном пространстве решений. Изменения координат частиц внутри пространства поиска обуславливаются природной социально-психологической тенденцией частиц конкурировать между собой. Следовательно, изменения в состоянии частицы зависят от опыта и знаний её соседей. В данном случае, слово “знания” является синонимом “информации”. Результат такого моделирования заключается в том, что процесс поиска мотивирует частицы недетерминированным образом возвращаться в оптимальные участки пространства решений.

Особи в рое частиц обладают очень простым поведением: они стремятся превзойти достижения соседних частиц и улучшить собственные. Таким образом, эмергентное свойство данной системы состоит в исследовании оптимальных участков многомерного пространства поиска.

PSO метод управляет роем частиц и каждая частица представляет из себя потенциальное решение [94, 95]. По аналогии с эволюционными стратегиями, *рой* можно трактовать как популяцию, а *частицу* как индивида (или хромосому). То есть частица “летает” в многомерном пространстве решений, и её позиция определяется исходя из собственного опыта, и опыта своих соседей.

С помощью  $x_i(t)$  обозначается позиция частицы  $i$  в пространстве поиска решений в момент времени  $t$  ( $t$  определяет дискретные значения времени). Позиция частицы изменяется с добавлением скорости  $v_i(t)$  к текущей позиции:

$$x_i(t+1) = x_i(t) + v_i(t+1). \quad (13.1)$$

Начальное состояние определяется следующим образом  $x_i(0) = U(x_{\min}, x_{\max})$ , где  $U(a, b)$  является функцией генерации случайных чисел из диапазона  $[a, b]$ . Формула (13.1) представляет собой вектор скорости и определяет сам оптимизационный процесс, а также отражает использование, как опытных знаний частицы, так и обмен информацией с соседними частицами. Собственные знания самой частицы, называемые также *когнитивным компонентом* формулы скорости, прямо пропорциональны текущему расстоянию частицы от её наилучшего положения, которое было найдено с момента старта её жизненного цикла. А обмен информацией данной особи с другими является *социальным компонентом* формулы скорости.

Исторически было разработано два подхода, которые фактически являются разновидностями базового PSO-метода: *gbest* и *lbest*; они различаются степенью связанности частиц в пространстве поиска.

### 13.2 Подход gbest PSO

В разновидности *gbest* PSO-метода каждая частица связана со всем роем [96, 97]. Частицы образуют так называемую социальную сеть, которая в *gbest* соответствует топологии типа “звезда” (рис. 13.1). Каждая частица может взаимодействовать со всеми другими частицами, и она тяготеет к лучшему решению целого роя. Частица имитирует общее оптимальное решение, поэтому её скорость зависит от информации, получаемой от всех остальных. В данном случае, социальным компонентом скорости является наилучшая достигнутая позиция роя (в пространстве решений), и она обозначается как  $y^*(t)$ .

В *gbest* PSO-методе скорость частицы рассчитывается по формуле:

$$v_{ij}(t+1) = v_{ij}(t) + c_1 r_{1j}(t)[y_{ij}(t) - x_{ij}(t)] + c_2 r_{2j}(t)[y^*_j(t) - x_{ij}(t)], \quad (13.2)$$

где  $v_{ij}(t)$  – это скорость частицы  $i$  в измерение  $j = 1, \dots, n_x$  в момент времени  $t$ ;  $x_{ij}(t)$  – это позиция частицы  $i$  в измерение  $j$ ;  $c_1$  и  $c_2$  – положительные константы ускорения, которые используются для варьирования весов когнитивной и социальной компонент скорости частицы соответ-



ственно;  $r_{1j}(t), r_{2j}(t) = U(0,1)$  являются случайными значениями из диапазона  $[0,1]$ . Эти случайные величины привносят стохастический элемент в работу метода.

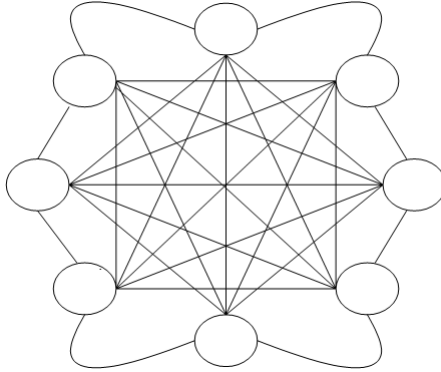


Рисунок 13.1 – Топология типа «звезда»

Величина  $y_i$  отображает наилучшую позицию частицы  $i$ , которую она посещала, начиная с первой итерации. Следующая оптимальная позиция частицы  $i$  в момент времени  $t + 1$  рассчитывается по формуле:

$$\begin{cases} y_i(t+1) = y_i(t), & \text{если } f(x_i(t+1)) \geq f(y_i(t)); \\ y_i(t+1) = x_i(t+1), & \text{если } f(x_i(t+1)) < f(y_i(t)), \end{cases} \quad (13.3)$$

где  $f : \mathcal{R}^{n_x} \rightarrow \mathcal{R}$  – фитнес-функция. Также, как и в эволюционных подходах, она является мерой близости данного решения к оптимальному, также фитнес-функция определяет производительность, или качество частицы (решения).

Глобальный наилучший оптимум  $y^*(t)$ , в момент времени  $t$ , определяется так:

$$y^*(t) \in \{y_0(t), \dots, y_{n_s}(t)\} \mid f(y^*(t)) = \min\{f(y_0(t)), \dots, f(y_{n_s}(t))\}, \quad (13.4)$$

где  $n_s$  – общее количество частиц в рое. Важно отметить, что согласно формуле (13.4),  $y^*$  – это наилучшая позиция, которая была найдена любой из частиц. Глобальный оптимум также может быть рассчитан на основе информации о частицах из данного роя:

$$y^*(t) = \min\{f(x_0(t)), \dots, f(x_{n_s}(t))\}. \quad (13.5)$$

Метод *gbest* PSO может быть описан в виде последовательности следующих шагов.

Шаг 1. Если условия окончания выполнены, то выход, в противном случае – переход на шаг 2.

Шаг 2. Создать и инициализировать  $n_x$ -мерный рой.

Шаг 3. Установить:  $i = 1$ .

Шаг 4. Определение локально наилучшей позиции. Если выполнено условие  $f(x_i) < f(y_i)$ , тогда установить  $y_i = x_i$ .

Шаг 5. Определение глобально наилучшей позиции. Установить:  $y^* = y_i$ .

Шаг 6. Установить:  $i = i + 1$ .

Шаг 7. Если  $i < n_s$ , то переход на шаг 3, в противном случае – на шаг 8.

Шаг 8. Установить:  $i = 1$ .

Шаг 9. Обновить скорость частицы, используя формулу (13.2).

Шаг 10. Обновить позицию частицы, используя формулу (13.1).

Шаг 11. Установить:  $i = i + 1$ .

Шаг 12. Если  $i < n_s$ , то выполнить переход на шаг 9, в противном случае на шаг 1.

### 13.3 Подход *lbest* PSO

В локально наилучшем PSO-методе, или *lbest* PSO, частицы образуют социально-сетевую топологию типа “кольцо” (рис. 13.2), и для каждой частицы задаётся меньшее количество соседей, чем при *gbest* подходе [97]. Каждая частица пытается имитировать лучшего из своих соседей, тяготея к нему. Социальный компонент скорости отражает обмен информацией между соседями частицы. Согласно формуле (13.1) вклад в скорость частицы пропорционален расстоянию между ней и наилучшим решением, найденным её соседями. Скорость рассчитывается по формуле

$$v_{ij}(t+1) = v_{ij}(t) + c_1 r_{1j}(t)[y_{ij}(t) - x_{ij}(t)] + c_2 r_{2j}(t)[y_{ij}^*(t) - x_{ij}(t)], \quad (13.6)$$

где  $y_{ij}^*(t)$  – наилучшая позиция, найденная соседями частицы  $i$  в измерение  $j$ . Локально наилучшая позиция частицы  $y_i^*$ , т.е. лучшая позиция, найденная в соседстве  $N_i$  и определяется как:

$$y_i^*(t+1) \in \{N_i \mid f(y_i^*(t+1))\} = \min\{f(x)\}, \forall x \in N_i\}, \quad (13.7)$$

где соседство определяется по следующей формуле:

$$N_i = \{y_{i-n_{N_i}}(t), y_{i-n_{N_i}+1}(t), \dots, y_{i-1}(t), y_i(t), y_{i+1}(t), \dots, y_{i+n_{N_i}}(t)\}, \quad (13.8)$$

для соседства размерностью  $n_{N_i}$ .

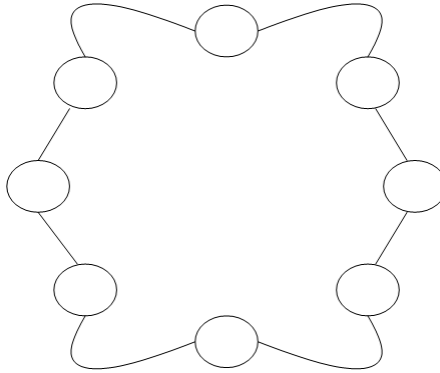


Рисунок 13.2 – Топология типа “кольцо”

Метод *lbest* почти эквивалентен *gbest*, с той лишь разницей, что на шаге 9 скорость частицы обновляется по формуле (13.6), а не (13.2), а связанные частицы определяются более сложным образом.

Шаг 1. Если условия окончания выполнены, то выход, в противном случае – переход на шаг 2.

Шаг 2. Создать и инициализировать  $n_x$ -мерный рой.

Шаг 3. Установить:  $i = 1$ .

Шаг 4. Определение локально наилучшей позиции. Если выполнено условие  $f(x_i) < f(y_i)$ , тогда установить  $y_i = x_i$ .

Шаг 5. Определение глобально наилучшей позиции. Установить:  $y^* = y_i$ .

Шаг 6. Установить:  $i = i + 1$ .

Шаг 7. Если  $i < n_s$ , то переход на шаг 3, в противном случае – на шаг 7.

Шаг 8. Установить:  $i = 1$ .

Шаг 9. Обновить скорость частицы, используя формулу (13.6).

Шаг 10. Обновить позицию частицы, используя формулу (13.1).

Шаг 11. Установить:  $i = i + 1$ .

Шаг 12. Если  $i < n_s$ , то выполнить переход на шаг 9, в противном случае на шаг 1.

Важно отметить, что в базовом варианте PSO-метода частицы в соседстве данной вообще не связаны. Выбор связанных частиц осуществляется на основе их индексов. Тем не менее, были разработаны стратегии, где соседние частицы определяются исходя их пространственного подобию.

Существует, по крайней мере, две причины, почему следуют отдать предпочтение индексам частиц в вопросе определения множества связанных частиц.

1. Это недорого в вычислительном плане, потому что не требуется пространственной нумерации частиц. В подходах, где соседство определяется на основе расчёта расстояний, требуется вычислять Евклидову дистанцию между всеми парами частиц, что имеет сложность около  $O(n_s^2)$ .

2. Данный подход обеспечивает информацией об оптимальных решениях все частицы, вне зависимости от их месторасположения в пространстве.

### 13.4 Компоненты скорости частицы в PSO-методе

Расчет скорости частицы выполняется по формулам (13.2) и (13.6). Рассмотрим компоненты скорости более детально [92, 93]:

1. *Предыдущая скорость*  $v_i(t)$  выступает в роли памяти частицы, т.е. содержит информацию об её перемещениях в прошлом. Эту память можно определить как импульс, который предотвращает скачкообразные изменения направления движения частицы. Данная скорость также является инерционным компонентом.

2. *Когнитивный компонент*  $c_1 r_1 (y_i - x_i)$  определяет производительность частицы  $i$  по отношению к прошлым результатам и выступает в роли индивидуальной памяти о наиболее оптимальных позициях данной частицы. Используя его, частица может возвращаться в состояния, которые были наилучшими для неё в прошлом. Это одно из эмергентных свойств системы в целом.

3. *Социальный компонент*  $c_2 r_2 (y^* - x_i)$  в *gbest* PSO или  $c_2 r_2 (y_i^* - x_i)$  в *lbest* PSO определяет производительность частицы по отношению к соседним или связанным с ней. Благодаря нему частица имеет возможность передвигаться в оптимальные позиции, которые были найдены соседними частицами.

Степень вклада когнитивного и социального компонентов определяется случайными величинами  $C_1 r_1$  и  $C_2 r_2$ , соответственно.

Эффект влияния скорости на поведение частицы может быть продемонстрирован графически [98]. Рассмотрим движение частицы в двумерном пространстве решений. Рисунок 13.3 а) отображает состояние единичного роя в момент времени  $t$ . В момент времени  $t + 1$ , как показано на рис. 13.3 б), индивидуальная наилучшая позиция частицы не изменилась, но остальные компоненты всё равно передвигают её по направлению лучшей частицы  $y^*(t)$ .

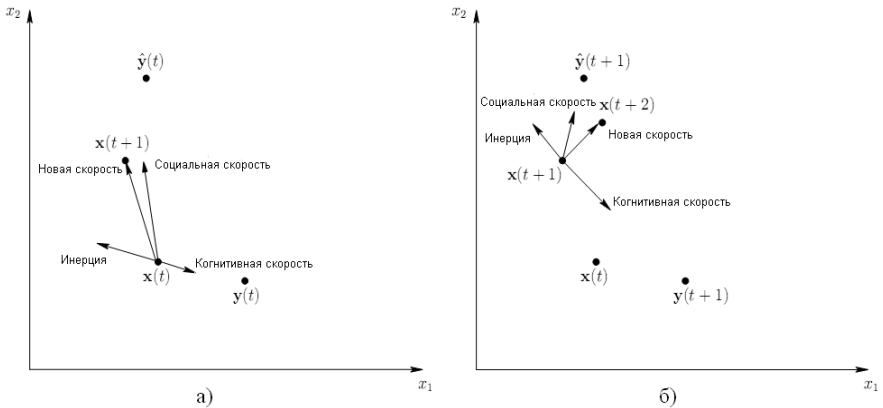


Рисунок 13.3 – Геометрическая иллюстрация изменения скорости и позиции частицы: а) момент времени  $t$ ; б) момент времени  $t + 1$ .

### 13.5 Инициализация и окончание PSO-метода

Согласно алгоритмам *gbest* и *lbest* оптимизационный процесс является итеративным и длится до тех пор, пока не будут выполнены условия окончания (хотя бы одно из них) [93, 94]. Каждая итерация заключается в выполнении всех шагов, начиная с первого, т.е. это определение индивидуальной лучшей позиции каждой частицы роя, глобального оптимума, и расчёт скоростей частиц. Внутри каждой итерации определённое количество раз происходит вызов фитнес-функции. В базовом варианте PSO-метода эта функция вызывается не более  $n_s$  раз, где  $n_s$  – это количество частиц в рое.

На первом шаге базового PSO-метода инициализируется рой и управляемые параметры, т.е. константы ускорения,  $c_1$  и  $c_2$ , начальные скорости, позиции частиц и их индивидуальные лучшие состояния. Алгоритм *lbest* также требует указания количества связей каждой частицы с соседними. Обычно, позиции частиц выбираются таким образом, чтобы равномерно покрывать всё пространство поиска. Также качество работы PSO-метода определяется начальной степенью разнообразия роя. Если некоторые локальные участки не покрываются частицами изначально, то с большой степенью вероятности процесс поиска может вообще не достигнуть этих подпространств. Данные участки могут быть исследованы, только если импульсы отдельных частиц перенесут их в эти участки.

Предположим, что оптимум должен быть найден в пространстве, ограниченном двумя векторами,  $x_{\min}$  и  $x_{\max}$ , которые соответственно снизу и сверху ограничивают диапазон значений каждого измерения. Тогда достаточно эффективный метод инициализации позиций частиц можно записать следующим образом (в данном случае  $r_j = U(0,1)$ ):

$$x(0) = x_{\min,j} + r_j(x_{\max,j} - x_{\min,j}), \forall j = 1, \dots, n_x, \forall i = 1, \dots, n_s. \quad (13.9)$$

Начальные скорости частиц можно принять нулевыми:

$$v_i(0) = 0. \quad (13.10)$$

Также возможно инициализировать скорости случайными значениями, но следует делать это с осторожностью. Фактически, согласно физической аналогии, нулевые скорости частиц говорят о том, что они стационарны, в противном случае – аналогия нарушается. Случайные значения скоростей означают, что алгоритм уже работает, и тогда частицы

имеют также случайные направления движения. Поэтому, даже если применяется подобный метод инициализации, не следует использовать слишком большие случайные числа, потому что частицы могут покинуть пространство поиска.

Индивидуальная наилучшая позиция каждой частицы при инициализации должна быть равна позиции данной частицы в момент времени  $t = 0$ , т.е.:

$$y_i(0) = x_i(0). \quad (13.11)$$

Окончание выполнения PSO-метода происходит, когда выполняется хотя бы один из критериев останова. Существуют два аспекта, которые должны быть учтены при выборе подобных мер:

1. Условие останова не должно обуславливать преждевременную сходимость PSO-метода. В противном случае будут получены субоптимальные решения.

2. При проверке критериев останова фитнес-функция должна вызываться минимальное количество раз, так как от этого сильно зависит скорость работы PSO-метода.

Обычно используют следующие критерии останова:

- достижение максимального количества итераций;
- нахождение удовлетворительного решения;
- выход при отсутствии улучшений на протяжении множества итераций;
- выход при условии, что первая производная фитнес-функции в окрестности частицы или частиц равна нулю.

### 13.6 Выбор параметров PSO-метода

Базовый PSO-метод зависит от выбора множества управляемых параметров таких как: количество частиц, коэффициенты ускорения, инерционные веса, количества связей между частицами, веса социальных и когнитивных компонентов [92, 93].

Рассмотрим влияние каждого из этих параметров по очереди:

Размер роя  $n_s$ . Очевидно, что чем больше частиц, тем большим разнообразием они обладают. Также объёмный рой способен более равномерно покрывать пространство поиска. Однако это также увеличивает вычислительную сложность на каждой итерации, снижая при этом их общее количество. Многие эмпирические исследования показали, что PSO-

метод может эффективно работать даже с малым количеством частиц: от 10 до 30 [99]. Но следует помнить, что гладкое, в отличие от неровного, пространство поиска требует меньшей плотности покрытия. Поэтому желательно величину  $n_s$  подбирать отдельно для каждой задачи.

Количество связей между частицами. Данная величина определяет степень социального взаимодействия частиц внутри роя. Чем она ниже, тем меньше кооперация. При малом количестве связей, PSO-метод сходится к глобальному решению более медленно, что предотвращает попадания в локальные оптимумы. Для того чтобы извлечь выгоды от этих двух полярных подходов, следует начинать поиск с малым количеством связей между частицами, а потом их количество равномерно увеличивать с ходом итераций.

Количество итераций. Эта величина, которая в конечном итоге определяет результат работы PSO-метода, и она также является проблемно-зависимой. Очевидно, что при малом количестве итераций работа метода может закончиться преждевременно, а при большом – безосновательно увеличится вычислительная сложность метода.

Коэффициенты ускорения,  $c_1$  и  $c_2$ . Вместе со случайными векторами  $r_1$  и  $r_2$ , они контролируют стохастическую степень влияния когнитивного и социального компонентов на общую скорость частицы. Коэффициенты  $c_1$  и  $c_2$  также называются доверительными весами. При  $c_1 = c_2 = 0$  частица передвигается до тех пор, пока не покинет пространство поиска. Если  $c_1 > 0$ , а  $c_2 = 0$ , то все частицы пытаются достичь оптимума вне зависимости друг от друга. Но успех PSO-метода обеспечивается только взаимной кооперацией частиц, поэтому важно, чтобы коэффициенты были сбалансированы, к примеру,  $c_1 \approx c_2$ . При малых значениях коэффициентов частицы перемещаются по гладким траекториям, при больших – наблюдается скачкообразное движение.

### 13.7 Многокритериальный PSO-метод

В настоящее время существует несколько модификаций базового PSO-метода, которые позволяют производить многокритериальный поиск.

В методе *динамического соседства* MOPSO (Multiple Objective Particle Swarm), который был предложен Хью и Эбернотом [100], динами-



чески определяются группы связанных частиц на каждой итерации, базируясь на их расстояниях в пространстве. Пусть  $f_1(x)$  будет первой целевой функцией, а  $f_2(x)$  – второй. Тогда, согласно методу, соседи данной частицы определяются как ближайшие в смысле значений целевой функции  $f_1(x)$ . А соседняя наилучшая частица определяется как ближайшая в смысле значений функции  $f_2(x)$ . Индивидуальная наилучшая позиция частицы изменяется только в том случае, если частица перешла в лучшее состояние.

Тем не менее, данный метод имеет несколько недостатков:

- метод может работать только с двумя целевыми функциями;
- метод предполагает использование априорных знаний (какая из двух функций является более простая);
- метод зависит от порядка следования целевых функций и отдаёт предпочтение второй при оптимизации.

Парсопулус и Врахатис разработали векторно-оценочный PSO-метод VEPSO (Vector Evaluation Particle Swarm Optimization) [101], который базируется на векторном генетическом алгоритме, разработанным Шафером. VEPSO использует две подпопуляции, где каждый рой оптимизирует одну целевую функцию. VEPSO представляет собой одну из разновидностей эволюционного подхода. Глобально наилучшее решение первого роя используется для расчёта скоростей второго роя, и наоборот. Таким образом:

$$\begin{aligned} S_1.v_{ij}(t+1) = & wS_1.v_{ij}(t) + c_1r_{1j}(t)(S_1.y_{ij}(t) - S_1.x_{ij}(t)) + \\ & + c_2r_{2j}(t)(S_2.y^*_i(t) - S_1.x_{ij}(t)), \end{aligned} \quad (13.12)$$

$$\begin{aligned} S_2.v_{ij}(t+1) = & wS_2.v_{ij}(t) + c_1r_{1j}(t)(S_2.y_{ij}(t) - S_2.x_{ij}(t)) + \\ & + c_2r_{2j}(t)(S_1.y^*_i(t) - S_2.x_{ij}(t)), \end{aligned}$$

где подпопуляция  $S_1$  производит вычисления на базе целевой функции  $f_1(x)$ , а  $S_2$  – на базе  $f_2(x)$ .

### 13.8 PSO-метод с управлением изменением скорости частиц

Хороший оптимизационный метод должен обладать двумя важными свойствами: всецело исследовать пространство решений задачи, и фиксировать поиск возле потенциальных оптимумов. Другими словами, алгоритм должен балансировать между двумя этими состояниями. В PSO-

методе баланс определяется формулой скорости частиц, а именно уравнениями (13.2) и (13.6) для  $gbest$  и  $lbest$  алгоритмов соответственно.

В ранних реализациях PSO-метода было обнаружено, что скорости могут лавинообразно возрастать, особенно скорости тех частиц, которые находятся далеко от своего собственного оптимума или общего оптимума своих соседей. В результате, такие частицы могут покидать пространство поиска, что исключительно отрицательно влияет на работу метода в целом. Для предотвращения подобных ситуаций следует контролировать диапазон изменения скоростей частиц. Если скорость частицы превышает максимально допустимую, то её следует снизить до разрешённого уровня. Пусть  $V_{\max, j}$  – максимально допустимая скорость частицы в измерение  $j$ . Тогда скорость частицы предлагается изменять следующим образом:

$$v_{ij}(t+1) = \begin{cases} \dot{v}_{ij}(t+1), & \text{если } v_{ij}(t+1) = V_{\max, j}, \\ V_{\max, j}, & \text{если } \dot{v}_{ij}(t+1) \geq V_{\max, j}, \end{cases} \quad (13.13)$$

где значение  $\dot{v}_{ij}$  рассчитывается по формуле (13.2) или (13.6).

Величина  $V_{\max, j}$  довольно важна, так как она контролирует общие масштабы поиска. Большие значения вынуждают PSO-метод исследовать пространство задачи более тщательно. Если значение  $V_{\max, j}$  достаточно мало, то PSO-метод может вообще не покинуть локальных участков, а также увеличивается количество итераций, нужных для достижения оптимума.

Исходя из вышесказанного понятно, что  $V_{\max, j}$  следует подбирать таким образом, чтобы балансировать между быстрым и медленным передвижениями частиц, и между фиксацией поиска и общим исследованием пространства. Предлагается рассчитывать  $V_{\max, j}$  следующим образом:

$$V_{\max, j} = \delta(x_{\max, j} - x_{\min, j}), \quad (13.14)$$

где  $x_{\max, j}$  и  $x_{\min, j}$  – величины, которые, соответственно, снизу и сверху ограничивают диапазон значений измерения  $j$ , а  $\delta \in (0, 1]$ . Очевидно, что величина  $\delta$  является проблемно-зависимой.

Отметим два важных свойства предложенной модификации PSO-метода:

1. Величина  $V_{\max,j}$  ограничивает не пространство поиска, внутри которого передвигаются частицы, а только скорости частиц (точнее диапазон их изменения за одну итерацию).

2. Максимальная скорость частиц определяется для каждого изменения в отдельности и обуславливается только его размерностью (формула (13.14)).

Несмотря на то, что в предложенной модификации скорости частиц контролируются с помощью величины  $V_{\max,j}$ , у данного метода есть и недостатки, которые надлежит рассмотреть. В первую очередь следует отметить, что ограничение скорости частицы также может вызывать изменение направления её передвижения. Этот эффект изображён на рис. 13.4. В данном случае,  $x_i(t+1)$  – это позиция частицы  $i$  без ограничений по скорости. А величина  $x'_i(t+1)$  уже отражает позицию частицы после того, как скорость стала равной  $V_{\max,j}$ . С другой стороны, такое «вмешательство» может направить частицы как раз по направлению к оптимуму.

Второй нежелательный эффект может возникнуть в случае, если скорости всех частиц станут равными  $V_{\max,j}$ . Для его предотвращения можно предложить с ходом итераций уменьшать значение  $V_{\max,j}$ . Тогда модифицированный PSO-метод будет последовательно сужать масштабы исследования пространства поиска, что делает оптимизационный процесс более эффективным.

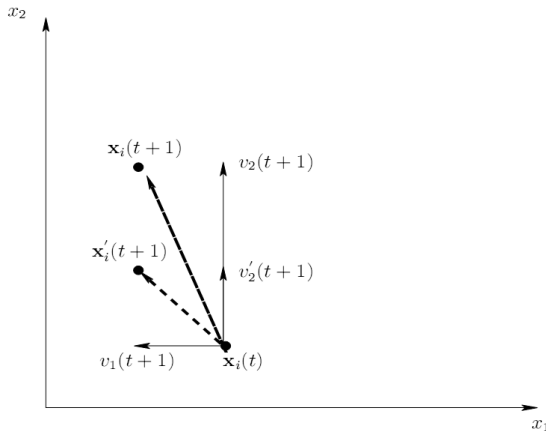


Рисунок 13.4 – Эффект использования ограничения скорости частицы

### 13.9 Сравнение разновидностей PSO-методов

Выполним сравнение разновидностей базового PSO-метода. Включим в анализ также модифицированный метод, который гибко регулирует скорости частиц в пространстве поиска. Сравнительный анализ проведём на основе наиболее важных и существенных различий, выявленных при их исследовании, а также на основе предложенных критериев. Анализ приведён в табл. 13.1.

Таблица 13.1 – Сравнительный анализ разновидностей PSO-метода

Критерии	Метод gbest PSO	Метод lbest PSO	PSO-метод с управлением изменением скорости частиц	Многокритериальный PSO-метод
Адаптивность	Высокая	Высокая	Высокая	Низкая
Масштабируемость	Средняя	Средняя	Средняя	Средняя
Гибкость модели	Средняя	Высокая	Высокая	Низкая
Скорость сходимости	Высокая	Низкая	Средняя	Сильно зависит от решаемой задачи
Качество оптимизации	Среднее	Высокое	Высокое	Низкое
Ошибкоустойчивость	Высокая	Высокая	Высокая	Низкая
Количество вычислений фитнес функции	Среднее	Низкое	Среднее	Высокое

Разновидности lbest и gbest базового PSO-метода подобны в том смысле, что в обоих случаях они двигаются к глобальному оптимуму, при обновлении компонентов скорости.

Но между этими подходами существует два важных различия, которые влияют на сходимость:

- из-за большей степени связности частиц между собой (топология «звезда») *gbest* PSO-метод сходится быстрее; однако быстрая сходимость ведёт к менее тщательному исследованию пространства решений;
- разновидность *lbest* PSO-метода имеет меньший шанс попасть в локальный минимум и найти, таким образом, только субоптимальное решение; с другой стороны, данный метод работает более медленно, чем *gbest*.

### 13.10 Приложения PSO-метода

Первые приложения PSO-метода следует отнести к обучению нейронных сетей (НС) прямого распространения [102]. С тех пор, в целом ряде работ [103–105] было показано, что PSO-метод также можно применять в качестве обучающего алгоритма НС самой различной архитектуры. В подобных приложениях следует определить нужный вид фитнес-функции, которая может быть представлена как функция ошибки, которую нужно минимизировать. Каждая частица в PSO-методе представляет собой отдельное решение оптимизационной проблемы, в данном случае – это целая нейронная сеть. А каждая координата частицы в пространстве поиска – это соответствующий весовой коэффициент.

Эберт и Кеннеди [105] предоставили первые результаты применения различных модификаций PSO-метода в задачах обучения НС, с учителем и без него. Мэндес [106] исследовал качество работы этих методов с использованием разнообразных топологий (кольцо, квадрат, звезда, пирамида).

Жанг и Шао [107] предложили использовать PSO-метод для непрерывной оптимизации весов НС и её архитектуры. Для этого следует использовать два роя частиц: первый ищет оптимальную архитектуру, а второй – оптимизирует веса. Частицы в первом рое передвигаются только в двух измерениях: первая координата – количество скрытых слоёв, а вторая – плотность связей. На первой итерации работы алгоритма частицы инициализируются случайным образом в заданных интервалах. Частицы второго роя ищут оптимальный набор весов НС. Данный PSO-метод для каждой частицы из первого роя, создаёт группу частиц во втором. После того как метод закончит свою работу, выбирается нужная архитектура и соответствующий ей оптимальное множество весов НС.

## РОЗДІЛ 14 МЕТОД ОПТИМИЗАЦИИ НА ОСНОВЕ МОДЕЛИРОВАНИЯ ПЕРЕМЕЩЕНИЯ БАКТЕРИЙ

### 14.1 Биологические основы

Одной из наиболее изученных бактерий является бактерия *E.Coli* [108–112]. Данная бактерия живёт в кишечнике большинства млекопитающих (в том числе и в кишечнике человека). Бактерия *E.Coli* при соответствующих условиях может самовоспроизводиться (расщепляться) за 20 мин. Способность двигаться бактерии *E.Coli* обеспечивается за счёт шести или более жгутиков, которые вращаются с частотой 100–200 оборотов в секунду, каждый из которых управляется с помощью своего собственного биологического мотора. Бактерия *E.Coli* чередует свои действия: то перемещается прямолинейно (со скоростью 10–20 мкм/сек.), то кувырывается (изменение направления). Когда жгутики вращаются по часовой стрелке (или против часовой стрелки), они работают как пропеллеры и, следовательно, *E.Coli* может плыть или кувыряться.

Хемотаксис – это двигательная реакция бактерии в ответ на появление в среде аттрактанта (аттрактант – вещество, привлекающее бактерии) или репеллента (репеллент – вещество, отпугивающее бактерий) [113, 114]. Понятно, что в естественных условиях аттрактантами являются вещества, полезные для бактерий, а репеллентами – те, которые бактериям вредны.

Примерная схема движения бактерии выглядит следующим образом: когда мотор работает в одну сторону, все жгутики у бактерии складываются и крутятся вместе, и бактерия движется прямолинейно. В конце пробега бактерия останавливается, мотор переключается и начинает работать в другую сторону. Жгутики растопыриваются и "бултыхаются" независимо друг от друга. Бактерия при этом переориентируется 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.

При наличии пространственных изменений концентрации аттрактантов или репеллентов частота кувыроканий, а, следовательно, и длина свободного пробега бактерии изменяются. Длина свободного пробега бактерии, плывущей в сторону возрастающей концентрации аттрактанта, увеличивается, а при движении в сторону возрастающей концентрации репеллента уменьшается.

Таким образом, можно выделить следующие хемотаксические действия бактерии *E. Coli* [115]:

- 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.

Таким образом, бактерия может перемещаться по полезным веществам и в то же время избегать опасных веществ. Датчики, используемые *E. Coli* – белковые рецепторы, которые являются очень чувствительными (незначительное изменение в концентрации полезных веществ может вызвать существенное изменение в поведении бактерии). От рецепторов сигналы поступают на метилакцептирующие белки, которые собирают все сигналы от рецепторов, и результирующий сигнал выходит на мотор жгутика, который управляет движением бактерии в зависимости от соотношения полезных и опасных веществ в окружающей среде.

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 устойчивые 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.

## **14.2 Метод оптимизации на основе 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 полезные и вредные 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 [116].

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,  $N_e$  – 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  $\varphi(j) \in R^p$ :

$$\varphi = \frac{\Delta}{\sqrt{\Delta^T \Delta}},$$

где  $\Delta$  – 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. Менее здоровые 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 (менее здоровые 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 постепенно (например, путём потребления полезных веществ), либо внезапно 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

щие значения целевой функции  $J^i$ ,  $i = 1, 2, \dots, S$ . Счётчики циклов сбрасываются в 0:  $j = 0$ ;  $k = 0$ ;  $l = 0$ .

Шаг 14. Останов.

Общая схема работы метода оптимизации на основе моделирования перемещения бактерий представлена на рис. 14.1.

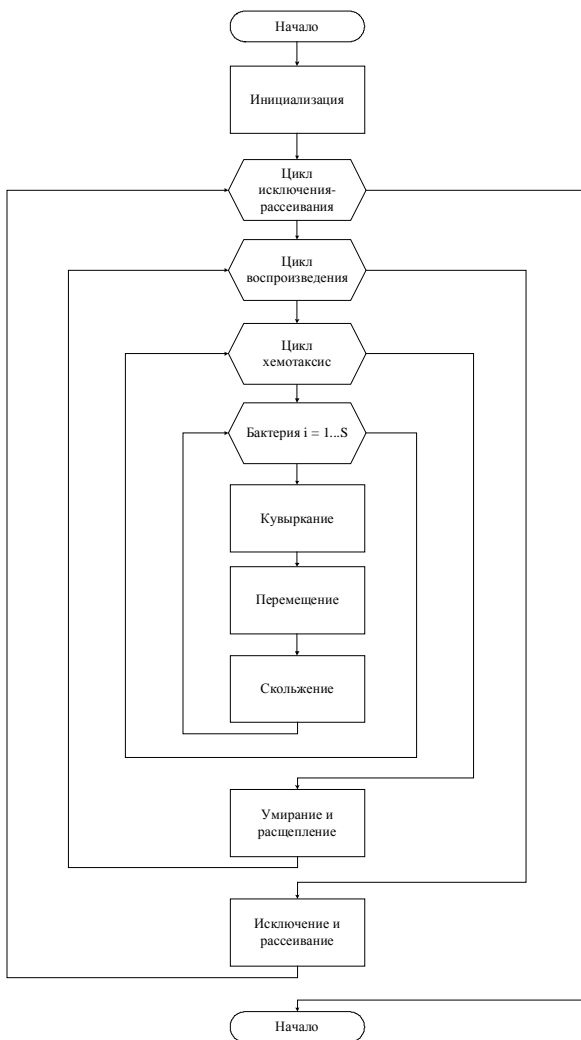


Рисунок 14.1 – Общая схема работы метода фуражировки бактерий

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 в дальнейшем было предложено несколько подходов, позволяющих учитывать в процессе моделирования перемещения бактерий и данную особенность их поведения, основными из которых являются: моделирование сигналов между клетками [116] и применение PSO-оператора (Particle Swarm Optimization, PSO) [117].

### 14.3 Метод оптимизации на основе моделирования перемещения бактерий с группировкой за счёт связи между клетками

Пусть  $d_{\text{attract}}$  – глубина аттрактанта (полезных веществ) клетки, и  $w_{\text{attract}}$  – мера ширины аттрактанта. Клетки исключают друг друга посредством локального потребления, а также за счёт того, что клетки не являются пищей друг для друга. Пусть  $h_{\text{repellent}}=d_{\text{attract}}$  – высота вредного вещества (репеллент), и  $w_{\text{repellent}}$  – мера ширины репеллента. Таким образом, можно использовать функции  $J_{cc}^i(X)$ ,  $i=1, 2, \dots, S$  для моделирования сигналов между клетками посредством выделения бактериями аттрактанта и репеллента:

$$J_{cc}(X) = \sum_{i=1}^S J_{cc}^i = \sum_{i=1}^S \left[ -d_{\text{attract}} \exp \left( -w_{\text{attract}} \sum_{j=1}^p (X_j - X_j^i)^2 \right) \right] + \\ + \sum_{i=1}^S \left[ h_{\text{repellent}} \exp \left( -w_{\text{repellent}} \sum_{j=1}^p (X_j - X_j^i)^2 \right) \right],$$

где  $X=[x_1, \dots, x_p]^T$  – точка в пространстве оптимизации.

Важными особенностями такого подхода к расчёту  $J_{cc}(X)$  являются:

- значение  $J_{cc}(X)$  не зависит от значения целевой функции в точке  $\theta$ ;
- за счёт варьирования коэффициентов  $w_{\text{attract}}$  и  $w_{\text{repellent}}$  можно влиять на поведение популяции бактерий в процессе поиска: при малых значениях  $w_{\text{attract}}$  и больших значениях  $w_{\text{repellent}}$  популяция склонна к большому рассеиванию в пространстве поиска, в обратном случае, то есть при малых значениях  $w_{\text{attract}}$  и больших значениях  $w_{\text{repellent}}$  популяция склонна к группировке в небольших областях поиска.

Очевидно, что сила выделения химических веществ бактериями зависит от окружающей среды, т.е. бактерия, находящаяся в клетке с высо-

кой концентрацией полезных веществ будет выделять более сильный аттрактант, чем та же бактерия в ячейке с низкой концентрацией полезных веществ. Поэтому в данном методе используется функция  $J_{ar}(\theta)$  для моделирования взаимодействия между ячейками с учётом особенностей окружающей среды:  $J_{ar}(X) = \exp(M - J(X))J_{cc}(X)$ , где  $M$  – настраиваемый параметр. Следовательно, для поиска оптимума необходимо минимизировать выражение  $(J(i, j, k, l) + J_{ar}(X^i(j, k, l)))$ , что позволяет бактериям искать полезные вещества, избегая областей с вредными веществами, и в то же время приближаясь к другим ячейкам, но не слишком близко к ним.

Таким образом, метод оптимизации на основе моделирования перемещения бактерий с группировкой за счёт связи между ячейками отличается от базового метода тем, что минимизируется не просто функция  $J(i, j, k, l)$ , а минимизируется сумма:  $J(i, j, k, l) + J_{ar}(X^i(j, k, l))$ , что позволяет учитывать положения всех бактерий в популяции.

Функция  $J_{ar}(X^i(j, k, l))$  обеспечивает следующую зависимость: чем меньше  $J(X)$ , тем больше  $J_{ar}(X)$ , и тогда большей является привлекательность, зависящая от  $J_{cc}(X)$ . Настраиваемый параметр  $M$  обычно выбирается из рассуждений, что если  $M$  достаточно большое, то  $J_{ar}(X)$  значительно больше, чем  $J(X)$  и, таким образом, доминирующим направлением в пространстве поиска является химический аттрактант, выделяемый *E. Coli*. С другой стороны, если  $M$  является очень малым, то  $J_{ar}(X)$  значительно меньше, чем  $J(X)$  и, таким образом, доминируют полезные вещества окружающей среды. Для  $J_{ar}(X)$  выбрана экспоненциальная зависимость, но она не является единственно возможным выбором. Любые другие функции, которые монотонно убывают и асимптотически приближаются к нулю – подходящие кандидаты, хотя могут быть необходимы и некоторые дополнительные ограничения.

Метод оптимизации на основе моделирования перемещения бактерий с группировкой за счёт связи между ячейками может быть описан в виде последовательности следующих шагов.

Шаг 1. Инициализация. Задаются параметры, влияющие на работу метода:  $S$ ,  $N_{re}$ ,  $N_s$ ,  $N_c$ ,  $N_{ed}$ ,  $P_{ed}$ . Случайным образом распределить начальные значения  $X^i$ ,  $i = 1, 2, \dots, S$  по пространству поиска. Рассчитываются начальные значения целевой функции для каждой бактерии  $J^i$ . Для каждой бактерии выполняется пересчёт:  $J^i = J^i + J_{ar}(X^i)$ .

Шаг 2. Установить:  $l = l + 1$ .

Шаг 3. Установить:  $k = k + 1$ .

Шаг 4. Установить:  $j = j + 1$ .

Шаг 5. Моделирование хемотаксиса.

Шаг 5.1. Установить:  $i = i + 1$ .

Шаг 5.2. Кувыркание и перемещение. Рассчитывается новое положение  $i$ -ой бактерии по формуле:

$$X^i(j+1, k, l) = X^i(j, k, l) + C(i) \frac{\Delta}{\sqrt{\Delta^T \Delta}},$$

Рассчитать значение целевой функции  $J(i, j+1, k, l)$ .

Установить:  $J(i, j+1, k, l) = J(i, j+1, k, l) + J_{ar}(X^i(j+1, k, l))$ .

Если  $J(i, j+1, k, l) < J_{best}$ , тогда установить:  $J_{best} = J(i, j+1, k, l)$ ,  $X_{best} = X^i(j+1, k, l)$ .

Шаг 5.3. Скольжение. Пока выполняется условие:  $J(i, j+1, k, l) < J(i, j, k, l)$ , повторяется шаг 5.2. Такое повторение может повторяться  $N_s$  раз. Если условие не выполняется, то переход к шагу 5.4.

Шаг 5.4 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. Все агенты сортируются в соответствии с полученными значениями целевой функции, после чего худшая половина отбрасывается, а лучшая – дублируется.

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

Шаг 9. Исключение и рассеивание. Проверяется условие:  $U^i < P_{ed}$ .

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  $X_{best}$ .

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  $X^i$ ,  $i = 1, 2, \dots, S$ , 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. Останов.

#### 14.4 Метод оптимизации на основе моделирования перемещения бактерий с группировкой за счёт использования PSO-оператора

PSO-метод основан на группировке частиц, поэтому логично было применить подходы, используемые в PSO, к методу оптимизации на основе моделирования перемещения бактерий, что и было предложено в [117].

Как известно, в PSO-методе связь между частицами достигается за счёт того, что скорость перемещения частицы зависит от решений, полученных другими частицами, входящими в популяцию. Поэтому в [117] предложено изменять положение бактерии в пространстве поиска в соответствии с PSO-оператором. Такое изменение выполняется в рамках цикла хемотаксиса, после моделирования скольжения бактериями.

Пусть  $V^i = (v_1^i, v_2^i, \dots, v_p^i)$  – вектор скоростей  $i$ -ой бактерии в измерениях  $1, 2, \dots, p$ , а  $X^i = (x_1^i, x_2^i, \dots, x_p^i)$  – позиция  $i$ -ой бактерии в пространстве поиска. Тогда в соответствии с PSO-методом скорость и позиция  $i$ -ой бактерии на  $j$ -ом хемотаксическом шаге,  $k$ -ом шаге воспроизведения и на  $l$ -ом событии исключения-рассеивания изменяется следующим образом:

$$\begin{aligned} V^i(j+1, k, l) &= \omega \cdot V^i(j, k, l) + C_1 \cdot \varphi_1 \cdot (X_{l\_best}^i - X^i(j, k, l)) + \\ &+ C_2 \cdot \varphi_2 \cdot (X_{g\_best} - X^i(j, k, l)), \\ X^i(j+1, k, l) &= X^i(j, k, l) + V^i(j+1, k, l), \end{aligned}$$

где  $\omega$  – коэффициент инерции,  $C_1, C_2$  – положительные константы ускорения, которые используются для варьирования весов когнитивной и социальной компонент скорости частицы соответственно;  $\varphi_1, \varphi_2 = rand(0,1)$  являются случайными значениями из диапазона  $[0,1]$ ;  $X_{g\_best}$  – глобально лучшая позиция, которая была выбрана из всех решений агентов, начиная с первой итерации поиска;  $X_{l\_best}^i$  – наилучшая позиция  $i$ -ой бактерии от начала поиска.

Метод оптимизации на основе моделирования перемещения бактерий с группировкой за счёт использования PSO-оператора может быть описан в виде последовательности следующих шагов.

Шаг 1. Инициализация. Задаются параметры, влияющие на работу метода:  $S, N_{re}, N_s, N_c, N_{ed}, P_{ed}, \omega, C_1, C_2$ . Случайным образом распределить начальные значения  $X^i, i = 1, 2, \dots, S$  по пространству поиска, задать нулевые начальные скорости бактерий:  $v_j^i = 0, i = 1, 2, \dots, S, j = 1, 2, \dots, p$ . Рассчитываются начальные значения целевой функции для каждой бактерии  $J^i$ . Для каждой бактерии устанавливается  $X_{l\_best}^i = X^i, J_{l\_best}^i = J^i$ , а также выбираются  $J_{g\_best}$  и  $X_{g\_best}$ .

Шаг 2. Установить:  $l = l + 1$ .

Шаг 3. Установить:  $k = k + 1$ .

Шаг 4. Установить:  $j = j + 1$ .

Шаг 5. Моделирование хемотаксиса.

Шаг 5.1. Установить:  $i = i + 1$ .

Шаг 5.2. Кувыркание и перемещение. Рассчитывается новое положение  $i$ -ой бактерии по формуле:

$$X^i(j+1, k, l) = X^i(j, k, l) + C(i) \frac{\Delta}{\sqrt{\Delta^T \Delta}}.$$

Рассчитать значение целевой функции  $J(i, j+1, k, l)$ . Если  $J(i, j+1, k, l) < J_{l\_best}^i$ , то установить:  $J_{l\_best}^i = J(i, j+1, k, l)$ ,  $X_{l\_best}^i = X^i(j+1, k, l)$ .

Шаг 5.3. Скольжение. Пока выполняется условие:  $J(i, j+1, k, l) < J(i, j, k, l)$ , повторяется шаг 5.2. Такое повторение может повторяться  $N_s$  раз. Если условие не выполняется, то переход к шагу 5.4.

Шаг 5.4 Если  $i < S$ , то переход к шагу 5.1, в противном случае – переход к шагу 6.

Шаг 6. Применение PSO-оператора. Для каждой бактерии рассчитывается скорость и новая позиция:

$$\begin{aligned} \forall i = \overline{1, S}: V^i(j+1, k, l) &= \omega \cdot V^i(j, k, l) + C_1 \cdot \varphi_1 \cdot (X_{l\_best}^i - X^i(j, k, l)) + \\ &+ C_2 \cdot \varphi_2 \cdot (X_{g\_best} - X^i(j, k, l)), \\ X^i(j+1, k, l) &= X^i(j, k, l) + V^i(j+1, k, l). \end{aligned}$$

Шаг 7. Если  $j < N_c$ , то переход к шагу 4, в противном случае – переход к шагу 8.

Шаг 8. Воспроизведение. Все агенты сортируются в соответствии с полученными значениями целевой функции, после чего худшая половина отбрасывается, а лучшая – дублируется.

Шаг 9. Если  $k < N_{re}$ , то выполняется переход к шагу 3, в противном случае – переход к шагу 10.

Шаг 10. Исклучение и рассеивание. 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 11. Если  $l < N_{ed}$ , то выполняется переход к шагу 2, в противном случае – к шагу 12.

Шаг 12. Выбирается и сохраняется лучшее решение  $J_{g\_best}$  и соответствующая позиция  $X_{g\_best}$ .

Шаг 13. Проверка на окончание поиска. Если были выполнены все циклы для всех агентов, то выполняется переход к шагу 15, в противном случае выполняется перезапуск – переход к шагу 14.

Шаг 14. Перезапуск агентов: выбираются новые случайные позиции для каждого агента  $X^i$ ,  $i = 1, 2, \dots, S$ , обнуляется вектор скоростей  $V^i$ , рассчитываются соответствующие значения целевой функции  $J^i$ ,  $i = 1, 2, \dots, S$ . Счётчики циклов сбрасываются в 0:  $j = 0$ ;  $k = 0$ ;  $l = 0$ .

Шаг 15. Останов.

Рассмотренный метод оптимизации на основе моделирования перемещения бактерий и его разновидности с группировкой, основанной на различных подходах, применялись для нахождения глобального оптимума многомерных функций.

Эксперименты, проведенные в [116, 117], показали, что результаты метода без группировки и его разновидностей с группировкой являются идентичными по величине найденных оптимальных значений целевой функции, однако скорость сходимости к оптимальному решению лучше у разновидностей, которые применяют группировку.

#### **14.5 Приложения и соответствующие модификации метода оптимизации на основе моделирования перемещения бактерий**

Поскольку метод оптимизации на основе моделирования перемещения бактерий проявил себя с хорошей стороны при решении задач оптимизации многомерных функций, то его начали применять для решения различных оптимизационных задач, разрабатывая при этом соответствующие модификации, учитывающие специфику той или иной задачи.

Метод оптимизации на основе моделирования перемещения бактерий применялся к следующим задачам.

1. Нахождение оптимального потока энергии [118].

Для нахождения оптимального потока энергии предложена модификация метода оптимизации с помощью моделирования перемещения бактерий с группировкой за счёт связи между ячейками, позволяющая ускорить сходимость.

Основные особенности предложенной модификации следующие:

- сортировка бактерий производится на основе минимальных значений целевой функции, которые были получены на данном шаге хемотаксиса для соответствующей бактерии. За счёт такого подхода бактерии с лучшими решениями переходят в новое поколение, что способствует более быстрой сходимости к глобально лучшему решению;
- для группировки рассчитывается расстояние не между всеми бактериями, а рассчитывается расстояние от каждой бактерии до текущего лучшего решения, которое было найдено от начала работы метода.



Проведенные эксперименты [118] показали, что предложенная модификация обеспечивает более высокую точность при решении задачи нахождения оптимального потока энергии по сравнению с традиционным методом IPSLP (Interior Point Successive Linear Programming).

## 2. Анализ независимых компонент [119].

Анализ независимых компонент (Independent Component Analysis, ICA) – статистический метод, цель которого заключается в линейном представлении данных, распределённых по негауссовскому закону, и таких, что компоненты независимы или независимы настолько, насколько это возможно. Существующие методы ICA застревают в локальных оптимумах, поэтому не находят глобальный оптимум. В связи с этим для анализа независимых компонент применяются методы, основанные на случайном подходе, например, довольно успешным было применение для данной задачи генетических алгоритмов [120]. Поскольку метод оптимизации на основе моделирования перемещения бактерий также является методом случайного поиска, то в работе [119] он был применён к анализу независимых компонент.

В [119] предлагается использовать эксцесс в качестве функции различия, которую следует максимизировать и, следовательно, в соответствии с методом BFO выполняется минимизация функции концентрации полезных веществ:

$$J = \frac{1}{\text{ContrastFunction}},$$

где *ContrastFunction* – функция различия.

Для нахождения минимума данной функции использовался базовый метод оптимизации на основе моделирования перемещения бактерий.

Результаты проведенных экспериментов [119] показали, что результаты, полученные с помощью BFO обладали более высокой точностью по сравнению с разработанными ранее подходами fast ICA и CGAICA (constrained genetic algorithm based ICA).

## 3. Настройка пропорционально-интегрально-дифференциального (ПИД) контроллера [121].

Несмотря на новые достижения в теории управления, пропорционально-интегрально-дифференциальный (ПИД) контроллеры продолжают широко применяться во многих промышленных процессах. Большинство подходов к настройке ПИД, разработанные за прошлые годы, используют традиционные методы, например, метод частотных характеристик [122]. Такие методы требуют большого количества технических экспериментов, что является трудоёмкой и дорогостоящей задачей. Поэтому в [121] для настройки параметров ПИД контроллера предлагается использовать метод оптимизации на основе моделирования перемещения бактерий.

В качестве обратной меры КПД предлагается использовать интеграл от взвешенного по времени квадрата ошибки:

$$PI = \int_0^{\infty} t(E(t))^2 dt ,$$

где  $E(t)$  – ошибка работы контроллера.  $E(t)$  зависит от параметров работы контроллера ( $p$ ). Тогда для наилучшей настройки ПИД контроллера надо найти такие значения параметров работы контроллера  $p$ , при которых будет достигаться минимум  $PI$ :  $\min_p PI(p)$ .

Поиск минимума  $PI$  осуществлялся методом оптимизации с помощью моделирования перемещения бактерий с группировкой за счёт связи между ячейками.

Полученные результаты сравнивались с результатами, полученными путём применения других методов, основанных на случайном подходе: генетические алгоритмы и метод моделирования иммунных систем. Результаты всех методов обладали достаточной точностью и незначительно отличались друг от друга и, таким образом, подтвердили эффективность применения метода оптимизации с помощью моделирования перемещения бактерий для настройки ПИД контроллера.

#### 4. Задача динамического распределения ресурсов [123].

Базовый метод BFO применялся в [123] для решения задачи динамического распределения ресурсов и тестировался с помощью экспериментальной платформы, которая характеризовалась различной температурой в разных областях платформы. Экспериментальная платформа состояла из двух частей: плата (состоящая из различных конструкторских элементов) и уровень получения данных, который снимает данные о температуре в различных областях платы.

Таким образом, моделировалось перемещение бактерий по экспериментальной платформе, и выполнялся поиск областей с наименьшей температурой.

Работа метода BFO сравнивалась с традиционными методами, а также с методом муравьиных колоний. Полученные результаты [123] показали, что метод BFO находит оптимальные решения быстрее по сравнению с другими методами.

Область применения метода BFO не ограничивается приведенными задачами. Описаны только основные и уже хорошо известные направления, в которых нашёл применение данный метод. Однако он используется и при решении других оптимизационных задач [124–126].

### 14.6 Метод оптимизации на основе моделирования перемещения бактерий для отбора информативных признаков

Задача отбора информативных признаков заключается в выборе набора признаков, которые наиболее полно характеризуют изучаемый сложный объект или процесс.

Для решения задачи отбора информативных признаков на основе метода моделирования перемещения бактерий агенты, моделирующие поведение бактерий, должны осуществлять оптимизацию в пространстве поиска  $X = (x_1, x_2, \dots, x_N)$ ,  $x_i = \{0, 1\}$ , где  $N$  – общее количество признаков, характеризующих выборку. Таким образом, каждая точка пространства поиска представляет собой битовую строку, координаты которой позволяют определить, считается ли признак информативным: информативным считается признак, если соответствующая координата равна 1, в противном случае – признак считается неинформативным. В качестве целевой функции  $J(X)$  предлагается использовать ошибку модели, построенной на основе признаков, которые являются информативными для данной точки пространства поиска.

Для выполнения перемещения по такому пространству поиска величины, определяющие процесс движения в пространстве поиска должны принимать следующие значения:

- переменные, определяющие размер шага в определённом направлении, для  $i$ -ой бактерии:  $C(i) \in (0; 1)$ ;
- вектор случайных чисел на  $j$ -ом шаге хемотаксиса:  $\varphi(j) = \{-1, 0, +1\}$ .

Пример перемещения по пространству поиска из трёх признаков, если агент в начале поиска находился в точке  $(0; 0; 0)$ , вектор размера шага для данной бактерии  $C = (1; 1; 1)$ , вектор случайных чисел на 1-ом шаге хемотаксиса  $\varphi(1) = (1; 0; +1)$ , а на 2-ом шаге  $\varphi(1) = (-1; +1; 0)$ , приведен на рис. 14.2.

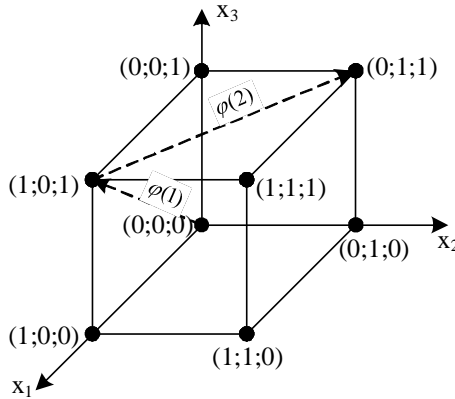


Рисунок 14.2 – Пример перемещения в пространстве поиска из трёх признаков

Тогда метод оптимизации на основе моделирования перемещения бактерий для отбора информативных признаков может быть представлен в виде последовательности следующих шагов.

Шаг 1. Инициализация. Задаются параметры, влияющие на работу метода:  $S$ ,  $N_{re}$ ,  $N_s$ ,  $N_c$ ,  $N_{ed}$ ,  $P_{ed}$ . Случайным образом распределить начальные значения  $X^i$ ,  $i = 1, 2, \dots, S$  по пространству поиска. Рассчитываются начальные значения целевой функции для каждой бактерии  $J^i$ . Для каждой бактерии выполняется пересчёт:  $J^i = J^i + J_{ar}(X^i)$ .

Шаг 2. Установить:  $l = l + 1$ .

Шаг 3. Установить:  $k = k + 1$ .

Шаг 4. Установить:  $j = j + 1$ .

Шаг 5. Моделирование хемотаксиса.

Шаг 5.1. Установить:  $i = i + 1$ .

Шаг 5.2. Кувыркание и перемещение.

Создаётся вектор  $C(i) = \text{rand}(0; 1)$ , где  $\text{rand}(0; 1)$  – случайное действительное число в диапазоне  $[0; 1]$ .

Генерация вектора случайных чисел  $\phi(j) = \text{rand\_int}(-1; +1)$ , где  $\text{rand\_int}(-1; +1)$  – случайное целое число в диапазоне  $[-1; +1]$ .

Рассчитывается новое положение  $i$ -ой бактерии по формуле:

$$X^i(j+1, k, l) = \begin{cases} 1, X^i(j, k, l) + C(i) \cdot \phi(j) \geq 1; \\ \text{round}(X^i(j, k, l) + C(i) \cdot \phi(j)), 0 < X^i(j, k, l) + C(i) \cdot \phi(j) < 1; \\ 0, X^i(j, k, l) + C(i) \cdot \phi(j) \leq 0. \end{cases}$$

Рассчитать значение целевой функции  $J(i, j+1, k, l)$ .

Установить:  $J(i, j+1, k, l) = J(i, j+1, k, l) + J_{ar}(X^i(j+1, k, l))$ .

Если  $J(i, j+1, k, l) < J_{best}$ , то установить:  $J_{best} = J(i, j+1, k, l)$ ,  $X_{best} = X^i(j+1, k, l)$ .

Шаг 5.3. Скольжение. Пока выполняется условие:  $J(i, j+1, k, l) < J(i, j, k, l)$ , повторяется шаг 5.2. Такое повторение может повторяться  $N_s$  раз. Если условие не выполняется, то переход к шагу 5.4.

Шаг 5.4 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. Все агенты сортируются в соответствии с полученными значениями целевой функции, после чего худшая половина отбрасывается, а лучшая – дублируется.

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

Шаг 9. Исключение и рассеивание. Проверяется условие:  $U^i < P_{ed}$ .

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

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

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

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  $X^i$ ,  $i = 1, 2, \dots, S$ , 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. Останов.

## 14.7 Преимущества и недостатки BFO

На основе рассмотренных применений метода BFO можно сделать вывод, что метод BFO хорошо применим для задач непрерывной оптимизации, в которых следует найти минимум некоторой целевой функции. Проведенные эксперименты по решению практических задач [118–126] засвидетельствовали эффективность использования метода BFO как базового, так и с группировкой бактерий.

Можно выделить следующие преимущества метода BFO по сравнению с традиционными методами многомерной оптимизации:

- метод не выдвигает абсолютно никаких требований к целевой функции, то есть функция может быть мультимодальной, негладкой, обладать разрывами;
- для нахождения оптимума целевой функции не надо знать градиент целевой функции;
- метод является методом глобального поиска, в то время как традиционные методы в часто являются методами локальной оптимизации;
- метод обладает большим количеством настраиваемых параметров, что позволяет его лучшей адаптации для решения той или иной практической задачи оптимизации;
- метод является мультиагентным методом, за счёт чего достигается снижение времени выполнения поиска, поскольку поиск выполняется некоторым количеством агентов параллельно.
- метод является хорошо расширяемым, то есть в его работу можно легко добавлять новые операторы и тем самым создавать гибридные системы. Одним из перспективных является добавление генетических алгоритмов, которые позволяли бы моделировать более правильные эволюционные процессы, которые наблюдаются в процессе деятельности бактерий.

К недостаткам метода можно отнести:

- хоть нахождение оптимума гарантируется, но скорость сходимости носит случайный характер;
- то, что при использовании метода BFO с группировкой бактерий возможно заикливание в локальных оптимумах в случае неправильной настройки параметров, влияющих на группировку бактерий;
- сильная зависимость работы метода от настраиваемых параметров;
- неполнота модели поведения бактерий и может быть доработана, поскольку процедуры воспроизведения и рассеивания ещё не доработаны.

### **Выводы по части III**

В данной части монографии рассмотрены интеллектуальные методы мультиагентной оптимизации, их особенности и возможные области применения различных методов, относящихся к ним. В результате проведенных исследований получены следующие результаты.

1. Выполнен аналитический обзор метода муравьиных колоний, его разновидностей и их особенностей. Представлена сравнительная характеристика вариаций метода муравьиных колоний.

2. Предложены четыре модификации метода муравьиных колоний для решения задачи отбора информативных признаков: на основе представления пунктов назначения признаками, на основе представления пунктов назначения в виде информативности, с использованием операций над чёткими множествами и с использованием операций над нечёткими множествами.

3. Разработана модификация метода муравьиных колоний для выполнения кластер-анализа.

4. Рассмотрен метод пчелиной колонии и произведен сравнительный анализ различных его модификаций, используемых при решении соответствующих задач оптимизации.

5. Выявлены особенности, преимущества и недостатки метода пчелиной колонии.

6. Предложены модификации метода пчелиной колонии для решения задачи отбора информативных признаков и для нахождения оптимума многомерной функции.

7. Рассмотрена оптимизация с использованием роя частиц (PSO-метод) и основные подходы (gbest и lbest), применяемые в PSO.

8. Разработан PSO-метод с управлением изменением скорости частицы и приведена сравнительная характеристика его с другими PSO-методами.

9. Приведен обзор метода оптимизации на основе моделирования перемещения бактерий и его разновидностей, выявлены основные направления применения, недостатки и преимущества метода.

10. Предложена модификация метода оптимизации на основе моделирования перемещения бактерий для решения задачи отбора информативных признаков.

## ЛИТЕРАТУРА К ЧАСТИ III

1. Beni G., Wang J. Swarm Intelligence // Annual Meeting of the Robotics Society: Proceedings of Seventh International Conference. – Tokyo: RSJ Press, 1989. – P. 425-428.
2. Bonabeau E., Dorigo M., Theraulaz G. Swarm Intelligence: From Natural to Artificial Systems. – New York: Oxford University Press, 1999. – 320 p.
3. Dorigo M., Maniezzo V., Colorni A. The Ant System: Optimization by a colony of cooperating agents // IEEE Transactions on Systems, Man, and Cybernetics. – 1996. – Part B, №26(1). – P. 29–41.
4. Bullnheimer B., Hartl R.F., Strauss C. A new rank-based version of the Ant System: A computational study // Central European Journal for Operations Research and Economics. – 1999. – №7(1). – P. 25–38.
5. Олейник Ал.А. Сравнительный анализ методов оптимизации на основе метода муравьиных колоний // Комп'ютерне моделювання та інтелектуальні системи: Збірник наукових праць / За ред. Д.М. Пізи, С.О. Субботіна. – Запоріжжя: ЗНТУ, 2007. – С. 147–159.
6. Camazine S., Sneyd J. A Model of Collective Nectar Source by Honey Bees: Self- organization Through Simple Rules // Journal of Theoretical Biology, 1991. – №149. – P. 547-571.
7. Sumpter D.J.T., Broomhead D.S. Formalising the Link between Worker and Society in Honey Bee Colonies // Lecture Notes In Computer Science: Proceedings of the First International Workshop on Multi-Agent Systems and Agent-Based Simulation. – MABS '98 LNAI, 1998. – P. 95-110.
8. Kennedy J., Eberhart R.C. Particle swarm optimization // International Conference on Neural Networks: Proceedings of the 1995 IEEE, Vol. 4, – NJ: Piscataway, 1995. – P. 1942–1948.
9. De Castro L.N., Von Zuben F.J. Artificial Immune Systems. Part I. Basic Theory And Applications. – Technical Report No. Rt Dca 01/99. – Brazil: Feec/Unicamp, 1999. – 95 p.
10. Gambardella L. M., Dorigo M. Solving symmetric and asymmetric TSPs by ant colonies // Proceedings of the 1996 IEEE International Conference on Evolutionary Computation (ICEC'96). – NJ: IEEE Press, 1996. – P. 622–627.
11. Dorigo M., Gambardella L.M. Ant colonies for the traveling salesman problem // BioSystems. – 1997. – №43. – P. 73–81.
12. Colorni A., Dorigo M., Maniezzo V., Trubian M. Ant system for job-shop scheduling // Belgian Journal of Operations Research, Statistics and Computer Science (JORBEL). – 1994. – №34. – P. 39–53.



13. Субботин С.А., Олейник А.А., Яценко В.К. Отбор информативных признаков на основе модифицированного метода муравьиных колоний // *Радіоелектроніка та інформатика*. – 2006. – № 1. – С. 65–69.
14. Субботин С.А., Олейник Ал.А. Сравнение модификаций метода муравьиных колоний при отборе информативных признаков // *Моделирование неравновесных систем: Материалы IX Всероссийского семинара (13-15 октября 2006) / Под ред. В.В. Слабко. Отв. за выпуск М.Ю. Сенашова*. – Красноярск: ИБМ СО РАН, 2006. – С. 175–177.
15. Subbotin S., Oleynik A. Modifications of Ant Colony Optimization Method for Feature Selection // *The experience of designing and application of CAD systems in Microelectronics: Proceedings of the IX International Conference CADSM–2007 (20–24 February 2007)*. – Lviv: Publishing house of Lviv Polytechnic, 2007. – P. 493–494.
16. Субботин С.А., Олейник А.А. Кластер-анализ на основе мультиагентного подхода // *Автоматика-2007: Тези доповідей чотирнадцятої міжнародної науково-технічної конференції (10–14 вересня 2007 р.)*. – Севастополь: СЧУЯЄтаП, 2007. – С. 50–53.
17. Maniezzo V., Colorni A., Dorigo M. The ant system applied to the quadratic assignment problem. – Bruxelles: Universite Libre de Bruxelles, 1994. – 24 p.
18. Bullnheimer B., Hartl R.F., Strauss C. An improved ant system algorithm for the vehicle routing problem. – Vienna: Institute of Management Science, University of Vienna, 1997. – 28 p.
19. Michel R., Middendorf M. An island model based ant system with lookahead for the shortest supersequence problem // *Proceedings of PPSN-V, Fifth International Conference on Parallel Problem Solving from Nature / In A. E. Eiben, T. Back, M. Schoenauer, and H.-P. Schwefel, editors*. – Verlag: Springer, 1998. – P. 692–701.
20. Costa D., Hertz A. Ants can colour graphs // *Journal of the Operational Research Society*. – 1997. – №48. – P. 295–305.
21. Fan X.P., Luo X., Yi S., Yang S.Y., Zhang H. Optimal Path Planning for Mobile Robots Based on Intensified Ant Colony Optimization Algorithm // *Proceedings of the International Conference on Robotics, Intelligent Systems and Signal Processing*. – Taipei, Taiwan: IEEE Press, 2003. – P. 131–136.
22. Chong S.C., Low M.Y.H. A Bee Colony Optimization Algorithm to Job Shop Scheduling // *Winter Simulation Conference: Proceedings of the 38th conference on Winter simulation*. – Monterey: Monterey Press, 2006. – P. 1954 – 1961.
23. Lučić P., Teodorović D. Bee System: Modeling Combinatorial Optimization Transportation Engineering Problems by Swarm Intelligence //

Transportation Analysis: Proceedings of the Triennial Symposium TRISTAN IV. – Sao Miguel: Azores Press, 2001. – P. 441-445.

24. Teodorović D., Dell'Orco M. Bee Colony Optimization – a Cooperative Learning Approach to Complex Transportation Problems // Advanced OR and AI Methods in Transportation: Proceedings of 16th Mini-EURO Conference and 10th Meeting of EWGT (13-16 September 2005). – Poznan: Publishing House of the Polish Operational and System Research, 2005. – P. 51-60.

25. Quijano N., Passino K.M. Honey Bee Social Foraging Algorithms for Resource Allocation: Theory and Application. – Columbus: Publishing house of the Ohio State University, 2007. – 39 p.

26. Nakrani S., Tovey C. On Honey Bees and dynamic allocation in an internet server colony // Adaptive Behavior. – 2004. – №12. – P. 223–240.

27. Passino K.M., Seeley T.D. Modeling and analysis of nest-site selection by honey bee swarms: The speed and accuracy trade-off // Behavioral Ecology and Sociobiology. – 2006. – №59. – P. 427–442.

28. Parunak H.V.D. “Go to the ant”: Engineering principles from natural multi-agent systems // Annals of Operation Research – 1997. – №75. – P. 69–101.

29. Ferber J., Müller J.P. Influences and reactions: A model of situated multiagent systems // Proceedings of the 2nd International Conference on Multi-Agent Systems (ICMAS-96). – Menlo Park: IEEE Computer Society Press, 1996. – P. 72-79.

30. Müller J.P. The Design of Intelligent Agents: A Layered Approach (Lecture Notes in Computer Science). – Berlin: Springer, 1996. – 227 p.

31. Gasser L., Hill R.W. Coordinated problem solvers // Annual Review of Computer Science, – 1990. – №4. – P. 203–253.

32. Hackman J.R. Groups That Work (and Those That Don't). – San Francisco, California: Jossey-Bass, 1990. – 544 p.

33. Majchrzak A., Gasser L. Toward a conceptual framework for specifying manufacturing workgroups congruent with technological change // International Journal of Computer Integrated Manufacturing. – 1992. – №5. – P. 118–131.

34. Griffin D.R. Animal Minds. – Chicago: The University of Chicago Press, 2001. – 376 p.

35. Wilson E.O. The Insect Societies. – Cambridge, Massachusetts: The Belknap Press, 1974. – 562 p.

36. Agah A., Bekey G.A. Phylogenetic and Ontogenetic Learning in a Colony of Interacting Robots // Autonomous Robots. – 1997. – №4. –P. 85–100.

37. Джонс М.Т. Программирование искусственного интеллекта в приложениях. – М.: ДМК Пресс, 2004. – 312 с.

38. Dornhaus A., Klügl F., Puppe F., Tautz J. Task Selection in Honeybees – Experiments Using Multi-Agent Simulation // Proceedings of GWAL'98. – Bochum: Verlag Harri Deutsch, 1998. – P. 171–183.
39. Bonabeau E., Theraulaz G., Deneubourg J.L., Aron S., Camazine S. Self-organization in social insects // Trends in Ecology and Evolution. – №12(50). – 1997. – P. 188–193.
40. Camazine S., Deneubourg J.L., Franks N.R., Sneyd J., Theraulaz G., Bonabeau E. Self-organization in biological systems. – NJ, Princeton: Princeton University Press, 2001. – 560 p.
41. Buhl J., Sumpter D.J.T., Couzin I.D., Hale J.J., Despland E., Miller E.R., Simpson S.J. From disorder to order in marching locusts // Science. – №312 (5778). – 2006. – P. 1402–1406.
42. Janson S., Middendorf M., Beekman M. Honeybee swarms: how do scouts guide a swarm of uninformed bees? // Animal Behaviour. – №70(2). – 2005. – P.349–358.
43. Hölldobler B., Wilson E. O. The ants. – Cambridge: Harvard University Press, 1990. – p.
44. Wilson E.O. Chemical communication among workers of the fire ants *Solenopsis saevisima* (Fr. Smith): I. The organization of mass foraging // Animal Behaviour. – №10. – 1962. – P. 134–147.
45. Downing H.A., Jeanne R.L. Nest construction by the paperwasp *Polistes*: a test of stigmergy theory // Animal Behaviour. – №36. – 1988. – P. 1729–1739.
46. Downing H.A., Jeanne R.L. The regulation of complex building behavior in the paperwasp *Polistes fuscatus* // Animal Behaviour. – №39. – 1990. – P. 105–124.
47. Karsai I., Theraulaz G. Nest building in a social wasp: postures and constraints // Sociobiology. – №26. – 1995. – P. 83–114.
48. Wenzel J.W. (1996). Natural history of paper-wasps (pp. 58–74). Oxford: University Press.
49. Bruinsma O.H. (1979). An analysis of building behaviour of the termite *Macrotermes subhyalinus*. Lanbouwhogeschool te Wageningen, The Netherlands.
50. Grasse P.P. La reconstruction du nid et les coordinations inter-individuelles chez *Bellicositermes natalensis* et *Cubitermes* sp. La theorie de la stigmergie : essai d'interpretation du comportement des termites constructeurs // Insectes Sociaux. – №6. – 1959. – P. 41–81.
51. Wojtusiak J., Godzinska E.J., Dejean A. Capture and retrieval of very large prey by workers of the African weaver ant *Oecophylla longinoda* (Latreille) // 12th congress of the international union for the study of social in-

sects, Paris, Sorbonne, 21–27 August 1994. – Paris: Universite Paris Nord, 1994. – P. 548.

52. Franks N.R. Teams in social insects: group retrieval of prey by army ants (*Eciton burchelli*, Hymenoptera: Formicidae) // Behavioral Ecology and Sociobiology. – №18. – 1986. – P. 425–429.

53. Chauvin R. Sur le transport collectif des proies par *Formica polyctena* // Insectes Sociaux. – №15. – 1968. – P. 193–200.

54. Sudd J.H. The transport of prey by ants // Behaviour. – №25. – 1965. – P. 234–271.

55. Moffett M.W. Cooperative food transport by an Asiatic ant // National Geographic Research. – №4. – 1988. – P. 386–394.

56. Seeley T.D., Camazine S., Sneyd J. Collective decision-making in honey bees: how colonies choose among nectar sources // Behavioural Ecology and Sociobiology. – №28. – 1991. – P. 277–290.

57. Gordon D.M. Dynamics of task switching in harvester ants // Animal Behaviour. – №38. – 1989. – P. 194–204.

58. Gordon D.M. The organization of work in social insect colonies // Nature. – №380 (6570). – 1996. – P. 121–124.

59. Dorigo M. Optimization, Learning and Natural Algorithms. – Milano: Politecnico di Milano, 1992. – 140 p.

60. Dorigo M., Maniezzo V., Colorni A. Positive feedback as a search strategy. – Milano: Politecnico di Milano, 1991. – 22 p.

61. Di Caro G., Dorigo M. Two ant colony algorithms for best routing in datagram networks // Proceedings of the Tenth IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS'98) / Eds: Y. Pan, S.G. Akl, K. Li. – Anaheim: IASTED/ACTA Press, 1998. – P. 541–546.

62. Leguizamón G., Michalewicz Z. A new version of Ant System for subset problems // Proceedings of the 1999 Congress on Evolutionary Computation (CEC'99). – New Jersey: IEEE Press, 1999. – P. 1459–1464.

63. Dorigo M., Gambardella L.M. Ant Colony System: A cooperative learning approach to the traveling salesman problem // IEEE Transactions on Evolutionary Computation. – 1997. – №1(1). – P. 53–66.

64. Stützle T., Hoos H.H. The MAX–MIN Ant System and local search for the traveling salesman problem // Proceedings of the 1997 IEEE International Conference on Evolutionary Computation (ICEC'97) / Eds: T. Back, Z. Michalewicz, X. Yao. – New Jersey: IEEE Press, 1997. – P. 309–314.

65. Stützle T. Local Search Algorithms for Combinatorial Problems: Analysis, Improvements, and New Applications. – Sankt Augustin: Infix, 1999. – 18 p.

66. Stützle T., Hoos H.H. MAX–MIN Ant System // Future Generation Computer Systems. – 2000. – №16(8). – P. 889–914.

67. Maniezzo V. Exact and approximate nondeterministic tree-search procedures for the quadratic assignment problem. – Bologna: Universita di Bologna, 1998. – 102 p.
68. Maniezzo V., Colorni A. The ant system applied to the quadratic assignment problem // *Knowledge and Data Engineering*. – 1999. – №5 (11). – P. 769–778.
69. Stützle T., Hoos H. MAX–MIN Ant system and local search for combinatorial optimization problems // *Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization* / Eds: S. Vo., S. Martello, I.H. Osman, C. Roucairol. – Boston: Kluwer, 1998. – P. 137–154.
70. Bullnheimer B., Strauss C. Tourenplanung mit dem ant system // *Technical reports of Universit at Wien*. – Wien: Instituts für Betriebswirtschaftslehre, 1996. – 26 p.
71. Bullnheimer B., Hartl R.F., Strauss C. Applying the ant system to the vehicle routing problem // *Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization* / Eds: I. H. Osman S. Vo., S. Martello, C. Roucairol. – Boston: Kluwer, 1998. – P. 109–120.
72. Gambardella L.M., Taillard E., Agazzi G. Macs-vrptw: A multiple ant colony system for vehicle routing problems with time windows // *New Methods in Optimisation* / Eds: D. Corne, M. Dorigo, F. Glover. – McGraw-Hill, 1999. – P. 63–79.
73. Michel R., Middendorf M. An ACO algorithm for the shortest common supersequence problem // *New Methods in Optimisation* / Eds: D. Corne, M. Dorigo, F. Glover. – McGraw-Hill, 1999. – P. 525–537.
74. Gambardella L.M., Dorigo M. HAS-SOP: An hybrid ant system for the sequential ordering problem. – Lugano: CH, 1997. – P. 237–255.
75. Branke J., Guntsch M. Solving the Probabilistic TSP with Ant Colony Optimization // *Journal of Mathematical Modeling and Algorithms*. – 2004. – №3. – P. 403–425.
76. Zhao D.B., Yi J.Q. A Framework of Mobile Robot Path Planning with Ant Colony Optimization // *Proceedings of the International Conference on Sensing, Computing and Automation*. – ChongQing: Watam Press, 2006. – P. 2009–2013.
77. Олейник Ал.А., Субботин С.А. Мультиагентный метод выделения информативных признаков для синтеза моделей объектов управления // *Сучасні проблеми і досягнення в галузі радіотехніки, телекомунікацій та інформаційних технологій: Тези доповідей Міжнародної науково-практичної конференції, м. Запоріжжя, 13-15 квітня 2006 р. / Під заг. ред. Д.М. Пізи. – Запоріжжя: ЗНТУ, 2006. – С. 181–183.*

78. Субботин С.А., Олейник А.А., Яценко В.К. Отбор информативных признаков на основе модифицированного метода муравьиных колоний // *Радіоелектроніка та інформатика*. – 2006. – № 1. – С. 65–69.

79. Субботин С.А., Олейник Ал.А. Сравнение модификаций метода муравьиных колоний при отборе информативных признаков // *Моделирование неравновесных систем: Материалы IX Всероссийского семинара (13-15 октября 2006)* / Под ред. В.В. Слабко. Отв. за выпуск М.Ю. Сенашова. – Красноярск: ИВМ СО РАН, 2006. – С. 175–177.

80. Олейник Ал.А., Олейник Ан.А., Субботин С.А., Яценко В.К. Синтез моделей коэффициента упрочнения деталей авиадвигателей после алмазного выглаживания на основе оптимизационного подхода // *Вісник двигунобудування*. – 2005. – № 3. – С. 25–30.

81. Олейник Ан.А., Олейник Ал.А. Эволюционный синтез моделей сложных объектов и процессов // 10-й міжнародний молодіжний форум “Радіоелектроніка і молодь в XXI сторіччі”: Зб. матеріалів форуму. – Харків.: ХНУРЕ, 2006. – С.417.

82. Олейник Ал.А. Сравнительный анализ модификаций метода муравьиных колоний для отбора признаков // *Радіоелектроніка і молодь в XXI сторіччі: Матеріали 11-го міжнародного молодіжного форуму (10–12 квітня 2007 р.)*. – Харків: ХНУРЕ, 2007. – Ч.1. – С. 353.

83. Subbotin S., Oleynik A. Modifications of Ant Colony Optimization Method for Feature Selection // *The experience of designing and application of CAD systems in Microelectronics: Proceedings of the IX International Conference CADSM–2007 (20–24 February 2007)*. – Lviv: Publishing house of Lviv Polytechnic, 2007. – P. 493–494.

84. Розробка математичного та інформаційного забезпечення інтелектуальної системи візуального контролю транспортних засобів: Звіт про НДР / ТОВ “МПА Групп”. – № ДР 0106U012013. – Запоріжжя, 2007. – 259 с.

85. Олейник Ал.А. Сравнительный анализ методов оптимизации на основе метода муравьиных колоний // *Комп’ютерне моделювання та інтелектуальні системи: Збірник наукових праць / За ред. Д.М. Пізи, С.О. Субботіна*. – Запоріжжя: ЗНТУ, 2007. – С. 147–159.

86. Субботин С.А., Яценко В.К., Олейник Ан.А., Олейник Ал.А. Эволюционный синтез нейромоделей коэффициента упрочнения лопаток авиадвигателей // *Нейроинформатика–2006: Труды VIII Всероссийской научно-технической конференции (26–28 января 2005 г.)*. – М.: МИФИ, 2006. – Ч. 2. – С.89–96.

87. Субботин С.А., Олейник Ал.А. Методы отбора информативных признаков для синтеза нейромоделей на основе метода муравьиных колоний // *Нейроинформатика–2007: Сборник научных трудов IX Всерос-*

сийской научно-технической конференции (23–26 января 2007 г.). – М.: МИФИ, 2007. – Ч. 2. – С.185–193.

88. Seeley T.D. The Wisdom of the Hive. – Cambridge: Harvard University Press, 1995. – 265 p.

89. Milner R. Communication and Concurrency. – Edinburgh: Prentice Hall, 1989. – 272 p.

90. Lučić P., Teodorović D. Computing with Bees: Attacking Complex Transportation Engineering Problems // International Journal on Artificial Intelligence Tools. – 2003. – №12. – P. 375- 394.

91. Zadeh L. Fuzzy Sets // Information and Control. – 1965. – №8. – P. 338-353.

92. Zadeh L. From computing with numbers to computing with words-from manipulation of measurements to manipulation of perceptions // IEEE Transactions on Circuits and Systems-I: Fundamental Theory and Applications. – 1999. – №45. – P. 105–119.

93. Eberhart R., Kennedy J. A New Optimizer using Particle Swarm Theory // In Proceedings of the Sixth International Symposium on Micro machine and Human Science 1995 – P. 39-43

94. Engelbrecht A. Computational intelligence: an introduction – John Wiley and Sons Ltd., 2007 – 597p.

95. Abraham A., Grosan G. Swarm Intelligence in Data Mining – Springer, 2006 – 267p.

96. Peer E., Engelbrecht A. Using Neighborhoods with the Guaranteed Convergence PSO //In Proceedings of the IEEE Swarm Intelligence Symposium, 2003 – P. 235-242.

97. Kennedy J. The Particle Swarm: Social Adaptation of Knowledge // In Proceedings of the IEEE International Conference on Evolutionary Computation, 1997 – P. 303-308.

98. Engelbrecht A. A Study of Particle Swarm Optimization Particle Trajectories // Information Sciences, 2006 – № 176(8) – P. 937-971

99. Clerc M., Kennedy J. The Particle Swarm-Explosion, Stability and Convergence in a Multidimensional Complex Space // IEEE Transactions on Evolutionary Computation, 2002 – P. 58, 73.

100. Coello C., Lechuga M. MOPSO: A Proposal for Multiple Objective Particle Swarm Optimization // In Proceedings of the IEEE Congress on Evolutionary Computation, volume 2002 – № 2 – P. 105-110, 56.

101. Parsopoulos K., Tasoulis D., Vrahatis M. Multiobjective Optimization using Parallel Vector Evaluated Particle Swarm Optimization // In Proceedings of the IASTED International Conference on Artificial Intelligence and Applications, 2004 – № 2 – P. 823-828.

102. Braendler D., Hendtlass T. The Suitability of Particle Swarm Optimization For Training Neural Hardware // In Proceedings of the Fifteenth International Conference on Industrial and Engineering, Applications of Artificial Intelligence And Expert Systems, Lecture Notes in Computer Science, 2002 – № 2358, P. 190-199.

103. Angmin Li, Xin Chen. A New Stochastic PSO Technique for Neural Network Training // Advances in neural networks : ( Part I-III : ISSN 2006 : Third International Symposium on Neural Networks, Chengdu, China, May 28-June 1, 2006 : proceedings ) International Symposium on Neural Networks No3, Chengdu , CHINE (2006) 2006, vol. 3973, pp. 564-569, [Note(s) : 3 vol.(1442, 1444, 1402 p.), // Lecture notes in computer science // Springer, Berlin, ALLEMAGNEETATS-UNIS (2006).

104. Elgallad, A.; El-Hawary, M.; Phillips, W.; Sallam, A. PSO-based neural network for dynamic bandwidth re-allocation // Power Engineering 2002 Large Engineering Systems Conference on, LESCOPE 02 Volume , Issue , 2002 Page(s): 98 - 102 Kennedy, J., and Eberhart, R. C. (1995). Particle swarm optimization. *Proceedings of the IEEE International Conference on Neural Networks, IV*, 1942–1948. Piscataway, NJ: IEEE Service Center.

105. Kennedy, J., and Eberhart, R. C. (1995). Particle swarm optimization. *Proceedings of the IEEE International Conference on Neural Networks, IV*, 1942–1948. Piscataway, NJ: IEEE Service Center.

106. Rui Mendes, *Member, IEEE*, James Kennedy and José Neves. The Fully Informed Particle Swarm: Simpler, Maybe Better. *IEEE Transactions of Evolutionary Computation*, vol. 1, no. 1, January 2005.

107. C. Zhang, H. Shao, Y. Li, Particle swarm optimization for evolving artificial network, in: Proceedings of IEEE International Conference on System, Man and Cybernetics, 2000, pp. 2487-2490.

108. Madigam, M. T., Martinko, J. M., Parker, J., *Biology of Microorganisms*, 8th Edition, Prentice Hall, Upper Saddle River, New Jersey, 1997.

109. Neidhardt, F. C., Ingraham, J. L., Schaechter, M., *Physiology of the Bacterial Cell: A Molecular Approach*, Sinauer Associates, Sunderland, Massachusetts, 1990.

110. Alberts, B., Bray, D., Lewis, J., Raff, M., Roberts, K., Watson, J. D., *Molecular Biology of the Cell*, 2nd Edition, Garland Publishing, New York, NY, 1989.

111. Berg, H. C., And Brown, D. A., *Chemotaxis in Escherichia coli Analyzed by Three-Dimensional Tracking*, *Nature*, Vol. 239, pp. 500–504, 1972.

112. Berg, H. C., *Motile Behavior of Bacteria*, *Physics Today*, Vol. 53, pp. 24–29, 2000.



113. Segall, J. E., Block, S. M., And Berg, H. C., *Temporal Comparisons in Bacterial Chemotaxis*, Proceedings of the National Academy of Sciences, Vol. 83, pp. 8987–8991, 1986.
114. Berg, H. C., *Random Walks in Biology*, New Expanded Edition, Princeton University Press, Princeton, New Jersey, 1993.
115. Woodward, D. E., Tyson, R., Myerscough, M. R., Murray, J. D., Budrene, E. O., Berg, H. C., *Spatio-Temporal Patterns Generated by Salmonella Typhimurium*, Biophysical Journal, Vol. 68, pp. 2181–2189, 1995.
116. K.M.Passino, “Biomimicry of bacterial foraging for distributed optimization and control”, IEEE Control System Magazine, pp.52-67, June 2002.
117. A. Biswas, S. Dasgupta, S. Das, A. Abraham. Synergy of PSO and Bacterial Foraging Optimization – A Comparative Study on Numerical Benchmarks // E. Corchado et al. (Eds.): Innovations in Hybrid Intelligent Systems, ASC 44, pp. 255–263, 2007.
118. M. Tripathy, S. Mishra, L.L. Lai, and Q.P. Zhang. Transmission Loss Reduction Based on FACTS and Bacteria Foraging Algorithm // T.P. Runarsson et al. (Eds.): PPSN IX, LNCS 4193, Springer-Verlag Berlin Heidelberg pp. 222 – 231, 2006.
119. D.P.Acharya, G.Panda, S.Mishra, Y.V.S.Lakshmi. Bacteria Foraging Based Independent Component Analysis // International Conference on Computational Intelligence and Multimedia Applications 2007. – IEEE Computer Society, 2007. – P. 527-531.
120. S.Mishra and C.N.Bhende, “Bacterial Foraging Technique-Based Optimized Active Power Filter for Load Compensation,” *IEEE Trans. Power Del.*, vol.22, no.1, pp.457-465, Jan.2007.
121. D.H. Kim. Robust Tuning of Embedded Intelligent PID Controller for Induction Motor Using Bacterial Foraging Based Optimization // Z. Wu et al. (Eds.): ICES 2004, LNCS 3605, Springer-Verlag Berlin Heidelberg 2005 pp. 137-142, 2005.
122. J.X. Xu, C. Liu, and C.C. Hang: Tuning of Fuzzy PI Controllers Based on Gain/Phase Margin Specifications and ITAE Index. *ISA Transactions* 35 (1996) 79-91.
123. Mario A. Muñoz, Jesús A. López, and Eduardo Caicedo. Bacteria Swarm Foraging Optimization for Dynamical Resource Allocation in a Multizone Temperature Experimentation Platform // P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, Springer-Verlag Berlin Heidelberg pp. 427–435, 2007.
124. Mishra, S.: A hybrid least square-fuzzy bacterial foraging strategy for harmonic estimation. *IEEE Trans. on Evolutionary Computation*, vol. 9(1): 61-73, (2005).

125. Kim, D.H., Cho, C. H.: Bacterial Foraging Based Neural Network Fuzzy Learning. IICAI 2005, 2030-2036.

126. Kim, D.H., Abraham, A., Cho, J.H.: A hybrid genetic algorithm and bacterial foraging approach for global optimization, Information Sciences, Vol. 177 (18), 3918-3937, (2007).

C. Zhang, H. Shao, and Y.Li, "Particle swarm optimization for evolving artificial neural networks", IEEE Int'l Conf on Systems, Man and Cybernetics vol. 4, 2000.