

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

*ПОМАЗКОВА Галина Владимировна*

Выпускная квалификационная работа

*Нейросетевое моделирование работы  
физиологических систем*

Уровень образования: магистратура

Направление: 01.04.02 "Прикладная математика и информатика"

Основная образовательная программа: ВМ.5691.2018 Прикладная математика и информатика в задачах медицинской диагностики

Научный руководитель:  
доцент кафедры КТПА,  
кандидат физ.-мат. наук,  
Степенко Н. А.

Рецензент:  
доцент факультета СУиР  
университета ИТМО,  
доктор физ.-мат. наук,  
Холодова С. Е.

Санкт-Петербург

2020

# Содержание

Введение	3
Постановка задачи	8
Обзор литературы	9
<b>Глава 1. Роевые алгоритмы искусственного интеллекта</b>	<b>11</b>
§1. Основной роевой алгоритм . . . . .	11
§2. Локальный роевой алгоритм . . . . .	14
§3. Важные моменты роевых алгоритмов . . . . .	16
§4. Основные модификации роевых алгоритмов . . . . .	19
<b>Глава 2. Математическая основа стайного движения в рое</b>	<b>22</b>
§1. Графы и Сети . . . . .	23
§2. $\alpha$ -Решётки и Квази $\alpha$ -Решётки . . . . .	25
§3. Энергия отклонения конфигурации сети . . . . .	27
<b>Глава 3. Моделирование стайного движения множественными объектами</b>	<b>31</b>
§1. Алгоритмы стайного движения в пространстве . . . . .	31
§2. Коллективная динамика роя . . . . .	34
§3. Результаты моделирования . . . . .	36
<b>Выводы</b>	<b>40</b>
<b>Заключение</b>	<b>41</b>
<b>Список литературы</b>	<b>42</b>

# Введение

Сейчас нейронные сети пользуются популярностью. Причиной этого является мощь вычислительной техники. С помощью множества компьютеров появилась возможность создания сети с огромной вычислительной мощью, которая сможет решить задачи, ранее не доступные ученым. Например, смоделировать вселенную, распознать речь и получение быстрого ответа на интересующий вопрос.

В конце 1940 появились электронно-вычислительные машины (ЭВМ), что привело к началу развития искусственной нейросети. В тоже время канадский физиолог и нейропсихолог Дональд Хебб создает алгоритм нейронной сети и закладывает принципы его работы в ЭВМ. Стоит отметить и другие даты в истории развития искусственных нейросетей:

1954 год — на рабочей ЭВМ впервые на практике применили нейросеть;  
1958 год — американским ученым по нейрофизиологии и искусственного интеллекта, Фрэнком Розенблаттом был разработан алгоритм распознавания образов и предоставлено его краткое изложение для общественности;  
1960 год — ЭВМ не могла должным образом из-за слабых мощностей выполнять сложные задачи, возложенные на нее, поэтому интерес к технологии немного угас; За двадцатилетний период полным ходом шла «компьютеризация», и мощности тогдашних компьютеров хватило, чтобы вновь разжечь интерес к нейросетям. В 1980 году появилась система с механизмом обратной связи и начались разработки алгоритмов по самообучению; Спустя следующие 20 лет, мощности компьютеров выросли настолько, что в 2000 году ученые-исследователи смогли применять нейросети во многих сферах. Появились программы распознавания речи, имитация зрения, когнитивного восприятия информации. Нейросеть, машинное обучение, робототехника, компьютеризация стали частью нечто большего, под названием «искусственный интеллект».

Нейронные сети — это современный тренд, применяемый в науке и технике. С их помощью улучшаются программы и создаются целые системы, способные автоматизировать, ускорять и помогать в работе человеку. Основная цель — научить систему самостоятельно принимать решения в сложных ситуациях так, как это делает человек.

В состав нейрона входит тело клетки и ядро. У тела клетки в свою

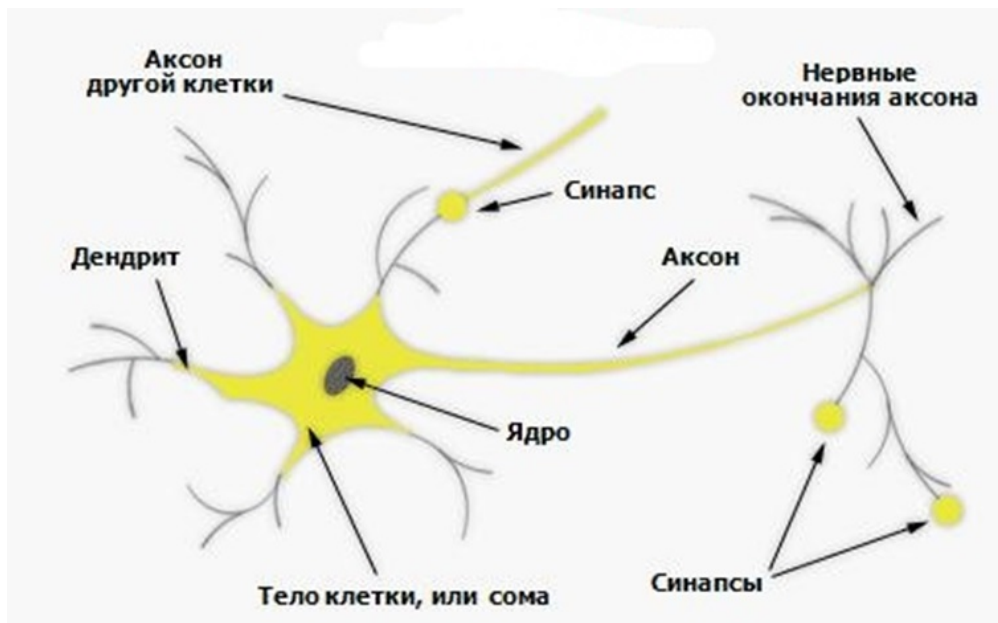


Рис. 1: Биологическая структура нейронной сети

очередь есть множество ответвлений, которое называется дендритами. Длинные дендриты имеют отдельное название — аксоны, которые связывают нейроны между собой с помощью синапсов (по ним проходит импульс). Таким образом, сигнал поступает на дендрит (вход), передается на другой нейрон по аксону (выход). При передачи импульса в точках синапса частоты и амплитуды могут изменяться. Тело нейрона представимо в виде сумматора, а кодом, 0 или 1, регулируются частота, импульс сигнала перед тем, как попасть на сумматор в раздел "Весы". Описанное является основной составляющей нейрона. На рис. 2 представлена математическая и графическая модель нейрона.

Искусственная нейронная сеть — это построенная по математическим правилам модель естественной нейросети, которая воплощена в жизнь с помощью программных и аппаратных составляющих.

Нейронные сети делятся на множество типов, переплетающихся между собой в задачах. Классификация происходит по принципу применения, по типу входной информации, по характерам обучения и связей и по сфере применения.

Сверточные: популярнейший тип сети, часто используемый для распознавания информации в фотографиях и видео, обработке языка, системах для рекомендаций. Основные характеристики: отличная масштабиру-

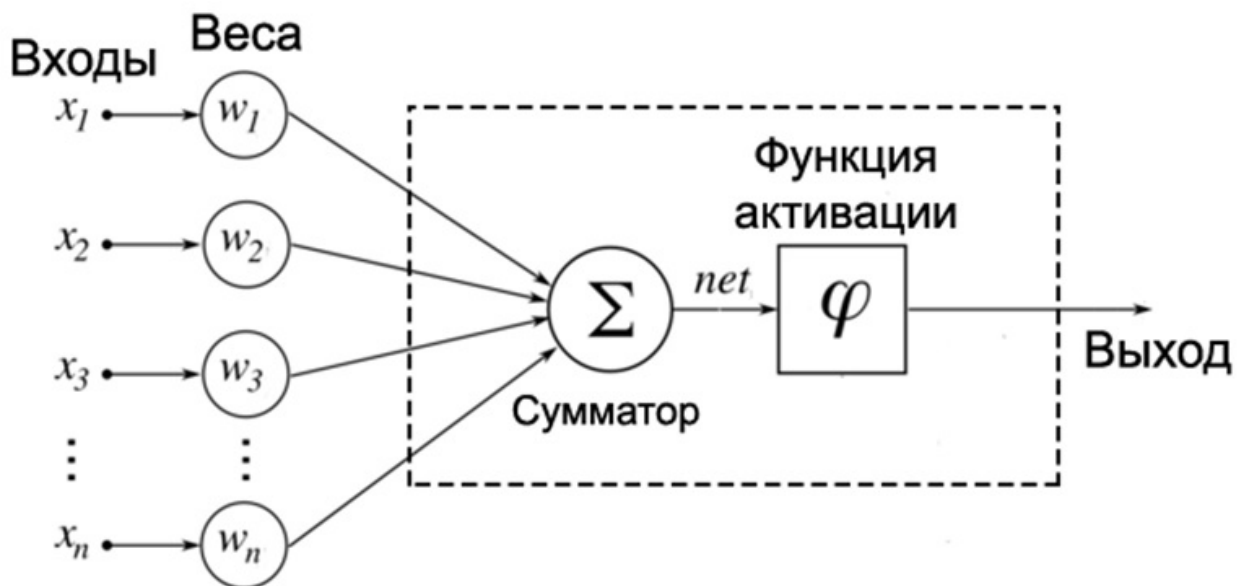


Рис. 2: Математическая и графическая модель нейрона.

емость (распознавание образов любого разрешения); использование объемных трехмерных нейронов; механизм пространственной локализации (работают нелинейные фильтры и охватывается большее число пикселей графического изображения).

Рекуррентные: связи между элементами могут обрабатывать серии различных событий во времени или работать с последовательными цепочками в пространстве. Такой тип часто применяют там, где что-то целое разбито на куски.

## Обучение нейронной сети

Самым главным и важным критерием является возможность обучения нейросети. Нейросеть же представляет совокупность нейронов с проходящими по ним сигналами. На входе подается конкретная информация, которая проходит через тысячи нейронов, а что будет на выходе остается неизвестным. Так вот, чтобы получить нужный результат, необходимо менять параметры сети.

Входной сигнал изменить мы не можем, как и повлиять на сумматор (или вывести его из системы), иначе система перестанет быть нейросетью. Единственное, чем мы можем манипулировать – это коэффициентами или коррелирующими функциями, после чего применить их на веса связей.

Таким образом, можно дать определение обучения нейронной сети — это поиск набора весовых коэффициентов, которые при прохождении через сумматор позволят получить на выходе нужный сигнал.

Коэффициенты весов вручную задавать нельзя. Иначе, нейросеть запомнит правильный выходной сигнал, что приведет к мгновенному выводу информации и иллюзии, что обучение прошло быстро. Но при малейшем изменении входного сигнала на выходе появится неправильный результат. Поэтому, вместо указания конкретных коэффициентов для одного входного сигнала, можно создать обобщающие параметры с помощью выборки. С помощью такой выборки можно обучать сеть, чтобы она выдавала корректные результаты. В этом моменте, можно поделить обучение нейросети на обучение с учителем и без учителя.

Обучение с учителем. Обучение таким способом подразумевает концепцию: даете выборку входных сигналов нейросети, получаете выходные и сравниваете с готовым решением. Сеть нужно тренировать сотнями и тысячами часов, поэтому в 99 процентов случаев тренировкой занимается компьютерная программа.

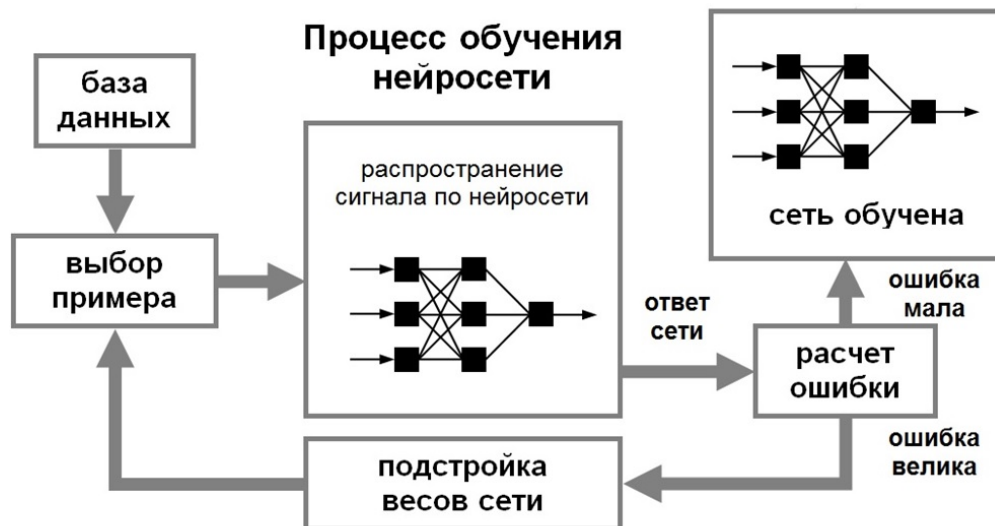


Рис. 3: Процесс обучения нейросети.

Обучение без учителя. Концепция состоит в том, что делается выборка входных сигналов, но правильных ответов на выходе вы знать не можете. Как происходит обучение? В теории и на практике, нейросеть начинает кластеризацию, то есть определяет классы подаваемых входных сигналов.

Затем, она выдает сигналы различных типов, отвечающие за входные объекты.

## Роевые алгоритмы искусственного интеллекта

Алгоритм роя частиц, разработанный Д. Кеннеди (James Kennedy) и Р. Эберхартом, моделирует многоагентную систему, где агенты-частицы двигаются к оптимальным решениям, обмениваясь при этом информацией с соседями.

Текущее состояние частицы характеризуется координатами в пространстве решений (то есть, собственно, связанным с ними решением), а также вектором скорости перемещения. Оба этих параметра выбираются случайным образом на этапе инициализации. Кроме того, каждая частица хранит координаты лучшего из найденных ей решений, а также лучшее из пройденных всеми частицами решений — этим имитируется мгновенный обмен информацией между птицами.

На каждой итерации алгоритма направление и длина вектора скорости каждой из частиц изменяются в соответствии со сведениями о найденных оптимумах:

$$v_i = v_i + a_1 + rand() * (pbest_i - x_i) + a_2 * rand() * (gbest_i - x_i), \quad (1)$$

где  $v$  — вектор скорости частицы,  $a_1$ ,  $a_2$  — постоянные ускорения,  $pbest$  — лучшая найденная частицей точка,  $gbest$  — лучшая точка из пройденных всеми частицами системы,  $x$  — текущее положение частицы, а функция  $rand()$  возвращает случайное число от 0 до 1 включительно.

После вычисления направления вектора  $v$ , частица перемещается в точку . В случае необходимости, обновляются значения лучших точек для каждой частицы и для всех частиц в целом. После этого цикл повторяется.

Модификации: LBEST (Кеннеди и Эберхарт, 1995) [5]; Inertia Weighted PSO (Юхи Ши (Yuhui Shi) и Рассел Эберхарт, 1998) [6]; Time-Varying Inertia Weighted PSO (Юхи Ши и Рассел Эберхарт, 1999) [7]; Canonical PSO (Марис Клер (Maurice Clerc) и Джеймс Кеннеди, 2002) [8]; Fully Informed Particle Swarm (Руи Мендес (Rui Mendes), Джеймс Кеннеди и Жозе Невес (José Neves), 2004) [9].

## Постановка задачи

Целью данной работы является разработка методов искусственного интеллекта по моделированию работы различных элементов физиологических систем организма. В качестве базовых математических алгоритмов были приняты алгоритмы роевого поведения элементов/частиц исследуемых систем. Поэтому, предпочтение отдавалось тем физиологическим системам организма, которые хотя бы в малой степени, позволяли сделать такие допущения. Каковым была взята за основу — кровеносная система. В данной работе, наиболее предпочтительным роевым алгоритмом был рассмотрен алгоритм моделирующий стайное поведение частиц — агентов системы.

Поставленной цели соответствует решению следующих задач:

1. Исследовать некоторые методы искусственного интеллекта: роевого интеллекта, стайного поведения;
2. Определить, как методы роевого интеллекта реализуют работу кровеносной системы (но возможно и технологической, информационной, социальной и др.);
3. Разработать нейросетевой алгоритм роевого интеллекта (РИ) для общей модели кровеносной системы;
4. Провести моделирование его практической реализации для стайного движения роевых частиц в неоднородной среде с препятствиями.



## Обзор литературы

"Искусственные иммунные системы и их применение"[2]. В одной из глав приводится описание различных механизмов иммунитета, их отношения к обработке информации и дается общее представление о быстро растущем направлении исследований под общим названием "искусственные иммунные системы". Соответствующие методы с успехом применяются для решения задач распознавания образов, диагностики и обнаружения неисправностей, а также для создания систем компьютерной безопасности и в некоторых других приложениях.

В статье Чернышева Ю. О., Григорьева Г. В., Венцова Н. Н. "Искусственные иммунные системы: обзор и современное состояние"[3] делается обзор современного состояния, относительно нового класса, биоинспирированных алгоритмов – искусственных иммунных систем. Рассматриваются их проблемы, недостатки и преимущества, актуальные разработки в области искусственных иммунных систем. Работа содержит обзор исследований в области иммунных сетей, иммунного ответа, соматической гипермутации, теории опасности и процессов отбора.

В статье "Flocks, Herds, and Schools: A Distributed Behavioral Model" 1987 Крейга Рейнольдса (Craig W. Reynolds) [10] рассматривается альтернативный подход, основанный на моделировании для того, чтобы описать пути каждой птицы в отдельности. Совокупное движение смоделированной стаи создается распределенной поведенческой моделью, очень похожей на ту, что работает в естественной стае; птицы выбирают свой собственный курс. Каждая имитируемая птица реализована в виде независимый субъект, который ориентируется в соответствии со своим локальным восприятием динамической среды, законы моделируемой физики, управляющие ее движением, и набор запрограммированных в ней моделей поведения на "аниматор.

В статье "Particle Swarm Optimization" Д. Кеннеди и Р. Эберхарта введена концепция оптимизации нелинейных функций с использованием методологии роя частиц. Намечается эволюция нескольких парадигм и обсуждается реализация одной из них. Описаны взаимосвязи между оптимизацией роя частиц и искусственной жизнью, а также генетическими алгоритмами.

В статье "Роевой интеллект и его наиболее распространённые методы реализации" Водолазский И. А., Егоров А. С., Краснов А. В. [4] проводится сравнение трёх наиболее популярных способов реализации роевого интеллекта. Особый акцент сделан на муравьином, пчелином алгоритмах и методе роя частиц. Рассмотрены их особенности, сферы применения и математический аппарат, их реализующий. Кроме того, приведено краткое описание некоторых иных, менее популярных, алгоритмов роевого интеллекта.

В работе "Flocks, herds, and schools: A quantitative theory of flocking" Тонера и Ту [11] представлена количественная континуальная теория "флокирования": коллективное когерентное движение большого числа самоходных организмов. Их модель предсказывает существование "упорядоченной фазы" стаи, в которой все члены даже сколь угодно большой стаи движутся вместе с одинаковой средней скоростью. Это когерентное движение стаи является примером спонтанно нарушенной симметрии: в модели априори не выбирается предпочтительное направление движения; скорее, каждому стаду позволено спонтанно выбирать какое-то совершенно произвольное направление для движения. Анализируя свою модель, они могут сделать детальные количественные предсказания для долгосрочного поведения этого "нарушенного состояния симметрии". Также они рассматривают анизотропную модель, в которой птицы перемещаются преимущественно в "легкой" (например, горизонтальной) плоскости, и делают аналогичные, но количественно отличные предсказания для этой модели. Для этой анизотропной модели получаются точные показатели масштабирования для всех пространственных измерений, включая физически значимый случай.

# Глава 1. Роевые алгоритмы искусственного интеллекта

Глобальный роевой алгоритм — алгоритм, основанный на правилах взаимодействия частиц в рое (учитывается изменения в положении и скорости). Локальный роевой алгоритм — алгоритм, в котором изменение в поведении конкретного члена зависит только от его локального окружения. Роевые алгоритмы (РА) имеют схожесть с эволюционными алгоритмами, они основываются на популяции особей, а именно, потенциальных решениях задачи и методе стохастической оптимизации, который моделируется социальным поведением птиц в стае или насекомых в рое.

## §1. Основной роевой алгоритм

Роевой алгоритм основан на рое частиц, каждая из которых имеет собственное решение задачи. Каждая частица находится в гиперпространстве поиска решения. Её поведение в данном пространстве зависит от своего опыта и опыта её соседей. Частица запоминает свою лучшую точку и лучшее локальное значение целевой функции, а также наилучшие точки своих соседей, в которых и достигается глобальное оптимальное значение в текущий момент. В ходе поиска решения задачи частицы делятся друг с другом информацией, какие лучшие результаты им удалось достигнуть. Получив сведения о локальных и глобальных значениях в настоящий момент, частицы корректируют свои позиции и скорости по определенным законам. Если какая-нибудь частица находит позицию, где глобальное значение целевой функции превосходит имеющееся значение, то происходит изменения в целом рое, чтобы достигнуть данного наилучшего глобального результата. При этом, каждая частица запоминает координаты своей траектории, с помощью которой достигается лучшее значение целевой функции. Обозначим эту траекторию  $y_i$ , когнитивная компонента. Отражение социальной компоненты будет выражать  $\hat{y}_i$ , то есть значение глобального оптимума, которое достигает весь рой. Итак, существуют простые правила поведения, которым следует каждая частица роя, которые подчиняются локальному успеху каждой особи и глобальному оптимуму всего роя.

Какие же характеристики есть у каждой частицы? В момент времени  $t$   $i$ -я частица имеет позицию  $x_i(t)$  в гиперпространстве поиска решений и скорость  $v_i(t)$ . Движение частицы можно записать следующей формулой:

$$x_i(t+1) = x_i(t) + v_i(t+1), \quad (2)$$

где  $x_i(0) \sim (x_{\min}, x_{\max})$ .

Когнитивная и социальная составляющие учитываются в компонентах вектора скорости  $v_i(t+1)$ , который влияет на процесс нахождения решения. Определим вектор скорости следующим образом:

$$v_{ij}(t+1) = v_{ij}(t) + c_1 r_{1j}(t)[y_{ij}(t) - x_{ij}(t)] + c_2 r_{2j}(t)[\hat{y}_j(t) - x_{ij}(t)]. \quad (3)$$

Здесь  $v_{ij}(t)$  —  $j$ -ая компонента скорости ( $j = 1, \dots, n_x$ )  $i$ -ой частицы в момент времени  $t$ ;  $y_{ij}(t)$  — личная лучшая позиция по  $j$ -ой координате  $i$ -ой частицы;  $x_{ij}(t)$  —  $j$ -я координата позиции  $i$ -ой частицы;  $c_1$  и  $c_2$  — положительные коэффициенты ускорения (в большинстве случаев приравниваются к 2), отвечают за влияние когнитивной и социальной компонент;  $r_{1j}(t)$  и  $r_{2j}(t)$  — числа подобранные случайным образом из диапазона  $[0, 1]$ , они влияют на случайность в процессе поиска решения и сгенерированы с учетом нормального распределения.

Решая задачу минимизации, в следующий момент времени личная наилучшая позиция  $i$ -ой частицы записывается следующим образом:

$$y_i(t+1) = \begin{cases} y_i(t) & \text{if } f(x_i(t+1)) \geq f(y_i(t)) \\ x_i(t+1) & \text{if } f(x_i(t+1)) < f(y_i(t)) \end{cases}, \quad (4)$$

где  $f : R^{n_\infty} \rightarrow R$  — целевая функция, которая считает разницу между нынешним решением и оптимальным.

Через  $\hat{y}_j(t)$  обозначим лучшую глобальную позицию роя, при этом значении достигается экстремальное значение целевой функции:

$$\hat{y}_j(t) \in \{y_0(t), \dots, y_{n_s}(t)\} | f(\hat{y}_j(t)) = \min\{f(y_0(t)), \dots, f(y_{n_s}(t))\}, \quad (5)$$

где  $n_s$  — количество частиц во всём рое.

Было рассмотрено как влияют разные компоненты на вычисление

скорости  $i$ -ой частицы согласно (3). Инерционной составляющей и препятствием резкому изменению направления скорости является первое слагаемое  $v_{ij}(t)$ , в ней хранится прошлое направление вектора скорости  $i$ -ой частицы. Когнитивной составляющей, хранящей наилучшую позицию  $i$ -ой частицы и определяющей её соответствующие характеристики относительно предыдущих данных, является второе слагаемое  $c_1 r_{1j}(t)[y_{ij}(t) - x_{ij}(t)]$ . Она же и стремится вернуть  $i$ -ую частицу в наилучшую достигнутую позицию. За социальную составляющую, характеристики  $i$ -ой частицы относительно её же соседей, отвечает третье слагаемое  $c_2 r_{2j}(t)[\hat{y}_j(t) - x_{ij}(t)]$ , прилагает усилия, чтобы направить частицы к найденному глобальному оптимальному значению целевой функции. Графически изображено на рис. 4 для двумерного случая.



Рис. 4: Геометрическая иллюстрация изменения позиции и скорости частицы.

## §2. Локальный роевой алгоритм

Локальный роевой алгоритм — алгоритм, который основывается на локальном оптимальном значении целевой функции, достигнутом ближайшими соседними частицами, а не всего роя. Отношения между соседними частицами задаются конкретной сетевой структурой, которая позволяет образовать пересекающиеся множества во всём рое. Частица входит не в одно множество, что позволяет обмениваться информацией между несколькими множествами. В каждом множестве частицы движутся в направлении локального оптимального решения в своей окрестности.

В роевом алгоритме применяются разнообразные социальные структуры, которые представлены на рис. 5. На рис. 5 а) изображена структура "звезда": каждая частица взаимодействует со всеми частицами (получается полный граф) и пытается переместиться в направлении глобального лучшего значения, которое удалось найти рою; у данного типа лучшая сходимость, но есть тенденция попадания в локальные экстремумы. На рис. 5 б)

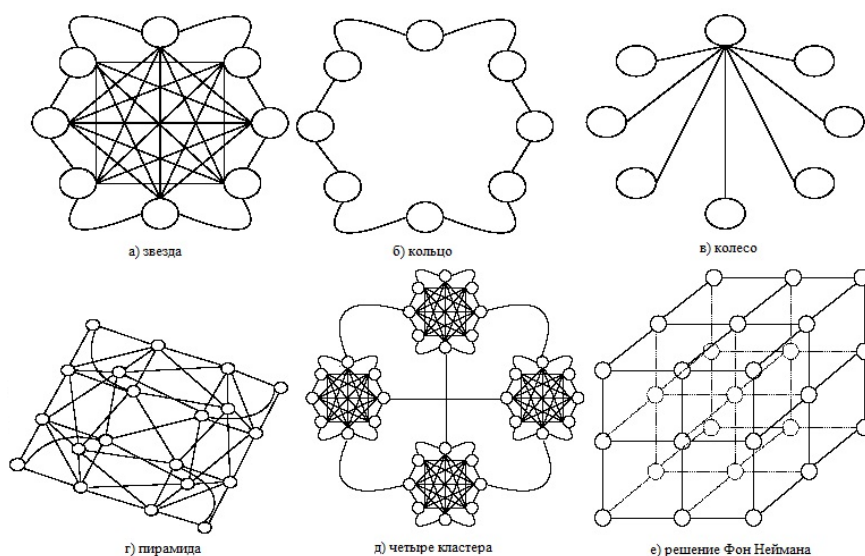


Рис. 5: Типовые социальные сетевые структуры.

структура "кольцо": частица взаимодействует только с  $n_N$  близлежащими частицами и стремится переместиться в направлении лучшей соседней частицы; скорость нахождения глобального экстремума ниже. Если  $n_N = 2$  (связь с двумя соседними частицами), то получается именно проиллюстрированная структура. На рис. 5 в) структура "колесо": частицы не связаны друг с другом, лишь одна частица взаимодействует со всем роем; она

же обрабатывает полученную информацию и перемещается в направлении лучшего соседа. Если достигнуто новое лучшее положение, то сообщается остальным соседям. Эта сеть медленная, обмен оптимальными решениями долог. На рис. 5 г) структура "пирамида": основывается на трёхмерном каркасе. На рис. 5 д) структура из четырёх кластеров, связанных между собой двумя соединениями. На рис. 5 е) структура Фон Неймана: частицы связаны решётчатым способом; выдает наилучшие показания.

Структура влияет на скорость частицы, она задает расстояние между частицами. Лепта каждой частицы пропорциональна расстоянию между положением самой частицы и наилучшей позицией её окружения. Вычислить скорость можно как:

$$v_{ij}(t + 1) = v_{ij}(t) + c_1 r_{1j}(t)[y_{ij}(t) - x_{ij}(t)] + c_2 r_{2j}(t)[\hat{y}_j(t) - x_{ij}(t)], \quad (6)$$

где  $\hat{y}_{ij}$  — наилучшая позиция, найденная по координате  $j$  окружением  $i$  ой частицы. Так же  $\hat{y}_{ij}$  является наилучшей позицией соседей  $N_i$  согласно выражению:

$$\hat{y}_i(t + 1) \in \{N_i | f(\hat{y}_i(t + 1)) = \min\{f(x)\}, \forall x \in N_i\}, \quad (7)$$

где

$$N_i = \{y_{i-nN_i}(t), y_{i-nN_i+1}(t), \dots, y_{i-1}(t), y_i(t), y_{i+1}(t), \dots, y_{i+nN_i}(t)\}, \quad (8)$$

при численности соседей  $nN_j$ .

В рассмотренных двух версиях роевого алгоритма есть схожести — фигурирует социальная составляющая преобразования скорости частицы, направляющей её в направлении глобальной наилучшей позиции. Но есть и отличия касательно их характеристик сходимости:

1. глобальный роевый алгоритм сходится быстрее локального из-за большего обмена информацией между частицами, ценностью уменьшения гиперпространства поиска решения;
2. в локальном роевом алгоритме шире множество возможных решений, из-за чего сходимость к локальному экстремуму не может быть досрочной.

### §3. Важные моменты роевых алгоритмов

Поиск решения в роевом алгоритме — итеративный метод, продолжающийся пока условие для остановки будет не выполнено. Каждая итерация включает в себя: нахождение личного и глобально наилучшего положения в момент времени  $t$ ; изменение скорости каждой частицы; оценивание значения целевой функции всего роя ( $n_s$ , количество членов в рое, значений целевой функции).

Роевый алгоритм начинается с описания роя и соответствующих характеристик: начальные скорости, положения, личные наилучшие позиции,  $c_1$  и  $c_2$ , а также отношение между частицами и окружение соседний частиц для локального роевого алгоритма. Гиперпространство поиска решения равномерно покрывается, ведь от этого разнообразия зависит эффективность роевого алгоритма. Если начальный рой не захватит какую-нибудь область гиперпространства, где будет расположено оптимальное решение, то алгоритм может не найти наилучшее решение, только если в некоторый момент частица изменит свое направление в закрытую область. Будем рассматривать, что оптимальное решение находится в области, задаваемой векторами  $x_{\min}$  и  $x_{\max}$  (максимальное и минимальное значения). Начальную позицию частицы для эффективности метода зададим следующим образом:

$$x(0) = x_{\min,j} + r_j(x_{\max,j} - x_{\min,j}), \forall j = 1, \dots, n_x, \quad \forall i = 1, \dots, n_x, \quad (9)$$

где  $r_j$  число из  $(0, 1)$ . Начальная скорость  $i$  ой частицы приравнивается к нулю,  $v_i(0) = 0$ . Если скорости задать случайным образом или из заданного диапазона, то частицы могут выйти из интересующей области, что ухудшит сходимость. Начальное движение  $i$  ой частицы, благодаря случайной позиции, определяется в случайном направлении. За личную наилучшую позицию для  $i$  ой частицы берется её позиция в момент времени  $t = 0$ ,  $y_i(0) = x_i(0)$ .

Условия для завершения итераций: не должно быть досрочной сходимости роевого алгоритма в локальном оптимальном значении; не допускать многочисленных вычислений из-за постоянной оценки целевой функции частиц. Рассмотрим основные критерии для завершения итераций:



- По максимальному числу итераций (задается максимальновозможное число итераций): при малом значении наилучшее решение может быть не найдено, остановится до. Поэтому требуется совместить с критерием для сходимости. Эффективен, если необходимо найти наилучшее решение в указанное время.

- По найденному приемлемому решению. Пусть при  $x^*$  достигается оптимальное значение целевой функции  $f$ . Критерием завершения итерации послужит  $f(x_i) \leq |f(x^*) - \varepsilon|$ , при малых значениях  $\varepsilon$ .

- По стремлению нормализованного радиуса роя (10) к нулю.

$$R_{norm} = \frac{R_{max}}{d(s)}, \quad (10)$$

где  $d(S)$  — диаметр начального роя, а  $R_{max}$  — максимальный радиус:

$$R_{max} = \|x_m - \hat{y}\|, \|x_m - \hat{y}\| \geq \|x_i - \hat{y}\| \quad (11)$$

для  $\forall i = 1, \dots, n_s$ , где  $m = 1, \dots, n_s$ . Критерием завершения будет  $R_{norm} < \varepsilon$ .

- По малому значению уклона целевой функции. Вышеописанные критерии принимают во внимание относительные позиции частиц в гиперпространстве поиска решения, но не уклон целевой функции. Для этого рассматривают отношение:

$$f'(t) = \frac{f(\hat{y}(t)) - f(\hat{y}(t-1))}{f(\hat{y}(t))}. \quad (12)$$

Критерием сходимости роевого алгоритма служит  $f'(t) < \varepsilon$ , что может достигаться в локальном оптимальном значении. Из-за этого стоит добавить и предыдущий критерий.

Главные составляющие роевого алгоритма: размерность задачи, количество частиц в рое, коэффициенты ускорения, вес инерции, характер и численность окружающих соседей, количество итераций, когнитивный и социальный коэффициенты.

Численность роя  $n_s$  является одним из главных параметров. Увеличивается количество частиц — шире гиперпространство поиска решения (при однородном расположении членов роя), больше частиц покрывает значительную часть гиперпространства за итерацию. Но повышается вычислительная часть итераций, может образоваться параллельная задача нахождения оптимального значения. В каждой задаче есть своя оптимальная численность роя и она определяется с помощью экспериментов.

Численность окружающих соседей влияет на социальную компоненту в локальном роевом алгоритме. Меньше окружение — меньше взаимодействий, медленная сходимости, но эффективнее поиск оптимального решения. Иногда начинают с небольшой численностью соседей и наращивают пропорционально количеству пройденных итераций, что приводит к начальной разнородности и быстрому приближению частиц к выгодной области.

Важно количество итераций. При недостаточном значении условия сходимости может не быть достигнутым. При большом значении повышается вычислительная часть.

Коэффициенты ускорения  $c_1$  и  $c_2$  и случайные вектора  $r_1$  и  $r_2$  влияют на когнитивную и социальную компоненты в результирующем значении скорости частицы. Если  $c_1 = c_2 = 0$ , то скорости частиц не изменяются до достижения ими границ гиперпространства поиска решения. При  $c_1 > 0$  и  $c_2 = 0$  нет взаимодействия между частицами, частицы ищут наилучшие положения в своей области. Если  $c_1 = 0$  и  $c_2 > 0$ , то все частицы роя стремятся к  $\hat{y}$ .

## §4. Основные модификации роевых алгоритмов

Для увеличения эффективности роевого алгоритма ограничивают скорость. Рассматривается гиперпространство поиска решения, задаваемое уравнением изменения скорости (3) (или 6), содержащее три компоненты, которые регулируют значение и направление скоростей частиц. Зачастую значения скоростей частиц увеличиваются, что приводит к большому приращению и покиданию частице интересующей области гиперпространства (расхождение частиц). Для этого и стоит ввести ограничение на изменения значения скорости каждой частицы роя в каком-нибудь диапазоне. При превышении скоростью некоторого значения пусть она принимает заданное допустимое максимальное значение. Обозначим  $V_{\max,j}$  это значение в  $j$  ой компоненте. Регулирование скорости частицы представим в виде:

$$v_{ij}(t+1) = \begin{cases} v'_{ij}(t+1) & \text{если } v'_{ij}(t+1) < V_{\max,j} \\ V_{\max,j} & \text{если } v'_{ij}(t+1) \geq V_{\max,j} \end{cases}, \quad (13)$$

где  $V'_{i,j}$  находится по формуле (3) (или 6).  $V_{\max,j}$  принимает наибольшее значение, способствующее глобальному изучению гиперпространства поиска решения. При малых значениях области принятия удовлетворительно решения, но есть вероятность попадания роя в ловушку локального экстремума. Но и в противном случае увеличивается шанс пропустить область, где находится лучшее решение. Положим  $V_{\max,j}$  как

$$V_{\max,j} = \delta(x_{\max,j} - x_{\min,j}), \quad (14)$$

где  $x_{\max,j}$  и  $x_{\min,j}$  соответственно максимальное и минимальное значение  $j$ -ой компоненты, а  $\delta \in (0, 1]$ .  $\delta$  определяется экспериментальным путем и находится в зависимости от проблематичной области. В этой модификации есть несколько ценных моментов: позиция частицы не ограничивается из-за сдерживания скорости в заданном диапазоне, изменяется размерность перемещения частицы; взаимосвязь максимального значения с каждой составляющей  $j$  и пропорциональность интересующей области. Происходит регулирование увеличения скорости, но всё равно необходимо отслеживать данный момент. Потому что кроме преобразования шага изменения, из-

меняется и движение частиц. На рис. 6 это и продемонстрировано для двумерного случая.  $x_i(t+1)$  — позиция  $i$ -ой частицы без ограничения ско-

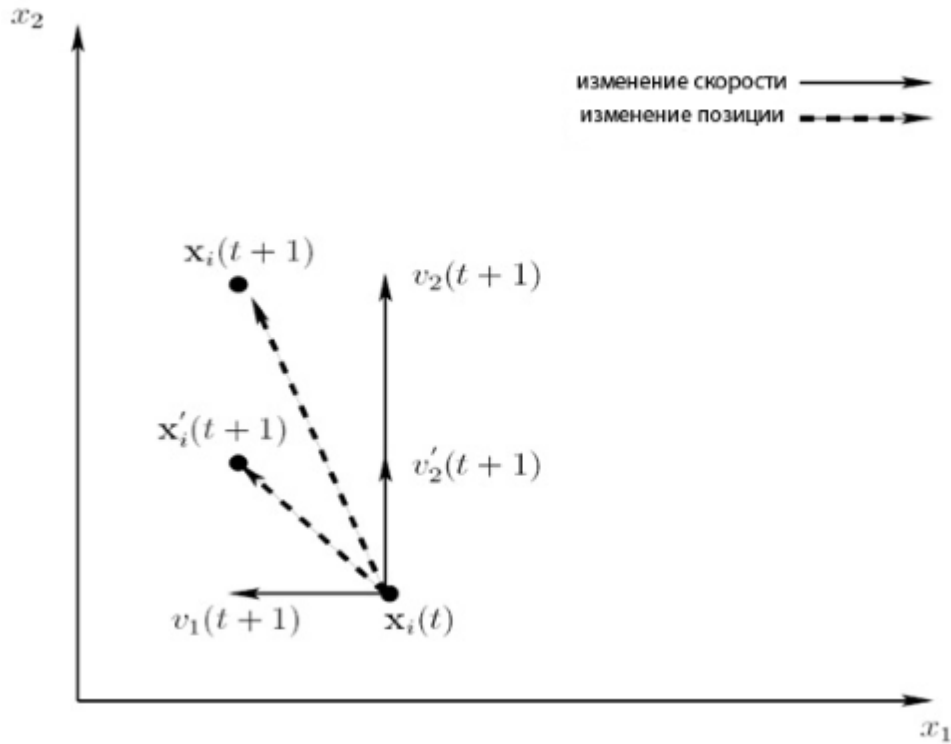


Рис. 6: Эффект ограничения скорости.

рости;  $x'_i(t+1)$  — позиция  $i$ -ой частицы с ограничениями на скорость по составляющей  $x_2$ . Стоит отметить, что произошло изменение направления и величины шага.

Другим способом является введение веса инерции, который контролирует моменты частиц с помощью предыдущего значения скорости. Математически записывается через формулу:

$$v_{ij}(t+1) = wv_{ij}(t) + c_1r_{1j}(t)|y_{ij}(t) - x_{ij}(t)| + c_2r_{2j}(t)|\hat{y}_j(t) - x_{ij}(t)|. \quad (15)$$

Коэффициент  $w$  находит оптимальное решение между изучением и прекращением поиска в гиперпространстве поиска решения. Когда  $w \geq 1$  скорости частиц возрастают и рой разлетается. Необходимо приложить усилия, чтобы члены вернулись к лучшей области гиперпространства. Когда же  $w < 1$  происходит затухание скорости до нуля. Итак, при больших значениях  $w$  наблюдается изучение гиперпространства, а при очень малых — прекращение поиска и увеличение значимости когнитивной и социальной компонент.

Для того, чтобы в процессе поиска решения веса инерции изменялись, существуют некоторые методы:

- Веса генерируются случайным образом на каждой итерации. Иногда применяется распределение по Гауссу, где  $w \sim N(0.72, \sigma)$  и  $\sigma$  малы, для неперевышения  $w$  единицы. Встречается и  $w = (c_1 r_1 + c_2 r_2)$ .

- Веса линейно уменьшаются, например

$$w(t) = (w(0) - w(n_t)) \frac{(n_t - 1)}{n_t} + w(n_t), \quad (16)$$

где  $w(t)$  — вес в момент времени  $t$ ,  $w(0)$  и  $w(n_t)$  начальный и конечный вес соответственно,  $n_t$  — количество итераций. К тому же  $w(0) > w(n_t)$ .

- Веса нелинейно уменьшаются, например

$$w(t + 1) = \frac{(w(t) - 0.4)(n_t - t)}{n_t + 0.4} \quad \text{с } w(0) = 0.9; \quad (17)$$

$$w(t + 1) = \alpha w(t) \quad \text{с } w(0) = 0.975. \quad (18)$$

- Веса генерируются неточным образом, базируясь на неточных множествах и правилах изменения.

## Глава 2. Математическая основа стайного движения в рое

Стайное движение — это форма коллективного поведения большого количества взаимодействующих агентов с общей групповой целью. Инженерные приложения стайного поведения включают массивное мобильное зондирование в окружающей среде; параллельная и одновременная перевозка транспортных средств или доставка полезных грузов; и выполнение военных задач, таких как разведка, наблюдение и борьба, используя кооперативную группу беспилотных летательных аппаратов (БПЛА). Стайное движение является примером скоординированной задачи, выполняемой динамическими агентами над (визуально невидимыми) самоорганизующимися сетями в природе. Самоорганизующаяся особенность стай может дать более глубокое понимание проектирования сенсорных сетей.

В ряде недавних работ по стайному движению делается ряд предположений, которые приводят к алгоритмам с некоторыми из следующих свойств:

- алгоритмы централизованы и требуют, чтобы каждый агент взаимодействовал с любым другим агентом;
- алгоритмы не обладают способностями избегать препятствий;
- алгоритмы приводят к нерегулярной фрагментации или коллапсу;
- неограниченные силы используются для предотвращения столкновений;
- алгоритмы не обладают возможностями отслеживания для групп.

Существует только два класса масштабируемых алгоритма стайного движения, которые приводят к стайному движению при наличии препятствий. Один из них использует коллективные потенциальные функции нескольких видов как для установления межагентских ограничений, так и для избежания препятствий.

## §1. Графы и Сети

Граф  $G$  - это пара  $(\mathcal{V}, \mathcal{E})$ , которая состоит из набора вершин  $\mathcal{V} = \{1, 2, \dots, n\}$  и набора ребер  $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ . Размерности  $|\mathcal{V}|$  и  $|\mathcal{E}|$  соответственно называются порядком и размером графа. В контексте сетевых динамических систем,  $|\mathcal{E}|$  упоминается как "сложность общения" сети с информационным потоком  $G$ .

Матрица смежности  $A = [a_{ij}]$  графа является матрицей с ненулевыми элементами, удовлетворяющими свойству  $a_{ij} \neq 0 \iff (i, j) \in \mathcal{E}$ . Граф называется взвешенным всякий раз, когда элементы его матрицы смежности отличны от 0-1 элементов. В работе в основном используются взвешенные графы с позиционно-зависимыми элементами смежности. Граф  $G$  называется неориентированным тогда и только тогда, когда матрица смежности  $A$  симметрична (или  $A^T = A$ ). Множество соседей узла  $i$  определяется как

$$N_i = \{j \in \mathcal{V} : a_{ij} \neq 0\} = \{j \in \mathcal{V} : (i, j) \in \mathcal{E}\}. \quad (19)$$

Во всей работе для простоты обозначений предполагается  $a_{ii} = 0$  для всех  $i$  (или график не имеют циклов, означающих  $(i, i) \notin \mathcal{E}$ ).

Пусть  $q_i \in \mathbb{R}^m$  обозначают положение узла  $i$  для всех  $i \in \mathcal{V}$ . Вектор  $q = (q_1, \dots, q_n)^T \in Q = \mathbb{R}^{mn}$  называется конфигурацией всех узлов графа. Фреймворк (или структура) — это пара  $(G, q)$ , которая состоит из графа и конфигурации его узлов.

Рассмотрим группу динамических агентов (или частиц) с уравнением движения

$$\begin{aligned} \dot{q}_i &= p_i, \\ \dot{p}_i &= u_i, \end{aligned} \quad (20)$$

где  $q_i, p_i, u_i \in \mathbb{R}^m$  (например  $m = 2, 3$ ) и  $i \in \mathcal{V}$ . Преимущество использования основанных на частицах моделей потоков, по сравнению с моделями континуума, состоит в том, что нельзя воспринимать межагентское восприятие, коммуникацию и вычислительные проблемы как должное. Пусть  $r > 0$  обозначает диапазон взаимодействия между двумя агентами. Открытый шар с радиусом  $r$  (см. рис. 10) определяет набор пространственных

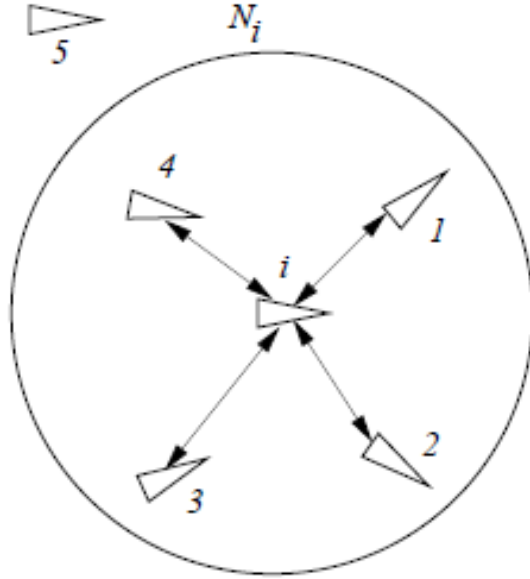


Рис. 7: Агент и его соседи в сферической окрестности

соседей агента  $i$  и обозначается

$$N_i = \{j \in \mathcal{V} : \|q_j - q_i\| < r\}, \quad (21)$$

где  $\|\cdot\|$  — евклидова норма в  $\mathbb{R}^m$ . Учитывая диапазон взаимодействия  $r > 0$ , пространственно индуцированный граф  $G(q) = (\mathcal{V}, \mathcal{E}(q))$  может быть указан  $\mathcal{V}$  и множеством рёбер

$$\mathcal{E}(q) = \{(i, j) \in \mathcal{V} \times \mathcal{V} : \|q_j - q_i\| < r, i \neq j\}, \quad (22)$$

явно зависящим от  $q$ . Мы ссылаемся на граф  $G(q)$  как сеть и к структуре  $(G(q), q)$  как фреймовая сеть.

Топология беспроводной сенсорной сети с радиодиапазоном  $r$  это пример сети. Если диапазон взаимодействия всех агентов одинаков, сеть  $G(q)$  становится неориентированным графом. Сеть  $G(q)$  обычно является орграфом при любом из следующих предположений: 1) сферические окрестности агентов не имеют одинакового радиуса, или 2) каждый агент использует коническую окрестность для определения своих соседей. В данной работе все сети являются неориентированными графами.



## §2. $\alpha$ -Решётки и Квази $\alpha$ -Решётки

Одной из целей является разработка алгоритмов стайного движения с возможностями, которые позволяют группе динамических агентов поддерживать идентичные межагентские расстояния по сети  $G(q)$ . Это побуждает нас рассмотреть следующий набор межагентных алгебраических ограничений:

$$\|q_j - q_i\| = d, \quad \forall j \in N_i(q). \quad (23)$$

Конфигурация  $q$  есть решение множества ограничений в (23). Следовательно, мы находим удобным определение этих решений как объектов типа решетки.

$\alpha$ -решетка — это конфигурация  $q$  удовлетворяющая множеству ограничений в (23). Мы ссылаемся на  $d$ , а также  $\kappa = r/d$  как шкала и отношение для  $\alpha$ -решетки соответственно.

Имеется в виду, что сеть, вызванная  $\alpha$ -решеткой не обязательна для подключения. Кроме того, все края структуры  $(G(q), q)$  вызванной  $\alpha$ -решеткой имеют одинаковую длину. Некоторые общие примеры 2-D и 3-D  $\alpha$ -решетки (с диапазоном взаимодействия  $r = \kappa d$  при  $\kappa = 1 + \epsilon$  и  $0 < \epsilon \ll 1$ ) включают квадратные решетки, гексагональные решетки и наборы вершин правильных многоугольников и платоновых тел. Рис. 8 иллюстрирует другие примеры  $\alpha$ -решеток.

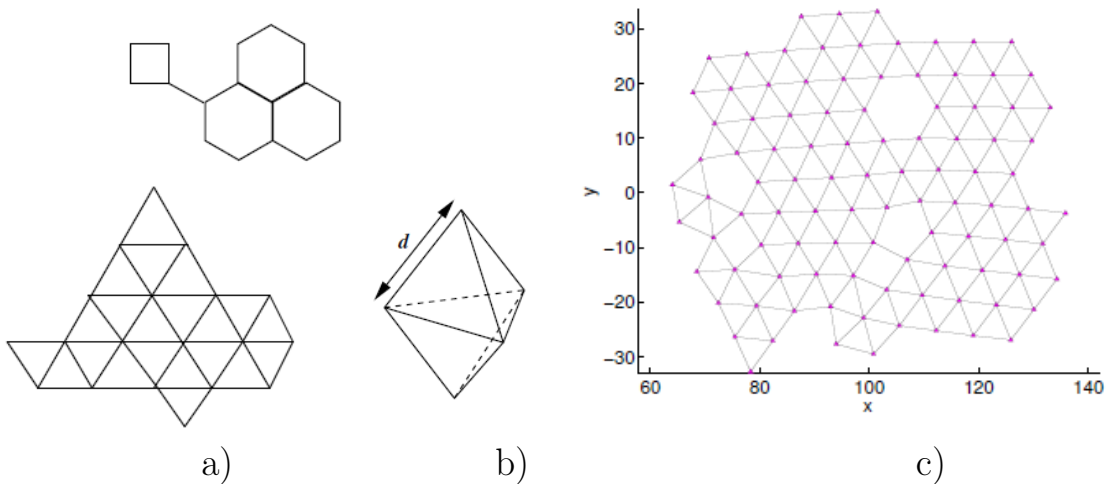


Рис. 8: Примеры  $\alpha$ -решетки и квази  $\alpha$ -решетки: а) двумерные  $\alpha$ -решетки, б) 3-D  $\alpha$ -решетка и в) 2-D квази  $\alpha$ -решетка с  $n = 100$  узлов

Нам также нужно использовать слегка деформированные версии  $\alpha$ -

решеток, которые определяются следующим образом:

Квази  $\alpha$ -решетка — это конфигурация  $q$ , удовлетворяющая следующему набору ограничений неравенства:

$$-\delta \leq \|q_j - q_i\| - d \leq \delta, \quad \forall (i, j) \in \mathcal{E}(q), \quad (24)$$

где  $\delta \ll d$  это неопределенность длины ребра,  $d$  это масштаб, и  $\kappa = r/d$  это отношение квази  $\alpha$ -решетки.

Рис. 8 с) иллюстрирует квази  $\alpha$ -решетку, которая вычисляется численно с использованием второго алгоритма стайного движения.

### §3. Энергия отклонения конфигурации сети

Чтобы измерить степень, в которой конфигурация  $q$  отличается от  $\alpha$ -решётки, мы используем следующую энергию отклонения

$$E(q) = \frac{1}{(|\mathcal{E}| + 1)} \sum_{i=1}^n \sum_{j \in N_i} \psi(\|q_j - q_i\| - d), \quad (25)$$

где  $\psi(z) = z^2$  называется парным потенциалом (можно использовать и другие скалярные потенциалы). Энергия отклонения может рассматриваться как негладкая потенциальная функция для системы  $n$  частиц.  $\alpha$ -решетки есть глобальные минимумы этой потенциальной функции, и они достигают минимального значения в поле. Для квази  $\alpha$ -решетки  $q$  с неопределенностью длины ребра  $\delta$ , энергия отклонения определяется

$$E(q) = \frac{|\mathcal{E}|}{|\mathcal{E}| + 1} \delta^2 \leq \delta^2 = \varepsilon^2 d^2, \quad \varepsilon \ll 1, \quad (26)$$

что значит квази  $\alpha$ -решетки низкоэнергетические конформации  $n$  точек. Порядок величины энергии отклонения квази  $\alpha$ -решетки на рис. 8 с) составляет  $10^{-3}$  (для  $d = 7$  и  $\kappa = 1.2$ ).

#### $\sigma$ -Нормы и $\text{Vintp}$ функции

$\sigma$ -норма вектора есть отображение  $\mathbb{R}^m \rightarrow \mathbb{R}_{\geq 0}$  (не норма), определяется как

$$\|z\|_{\sigma} = \frac{1}{\varepsilon} \left| \sqrt{1 + \varepsilon \|z\|^2} - 1 \right| \quad (27)$$

с параметром  $\varepsilon > 0$  и градиентом  $\sigma_{\varepsilon}(z) = \nabla \|z\|_{\sigma}$ , который записывается следующим образом:

$$\sigma_{\varepsilon}(z) = \frac{z}{\sqrt{1 + \varepsilon \|z\|^2}} = \frac{z}{1 + \varepsilon \|z\|_{\sigma}}. \quad (28)$$

Параметр  $\varepsilon$  из  $\sigma$ -нормы остается неизменным. Можно задаться вопросом, зачем нам вообще определять новую норму. Отображение  $\|z\|_{\sigma}$  дифференцируемо везде, но  $\|z\|$  не дифференцируемо в  $z = 0$ . Позже это свойство  $\sigma$ -нормы используется для построения гладких коллективных потенциальных функций для групп частиц.

Отображение  $\sigma_\epsilon(z)$  является равномерно ограниченной функцией, удовлетворяющей следующим свойствам:

- 1)  $z^T \sigma_\epsilon(z) > 0, \forall z \neq 0,$
- 2)  $\|\sigma_\epsilon(z)\| < 1/\sqrt{\epsilon}$  для всех  $z,$
- 3)  $(x - y)^T (\sigma_1(x) - \sigma_1(y)) > 0, \forall x \neq y.$

Функция  $\text{bump}$  — скалярная функция  $\rho_h(z)$ , которая плавно изменяется между 0 и 1. Функции  $\text{bump}$  используются для построения гладких потенциальных функций с конечными сечениями и гладкими матрицами смежности. Одним из возможных вариантов является следующая  $\text{bump}$  функция

$$\rho_h(z) = \begin{cases} 1, & z \in [0, h), \\ \frac{1}{2} \left( 1 + \cos \left( \pi \frac{(z - h)}{(1 - h)} \right) \right), & z \in [h, 1], \\ 0 & \text{в противном случае,} \end{cases} \quad (29)$$

где  $h \in (0, 1)$ . Можно показать, что  $\rho_h(z)$  это  $C^1$ -гладкая функция со свойством, что  $\rho'_h(z) = 0$  за интервал  $[1, \infty)$  и  $|\rho'_h(z)|$  равномерно ограничен в  $z$ . Используя эту функцию, мы можем определить матрицу пространственной смежности  $A(q)$  через его элементы

$$a_{ij} = \rho_h(\|q_j - q_i\|_\sigma / r_\alpha) \in [0, 1], \quad j \neq i, \quad (30)$$

где  $r_\alpha = \|r\|_\sigma$  и  $a_{ii}(q) = 0$  для всех  $i$  и  $q$ . Для  $h = 1$ ,  $\rho_h(z)$  является индикаторной функцией равной 1 в течение интервала  $[0, 1)$  и 0, в противном случае. Использование индикаторной функции приводит к созданию сети с 0-1 позиционно-зависимыми элементами смежности.

## Коллективные потенциальные функции

Функция коллективного потенциала  $V(q)$  группы агентов является неотрицательной функцией  $V : \mathbb{R}^{mn} \rightarrow R_{\geq 0}$  с тем свойством, что любое решение множества алгебраических ограничений в (23) «тесно связано с» локальными минимумами  $V(q)$  и наоборот. Смысл, в котором эти две величины тесно связаны, измеряется с использованием энергии отклонения, которая действует как функция расстояния. В данной работе коллектив-

ный потенциал является гладкой версией функции энергии отклонения со скалярным попарным потенциалом, который имеет конечное сечение. Это означает, что существует конечный диапазон взаимодействия  $r > 0$  такой, что  $\psi'(z) = 0, \forall z \geq r$ . Эта особенность оказывается основным источником масштабируемости наших алгоритмов стайного движения.

Обобщенные функции Леннарда-Джонса и экспоненциально исчезающие отображения не имеют конечных сечений и не подходят для наших целей без каких-либо изменений. Распространенным подходом к созданию парного потенциала с конечным вырезом является «мягкая резка», при которой парный потенциал умножается на функцию выпуклости. Здесь мы используем альтернативный подход, мягко вырезая функции действия, а затем используя их интегралы в качестве парных потенциалов. Таким образом, производная bump-функции никогда не появляется в функции действия (т.е. производная от парного потенциала), и, таким образом, отрицательный bump в функции действия вблизи  $z = r$  избегается.

Пусть  $\psi(z) : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$  есть притягивающий/отталкивающий попарный потенциал с глобальным минимумом в  $z = d$  и конечным сечением  $r$ . Тогда следующая функция

$$\phi(q) = \frac{1}{2} \sum_i \sum_{j \neq i} \psi(\|q_j - q_i\| - d) \quad (31)$$

является коллективным потенциалом, который не дифференцируем в единичных конфигурациях, в которых два различных узла совпадают, или  $q_j = q_i$ . Чтобы решить эту проблему, мы используем набор алгебраических ограничений (23), которые переписаны в терминах  $\sigma$ -норм как

$$\|q_j - q_i\|_\sigma = d_\alpha, \quad j \in N_i(q), \quad (32)$$

где  $d_\alpha = \|d\|_\sigma$ . Эти ограничения дают гладкую коллективную потенциальную функцию в виде:

$$V(q) = \frac{1}{2} \sum_i \sum_{j \neq i} \psi_\alpha(\|q_j - q_i\|_\sigma) \quad (33)$$

где  $\psi_\alpha(z)$  является гладким попарно притягивающим/отталкивающим по-

тенциалом (определенным в уравнении (35)) с конечным срезом при  $r_\alpha = \|r\|_\alpha$  и глобальный минимум в  $z = d_\alpha$ .

Чтобы построить гладкий попарный потенциал с конечным сечением, мы интегрируем функцию действия  $\psi_\alpha(z)$ , которая исчезает для всех  $z \geq r_\alpha$ . Определим эту функцию действия как

$$\begin{aligned}\phi_\alpha(z) &= \rho_h(z/r_\alpha)\phi(z - d_\alpha), \\ \phi(z) &= \frac{1}{2}\left((a+b)\sigma_1(z+c) + (a-b)\right),\end{aligned}\tag{34}$$

где  $\sigma_1(z) = z/\sqrt{1+z^2}$  и  $\phi(z)$  является неравномерной сигмоидальной функцией с параметрами, которые удовлетворяют  $0 < a \leq b$ ,  $c = |a-b|/\sqrt{4ab}$ , гарантирующим  $\phi(0) = 0$ . Попарно притягивающий/отталкивающий потенциал  $\psi_\alpha(z)$  в (33) определяется как

$$\psi_\alpha(z) = \int_{d_\alpha}^z \phi_\alpha(s) ds.\tag{35}$$

функции  $\phi_\alpha(z)$  и  $\psi_\alpha(z)$  изображены на рис. 9.

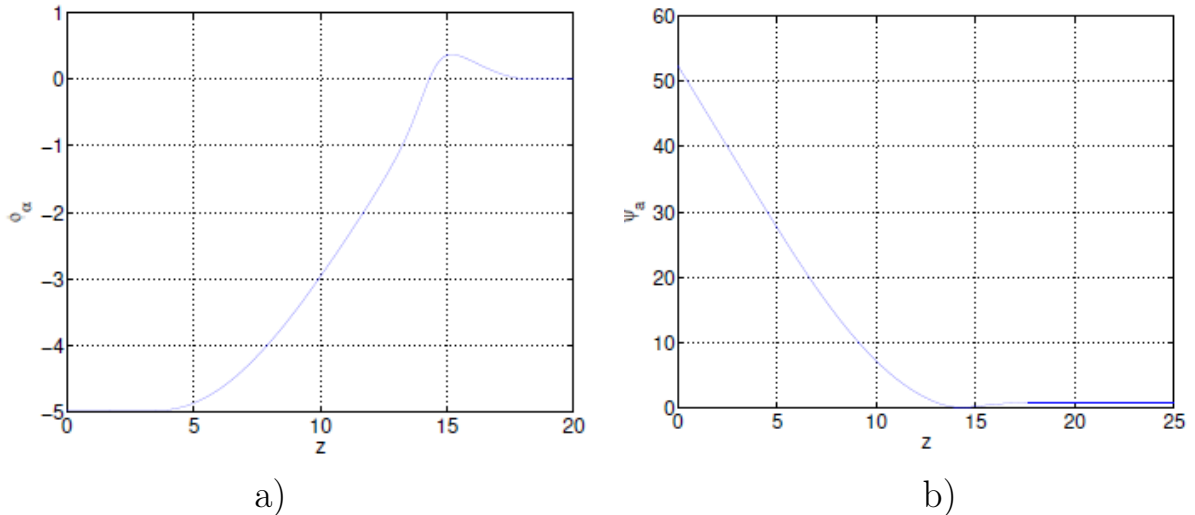


Рис. 9: Функция действия и потенциальная функция с конечными сечениями: а)  $\phi_\alpha(z)$  и б)  $\psi_\alpha(z)$

# Глава 3. Моделирование стайного движения множественными объектами

## §1. Алгоритмы стайного движения в пространстве

Рассмотрим теперь набор распределенных алгоритмов для стайного движения в свободном пространстве. Примем динамику движения физической частицы роя —  $\alpha$ -агент, заданной уравнением движения

$$\ddot{q}_i = u_i, \quad (36)$$

где  $\alpha$ -агенты соответствуют птицам, рыбам или как в исследуемом случае — клеткам крови. Будем также считать, что  $\alpha$ -агент имеет тенденцию оставаться на расстоянии  $d > 0$  со всеми соседними  $\alpha$ -агентами. В свободном движении каждый  $\alpha$ -агент находится под управляющим воздействием:

$$u_i = f_i^g + f_i^d + f_i^\gamma, \quad (37)$$

где  $f_i^g = -\nabla_{q_i} V(q)$  — градиентный термин,  $f_i^d$  — согласование скорости/время выравнивания который действует как демпфирующая сила и  $f_i^\gamma$  — навигационная обратная связь от следования групповой цели. Примерами групповой цели может быть движение пункту назначения (миграции) или рассредоточение из скученного пространства. Приведём два распределенных алгоритма, которые могут быть использованы для создания стайного движения в  $\mathbb{R}^m$  (размерности 1,2,3 представляют наибольший интерес):

1. Алгоритм 1:

$u_i = u_i^\alpha$  при

$$u_i^\alpha = \sum_{j \in N_i} \phi_\alpha(\|q_j - q_i\|_\sigma) n_{ij} + \sum_{j \in N_i} a_{ij}(q)(p_j - p_i), \quad (38)$$

где  $n_{ij}$  есть вектор вдоль линии, соединяющей  $q_i$  с  $q_j$  и определяемый как

$$n_{ij} = \sigma_\epsilon(q_j - q_i) = \frac{q_j - q_i}{\sqrt{1 + \epsilon\|q_j - q_i\|^2}} \quad (39)$$

и  $0 < \epsilon < 1$  является фиксированным параметром  $\sigma$ -нормы. Алгоритм 1 не имеет навигационной обратной связи и известен как  $(\alpha, \alpha)$  протокол стайного движения, потому что он устанавливает правила взаимодействия между двумя  $\alpha$ -агентами.

1. Алгоритм 2:

$$u_i = u_i^\alpha + u_i^\gamma$$

или

$$u_i = \sum_{j \in N_i} \phi_\alpha(\|q_j - q_i\|_\sigma) n_{ij} + \sum_{j \in N_i} a_{ij}(q)(p_j - p_i) + f_i^\gamma(q_i, p_i), \quad (40)$$

где  $u_i^\gamma$  это навигационная обратная связь, заданная как

$$u + i^\gamma = f_i^\gamma(q_i, p_i, q_r, p_r) = -c_1(q_i - q_r) - c_2(p_i - p_r), \quad c_1, c_2 > 0. \quad (41)$$

Пара  $(q_r, p_r) \in \mathbb{R}^m \times \mathbb{R}^m$  есть состояние  $\gamma$ -агента. Здесь  $\gamma$ -агент является динамическим/статическим агентом, который представляет цель группы. Пусть  $(q_d, p_d)$  — фиксированная пара  $m$ -векторов, которые обозначают начальное положение и скорость  $\gamma$ -агента. Динамический  $\gamma$ -агент имеет следующую модель

$$\begin{cases} \dot{q}_r = p_r, \\ \dot{p}_r = f_r(q_r, p_r), \end{cases} \quad (42)$$

с начальными условиями  $(q_r(0), p_r(0)) = (q_d, p_d)$ . Статический  $\gamma$ -агент имеет фиксированное состояние, равное  $(q_d, p_d)$  за все время.

Функция  $f_r(q_r, p_r)$  для динамического  $\gamma$ -агента является частью системы контроля слежения за группой агентов. Например, выбор  $f_r \equiv 0$  приводит к  $\gamma$ -агенту, который движется по прямой с желаемой скоростью  $p_d$ .

В ходе анализа этих алгоритмов находим, что несмотря на сходство определенных терминов в этих алгоритмах, коллективное поведение группы агентов, использующих алгоритм 1, существенно отличается от группы агентов, применяющих алгоритм 2.

Покажем далее, что алгоритм 1 объединяет все три правила Рейнольдса в одном уравнении. Это не результат применения алгоритма на основе градиента. Это связано с тем, что основанный на градиенте термин



алгоритма может быть выражен через зависящие от положения элементов напряжения графа.

Алгоритм 1 приводит к поведению стайного движения только для очень ограниченного набора начальных состояний. Для общего набора начальных состояний и большого количества агентов (например,  $n > 10$ ), алгоритм 1, скорее всего, не в состоянии воспроизводить стайное движение и вместо этого приводит к регулярной фрагментации (рис. 4).

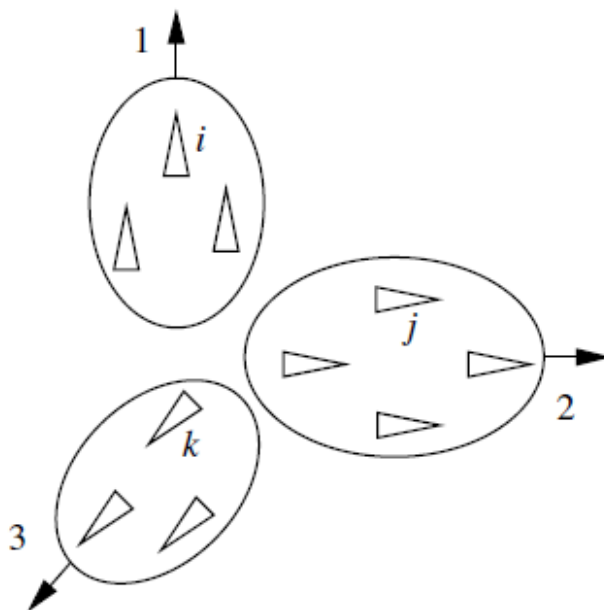


Рис. 10: Феномен фрагментации

Напротив, алгоритм 2 никогда не приводит к фрагментации. В результате только алгоритм 2 действительно является алгоритмом стайного поведения.

## §2. Коллективная динамика роя

Коллективная динамика группы  $\alpha$ -агентов, действующие по алгоритму 2 (или 1) орпеделим в форме:

$$\begin{cases} \dot{q} = p, \\ \dot{p} = -\nabla V(q) - \hat{L}(q)p + f_\gamma(q, p, q_r, p_r), \end{cases} \quad (43)$$

где  $V(q)$  является гладкой коллективной потенциальной функцией, заданной выше, и  $\hat{L}(q)$  это  $m$ -мерный Лапласиан сети  $G(q)$  с позиционно-зависимой матрицей смежности  $A(q) = [a_{ij}(q)]$ . Для алгоритма 1 значение  $f_\gamma \equiv 0$ .

Первый ожидаемый результат заключается в том, что если  $f_\gamma \equiv 0$ , то система динамики является диссипативной системой частиц с Гамильтонианом

$$H(q, p) = V(q) + \sum_{i=1}^n |p_i|^2. \quad (44)$$

Это связано с тем, что  $\dot{H} = -p^T \hat{L}(q)p \leq 0$  и фактом, что многомерный граф Лапласианв  $\hat{L}(q)$  это положительная полуопределённая матрица для всех  $q$ .

Анализ устойчивости коллективной динамики зависит от использования правильной системы координат, которая позволяет использовать принцип инвариантности Ласалля. Простой подход заключается в использовании  $H(q, p)$  в  $(q, p)$ -координатах. Причина, по которой такой подход не работает, заключается в том, что невозможно установить ограниченность решений. Во время фрагментации решение не может оставаться ограниченным.

Рассмотрим движущуюся систему отсчёта, которая центрирована на  $q_c$ -центр масс (ЦМ) всех частиц. Пусть

$$Ave(z) = \frac{1}{n} \sum_{i=1}^n z_i \quad (45)$$

обозначает среднее значение  $z_i$ , где  $z = (z_1, \dots, z_n)^T$ . Примем, что  $q_c = Ave(q)$  и  $p_c = Ave(p)$  являются положением и скоростью центра движущейся системы отсчета. Затем примем  $\dot{q}_c(t) = p_c(t)$  и  $\dot{p}_c(t) = Ave(u(t))$ .

Рассмотрим движения центра группы относительно системы отсчета

отдельно от коллективного движения частиц в движущейся системе. Положение и скорость агента  $i$  движущейся системе отсчёта задается

$$\begin{cases} x_i = q_i - q_c, \\ v_i = p_i - p_c. \end{cases} \quad (46)$$

Относительные положения и скорости остаются неизменными в движущейся системе, т.е.  $x_j - x_i = q_j - q_i$  и  $v_j - v_i = p_j - p_i$ . Таким образом,  $V(q) = V(x)$  и  $\nabla V(q) = \nabla V(x)$ . Закон управления  $(\alpha, \alpha)$  в движущейся системе отсчёта может быть выражен как

$$u_i^\alpha = \sum_{j \in N_i} \phi_\alpha(\|x_j - x_i\|_\sigma) n_{ij} + \sum_{j \in N_i} a_{ij}(x)(v_j - v_i) \quad (47)$$

при  $a_{ij}(x) = \rho_h(\|x_j - x_i\|_\sigma / r_\alpha)$ .

Предположим, что навигационная обратная связь  $f_\gamma(q, p)$  является линейной, то есть существует разложение  $f_\gamma(q, p)$  в следующей форме

$$f_\gamma(q, p, q_r, p_r) = g(x, v) + e_1 \otimes h(q_c, p_c, q_r, p_r). \quad (48)$$

Тогда коллективная динамика группы агентов, применяющих алгоритм 2 (или 1) может быть разложена на  $n$  систем второго порядка в движущейся системе отсчёта

$$\begin{cases} \dot{x} = v, \\ \dot{v} = -\nabla V(x) - \hat{L}(x)v + g(x, v) \end{cases} \quad (49)$$

и одну система второго порядка в ориентированной системе отсчета

$$\begin{cases} \dot{q}_c = p_c, \\ \dot{p}_c = h(q_c, p_c, q_r, p_r), \end{cases} \quad (50)$$

где

$$\begin{aligned} g(x, v) &= -c_1 x - c_2 v, \\ h(q_c, p_c, q_r, p_r) &= -c_1(q_c - q_r) - c_2(p_c - p_r) \end{aligned} \quad (51)$$

и  $(q_c, p_c)$  — состояние  $\gamma$ -агент.

### §3. Результаты моделирования

В данном разделе представлены несколько результатов моделирования плоского стайного движения. Следующие параметры остаются неизменными на протяжении всего моделирования:  $d = 7$ ,  $r = 1.2d$  (или  $\kappa = 1.2$ ),  $d' = 0.6d$ ,  $r' = 1.2d'$ ,  $\epsilon = 0.1$  (для  $\sigma$ -норма),  $a = b = 5$  для  $\phi(z)$ ,  $h = 0.2$  для  $\text{bump}$ -функции  $\phi_\alpha(z)$ ,  $h = 0.9$  для  $\text{bump}$ -функции  $\phi_\beta(z)$ , и размер шага во всех моделях колеблется от 0,01 до 0,03 секунды. Параметры алгоритмов стайного движения и типы начальных состояний задаются отдельно для каждого эксперимента. Набор  $l$  сферических препятствий определяются с помощью  $(m + 1) \times l$  матриц  $M_s$ , где каждый столбец  $M_s$  это вектор  $(y_k, R_k) \in \mathbb{R}^{m+1}$ .

Во всех результатах моделирования угол направления (или ориентация) каждого  $\alpha$ -агента определяет направление скорости этого агента. Кроме того, положение динамического  $\gamma$ -агента помечен знаком  $\times$ .

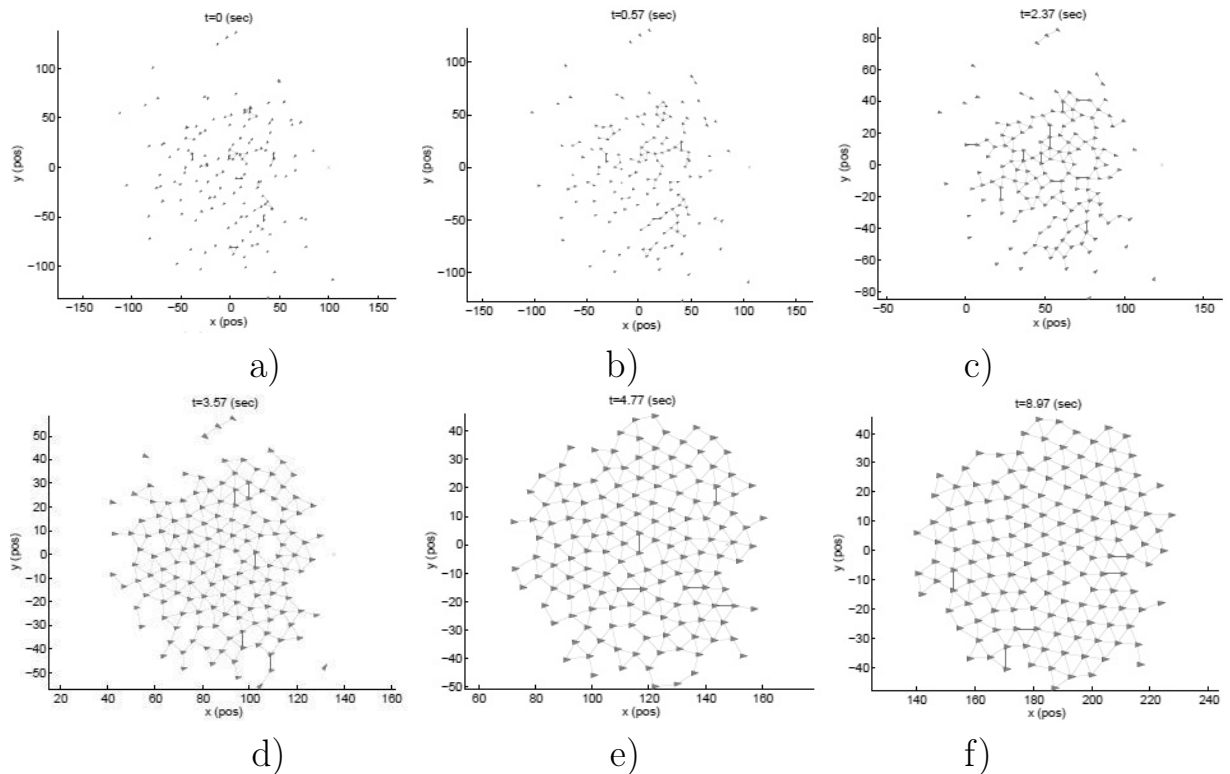


Рис. 11: 2-D стая  $n = 150$  агентов с топологией динамики

## Плоское движение стаи в свободном пространстве ( $n = 150$ )

На рис. 11 показаны последовательные снимки двухстороннего захвата для 150 агентов в свободном пространстве с использованием Алгоритма 2. Начальные положения выбираются случайным образом из гауссовского распределения с дисперсией 2500. Начальные координаты скорости выбираются случайным образом из  $[-2, -1]$ . На рис. 11 показаны кадры, соответствующие расположению агентов сформированной стаи. Число ребер динамического графа  $G(q(t))$  увеличивается со временем и имеет тенденцию связывать сеть. Набор начальных позиций выбирается случайным образом равномерно, так что сеть  $G(q(0))$  сильно рассоединена (т.е. имеет слишком много компонентов). Это делает задачу стайного движения более сложной. Численные измерения показывают, что конечная расположение стаи является низкоэнергетической квази  $\alpha$ -решеткой.

## 2-D фрагментация в свободном пространстве ( $n = 40$ )

Рис. 12 демонстрирует явление фрагментации для  $n = 40$  агентов, применяющих Алгоритм 1. Неудивительно, что при случайном наборе начальных состояний поведение стаи не создается. На рис. 12 f) можно идентифицировать 9 различных небольших потоков (каждый из которых содержит по два агента) и три отдельных агента.

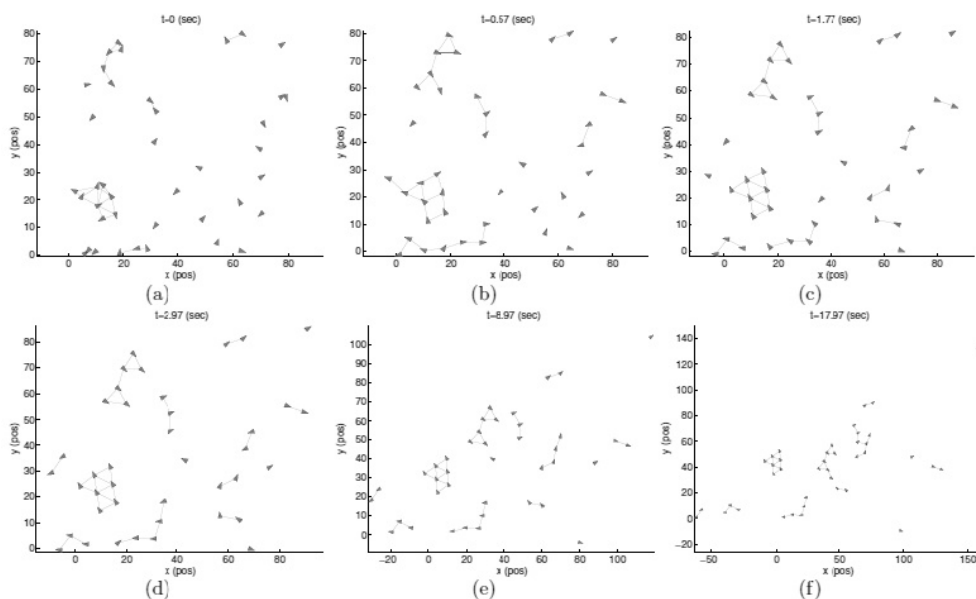


Рис. 12: 2-D стая  $n = 40$  агентов.

В Алгоритме 1 (или  $(\alpha, \alpha)$  протокол), не существует  $\gamma$ -агентов из-за отсутствия групповой цели. Этот результат моделирования является еще одним доказательством того, что создание движущихся движений довольно нетривиально, потому что идея использования основанного на градиенте закона управления плюс условие согласования скорости не обязательно работает!

Фрагментация — это общая форма коллективного поведения агентов, применяющих Алгоритм 1. Такое поведение нечувствительно к типу распределения вероятностей исходного положения агентов. По-видимому, для случая очень плотной исходной сети с небольшим начальным несоответствием скорости можно ожидать, что группа агентов образует гигантский компонент (т.е. связанный подграф с большим порядком).

На рис. 12 d)–f), можно заметить, что два агента, которые принадлежат двум разным стадам, со временем удаляются друг от друга дальше друг от друга. Опять же, это общее свойство фрагментации. Феномен фрагментации можно рассматривать как отсутствие когезии в группе частиц.

### **Маневр разделения/воссоединения и обтекание препятствий**

Рассмотрим группу агентов, которые намереваются перемещаться/мигрировать из точки  $A$  в  $B$ . Здесь,  $A$  и  $B$  позиции центра масс (ЦМ) группы в источнике и пункте назначения группы. Всякий раз, когда на прямой линии, соединяющей  $A$  с  $B$ , агенты не могут пройти через препятствия. В результате они могут разделиться на две или несколько небольших групп.

Цель в выполнении манёвра разделиться/вернуться состоит в том, чтобы собрать различные группы, которые изначально раскололись из-за наличия препятствий или противоборствующих агентов. Маневр разделения/воссоединения показан на рис. 13 для группы  $n = 150$  агентов в присутствии  $l = 6$  препятствий. Рис. 13 демонстрирует эволюцию топологии межагентских взаимодействий. Исходя из этой цифры, становится ясно, что топология сети динамических агентов во время стайного движения часто меняется. Другими словами, стайное движение включает в себя анализ устойчивости сети динамических агентов с топологией переключения.

Исходная позиция агентов выбирается случайным образом из интервала  $[-40, 80]$ . Начальные скорости установлены на ноль. Цель группы за-

дается статическим  $\gamma$ -агентом с  $q_d = (200, 30)^T$  и  $p_d = (5, 0)^T$ . Более того,  $c_1^\alpha < c_1^\gamma < c_1^\beta$  и  $c_2^\nu = 2\sqrt{c_1^\nu}$  для всех видов. Матрица препятствий

$$M_s = \begin{bmatrix} 100 & 110 & 120 & 130 & 150 & 160 \\ 20 & 60 & 40 & -20 & 40 & 0 \\ 10 & 4 & 2 & 5 & 5 & 3 \end{bmatrix}. \quad (52)$$

Согласно рис. 13, после того, как все агенты преодолевают препятствия на своем пути, группа образует гигантский стая. Численно подтверждено, что ни один из агентов никогда не сталкивался ни с одним из шести препятствий для полной траектории движения частиц. Рис. 13 иллюстрирует маневр разделения/воссоединения без рисования краев  $(\alpha, \beta)$ -сеть в каждый момент.

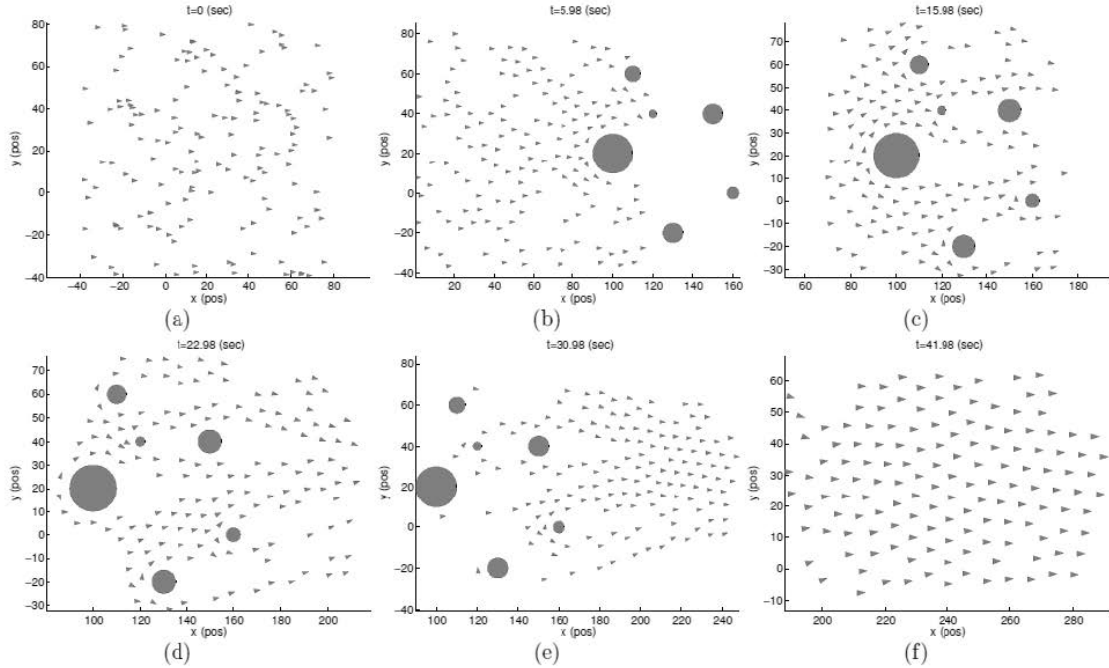


Рис. 13: Маневр разделения/воссоединения для  $n = 150$  агентов.

## Выводы

В данной работе проведено математическое моделирование стайного движения частиц кровеносной системы.

- Исследованы некоторые методы искусственного интеллекта на основе роевого интеллекта с учётом их применения для моделирования движения крови;
- В качестве основного алгоритма был принят метод стайного движения;
- Предложены два математических алгоритма выстраивания стаи, позволяющие смоделировать как её сбор, так и фрагментацию перед обходом препятствий;
- Проведено нейросетевое моделирование предложенных алгоритмов для стайного движения роевых частиц в неоднородной среде с препятствиями.



## Заключение

В работе представлен математический аппарат для разработки и анализа алгоритмов стайного движения роя. Рассмотрены два случая сбора роя в стаю в пространстве и фрагментации перед обходом множества препятствий. Математическим моделированием показана возможность использования такого рода алгоритмов для описания роевых механизмов взаимодействия.

## Список литературы

- [1] Берд Р., Стюарт В., Лайтфут Е. Явления переноса. М.: Химия, 1974. 688 с.
- [2] Под ред. Д. Дасгупты. Пер. с англ. под ред. А. А. Романюхи. Искусственные иммунные системы и их применение. М.: ФИЗМАТЛИТ, 2006. 344 с.
- [3] Чернышев Ю., Григорьев Г., Венцов Н. Искусственные иммунные системы: обзор и современное состояние. 2014г.
- [4] Роевой интеллект и его наиболее распространённые методы реализации Водолазский И. А, Егоров А. С., Краснов А. В. 2017
- [5] R. C. Eberhart, J. Kennedy, "A new optimizer using particle swarm theory" // Proceedings of the Sixth International Symposium on Micro Machine and Human Science MHS'95, стр. 39–43, 1995 г.
- [6] Y. Shi, R. Eberhart, "A modified particle swarm optimizer" // The 1998 IEEE International Conference on Evolutionary Computation Proceedings, стр. 69–73, 1998 г.
- [7] Y. Shi, R. Eberhart, "Empirical study of particle swarm optimization" // Proceedings of the 1999 IEEE Congress on Evolutionary Computation, стр. 1945–1950, 1999 г.
- [8] M. Clerc, J. Kennedy, "The particle swarm – explosion, stability, and convergence in a multidimensional complex space" // IEEE Transactions on Evolutionary Computation, №6 (1), стр. 58–73, 2002 г.
- [9] R. Mendes, J. Kennedy, J. Neves, "The fully informed particle swarm: Simpler, maybe better" // IEEE Transactions on Evolutionary Computation, №8 (3), стр. 204–210, 2004 г.

- [10] C. W. Reynolds, "Flocks, herds, and schools: a distributed behavioral model"// Computer Graphics (ACM SIGGRAPH '87 Conference Proceedings, Anaheim, California, July 1987), 21(4), стр. 25-34, Июль 1987 г.
- [11] J. Toner, Y. Tu, "Flocks, herds, and schools: A quantitative theory of flocking". Physical Review E, 58(4), стр. 4828-4858, Октябрь 1998г.
- [12] Ishida Y. "Fully distributed diagnosis by pdp learning algorithm: towards immune network pdp model"// IEEE Intern. Joint Conf. on Neural Networks, San Diego, 1990, vol. 1, стр. 777–782.
- [13] Timmis J., Neal M., Hunt J. An artificial immune system for data analysis. Biosystems, 2000, vol. 55, стр. 143–150.
- [14] Puteh M., Hamdan A. R., Omar K., Bakar A. A. Flexible immune network recognition system for mining heterogeneous data. 7th Intern. Conf. ICARIS-08, Springer, Berlin, Heidelberg, 2008, стр. 232–241.