

Лекция 6 :: Алгоритмы роевой робототехники

Ершов Н.М.

ershovnm@gmail.com

Роевая робототехника

- ▶ Роевая робототехника — это раздел теории роевого интеллекта, в котором рассматриваются вопросы построения распределенных робототехнических систем, а также разработки алгоритмов их управления.
- ▶ Роевые робототехнические системы состоят из большого числа одинаковых просто устроенных роботов.
- ▶ Желаемое коллективное поведение в таких системах должно возникать на основе локального взаимодействия роботов между собой и с окружающей их средой.



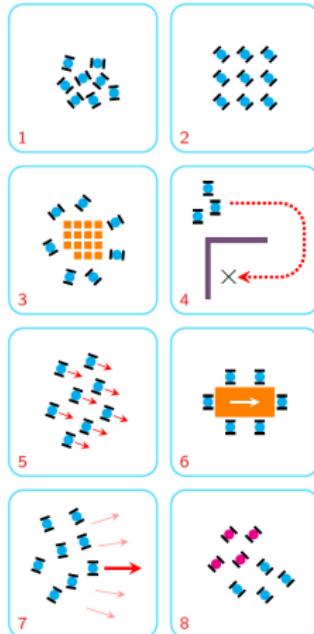
Ключевыми особенностями роевых систем роботов являются надежность, масштабируемость и адаптивность.

- ▶ *Надежность*: в роевых системах отсутствует встроенная специализация роботов, нет центрального управления, центрального хранилища данных и т.д. Такие системы слабо чувствительны к потерям отдельных роботов.
- ▶ *Масштабируемость*: размер роя роботов влияет только на скорость решения задачи и слабо влияет на возможность ее решения.
- ▶ *Адаптивность*: рой роботов способен решать поставленную задачу в широком диапазоне изменения ее параметров.

Роевая робототехника

Паттерны коллективного поведения в
роевой робототехнике:

1. агрегация роботов;
2. формообразование;
3. кластеризация и сортировка
предметов;
4. коллективная навигация;
5. координированное движение роя;
6. транспортировка объектов;
7. выработка консенсуса;
8. динамическая специализация.



- ▶ Актуальной задачей в области роевой робототехники остается *разработка алгоритмов управления такого рода системами*.
- ▶ Проблемой в данном случае является то, что мы хотим добиться от роя роботов желаемого *коллективного поведения*, программируя *локальное и одинаковое для всего роя поведение отдельных роботов*.
- ▶ По этой причине многие алгоритмы в роевой робототехнике строятся по аналогии и на основе моделей *коллективного поведения*, наблюдавшихся в живой природе (прежде всего у социальных насекомых).

Сортировка предметов

- ▶ Одной из первых задач, для которой был предложен распределенный алгоритм решения с помощью роя роботов, была задача кластеризации и сортировки объектов, рассмотренная Денебургом и его коллегами.
- ▶ В этой задаче имеется некоторая область, в случайных местах которой расположены предметы нескольких типов (например, под типом можно считать цвет объекта), в простейшем случае все предметы одинаковы, т.е. имеют одинаковый цвет.
- ▶ Требуется собрать все предметы в кластеры, согласно их типу. Например, в одной группе должны оказаться собранными предметы красного цвета, во второй — синего и т.д.

Муравьиный алгоритм сортировки

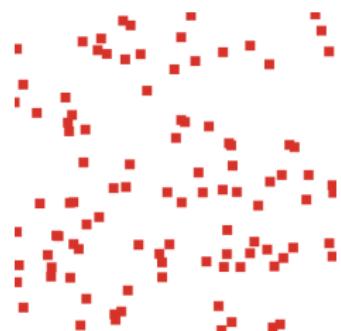
- ▶ Идея решения была основана на моделировании аналогичного поведения у муравьев и терmitов.
- ▶ Роботы (искусственные муравьи) выполняют случайные перемещения в заданной области. Каждый отдельный муравей может выполнять следующие два действия:
 1. поднять найденный предмет;
 2. сбросить предмет в текущем месте, если оно свободно.
- ▶ Эти действия выполняются муравьями вероятностным образом на основе анализа своей локальной окрестности: муравей поднимает найденный им предмет с вероятностью тем большей, чем меньше предметов того же типа он видит вокруг; муравей сбрасывает предмет, который он несет, с вероятностью, которая тем больше, чем больше аналогичных предметов он видит вокруг себя.

Случай одинаковых объектов

Для простейшего случая предметов одного типа вероятности подъема p и сброса q предмета вычисляются на основе числа n предметов, находящихся в области видимости заданного муравья:

$$p = \left(\frac{k_1}{k_1 + n} \right)^2 \text{ и } q = \left(\frac{n}{k_2 + n} \right)^2,$$

где $k_1 > 0$ и $k_2 > 0$ — управляющие параметры алгоритма.



Случай одинаковых объектов

Для простейшего случая предметов одного типа вероятности подъема p и сброса q предмета вычисляются на основе числа n предметов, находящихся в области видимости заданного муравья:

$$p = \left(\frac{k_1}{k_1 + n} \right)^2 \text{ и } q = \left(\frac{n}{k_2 + n} \right)^2,$$

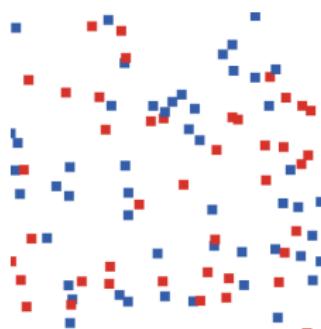
где $k_1 > 0$ и $k_2 > 0$ — управляющие параметры алгоритма.

Случай различных объектов

В задаче сортировки (когда имеются предметы нескольких типов) можно использовать те же две формулы, но параметр n вычислять следующим образом:

$$n = \max(0, n_0 - n_1),$$

где n_0 — число предметов в области видимости того же типа, что и текущий предмет (который муравей хочет взять или сбросить), а n_1 — число предметов других типов.



Случай различных объектов

В задаче сортировки (когда имеются предметы нескольких типов) можно использовать те же две формулы, но параметр n вычислять следующим образом:

$$n = \max(0, n_0 - n_1),$$

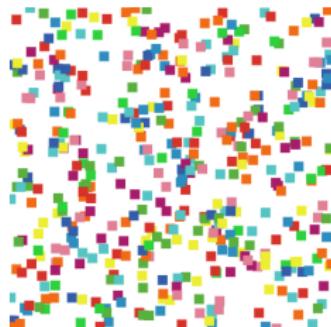
где n_0 — число предметов в области видимости того же типа, что и текущий предмет (который муравей хочет взять или сбросить), а n_1 — число предметов других типов.

Случай различных объектов

В задаче сортировки (когда имеются предметы нескольких типов) можно использовать те же две формулы, но параметр n вычислять следующим образом:

$$n = \max(0, n_0 - n_1),$$

где n_0 — число предметов в области видимости того же типа, что и текущий предмет (который муравей хочет взять или сбросить), а n_1 — число предметов других типов.



Случай различных объектов

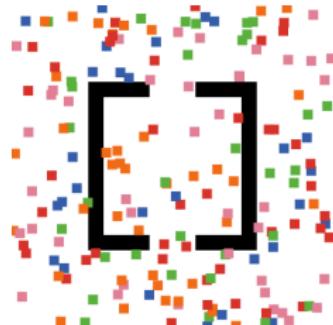
В задаче сортировки (когда имеются предметы нескольких типов) можно использовать те же две формулы, но параметр n вычислять следующим образом:

$$n = \max(0, n_0 - n_1),$$

где n_0 — число предметов в области видимости того же типа, что и текущий предмет (который муравей хочет взять или сбросить), а n_1 — число предметов других типов.

Препятствия

Представленный алгоритм работает и при наличии в рассматриваемой области различного рода препятствий — стен, неподвижных объектов и т.п.



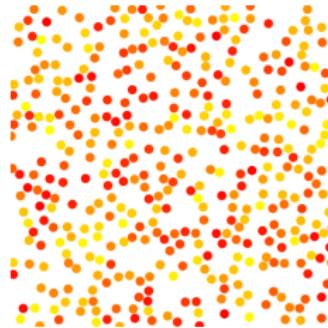
Препятствия

Представленный алгоритм работает и при наличии в рассматриваемой области различного рода препятствий — стен, неподвижных объектов и т.п.

Сортировка объектов с непрерывным типом

Если тип объекта v является не дискретной величиной, а непрерывной, например из диапазона от 0 до 1, то параметр n можно вычислять по правилу

$$n = \max \left(\sum_i \rho(v_i, v_0), 0 \right),$$



где суммирование ведется по всем свободным предметам в окрестности текущего предмета (с типом v_0), а функция $\rho(v, u) = 1 - 2|v - u|$ задает меру схожести двух предметов со значениями типа v и u .

Сортировка объектов с непрерывным типом

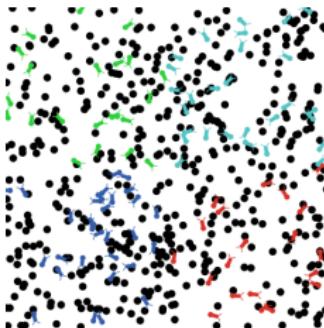
Если тип объекта v является не дискретной величиной, а непрерывной, например из диапазона от 0 до 1, то параметр n можно вычислять по правилу

$$n = \max \left(\sum_i \rho(v_i, v_0), 0 \right),$$

где суммирование ведется по всем свободным предметам в окрестности текущего предмета (с типом v_0), а функция $\rho(v, u) = 1 - 2|v - u|$ задает меру схожести двух предметов со значениями типа v и u .

Алгоритм прямого возвращения

Каждый робот в процессе перемещения по модели запоминает лучшее место для каждого типа предметов и при обнаружении свободного предмета поднимает его и несет прямо в запомненное место, где его и сбрасывает. При работе этого алгоритма возникает интересный эффект, когда роботы образуют конкурирующие группы, каждая из которых пытается собрать предметы в своем месте, «воруя» их из других мест.



Алгоритм прямого возвращения

Каждый робот в процессе перемещения по модели запоминает лучшее место для каждого типа предметов и при обнаружении свободного предмета поднимает его и несет прямо в запомненное место, где его и сбрасывает. При работе этого алгоритма возникает интересный эффект, когда роботы образуют конкурирующие группы, каждая из которых пытается собрать предметы в своем месте, «воруя» их из других мест.

Агрегация

- ▶ Алгоритм основан на моделировании аналогичного поведения у колоний тараканов.
- ▶ Роботы в такой модели могут находиться в двух состояниях — двигаться и покойться. Переход между состояниями происходит на вероятностной основе: чем больше покоящихся роботов вокруг текущего робота, тем больше вероятность того, что последний перейдет от движения к покоя и меньше вероятность обратного перехода.
- ▶ Если добавить к модели понятие освещенности и учесть ее в вероятностях перехода между состояниями (тараканы предпочитают останавливаться в темных местах), то рой будет с большой вероятностью выбирать одно из затемненных мест в качестве точки агрегации, т.е. децентрализованно вырабатывать консенсус

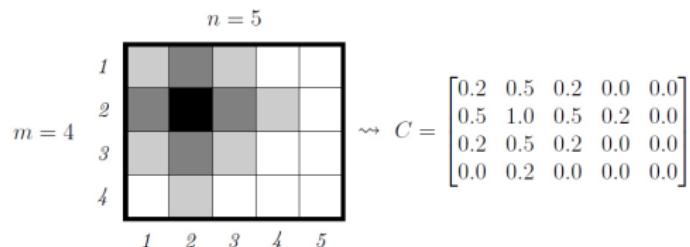
Агрегация

Задача уборки территории

- ▶ Рассмотрим практически важную задачу уборки некоторой ограниченной территории при наличии на этой территории разного рода препятствий.
- ▶ Для уборки небольших помещений в настоящее время активно используются коммерческие роботы-пылесосы.
- ▶ Однако, применение сразу большого числа такого типа устройств для уборки больших площадей со сложной геометрией является проблематичным, т.к. для организации их совместной деятельности требуется разработка *специальных алгоритмов управления*, чтобы работа такой группы роботов удовлетворяла сформулированным выше принципам роевой робототехники (надежность, гибкость и масштабируемость).

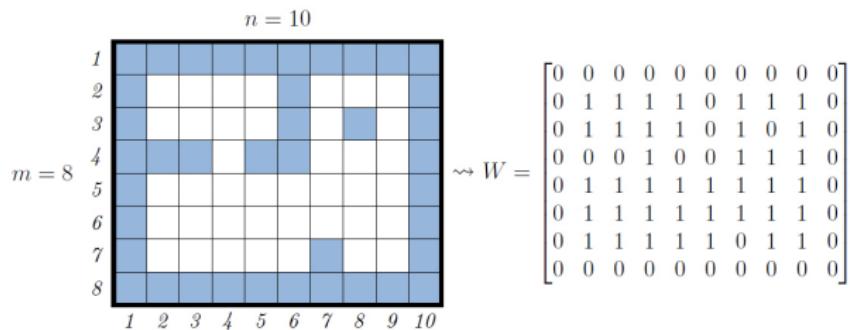
Задача уборки территории

- ▶ Предполагается, что территория, которая подлежит уборке, имеет прямоугольную форму.
- ▶ Для моделирования загрязненности вся область делится на конечное число квадратных клеток, каждая из которых характеризуется своей величиной загрязненности.



Задача уборки территории

- ▶ Часть клеток может быть закрытой для перемещения и коммуникации роботов — такие клетки формируют препятствия.



Задача уборки территории

- ▶ Предполагается, что все роботы роя имеют одинаковую круглую форму.
- ▶ Роботы находятся в непрерывном движении с постоянной скоростью, в процессе движения роботы могут менять направление движения.
- ▶ Роботы в рассматриваемой модели оснащены несколькими датчиками, сигналы которых могут использоваться для их управления.
- ▶ При своем движении каждый робот производит уборку того места, где он находится в текущий момент времени.

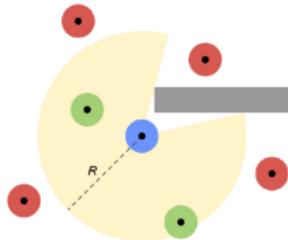
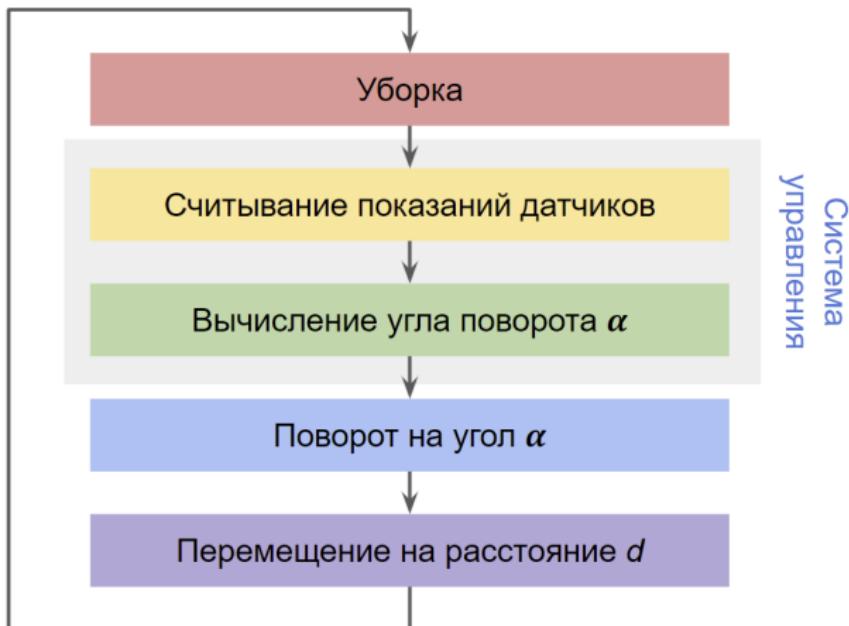


Схема функционирования робота



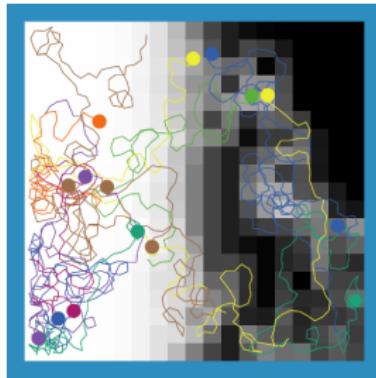
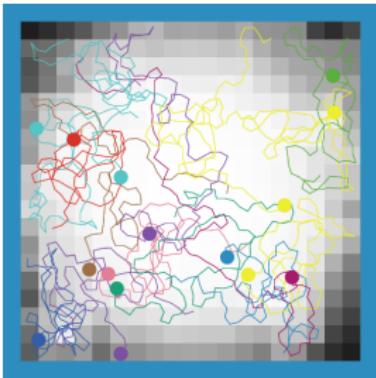
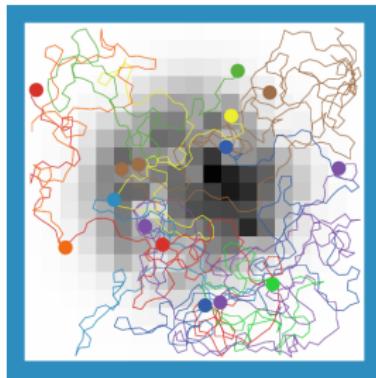
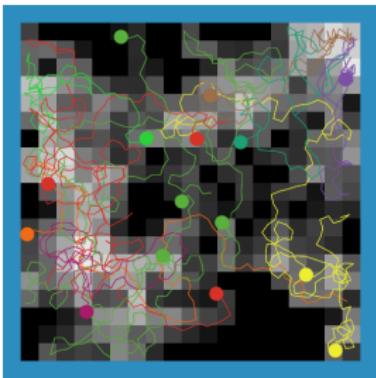
Алгоритмы

- ▶ Исследовались четыре алгоритма управления направлением движения роботов в описанной модели:
 1. случайный поиск (RND);
 2. алгоритм развертывания (DEP);
 3. бактериальный алгоритм на основе хемотаксиса (BAC);
 4. стохастический градиентный поиск (SGD).
- ▶ Общей частью всех алгоритмов является обработка столкновений роботов с препятствиями и с другими роботами.

Случайный поиск

- ▶ Простейшим алгоритмом управления является алгоритм случайного поиска, основанный на моделях броуновского движения и случайных блужданий.
- ▶ Работа этого алгоритма сводится, по сути, к случайному выбору нового направления движения робота в каждый дискретный момент времени.
- ▶ Формально, происходит выбор случайного угла α отклонения от текущего направления из интервала $[-90^\circ, 90^\circ]$.

Случайный поиск

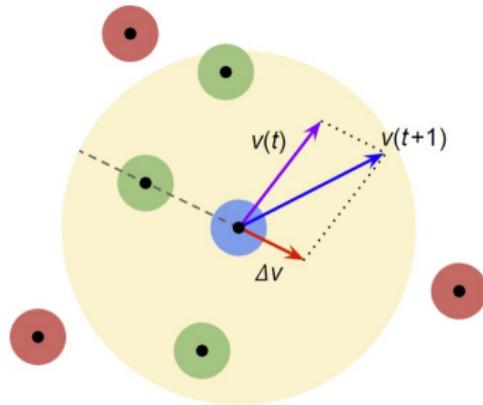


Случайный поиск

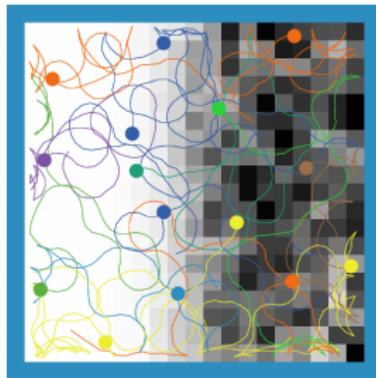
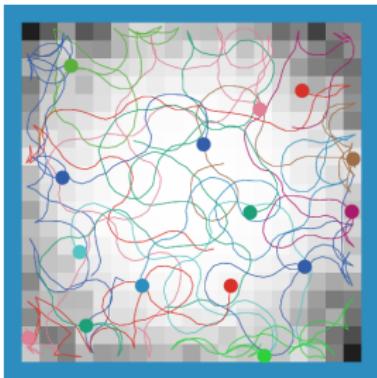
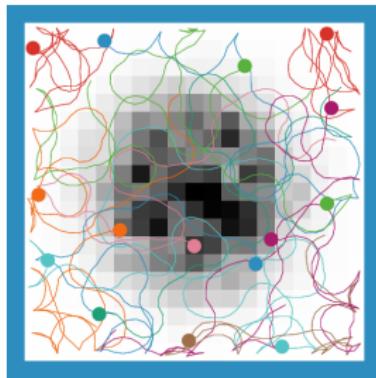
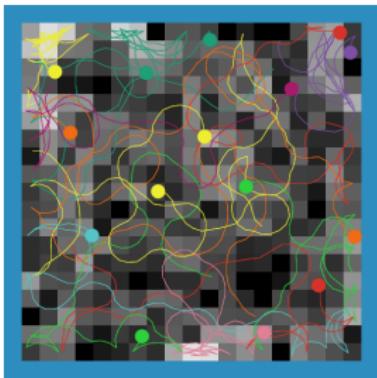
Алгоритм развертывания

- ▶ Алгоритм основан на использовании первого принципа Рейнолдса: находим ближайшего соседа заданного робота в его области видимости и, если такой робот есть, то изменяем скорость на некоторый вектор, направленный от найденного соседа.
- ▶ Аналогичная задача в области роевой робототехники известна, как задача *развертывания* роя, когда все роботы роя должно разместиться максимально равномерно по всей доступной этому рою области.

Алгоритм развертывания



Алгоритм развертывания

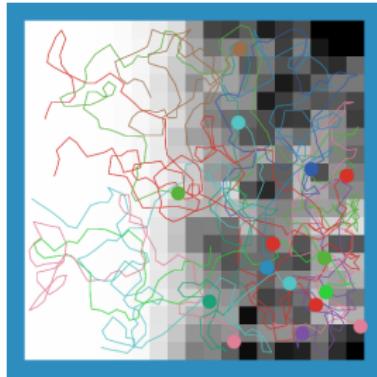
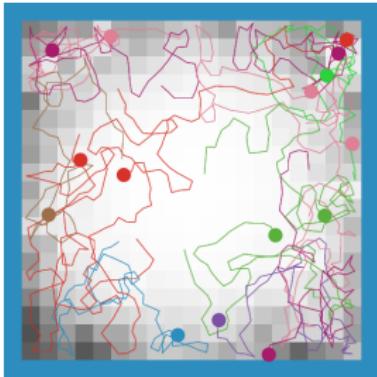
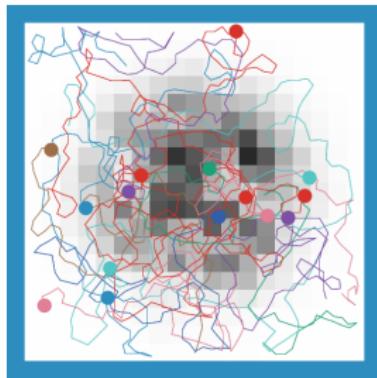
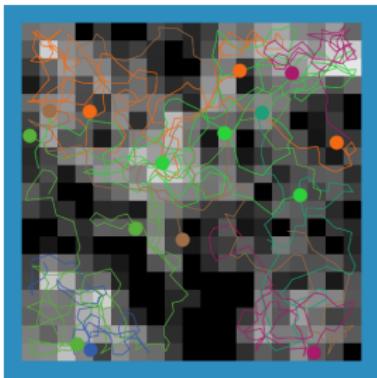


Алгоритм развертывания

Бактериальный алгоритм

- ▶ Первые два алгоритма являются *не адаптивными*, т.е. они работают в предположении, что должна быть убрана вся доступная роботу территория. В случаях, когда источник загрязнения локализован, такая стратегия будет скорее всего не эффективной — большая часть роботов будет перемещаться в областях, не требующих очистки.
- ▶ Алгоритм бактериального поиска построен на модели поведения колонии бактерий, выполняющих поиск питательных веществ. Каждая бактерия в процессе движения отслеживает градиент концентрации питательных веществ: если в новой точке концентрация питательных веществ ниже, чем в предыдущей, то бактерия с большой вероятностью выполняет поворот на случайный угол; если новая концентрация выше, то бактерия движется в правильном направлении и не меняет свое направление.

Бактериальный алгоритм

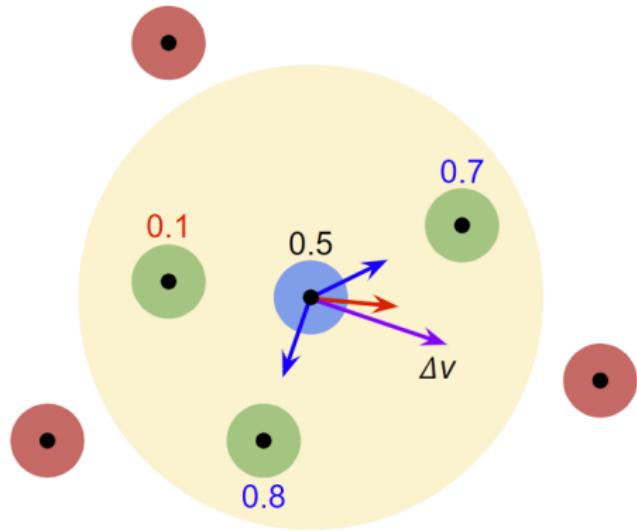


Бактериальный алгоритм

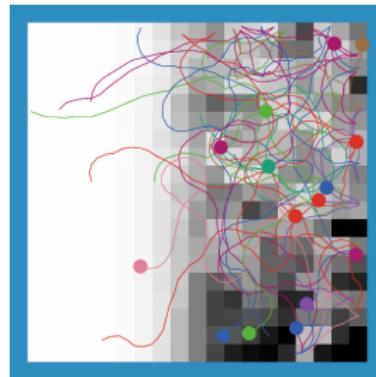
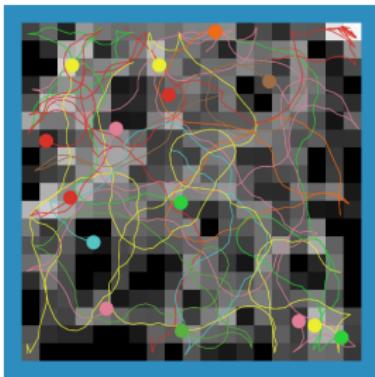
Стохастический градиентный поиск

- ▶ Более надежным является оценка градиента с помощью измерений уровня загрязненности сразу в нескольких точках области, для этого можно использовать измерения загрязненности других роботов, находящихся в области видимости заданного робота.
- ▶ По сути, такой подход является некоторой вариацией широко применяющегося в машинном обучении метода стохастического градиентного спуска.

Стохастический градиентный поиск



Стохастический градиентный поиск

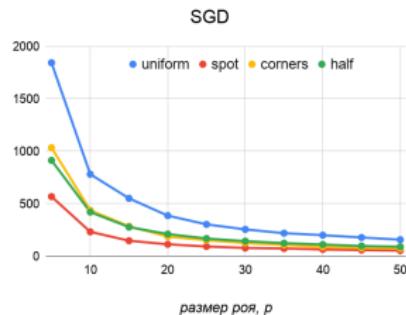
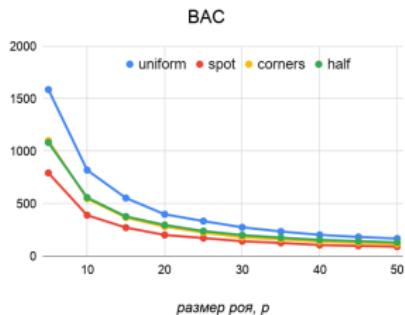
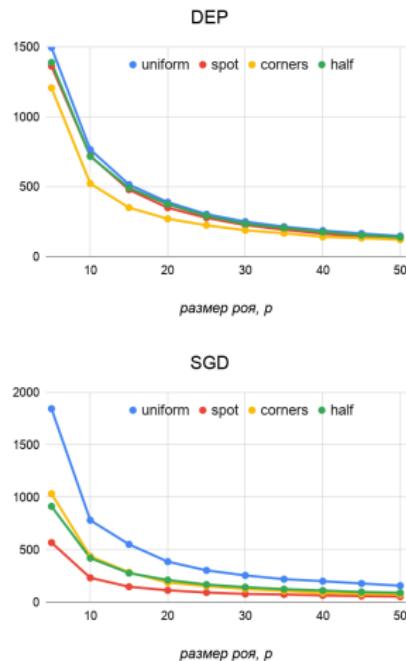
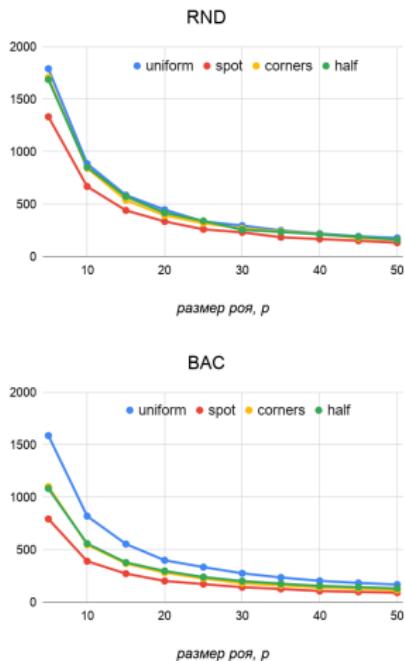


Стохастический градиентный поиск

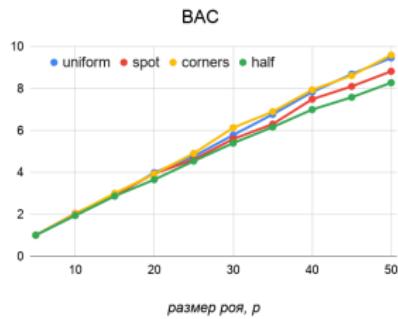
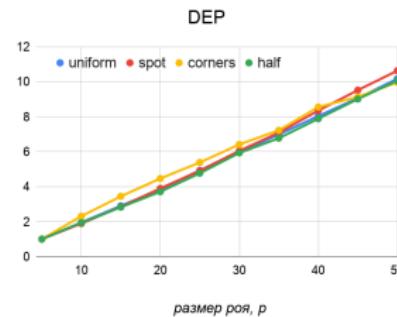
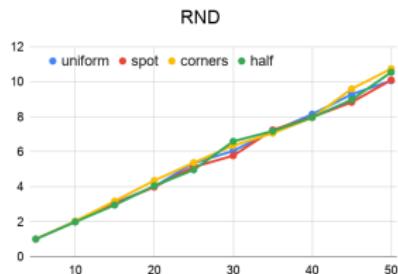
Параллельная обработка информации

- ▶ Во многих работах по роевым робототехническим системам эти системы рассматриваются как некоторый специальный подкласс параллельных вычислительных систем.
- ▶ Параллелизм в решении некоторой «вычислительной» задачи.
- ▶ Особенность: динамическая схема коммуникации.
- ▶ Поведение подобных систем можно описывать с использованием соответствующих методик анализа эффективности параллельных алгоритмов.

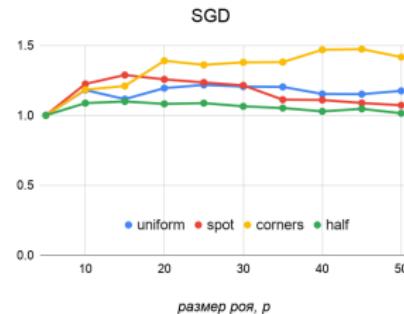
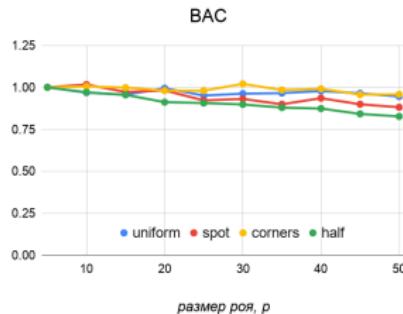
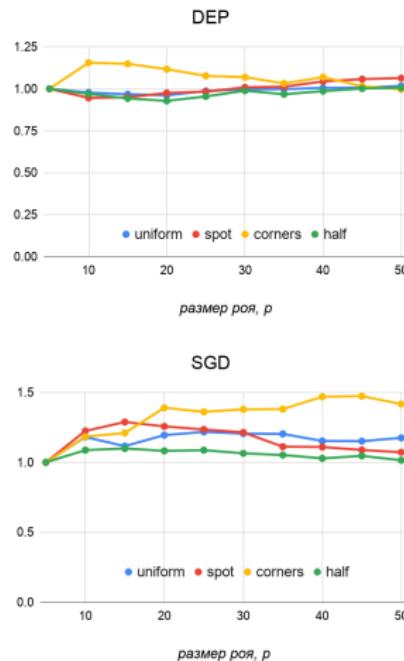
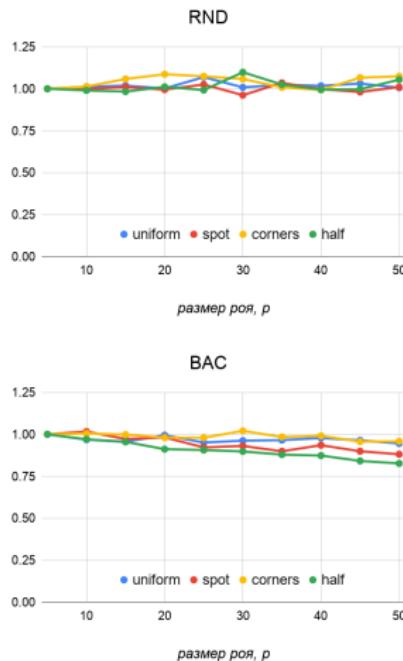
Время работы



Ускорение



Эффективность



Учет препятствий

Библиография

1. Beni G., Wang J., Swarm Intelligence in Cellular Robotic Systems, 1989.
2. <https://youtu.be/JmyTJSYw77g> — Kilobot swarm
3. <https://youtu.be/fjyTkagc8BI> — How Crazy Ants Carry Heavy Loads
4. J. L. Deneubourg et al., The dynamics of collective sorting robot-like ants and ant-like robots, 1990.
5. S. Garnier et al., Aggregation behaviour as a source of collective decision in a group of cockroach-like robots, 2005.

Спасибо за внимание!