

Санкт-Петербургский государственный университет
Прикладная математика и информатика
Управление и обработка информации в кибернетических и робототехнических
системах

Сухоруков Сергей Алексеевич

ОПТИМАЛЬНОЕ ДЕЦЕНТРАЛИЗОВАННОЕ САМОРАСПРЕДЕЛЕНИЕ
МОБИЛЬНОЙ СЕНСОРНОЙ СЕТИ В ПЛОСКОЙ ОБЛАСТИ

Выпускная квалификационная работа

Научный руководитель:

д. ф.-м. н., профессор А. С. Матвеев

Рецензент:

к. ф.-м. н., доцент С. В. Гусев

Санкт-Петербург

2018

Saint Petersburg State University
Applied Mathematics and Computer Science
Control and Processing of Information in Cybernetical and Robotic Systems

Sukhorukov Sergei

OPTIMAL DISTRIBUTED CONTROL OF SELF-DEPLOYING ROBOTIC SENSOR
NETWORK IN A PLANE

Graduation Project

Scientific Supervisor:
Professor Aleksei Matveev

Reviewer:
Associate Professor Sergei Gusev

Saint Petersburg
2018

Оглавление

Введение	4
Глава 1. Алгоритм самораспределения сенсоров для покрытия области на плоскости	6
1.1. Краткий обзор предшествующих результатов	6
1.2. Постановка задачи	7
1.3. Основные предположения	9
1.4. Базовый алгоритм решения задачи	12
1.4.1. Общая схема алгоритма	12
1.4.2. Описание алгоритма	13
1.5. Предложенная модификация базового алгоритма	15
1.5.1. Общая схема	15
1.5.2. Первая половина шага (целые моменты)	15
1.5.3. Вторая половина шага (дробные моменты)	17
Глава 2. Анализ предложенного алгоритма	23
2.1. Сходимость предложенного правила достижения консенсуса	23
2.2. Экспериментальное сравнение двух алгоритмов достижения консенсуса	26
2.3. Сравнение базового и модифицированного алгоритмов покрытия области методом компьютерного моделирования	28
Заключение	31
Список литературы	32
Приложение А. Иллюстрации к работе	34
Приложение Б. Полное описание алгоритма	46
Б.1. Пояснение	46
Б.2. Целые моменты	46
Б.3. Дробные моменты	47

Введение

Важным этапом в развитии экономики стало формирование разделения труда и образование мануфактур. Практика доказала эффективность разделения производства на специальные виды деятельности, поручаемые отдельным коллективам, специализирующимся в этой области, дальнейшего дробления всего цикла работ на части, поручаемые отдельным работникам, и итогового объединения результатов в конечный продукт. Аналогично многие современные технологии формируются, опираясь на принципы эффективной организации, распределения и использования ресурсов. В качестве прорывного аналога разделения труда и организации мануфактурного производства можно рассматривать современную парадигму интеллектуальных децентрализованных мультиагентных систем, которая применима к широкому классу задач управления и наблюдения. К ее основным принципам относится достижение определенной «глобальной» цели за счет композиции децентрализованных суверенных действий и решений членов команды (агентов). Как и в мануфактурном производстве, отдельный агент может не знать, откуда к нему поступают материалы (агенты не обладают полной информацией о глобальной проблеме), а весь цикл производства продукции оказывается согласован между участниками (взаимодействия между агентами синхронизированы). Чем лучше организован процесс распределения обязанностей, передачи готовых результатов в конвейерной цепочке создания продукции, тем более продуктивно работает мануфактура. Решение с мультиагентной системой должно также основываться на максимально эффективном взаимодействии агентов.

В совокупности все алгоритмы с участием мультиагентной системы можно разделить на два класса:

1. Координация по распределению задач между агентами и последующая композиция решений происходит под контролем некоторого общего координационного «центра».
2. Поведение каждого агента подчиняется общему правилу, распределение задач происходит в процессе взаимодействия агентов между собой, а координирующий «центр» отсутствует.

У каждого подхода имеются свои плюсы и минусы. Наличие центра упрощает процесс решения задачи, так как следить за протекающими процессами в системе проще из «од-

ной точки», нежели оценивать поведение каждого агента по отдельности. Децентрализация в свою очередь дает ослабление таких эффектов как потеря данных, задержки в передаче сообщений в сети, скорость принятия решения и т. д. Эти проблемы переносятся на локальный уровень. Однако, более сложным для исследования оказывается вопрос о синхронизации действий и достижении конечной цели. Подход мультиагентных систем хорош еще и тем, что позволяет не сосредотачивать все вычислительные ресурсы в одной области, а по возможности распределить нагрузку равномерно по всей сети. Тем самым, один мощный по производительности ресурс можно заменить несколькими более слабыми.

При решении реальных задач встречается проблема «контроля» некоторой области. Под «контролем» области можно понимать отслеживание движения определенных объектов в некоторой области. Например определение области разлившегося в море нефтяного пятна, отслеживание траектории движения охраняемых животных в заповедной зоне. Еще одной интерпретацией может служить задача перемещения одного или нескольких объектов области из одной точки в другую за сравнительно небольшой промежуток времени. Иллюстрацией может служить задача разбора образовавшихся завалов при стихийном бедствии или вариант организации получения и сдачи на хранения вещей для временно образованной складской зоны.

Для достижения «контроля» области во всех случаях необходимо и достаточно, чтобы любая точка из области лежала в операционной зоне хотя бы одного из агентов. В данной работе будем рассматривать случай, когда границы области не меняются со временем, но заранее их точное положение неизвестно. В силу ограниченности коммуникационных возможностей и других ресурсов, построение алгоритма будет децентрализованным. Для конкретности рассмотрен случай, когда задача заключается в мониторинге области. В этой связи речь пойдет о сети сенсоров, а операционная зона каждого из них именуется зоной мониторинга. Вместе с тем результаты работы равноприменимы и в иных ситуациях, например, когда сеть образована актуаторами и операционная зона - это зона, в пределах которой данный актуатор может совершать определенные действия.

Глава 1

Алгоритм самораспределения сенсоров для покрытия области на плоскости

1.1. Краткий обзор предшествующих результатов

В основополагающей работе [1] были введены следующие три основные типа организации сенсорной сети в задачах покрытия геометрических объектов на плоскости:

1. Покрытие области — статическое распределение сенсоров, доставляющее максимум вероятности обнаружения целей в заданной области. [2] [3]
2. Барьерное покрытие — статическое распределение сенсоров, минимизирующее вероятность незамеченного прохода целей через «барьер», то есть кривую или узкую полосу, ограничивающую определенную область. [4] [5]
3. Заметающее покрытие — динамическое (меняющееся со временем) распределение сенсоров, одновременно минимизирующее количество пропусков целей на единицу площади и максимизирующее число обнаружений в единицу времени. [6]

Проблеме статического покрытия области плоскости посвящено значительное число работ; с их обзором можно ознакомиться в [7]. Вместе с тем в их фокусе было выявление оптимального или эффективного в том или ином смысле итогового статического распределения, а алгоритмы достижения этого распределения за счет децентрализованного самораспространения команды сенсоров на основе локальных данных в основном оставались за рамками исследования. Одной из первых работ, в которой был предложен и обоснован алгоритм такого рода, является статья [8], впоследствии доработанная в [9]. Алгоритм сочетает метод децентрализованного достижения консенсуса между сенсорами относительно итогового распределения команды и правило перемещения сенсоров. Консенсус достигается за счет обмена мнениями по каналам связи между текущими ближайшими соседями, то есть сенсорами, оказавшимися в пределах априори заданной конечной дистанции коммуникации друг от друга. Перемещение производится в ближайšie «вакантные» позиции согласованного на основе консенсуса распределения.

При этом была использована частичная рандомизация алгоритма для преодоления негативных последствий децентрализации, например, в форме перемещения нескольких сенсоров в одну и ту же вакантную позицию или наоборот, «опустошение» позиции в результате массового бегства из неё всех сенсоров. Обсуждаемый алгоритм снабжен математически строгими гарантиями сходимости (с вероятностью единица) к эффективному статическому покрытию области. Вместе с тем компьютерное моделирование продемонстрировало невысокую скорость сходимости в определенных сценариях.

Целью дипломной работы является модификация указанного алгоритма с целью повышения скорости его сходимости.

Предложенная модификация лежит в русле следующих идей:

1. Сенсоры по каналам связи передают друг другу «подсказки», разъясняющие направление на вакантные позиции.
2. Обеспечив покрытие некоторой подобласти, сенсорная сеть стремится не терять контроль над нею на следующих шагах. Для этого сенсоры делятся на две группы; задача одной из них — контролировать уже «покрытую» подобласть, задача другой — расширить её.
3. Для достижения консенсуса (относительно итогового распределения) целесообразно использовать правило с более высокой скоростью сходимости.

Важным требованием является вычислительная простота предлагаемой модификации ввиду типичной для сенсорных сетей малой вычислительной мощности их элементов.

1.2. Постановка задачи

На плоскости задано ограниченное, непустое, связное множество G произвольной формы, граница которой априори неизвестна. Имеется группа роботизированных мобильных сенсоров. Каждый из них снабжен собственной декартовой системой отсчета и способен выполнить следующие операции:

- Осуществить мониторинг диска (именуемого зоной мониторинга), образованного точками, удаленными от сенсора на расстояние не более r_0 . В частности, сенсор

определяет относительное положение и ориентацию «компаньонов» (точнее, ориентацию связанной с каждым из них системы отсчета), попавших в эту зону.

- Обмениваться данными по каналу связи с «компаньонами», удаленными от него на расстояние не более r_c .
- Определить, принадлежит ли точка из зоны мониторинга области G и её δ -окрестности, где величина $\delta > 0$ уточняется далее.
- Выбрав точку из зоны мониторинга, переместиться в неё за единичное время или быстрее.

Предположение 1. *Справедливы следующие утверждения:*

- *Роботизированные сенсоры имеют одинаковые характеристики, в частности, одинаковые радиусы мониторинга r_0 и коммуникации r_c ;*
- *Роботизированные сенсоры не различают «компаньонов» друг от друга;*
- *В начальный момент времени сенсоры находятся внутри области G .*

В остальном их начальное расположение произвольно.

Требуется разработать алгоритм перемещения сенсоров по плоскости со следующими свойствами:

- a) Децентрализованность: каждый сенсор принимает решения индивидуально на основе данных, получаемых за счет мониторинга доступной ему зоны и по каналам связи;
- b) Однородность: сенсоры принимают суверенные решения, руководствуясь общим для всех правилом;
- c) Покрытие: по истечении некоторого конечного отрезка времени t_f каждая точка области G оказывается в зоне мониторинга хотя бы одного сенсора;
- d) Прекращение перемещений: по истечении конечного времени, сенсоры прекращают движения в позициях, обладающих свойством c).

Требования к области G не являются ограничительными, так как в случае несколько компонент связности решение задачи можно получить за счет покрытия этих компонент по отдельности. Конечность радиусов r_0 и r_c обычно обусловлена физическими или техническими ограничениями агентов, например, мощностью излучаемого сигнала, используемого для мониторинга окрестностей, разрешением видеокамеры, радиусом уверенного приёма радиоволн и т.п. Менее тривиальна возможность классификации точек на предмет принадлежности той или иной окрестности области G . Анонимность сенсоров связана с их стандартизацией и нежеланием расходовать ресурсы на решение задачи идентификации и согласуется с концепцией построения децентрализованного алгоритма для мультиагентной системы с полностью взаимозаменяемыми элементами.

1.3. Основные предположения

Прежде чем сформулировать алгоритм, введем некоторые обозначения и определения. Под мощностью конечного множества F будем понимать число его элементов и обозначать как $|F|$. Стандартную евклидову норму двумерного вектора $\mathbf{v} = (v_1, v_2)$ будем обозначать как

$$\|\mathbf{v}\|_2 = \sqrt{v_1^2 + v_2^2}.$$

Под g -окрестностью области G понимается множество

$$G^g := \{\mathbf{x} \in \mathbb{R}^2 : \exists \mathbf{y} \in G \ \& \ \|\mathbf{y} - \mathbf{x}\|_2 < g\}.$$

Для любого $g > 0$ имеет место включение $G \subset G^g$.

Определение 1. *Триангуляционной решеткой (сеткой) T мелкости g будем называть произвольное разбиение плоскости на равносторонние треугольники со стороной $\sqrt{3}g$ (Рис 1.1). Вершины этих треугольников будем называть узлами триангуляционной решетки (узлами сетки) и будем обозначать как $Q(T)$.*

Замечание 1. *Мелкость решетки инвариантна относительно её поворотов и параллельного переноса, а также инвариантна по отношению к выбору декартовой системы координат.*

Стороны всех треугольников триангуляционной решетки T образуют три семейства прямых с тремя разными углами наклона, отличающимися друг от друга поворотом на угол $\pi/3$.

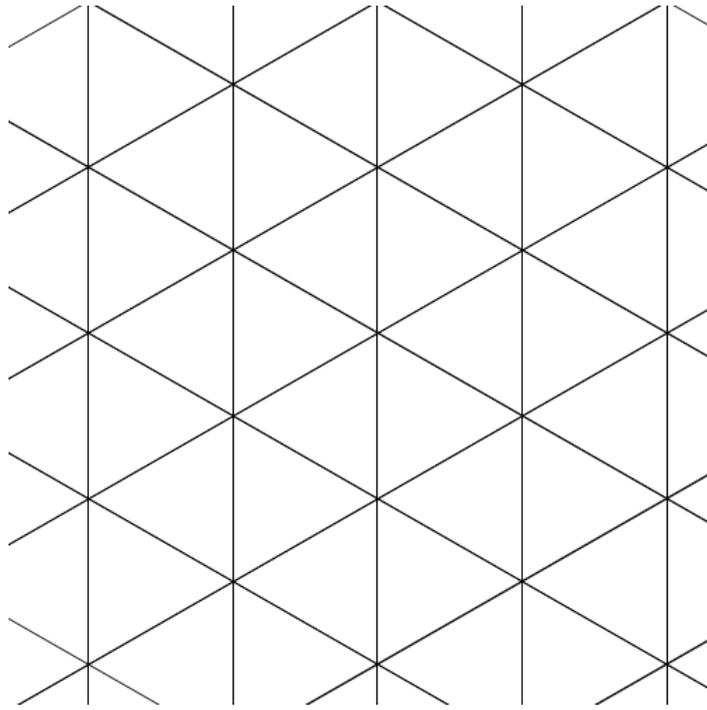


Рис. 1.1. Пример триангуляционной сетки

Определение 2. Угол наклона одной из прямых будем называть углом направляющей сетки или просто углом сетки.

Фактически этот параметр сетки определен с точностью до $\pi/3$.

Замечание 2. Для задания триангуляционной решетки T достаточно указать следующие величины:

1. Мелкость решетки g ;
2. Угол наклона направляющей сетки θ ;
3. Координаты одного из узлов сетки (x, y) .

Доказательство. Пусть на плоскости задан один равносторонний треугольник. Очевидно, что по нему однозначно восстанавливается вся триангуляционная решетка. В свою очередь, для задания одного равностороннего треугольника на плоскости необходимо и достаточно указать длину его стороны, угол наклона прямой, содержащей одну из его сторон, и расположение одной из вершин. Все перечисленные параметры из замечания как раз и задают эти величины. \square

Для любой триангуляционной сетки T можем рассмотреть множество ее узлов $Q(T)$, лежащих в некоторой области G

$$Q_G(T) := \{\mathbf{x} \in \mathbb{R}^2 : \mathbf{x} \in G \& \mathbf{x} \in Q(T)\}.$$

Замечание 3. Если множество G ограничено, то $Q_G(T)$ является конечным множеством для любой триангуляционной решетки T .

Определение 3. Назовем конечный набор точек плоскости W , полным g -покрытием области G если для любой точки \mathbf{x} из G найдется некоторая точка \mathbf{y} из W , что расстояние между ними не превосходит g

$$\|\mathbf{x} - \mathbf{y}\|_2 \leq g.$$

Таким образом, для достижения цели сенсоры в конечном итоге должны образовать полное r_0 -покрытие области G .

Лемма 1. Любое множество $Q_{G^g}(T)$ при $g \leq r_0$ образует полное r_0 -покрытие G .

Доказательство. Возьмем произвольную точку \mathbf{x} из G . Для нее очевидно существует узел q из $Q(T)$, расстояние до которого не превосходит g , а значит $q \in G^g$. Таким образом, нашелся элемент $q \in Q_{G^g}(T)$, для которого

$$\|\mathbf{x} - q\|_2 \leq g \leq r_0.$$

□

Замечание 4. Пусть задано значение $R_{tr} > 0$ и $g + R_{tr} \leq r_0$. После сдвига каждой точки из $Q_{G^g}(T)$ на некоторое расстояние, не превосходящее R_{tr} , множество $Q_{G^g}(T)$ останется полным r_0 -покрытием области G .

Доказательство. Так как по замечанию 3 $Q_{G^g}(T)$ конечно, достаточно доказать, что сдвиг одной точки оставляет множество $Q_{G^g}(T)$ r_0 -покрытием области G . Для большего числа сдвигов можно воспользоваться индукцией. Пусть $q \in Q_{G^g}(T)$ перешла в точку q' . Тогда, для всех точек $\mathbf{x} \in G$, для которых точка q была ближайшей из множества $Q_{G^g}(T)$ будет выполнено следующее соотношение

$$\|\mathbf{x} - q'\|_2 \leq \|\mathbf{x} - q\|_2 + \|q - q'\|_2 \leq g + R_{tr} \leq r_0.$$

Для остальных же точки G расстояние до не сдвинутых элементов множества $Q_{G^g}(T)$ не изменилось, а значит это по прежнему r_0 -покрытие области G . □

Обозначим максимум числа элементов множества $Q_{G^g}(T)$ по всем возможным триангуляционным решеткам T мелкости g как

$$M_{G^g} = \max_T |Q_{G^g}(T)|,$$

а минимально возможное число узлов в g -покрытии множества G^g за m_{G^g} . Тогда верна следующая теорема, доказанная в [10].

Теорема 1. *Для любого ограниченного и измеримого по Лебегу множества G верно предельное соотношение*

$$\lim_{g \rightarrow 0} g^2 \cdot M_{G^g} = \lim_{g \rightarrow 0} g^2 \cdot m_{G^g} = \frac{2\sqrt{3}}{9} \mu(G),$$

где за $\mu(G)$ обозначена мера Лебега множества G .

Таким образом, расположение агентов по узлам некоторой триангуляционной сетки можно считать (почти) оптимальным для произвольной области G и при достаточно малых радиусах мониторинга r_0 и, следовательно, мелкости $g \leq r_0$ используемой триангуляционной сетки.

1.4. Базовый алгоритм решения задачи

Данный параграф дает описание предложенного в [9] алгоритма, модификация которого — цель данной работы.

1.4.1. Общая схема алгоритма

Для удобства формулировки алгоритма используем нумерацию сенсоров, предполагая, что она доступна внешнему наблюдателю, но неизвестна самим сенсорам. Корректность использования такой нумерации для формулировки алгоритма подразумевает независимость предлагаемых решений от порядка нумерации. Предполагаем, что масштаб осей у системы координат, связанной с сенсором, общий для всех сенсоров. Каждый сенсор в своей системе координат строит свою собственную триангуляционную решетку общей для всех сенсоров мелкости g .

Во времени алгоритм функционирует по-тактово, такты последовательно занумерованы целыми кратными числа $1/2$ и разделены единичным отрезком времени. Пара, состоящая из одного целого и одного дробного такта, образует один шаг алгоритма. В

целые моменты времени (такты) агенты занимаются согласованием параметров сеток и сохранением собственной близости к своей сетке, в дробные происходит перераспределение агентов по узлам их сеток.

Обозначим через $\theta_i(k)$ угол направляющей сетки у i -го сенсора в начале шага k , а через $d_i(k) = (x_i(k), y_i(k))$ — декартовы координаты точки узла сетки у i -го сенсора. За $\mathcal{N}_i(k)$ будем обозначать множество сенсоров (представленных своими номерами), которые лежат в зоне мониторинга и в зоне коммуникации i -го сенсора в начале шага k . За $T_i(k)$ будем обозначать триангуляционную решетку, построенную по параметрам $d_i(k)$ и $\theta_i(k)$. Задана допустимая погрешность $R_{tr} \ll g$ локализации сенсора около узла сетки.

1.4.2. Описание алгоритма

В целые моменты k

Каждый сенсор i выполняет последовательно следующие операции:

1. Передает текущие значения параметров своей сетки соседям из $\mathcal{N}_i(k)$;
2. Получает параметры сеток от сенсоров из $\mathcal{N}_i(k)$;
3. Определяет относительное положение сенсоров из $\mathcal{N}_i(k)$ в своей системе отсчета и ориентацию их систем отсчета относительно собственной системы;
4. Для всех соседей j из $\mathcal{N}_i(k)$ производит пересчет полученных значений параметров $\theta_j(k)$, $d_j(k)$ из чужой системы отсчета в свою систему, используя данные из п. 2 и 3, пересчитанные таким образом значения параметров обозначим через $\tilde{\theta}_j(k)$, $\tilde{d}_j(k)$;
5. Обновляет свою триангуляционную сетку¹ по правилу

$$\theta_i(k+1) = \frac{\sum_{j \in \mathcal{N}_i(k)} \tilde{\theta}_j(k)}{|\mathcal{N}_i(k)|}, \quad d_i(k+1) = \frac{\sum_{j \in \mathcal{N}_i(k)} \tilde{d}_j(k)}{|\mathcal{N}_i(k)|};$$

6. Если расстояние до ближайшего узла новой сетки не превосходит R_{tr} , то остается на месте;
7. Иначе

¹ $d_i(k)$ является двумерным вектором; алгебраические операции над такими векторами выполняются покомпонентно.

- а. Определяется множество $S_i(k)$ узлов новой сетки, расстояние до которых не превосходит $2g + R_{tr}/2$;
- б. Выбрасывает из $S_i(k)$ узлы, $(R_{tr}/2)$ -окрестность которых не пересекается с g -окрестностью области G ;
- в. С вероятностью $\frac{1}{|S_i(k)|}$ выбирается один из узлов $S_i(k)$;
- г. Перемещается в пересечение $(R_{tr}/2)$ -окрестности этого узла и g -окрестности области G .

К началу дробного момента шага k все вычисления и перемещения завершаются. Далее в дробный момент времени шага k определяется узел сетки, в окрестность которого сенсор затем перемещается к началу шага $k + 1$. Для этого сенсор i выполняет следующие операции:

1. Формирует множество $F_i^*(k)$ узлов-кандидатов, последовательно выполняя следующие шаги
 - а. Определяется ближайший к сенсору i узел сетки $f_i(k)$;
 - б. Определяет шесть узлов текущей сетки $Q(T_i(k + 1))$, соседних узлу $f_i(k)$; в совокупности они образуют множество, обозначаемое через $F_i^*(k)$;
 - в. Удаляется из $F_i^*(k)$ узлы, в R_{tr} -окрестности которых есть сенсор;
 - г. Удаляет из $F_i^*(k)$ узлы, R_{tr} -окрестность которых не пересекается с g -окрестностью области G .
2. Если $|F_i^*(k)| = 0$, остается на месте;
3. В противном случае
 - а. Добавляем к множеству $F_i^*(k)$ ранее найденный узел $f_i(k)$ и с вероятностью $\frac{1}{|F_i^*(k)|}$ выбирается один из узлов $F_i^*(k)$;
 - б. Сенсор перемещается в пересечение $R_{tr}/2$ окрестности этого узла и g -окрестности области G .

К началу следующего шага $k + 1$ все агенты заканчивают свои вычисления и перемещения.

1.5. Предложенная модификация базового алгоритма

1.5.1. Общая схема

Общая структура модифицированного алгоритма совпадает со структурой базового алгоритма. Как и ранее, в целые моменты сенсоры выполняют действия, направленные на синхронизацию параметров θ и d своих сеток и при необходимости переходят в малую окрестность некоторого узла обновленной сетки. В дробные моменты сенсоры принимают решения о перемещении в другой узел сетки. При этом в отличие от базового алгоритма, они опираются на имеющееся у каждого из них индивидуальное значение дополнительного параметра p , который характеризует удаленность соответствующего сенсора от пустых узлов сеток. В приложении Б представлено полное описание модифицированного алгоритма.

Перейдем к описанию модифицированного алгоритма, сохраняя обозначения и условия предыдущего параграфа. Систему координат, связанную с сенсором, и ассоциированные с ней величины будем нумеровать тем же индексом, что и сенсор.

1.5.2. Первая половина шага (целые моменты)

Действия сенсора i в целые моменты времени k те же, что и в случае базового алгоритма, за исключением пересчета параметров сеток. Именно, сенсор выполняет следующие операции:

1. Передает значения параметров своей сетки $\theta_i(k)$, $d_i(k)$ сенсорам из $\mathcal{N}_i(k)$;
2. Принимает параметры сеток от сенсоров из $\mathcal{N}_i(k)$;
3. Определяет относительное положение и ориентацию сенсоров из $\mathcal{N}_i(k)$ в своей системе отсчета;
4. Сенсор строит правую декартову систему отсчета, ось абсцисс которой ориентирована по направляющей решетки данного сенсора в начале шага k ;
5. Для всех соседей j из $\mathcal{N}_i(k)$ производит пересчет полученных значений параметров $\theta_j(k)$, $d_j(k)$ в только что полученную систему координат, используя данные из пунктов 2 и 3; рассчитанные таким образом параметры обозначим за $\tilde{\theta}_j(k)$, $\tilde{d}_j(k)$;
6. Изменяет параметры своей сетки по правилам:

а. Для нахождения абсциссы x_i использует формулу

$$\tilde{x}_i(k+1) = \min_{j \in \mathcal{N}_i(k)} \tilde{x}_j(k);$$

и полученное значение $\tilde{x}_i(k+1)$ переносит в свою систему отсчета.

б. Ордината $y_i(k+1)$ и угол $\theta_i(k+1)$ определяются аналогично (с точностью до замены x на y или θ);

7. Определяет расстояние до ближайшего узла обновленной сетки $d_i(k+1)$ и если оно не превосходит R_{tr} , то остается на месте и заканчивает целый момент;

8. Если указанное расстояние превосходит R_{tr} ,

а. Определяет множество $S_i(k)$ узлов обновленной сетки, для которых расстояние до сенсора не превосходит $2g + R_{tr}/2$;

б. Выбрасывает из $S_i(k)$ узлы, $(R_{tr}/2)$ -окрестность которых не пересекается с g -окрестностью области G ;

в. С вероятностью $\frac{1}{|S_i(k)|}$ выбирает один из узлов $S_i(k)$;

г. Перемещается в пересечение $(R_{tr}/2)$ -окрестность этого узла и g -окрестности области G .

Замечание 5. Множество $\mathcal{N}_i(k)$ на каждом шаге не пусто, так как сам сенсор i всегда в него попадает.

Замечание 6. К началу дробного момента шага k , агенты остаются в g -окрестности области G и в R_{tr} -окрестности узла сетки.

Замечание 7. Пересчет координат узлов сеток и углов направляющих сеток в пункте 6 инвариантен относительно аффинных преобразований исходной декартовой системы.

Доказательство. Поворот и сдвиг исходной системы координат не влияет на ориентацию строящегося в пункте 4 базиса, а определение минимального элемента инвариантно относительно сдвига центра системы координат. \square

Вывод. Для каждого агента результат пересчета сетки не зависит от выбора декартовой системы координат. В частности, можно считать, что вычисления производятся в некоторой абсолютной системе координат.

1.5.3. Вторая половина шага (дробные моменты)

Описание алгоритма

По завершению первой половины шага каждый агент находится вблизи узла своей сетки. В начале второй половины шага каждый агент принимает решение, останется ли он на месте чтобы «контролировать» этот узел или будет искать новый узел для перемещения в его окрестность. При этом сенсор использует дополнительные параметры $p_i(k) \in \mathbb{Z}_+$ и $pLast_i(k)$, которые пересчитываются от шага k к шагу индивидуально для каждого сенсора. Первый из них характеризует удаленность ближайшего пустого узла сетки, смысл второго будет объяснен позже. В начальный момент времени у всех агентов значения параметров p и $pLast$ равны 0.

Для выделения основных идей вначале обсудим их в предположении, что параметры сеток всех сенсоров уже согласованы и таким образом совпадают с параметрами θ, x, y некоторой общей сетки T .

В начале дробного момента шага k в R_{tr} -окрестности каждого узла v решетки $Q_{G^g}(T)$ находится некоторое множество $S_v(k)$ сенсоров (возможно, пустое); причем каждый сенсор попадает в одно из таких множеств. Предполагая, что

$$2R_{tr} < g < \min\{r_o, r_c\},$$

закключаем, что сенсоры из $S_v(k)$ обмениваются данными и определяют взаимное положение друг друга. В частности, каждый из них i определяет состав своей группы $S_v(k)$, для обозначения которой далее параллельно используем символ $S_i(k)$.

1. Если помимо i множество $S_i(k)$ не содержит других сенсоров, сенсор i остается на месте, чтобы «контролировать» ближайший узел;
2. В противном случае сенсор определяет «контролера» среди элементов $S_i(k)$. Если «контролером» оказывается он сам, действует в соответствии с 1. В противном случае сенсор ищет узел для последующего перемещения.

Для поиска «контролера» сенсор использует правую декартову систему отсчета, ось абсцисс которой ориентирована по направляющей решетки T . «Контролером» является максимальный элемент множества $S_i(k)$ относительно лексикографического упорядочения плоскости декартовыми координатами в указанной системе. Таким образом, для

поиска максимального элемента сначала отбираются представители множества $S_i(k)$ с наибольшим значением абсциссы, а среди них - элемент с наибольшей ординатой.

Замечание 8. *Изложенный алгоритм выбора «контролера» подразумевает, что сенсоры не могут физически находиться в одной точке пространства. Поэтому в предположении абсолютной согласованности сеток все входящие в $S_i(k)$ сенсоры приходят к консенсусному выбору «контролера». В реальной ситуации неидентичности сеток решения разных сенсоров по рассматриваемому вопросу могут не совпадать. Это не означает прекращение работы алгоритма, но усложняет анализ его сходимости.*

Подводя итог применительно к реальной ситуации неидеально согласованных решеток, агент i на втором полушаге k выполнит следующие действия:

1. Определяет ближайший узел ζ своей решетки и множество $S_i(k)$ сенсоров, лежащих в R_{tr} -окрестности этого узла ζ ;
2. Если этому множеству $S_i(k)$ принадлежит только сенсор i , принимает решение о постановке узла ζ под собственный «контроль»;
3. Иначе
 - а. Сенсор строит правую декартову систему отсчета, ось абсцисс которой ориентирована по направляющей решетки данного сенсора;
 - б. Определяет множество сенсоров из $S_i(k)$, у которых абсцисса превосходит его собственную абсциссу \bar{x}_i .
 - i. Если оно непусто, принимает решение искать пустой узел и заканчивает п. 3.;
 - ii. В противном случае определяет множество сенсоров из $S_i(k)$, у которых абсцисса равна \bar{x}_i .
 - А. Если среди них есть сенсор, ордината которого превосходит ординату сенсора i , принимает решение искать пустой узел и заканчивает п. 3.;
 - Б. В противном случае принимает решение «контролировать» узел ζ .

Замечание 9. *После выполнения всеми сенсорами второй половины шага, все узлы $Q_{G^g}(T)$ подразделяются на два класса: «подконтрольные» какому-то сенсору и «неподконтрольные» ни одному из них, которые далее называются свободными.*

Дальнейшие действия сенсоров, принявших решение «контролировать» узел сетки, отличаются от действий прочих сенсоров. Поэтому рассмотрим эти две группы сенсоров по отдельности.

Сенсоры, принявшие решение контролировать узел сетки

Их дальнейшая деятельность сосредоточена на итерационном вычислении параметра p , который характеризует расстояние от подконтрольного узла до ближайшего свободного (в рассматриваемый момент) узла из $Q_{G^g}(T_i)$, где T_i - триангуляционная сетка рассматриваемого сенсора i . При этом каждый сенсор использует данные об объектах из пересечения (именуемого далее соседней зоной) зоны мониторинга и зоны коммуникации, которые он получает как в результате сенсорного контакта с ними, так и по каналам связи.

Во избежание неоправданно больших значений p вводится верхний порог p_{max} этих значений. В качестве него можно взять максимальную длину пути в графе $Q_{G^g}(T)$, максимизированную по всем триангуляционным сеткам T рассматриваемой мелкости, либо общее число сенсоров. Напомним, что начальные значения параметров p и $pLast$ равны нулю для всех сенсоров.

Агент i , принявший на шаге k решение контролировать ближайший узел ζ своей сетки, выполняет следующие действия:

1. Обнуляет вспомогательный параметр $pLast_i(k+1) = 0$;
2. Формирует множество $V_i(k)$ узлов сетки T_i из g -окрестности области G на расстоянии одного ребра (сетки) от ζ ;
3. Если в R_{tr} -окрестности некоторого узла из $V_i(k)$ нет агентов, полагает $p_i(k+1) = 1$;
4. Иначе
 - а. Если текущее значение $p_i(k) = 1$, то полагает $p_i(k+1) = 0$ и заканчивает шаг;
 - б. Если $p_i(k) \neq 1$, то определяет множество $W_i(k)$ сенсоров j с $p_j(k) \neq 0$, лежащих в R_{tr} -окрестности одного из узлов из $V_i(k)$;
 - в. Если $W_i(k)$ пусто, то принимает $p_i(k+1) = 1$ и заканчивает шаг k ;

г. Если $W_i(k)$ не пусто, использует формулу

$$p_i(k+1) = 1 + \min_{j \in W_i(k)} p_j(k);$$

д. Если $p_i(k+1) > p_{max}$, принимает $p_i(k+1) = 0$.

Сенсоры, ищущие новый узел,

руководствуются следующими простыми правилами:

- Если вблизи (т.е. примерно на расстоянии ребра сетки) есть пустой узел, его следует занять;
- Если таких узлов несколько, осуществить случайный равновероятный выбор между ними;
- Если вблизи все узлы заняты, выяснить, к какому из них ближе всего пустые места сетки;
- Для этого использовать значения параметра p находящихся в этих узлах сенсоров, исключая «неосведомленных» j , признаком которых является равенство $p_j = 0$;
- Переместиться в узел, ближайший к пустым местам;
- Если таких узлов несколько, осуществить случайный равновероятный выбор между ними.

Решая вопрос о перемещении в выбранный узел, сенсор предпринимает дополнительные предосторожности. Они мотивированы недоверием к информации, предоставляемой параметром p . В основе этого скепсиса два соображения. Во-первых, для достоверности параметр p должен пройти через период итерационного «созревания», в ходе которого он увеличивает свое значение от 0 до «расстояния до пустых мест». Вместе с тем используется этот параметр начиная с первого шага, не дожидаясь окончания указанного периода. Во-вторых, ввиду того, что сенсоры перемещаются в пустые узлы, множество «пустых мест» сетки меняется от шага к шагу. Это обстоятельство осложняет обеспечение достоверности оценки p расстояния до этих мест; скажем, несложно привести пример, когда оценка за один шаг теряет достоверность.

Для подавления обсуждаемых негативных последствий используется процедура перепроверки, в которой используется параметр $pLast$. Именно, сенсор не сразу переходит в выбранный узел, а предварительно сравнивает полученное в процессе выбора текущее значение параметра p с его значением на предыдущем шаге, которое хранится в $pLast$. И только если они одинаковы, перемещается в выбранный узел.

Идея введения $pLast$ связана с желанием сократить число лишних передвижений в ситуации, когда значения параметров p еще не «созрели» и плохо согласованы с реальностью. Другое соображение адресовано ситуации, когда покрытие области уже достигнуто, но в силу инертности процедуры построения параметра p его значения все еще увеличиваются (для отдельных сенсоров), давая ложную надежду найти еще один пустой узел.

Подытожим изложенное описанием применяемого алгоритма. Сенсор i , принявший решение искать новый узел на шаге k , производит следующие операции:

1. Обнуляет параметр $p_i(k) = 0$;
2. Находит ближайший к себе узел ζ своей сетки;
3. Формирует множество $V_i(k)$ узлов сетки T_i из g -окрестности области G на расстоянии одного ребра (сетки) от ζ ;
4. Если в R_{tr} -окрестности некоторого узла из $V_i(k)$ нет агентов, осуществляет равновероятный случайный выбор одного из таких узлов ;
5. Перемещается в пересечение $R_{tr}/2$ -окрестности этого узла и g -окрестности области G и заканчивает шаг k ;
6. В противном случае
 - а. Формирует множество $W_i(k)$ сенсоров с $p_i(k) \neq 0$, лежащих в объединении R_{tr} -окрестностей узлов из $V_i(k)$;
 - б. Если $W_i(k)$ пусто, то принимает $pLast_i(k + 1) = 0$ и заканчивает шаг k ;
 - в. Если $W_i(k)$ не пусто, использует формулу

$$pLast_i(k + 1) = \min_{j \in W_i(k)} p_j(k);$$

- г. Если $pLast_i(k + 1) \neq pLast_i(k)$, заканчивает шаг k ;

д. Если $pLast_i(k+1) = pLast_i(k)$, то

- i. Находит все узлы из $V_i(k)$, в R_{tr} -окрестности которых расположен сенсор j с $p_j(k) = pLast_j(k+1)$;
- ii. Равновероятно выбирает один из них;
- iii. Перемещается в пересечение $(R_{tr}/2)$ -окрестности этого узла и g -окрестности области G .

Замечание 10. *К началу целого момента шага $k+1$, агенты всегда остаются в g -окрестности области G и в R_{tr} -окрестности узла сетки.*

Оставшаяся часть работы посвящена анализу сформулированного в данном разделе (модифицированного) алгоритма и его сравнению с базовым алгоритмом.

Глава 2

Анализ предложенного алгоритма

2.1. Сходимость предложенного правила достижения консенсуса

В базовом алгоритме для согласования параметров $z = \theta, x, y$ сеток сенсоры применяли стандартное правило ближайших соседей

$$z(k+1) = \frac{\sum_{j \in \mathcal{N}_i(k)} z_j(k)}{|\mathcal{N}_i(k)|}, \quad (2.1)$$

которое далее также будем называть правилом среднеарифметического для выделения его основного отличия от метода, использованного в предложенном алгоритме. Он заключается в замене среднеарифметического значения минимальным значением

$$z(k+1) = \min_{j \in \mathcal{N}_i(k)} z_j(k) \quad (2.2)$$

и далее называется правилом экстремума.

Изучение правила среднеарифметического и его расширений является одним из основных фокусов интенсивных исследований в области динамики многоагентных систем в течение последних двух десятилетий; с обзорами главных достижений можно ознакомиться в [11, 12, 13, 14, 15, 16, 17, 18]. В частности были выявлены необходимые и достаточные условия сходимости данного правила (при произвольных начальных данных). Вместе с тем правило экстремума не вполне укладывается в русло известных результатов, что мотивирует необходимость отдельного исследования его сходимости, проведенного в данном параграфе.

Следуя традициям, сложившимся в литературе по многоагентному консенсусу, исследуем абстрактную модель процесса достижения консенсуса. Именно, рассматриваем n «абстрактных» агентов, занумерованных индексом $i = 1, \dots, n$. Функционируя в дискретном времени $k = 0, 1, 2, \dots$, каждый агент i формирует в момент k собственное значение $z_i(k) \in \mathbb{R}$ некоторой скалярной величины z . В процессе его формирования он использует данные о значении этой величины для агентов из некоторого зависящего от времени k и номера i множества $\mathcal{N}_i(k)$ агентов j , именуемых его соседями; при этом $i \in \mathcal{N}_i(k)$. Предполагаем, что отношение $j \in \mathcal{N}_i(k)$ соседства выражено зависящим от времени k ненаправленным графом $\Gamma(k)$: его вершины ассоциированы с агентами и

агент $j \neq i$ соединен с i ребром в том и только том случае, если считается соседом i . Предположение ненаправленности графа означает, что отношение соседства симметрично: $(j \text{ сосед } i) \Leftrightarrow (i \text{ сосед } j)$. (Это условие выполнено в случае сети мобильных сенсоров ввиду того, что у всех сенсоров один и тот же радиус коммуникации и мониторинга, причем j является соседом i в том и только том случае, когда расстояние между ними меньше обоих указанных радиусов.)

Определение 4. Скажем, что агенты достигают консенсуса, если существует такое число z_∞ , что $z_i(k) \rightarrow z_\infty$ при $k \rightarrow \infty$ для всех $i = 1, \dots, n$. Скажем, что они достигают консенсуса за конечное число шагов, если наряду с z_∞ существует такое целое $k_* \geq 0$, что $z_i(k) = z_\infty$ при $k \geq k_*$ для всех $i = 1, \dots, n$.

Определение 5. Графом существенных взаимодействий Γ_{ess} называется неориентированный граф с вершинами $i = 1, \dots, n$, в котором вершины $i \neq j$ соединены ребром в том и только том случае, если агенты i и j являются соседями бесконечно много раз, то есть соединяющее i и j ребро присутствует в графе $\Gamma(k)$ для бесконечного множества моментов времени k .

Относительно правила среднеарифметического (2.1) известен следующий результат [11, 12, 13, 14, 15, 16, 17, 18].

Теорема 2. Для того, чтобы при применении правила (2.1) агенты достигали консенсуса независимо от начальных значений $z_i(0), i = 1, \dots, n$ необходимо и достаточно, чтобы граф существенных взаимодействий Γ_{ess} был связан.

В выпускной работе эта теорема дополнена следующим результатом.

Теорема 3. Пусть граф существенных взаимодействий Γ_{ess} связан и агенты используют правило экстремума (2.2). Тогда они достигают консенсуса за конечное число шагов независимо от начальных значений $z_i(0), i = 1, \dots, n$.

При этом опираясь на аргументы из [11, 12, 13, 14, 15, 16, 17, 18] легко заметить, что условие связности Γ_{ess} по-прежнему является необходимым. Подчеркнем, что за исключением редких ситуаций (точнее, за исключением наборов начальных значений, образующих множество нулевой лебеговой меры) правило среднеарифметического (2.1) не обеспечит достижение консенсуса за конечное число шагов. В этом смысле теорема 3 указывает на определенное преимущество предложенного правила (2.2) над классическим алгоритмом (2.1).

Доказательство теоремы 3. Определим следующую величину

$$z_{\min} = \min_{i \in \{1, \dots, n\}} z_i(0).$$

Для любого агента i_0 , на котором достигается этот минимум, очевидно $z_{i_0}(k) = z_{\min} \forall k$.

Для каждого агента i на каждом шаге k справедливо следующее соотношение

$$z_i(k+1) = \min_{j \in \mathcal{N}_i(k+1)} z_j(k) \leq z_i(k),$$

так как сам агент включается в множество $\mathcal{N}_i(k+1)$. Таким образом, значение параметра z у каждого агента с ростом шага не увеличивается. Введем понятие меры рассогласованности для каждого шага

$$\mu(k) = \sum_{i=1}^n (z_i(k) - z_{\min}).$$

Заметим, что эта мера может только убывать, так как параметры у каждого агента не возрастают с ростом шага

$$\mu(k+1) = \sum_{i=1}^n (z_i(k+1) - z_{\min}) \leq \sum_{i=1}^n (z_i(k) - z_{\min}) = \mu(k). \quad (2.3)$$

Рассмотрим шаг $k = k_*$, для которого $\mu(k_*) > 0$, и покажем, что на некотором шаге $k \geq k_*$ значение $\mu(k)$ строго уменьшается. Рассуждаем от противного, то есть допустим, что это утверждение неверно. Заметим, что на каждом шаге $k \geq k_*$ множество агентов разбивается на два непустых подмножества:

- $\mathcal{N}_{\min}(k)$ состоит из агентов i , для которых $z_i(k) = z_{\min}$,
- $\mathcal{N}_{>\min}(k)$ состоит из агентов i , для которых $z_i(k) > z_{\min}$.

Здесь с ростом k первое множество может только расширяться. Этого однако не происходит, так как расширение, то есть переход некоторого агента i из $\mathcal{N}_{>\min}(k)$ в $\mathcal{N}_{\min}(k)$, сопровождается уменьшением соответствующего слагаемого в (2.3) и значит падением значения $\mu(k)$ в нарушение принятой гипотезы. В свою очередь отсутствие переходов из $\mathcal{N}_{>\min}(k)$ в $\mathcal{N}_{\min}(k)$ в силу (2.2) означает, что при $k \geq k_*$ между любыми агентами $i \in \mathcal{N}_{>\min}(k)$ и $j \in \mathcal{N}_{\min}(k)$ нет отношения соседства поэтому в графе Γ_{ess} между ними нет ребра. Но тогда непустые множества $\mathcal{N}_{>\min}(k)$ и $\mathcal{N}_{\min}(k)$ лежат в разных компонентах связности этого графа, что противоречит предположению о его связности из формулировки теоремы.

Итак если $\mu(k_*) > 0$, то на некотором шаге $k \geq k_*$ значение $\mu(k)$ строго уменьшается. Вместе с тем на каждом шаге множество значений $z_i(k), i = 1, \dots, n$ является подмножеством конечного множества $z_i(0), i = 1, \dots, n$. Отсюда следует, что мера рассогласованности μ также может принимать только конечное число значений, так как является функцией от набора параметров $\{z_i(k)\}_{i=1}^n$ на каждом шаге алгоритма. В итоге имеется функция шага k , которая может принимать только конечное число значений, монотонно убывает и за каждым моментом k , когда она положительна, следует момент её строгого убывания. Очевидно, она рано или поздно достигает нулевого значения. Значит мера рассогласованности в какой-то момент обязательно примет нулевое значение и агенты достигнут консенсуса. \square

Вывод. *Алгоритм сходится за конечное число шагов.*

2.2. Экспериментальное сравнение двух алгоритмов достижения консенсуса

Теоретическое преимущество правила экстремума над правилом среднеарифметического, отражаемое теоремой 3, это точная сходимость первого правила к предельному значению за конечное число шагов, в то время как второе правило обеспечивает сколь угодно точное приближение к пределу лишь в асимптотике. Вместе с тем на практике играет роль приближение не со сколь угодно высокой, а с требуемой (заданной и возможно высокой) точностью, которое достигается для второго правила также за конечное число шагов. В данном параграфе указанные два правила сравниваются с более «практической», чем в предыдущем разделе, точки зрения: с позиции числа шагов до достижения момента, когда для любых двух агентов разница значений их параметров по абсолютной величине не превосходит заданного числа ε , которое выбиралось разным для разных серий экспериментов. Сравнение производится методом компьютерного моделирования.

В процессе моделирования граф $\Gamma(k)$ задавался бинарной симметричной матрицей инцидентности $A(k)$ со следующей интерпретацией: если в i -ом столбце и j -ой строке матрицы стоит 1, то агенты i и j связаны ребром графа, если 0, то ребра нет. Так как каждый агент всегда является себе соседом, то на главной диагонали всегда стоят 1.

Формирование меняющейся во времени симметричной матрицы $A(k)$ был органи-

зовано как случайный процесс, в ходе которого наддиагональные элементы $a_{ij}(k)$ независимо друг от друга и предшествующих значений $A(l), l < k$ с вероятностью $P \in (0, 1)$ принимают значение 1 и с дополнительной вероятностью $1 - P$ — значение 0; после этого поддиагональные элементы определяются однозначно ввиду симметрии матрицы. Таким образом, P имеет смысл вероятности (средней частоты) соседства двух различных агентов. При первом же достижении заданной точности ε процесс согласования считался завершенным. Для первого правила середина интервала, в котором содержались все значения параметров при окончании процесса, интерпретировалась как предельное значение, для второго соответственно это была левая граница интервала.

Было проведено имитационное моделирование, в ходе которого сравниваемые алгоритмы запускались с одинаковыми данными, включая начальные значения $z_i(0)$, а также P и ε . Для каждой из рассмотренных пар (P, ε) было проведено 2500 экспериментов с участием 100 агентов. Параметры z_i рассматривались как угловые величины, независимо друг от друга они случайно и равновероятно выбирались из отрезка $(0, \pi)$. В каждом эксперименте определялось число шагов первого и второго алгоритма, за которое достигалась требуемая точность.

Ввиду большого числа (2500) экспериментов с фиксированными P и ε использовались следующие интегральные числовые характеристики их результатов (которые таким образом оказывались функциями (P, ε)).

ch.1) Вероятность того, что алгоритм среднеарифметического сошелся быстрее алгоритма экстремума, оцениваемая отношением числа реализаций этого события к общему числу экспериментов. Обозначим эту величину за $\Omega(P, \varepsilon)$.

ch.2) Отношение среднего числа шагов алгоритма среднеарифметического к среднему числу шагов алгоритма экстремума. Обозначим эту величину за $\Phi(P, \varepsilon)$.

В приложении на Рис. А.11 представлен график зависимости вероятности Ω из ch. 1, выраженная в процентном соотношении от P и ε в двух ракурсах. На графике видно, что максимальное значение вероятности не превосходит 15% и с уменьшением допустимой погрешности ε приближается к нулю. Иными словами, при достаточно малых ε метод экстремума сходится как минимум не медленнее, чем алгоритм среднеарифметического для любого P .

В приложении на Рис. А.12 изображен график зависимости характеристики Φ из ch. 2 от P и ε представленный в двух ракурсах. С уменьшением ε она волнообразно

возрастает. При требуемой точности до второго знака после запятой, эта характеристика оказывается строго больше единицы. Данный график показывает, во сколько раз в среднем алгоритм выбора минимума сошелся быстрее, чем алгоритм среднеарифметического. Отношение среднего числа шагов не опускается ниже 0.95 и в целом превосходит 1.

При этом при $P > 0.15$ стандартное отклонение у исследуемой выборки достаточно маленькое, основная масса находится в пределах от 0 до 0.5 и встречается не больше 3 различных значений числа шагов. Так как дисперсия довольно мала при $P > 0.15$, среднее значение достаточно хорошо характеризует скорости сходимости методов.

Вывод. *Правило экстремума продемонстрировало преимущество над правилом среднеарифметического и с точки зрения скорости практической сходимости.*

2.3. Сравнение базового и модифицированного алгоритмов покрытия области методом компьютерного моделирования

Анализ сходимости модифицированного алгоритма выполнен методом компьютерного моделирования. Основной акцент был сделан на невыпуклые области G , а также ситуации, когда до какой-то части области добраться трудно из-за наличия узкого коридора, соединяющего её основные части. Поэтому для экспериментов были выбраны:

1. Не выпуклая область, напоминающая «конфету»;
2. Два прямоугольника, соединенные узким коридором.

Их границы изображены синим цветом в приложении на Рис. А.1 и Рис. А.2 соответственно. Красным цветом нарисована граница g -окрестности области. Зеленым цветом изображены узлы триангуляционной сетки мелкости g . Для ускорения процесса сходимости и для экономии запасов энергии агентов на практике часто первоначально высаживают несколькими группами вблизи разных точек области. Поэтому в экспериментах были выбраны некоторые две точки области. Все агенты располагались в начальный момент времени на случайном расстоянии друг от друга и были удалены от одной из заданных двух точек не более чем на $r = 2$ (м). Начальные координаты узла сетки имели нулевое значение в системе отсчета, связанной с агентом, поэтому расположение узла сетки $d_i(0)$ в абсолютной системе координат у каждого агента i совпадало

№	Область	Число агентов	Значения параметров
1	1	120	$r_0 = 1$ (м), $r_c = 1$ (м), $g = 0.5$ (м), $R_{tr} = 0.04$ (м)
2	2	120	$r_0 = 1$ (м), $r_c = 1$ (м), $g = 0.5$ (м), $R_{tr} = 0.04$ (м)
3	1	250	$r_0 = 1$ (м), $r_c = 1$ (м), $g = 0.5$ (м), $R_{tr} = 0.04$ (м)
4	2	250	$r_0 = 1$ (м), $r_c = 1$ (м), $g = 0.5$ (м), $R_{tr} = 0.04$ (м)
5	1	400	$r_0 = 1$ (м), $r_c = 1$ (м), $g = 0.5$ (м), $R_{tr} = 0.04$ (м)
6	2	400	$r_0 = 1$ (м), $r_c = 1$ (м), $g = 0.5$ (м), $R_{tr} = 0.04$ (м)
7	1	400	$r_0 = 2$ (м), $r_c = 2$ (м), $g = 0.5$ (м), $R_{tr} = 0.1$ (м)
8	2	400	$r_0 = 2$ (м), $r_c = 2$ (м), $g = 0.5$ (м), $R_{tr} = 0.1$ (м)
9	1	800	$r_0 = 0.5$ (м), $r_c = 0.5$ (м), $g = 0.25$ (м), $R_{tr} = 0.04$ (м)
10	2	800	$r_0 = 0.5$ (м), $r_c = 0.5$ (м), $g = 0.25$ (м), $R_{tr} = 0.04$ (м)

Таблица 2.1. Номера серий и используемые параметры

с его собственным начальным положением. Начальный угол направляющей сетки $\theta_i(0)$ выбирался равновероятно из диапазона $(0, 2\pi)$ и независимо от выбора других агентов.

Было проведено 10 серий экспериментов для различных наборов параметров, их список приведен в таблице 2.1. Серии содержали по 500 экспериментов для каждого из двух рассматриваемых алгоритмов. Стоит отметить, что в каждом эксперименте модифицированный алгоритм сошелся по числу шагов как минимум в 4 раз быстрее, чем базовый. Рассмотрим несколько экспериментов, с целью демонстрации развития процессов в алгоритмах.

Синими точками в приложении на Рис. А.3 и Рис. А.4 представлена динамика расположения агентов из серии 1 для соответственно базового и модифицированного алгоритмов. В сети участвовало 120 агентов, что примерно соответствует минимально необходимому числу агентов для получения полного покрытия обеих рассмотренных областей при данных параметрах. Результаты эксперимента для второй серии представлены соответственно в приложении на Рис. А.5 и Рис. А.6. При выбранных условиях агенты не достигают покрытия при использовании базового алгоритма даже к 10000 шагу, в отличие от модифицированного алгоритма, которому потребовалось всего соответственно 51 и 76 шагов для каждой представленной области.

Пронаблюдаем по представленному ряду картинок динамику процессов происходя-

щих в сети агентов, при использовании ими модифицированного алгоритма. В каждой из двух групп выброски агентов процесс согласования параметров сеток произошел достаточно быстро как для первой области, так и для второй. В дальнейшем примерно к 20 шагам почти получено полное покрытие первой области и только к 50 шагам занимаетесь весь «коридор» второй области. Как упоминалось ранее, весь процесс перемещений агентов в обеих областях закончился на 51 и соответственно на 76 шагах. Таким образом, модификация устранила проблемы со сходимостью для случая экстремально малого числа агентов в сети.

Рассмотрим происходящие процессы в сети для увеличенного числа агентов. На Рис. А.7 представлен один из экспериментов для базового алгоритма в серии под номером 5. Уже к 10 шагу почти получено покрытие области, но из-за несогласованности параметров сеток процесс передвижения не останавливается. Примерно к сотому шагу параметры сеток достаточно хорошо согласованы и агенты разошлись по окрестностям некоторых узлов сеток. В дальнейшем для сходимости метода требовалось наступление момента времени, когда в окрестности каждого из узлов сетки окажется хотя бы один из агентов. Этот момент наступил только к 210 шагу. В течение этого времени освобождались и занимались узлы построенной агентами сетки.

Результаты динамики для модифицированного алгоритма для той же серии 5 представлены в приложении на Рис. А.8. Из-за процесса согласования параметров сеток почти полное покрытие было получено позже, чем у базового алгоритма. Но для остановки потребовалось всего 42 шага.

В аналогичных условиях, но для другой области G на Рис. А.9 и на Рис. А.10 представлены эксперименты из серии 6 для соответственно базового и модифицированного алгоритмов. В целом, динамика схожа с серией 5, но усугубляется наличием узкого места в области. В следствие чего, базовый алгоритм требует существенно больше шагов для остановки, чем для первой области.

Хотя результаты проведенного моделирования не позволяют точно определить скорость сходимости методов, они позволяют определить общие тенденции для их сравнительной характеристики. Именно наблюдаемая динамика уменьшения числа шагов, необходимых для получения покрытия области у модифицированного алгоритма относительно базового, дает основания считать его более экономичным.

Заключение

Рассмотрена задача мониторинга априори неизвестной области плоскости сетью роботизированных мобильных сенсоров. С целью повышения скорости сходимости разработана модификация алгоритма решения этой задачи, предложенного ранее в [9]. Методом компьютерного моделирования произведен сравнительный анализ базового и модифицированного алгоритмов; эксперименты продемонстрировали достижение поставленной цели ускорения алгоритма. Сходимость предложенного в работе алгоритма согласования ключевых параметров агентов обоснована теоретически и подтверждена результатами компьютерного моделирования.

Среди возможных направлений дальнейших исследований отметим задачу покрытия трехмерных областей, а также исследование проблемы робастности алгоритма к

1. потерям данных в каналах связи между сенсорами;
2. погрешностям измерений и вычислений;
3. потере части сенсоров.

Список литературы

1. Gage D. Command control for many - robot systems // in Proceedings of the 19th Annual AUVS Technical Symposium. — Vol. 4. — Huntsville, Alabama, USA, 1992. — P. 22–24.
2. Wang G., Cao G., Porta T. F. L. Movement-assisted sensor deployment // IEEE Transactions on Mobile Computing. — 2006. — Vol. 5, no. 6. — P. 640–652.
3. Wang W., Srinivasan V., Chua K. Coverage in hybrid mobile sensor networks // IEEE Transactions on Mobile Computing. — 2008. — Vol. 7, no. 11. — P. 1374–1387.
4. Cheng T. M., Savkin A. V. A distributed self-deployment algorithm for the coverage of mobile wireless sensor networks // IEEE Communications Letters. — 2009. — Vol. 13, no. 11. — P. 877–879.
5. Kumar S., Lai T. H., Arora A. Barrier coverage with wireless sensors // Wireless Networks. — 2007. — Vol. 13, no. 6. — P. 817–834.
6. Choset H. Coverage for robotics - a survey of recent results // Wireless Networks. — 2001. — Vol. 31, no. 1–4. — P. 113–126.
7. Bisht M., Chhetri S. A survey on the coverage of wsns // International Journal of Advanced Research in Computer Science and Software Engineering. — 2013. — Vol. 3, no. 3. — P. 295–300.
8. Savkin A., Javed F., Matveev A. Optimal distributed blanket coverage self-deployment of mobile wireless sensor networks // IEEE Communications Letters. — 2012. — Vol. 16, no. 6. — P. 949–951.
9. Matveev A., Ovchinnikov K., Nasimov M. Suboptimal distributed selfdeployment of robotic sensor networks: Blanket coverage with guarantees of global convergence // Robotica. — accepted for publication (to be issued in 2018).
10. Kershner R. The number of circles covering a set // American Journal of Mathematics. — 1939. — Vol. 61. — P. 665–671.
11. Olfati-Saber R., Murray R. Consensus problems in networks of agents with switching topology and time-delays // IEEE Transactions on Automatic Control. — 2004. — Vol. 49. — P. 1520–1533.
12. Ren W., Beard R. Consensus seeking in multiagent systems under dynamically changing interaction topologies // IEEE Transactions on Automatic Control. — 2005. — Vol. 50. —

- P. 655–661.
13. Moreau L. Stability of multiagent systems with time-dependent communication links // IEEE Transactions on Automatic Control. — 2005. — Vol. 50. — P. 169–182.
 14. Ren W., Beard R., Atkins E. Information consensus in multivehicle cooperative control // IEEE Control Systems Magazine. — 2007. — Vol. 27. — P. 71–82.
 15. Cao M., Morse S., Anderson B. Agreeing asynchronously // IEEE Transactions on Automatic Control. — 2008. — Vol. 53. — P. 1826–1838.
 16. Swarm robotics: A review from the swarm engineering perspective / M. Brambilla, E. Ferrante, M. Birattari, M. Dorigo // Swarm Intelligence. — 2013. — Vol. 7. — P. 1–41.
 17. Chen C., Chen G., Guo L. Consensus of flocks under m -nearest-neighbor rules // Journal of Systems Science and Complexity. — 2015. — Vol. 28. — P. 1–15.
 18. Multi-agent cooperative control consensus: A comparative review / M. Gulzar, S. Rizvi, M. Javed et al. // Electronics. — 2018. — Vol. 22. — doi:10.3390/electronics7020022.

Приложение А

Иллюстрации к работе

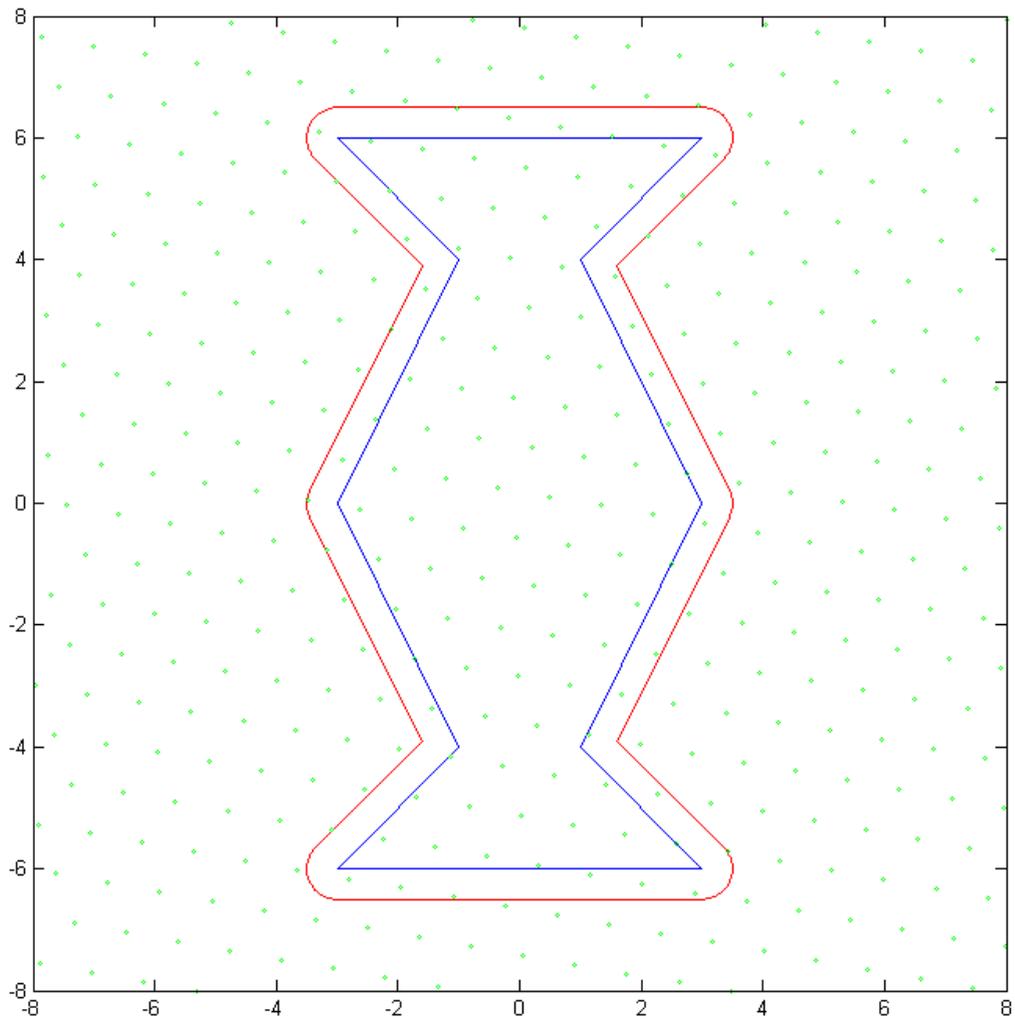


Рис. А.1. Изображение области «конфета».

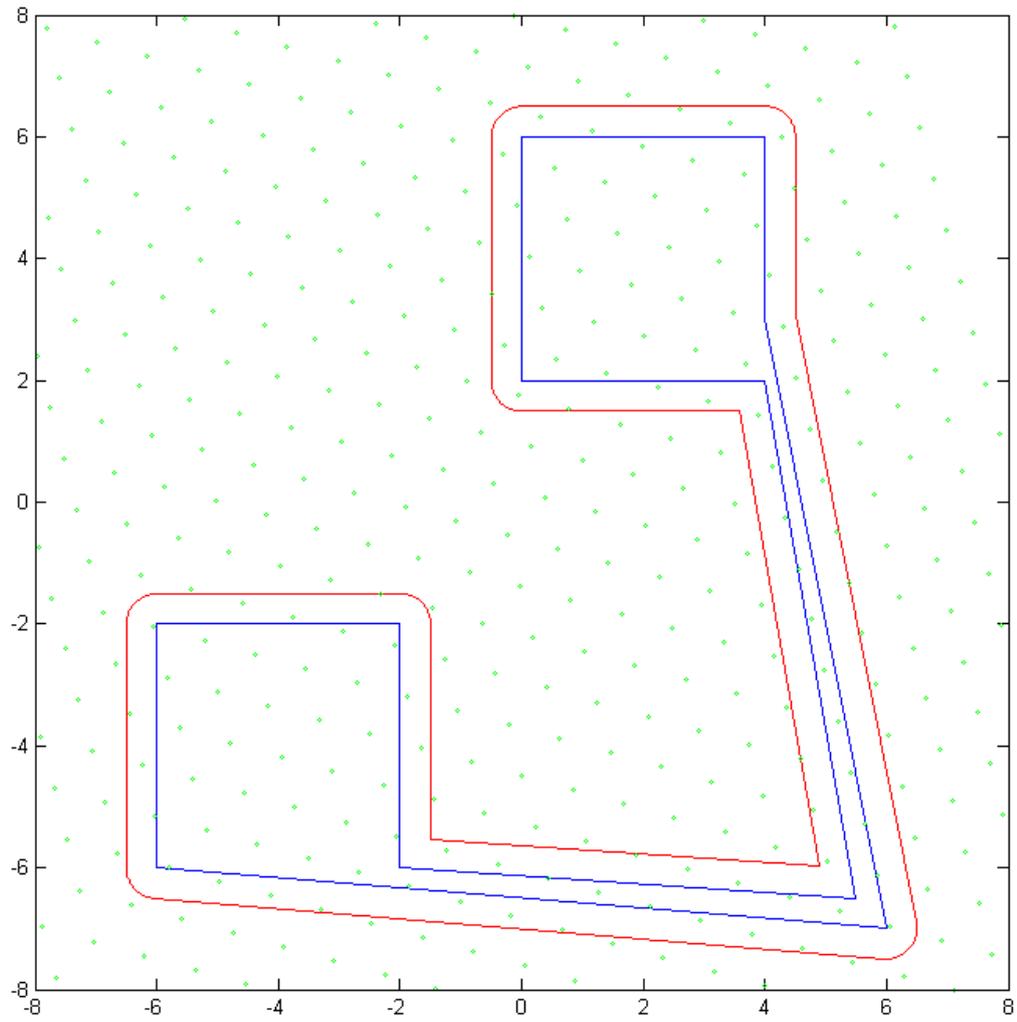


Рис. А.2. Изображение области с узким коридором.

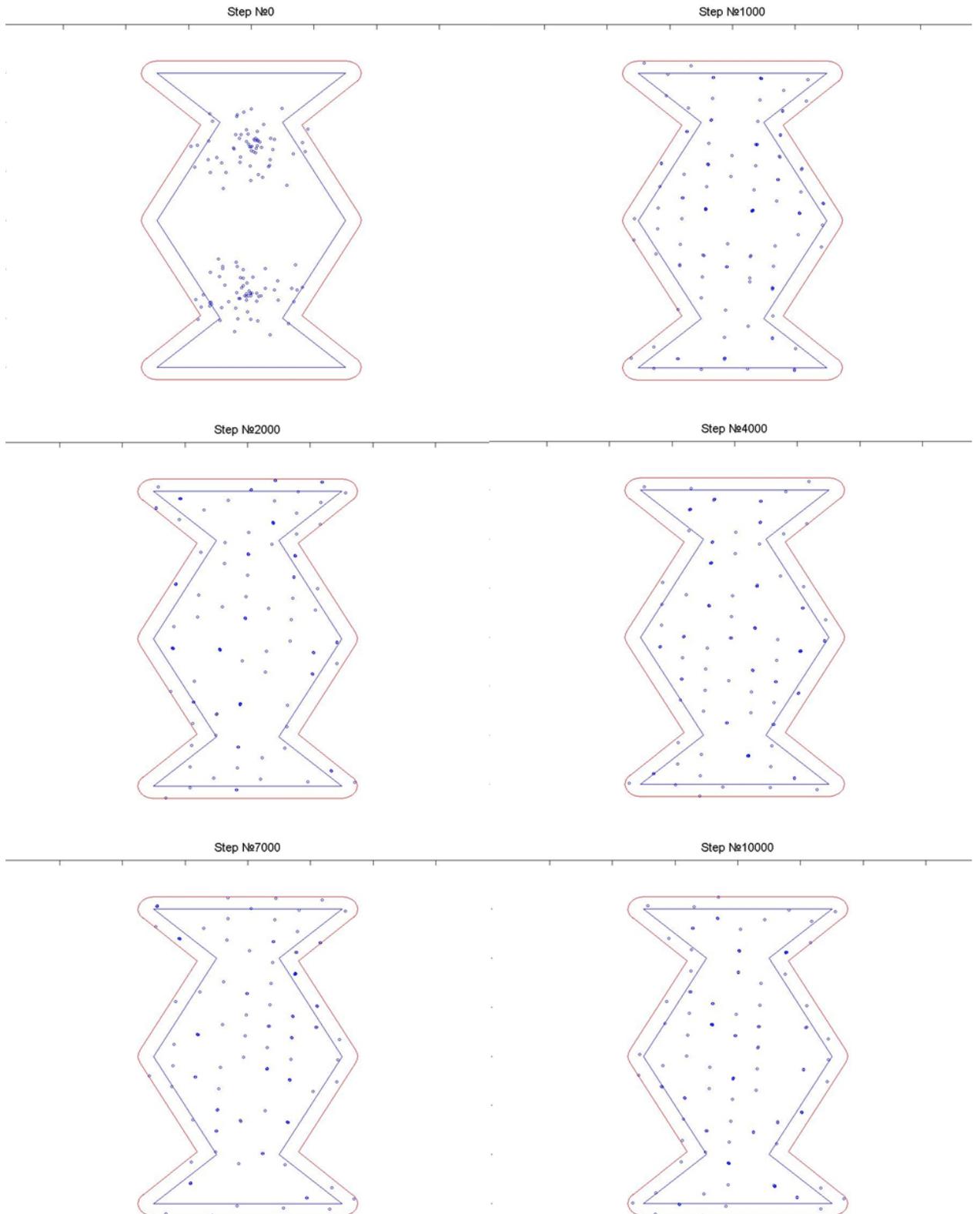


Рис. А.3. Динамика базового алгоритма при $n = 120$ для области «конфета».

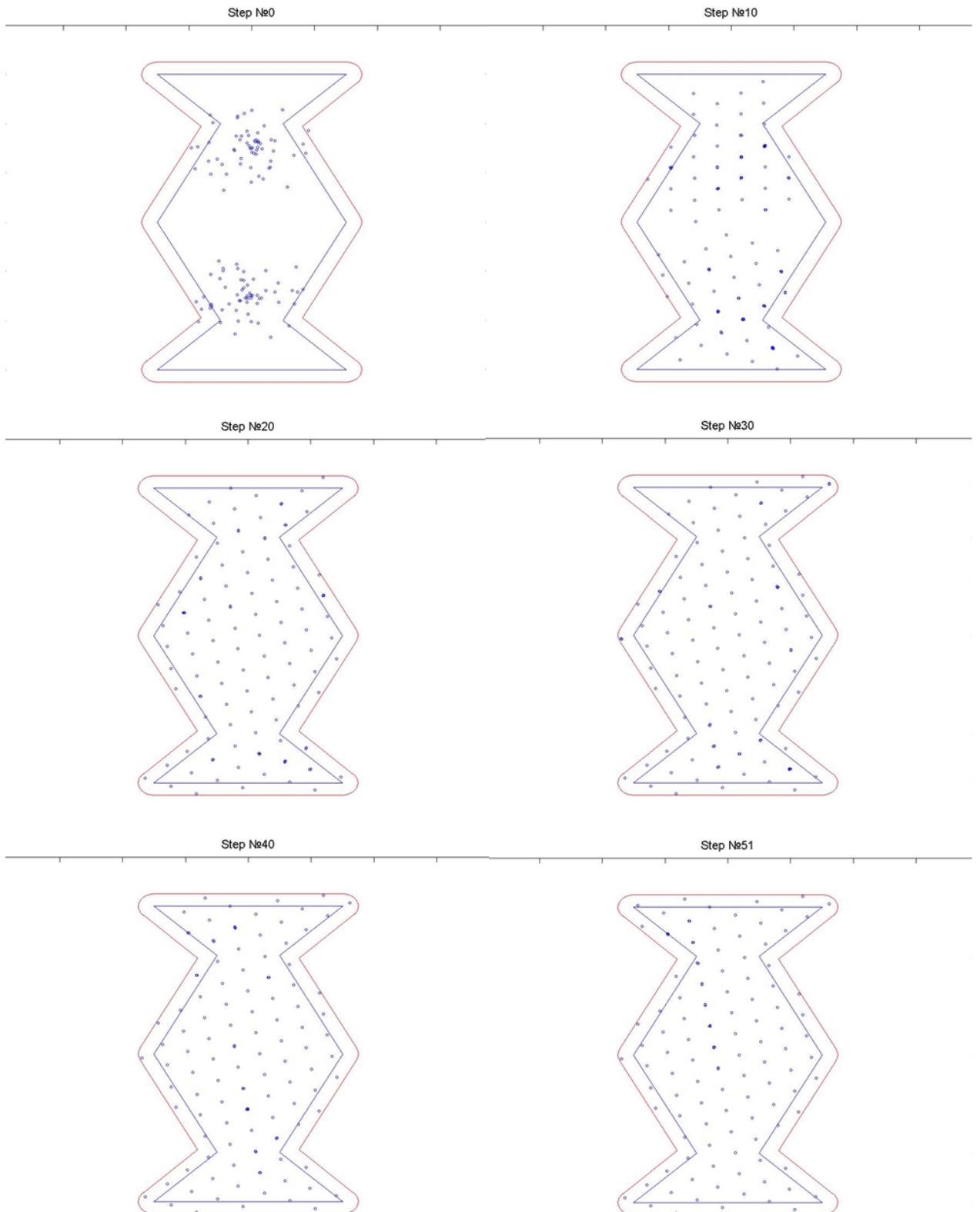


Рис. А.4. Динамика модифицированного алгоритма при $n = 120$ для области «конфета».

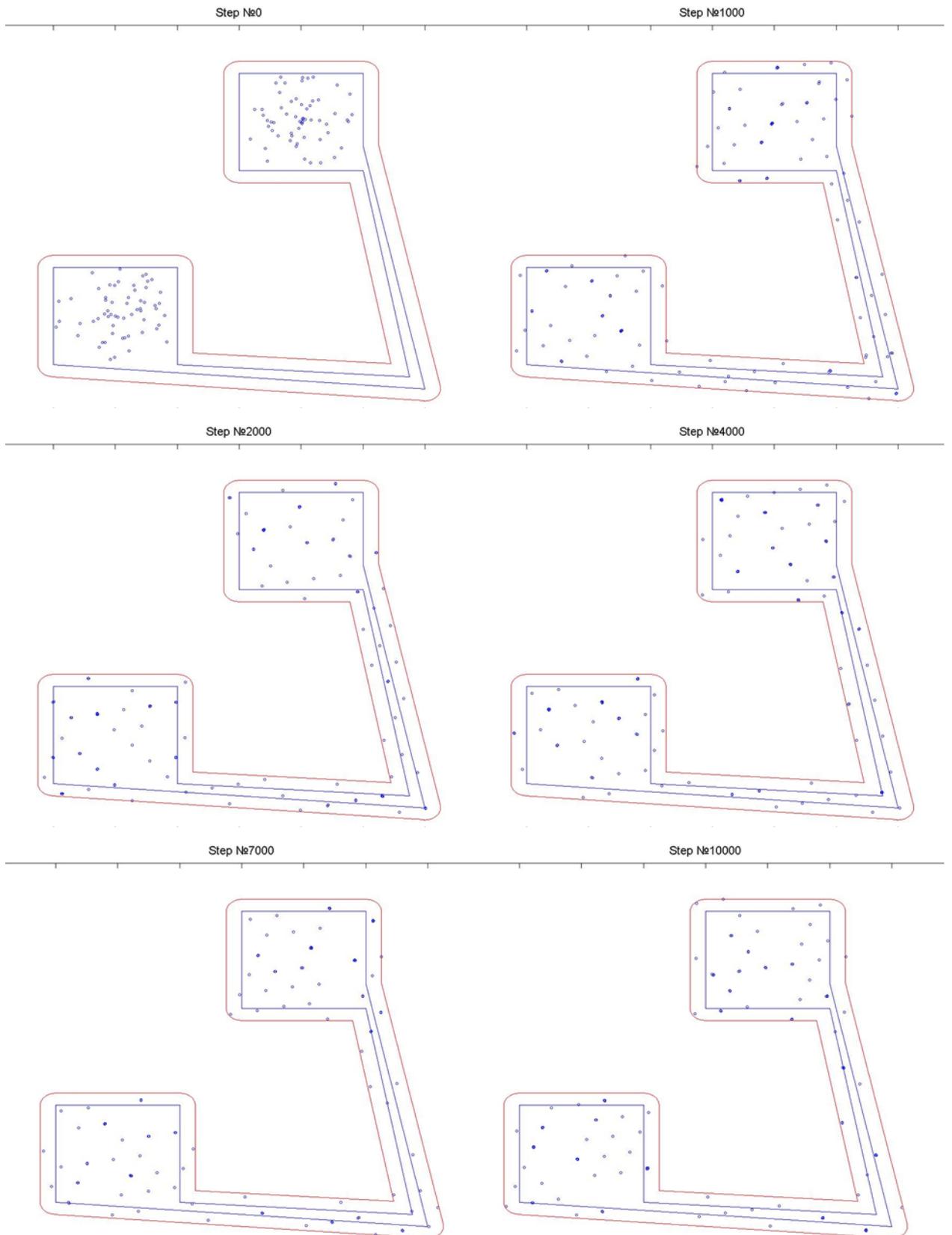


Рис. А.5. Динамика базового алгоритма при $n = 120$ для области с узким коридором.

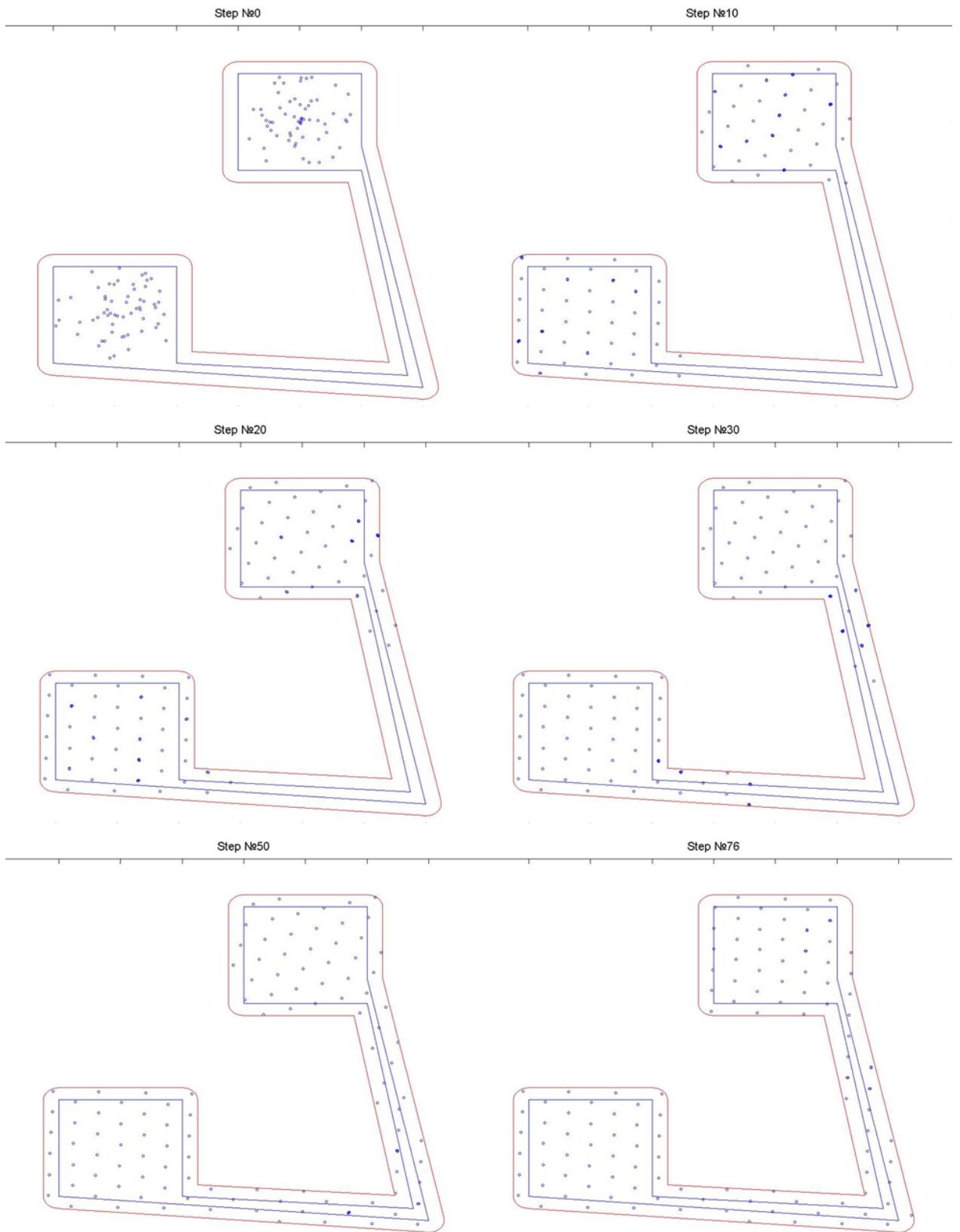


Рис. А.6. Динамика модифицированного алгоритма при $n = 120$ для области с узким коридором.

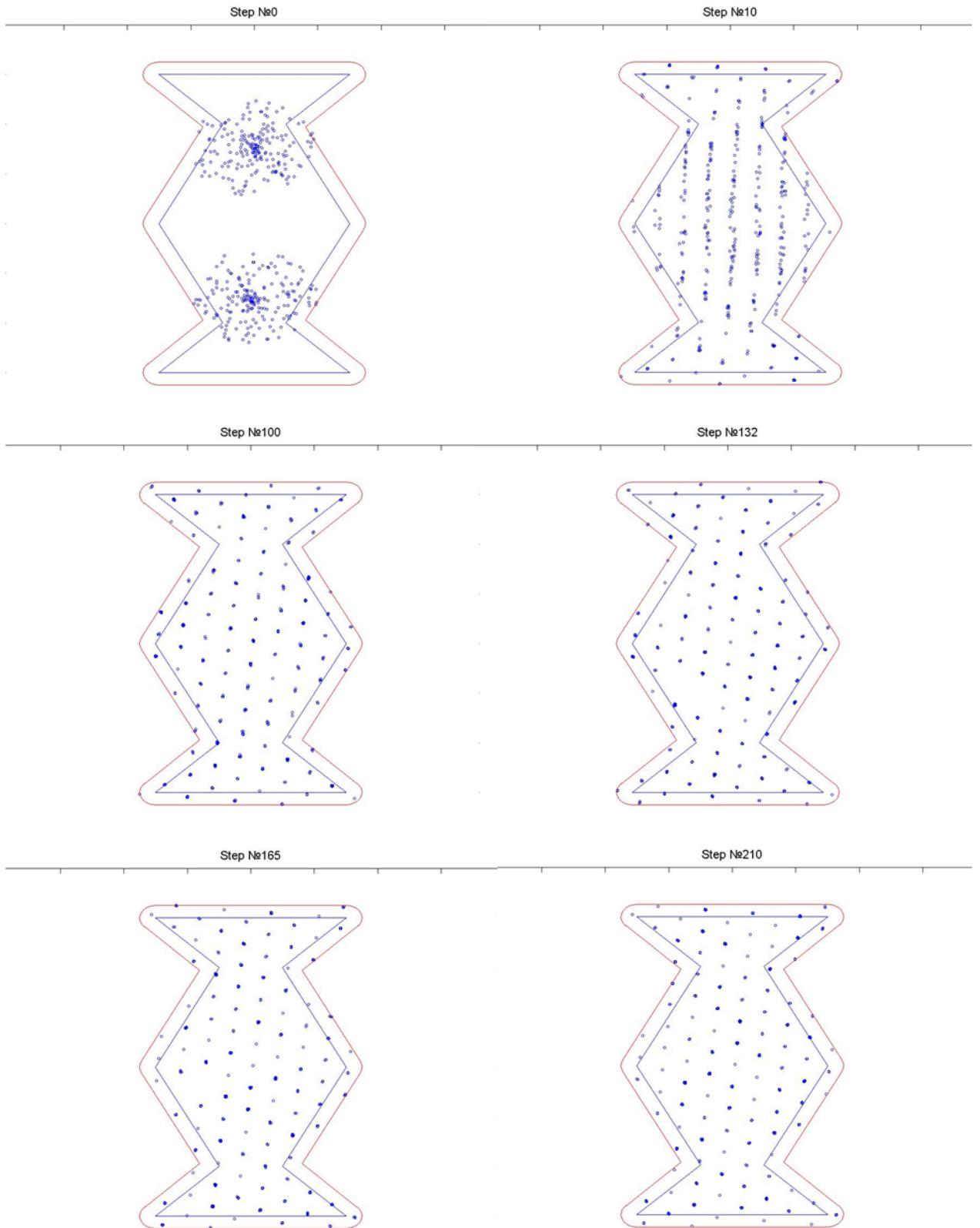


Рис. А.7. Динамика базового алгоритма при $n = 400$ для области «конфета».

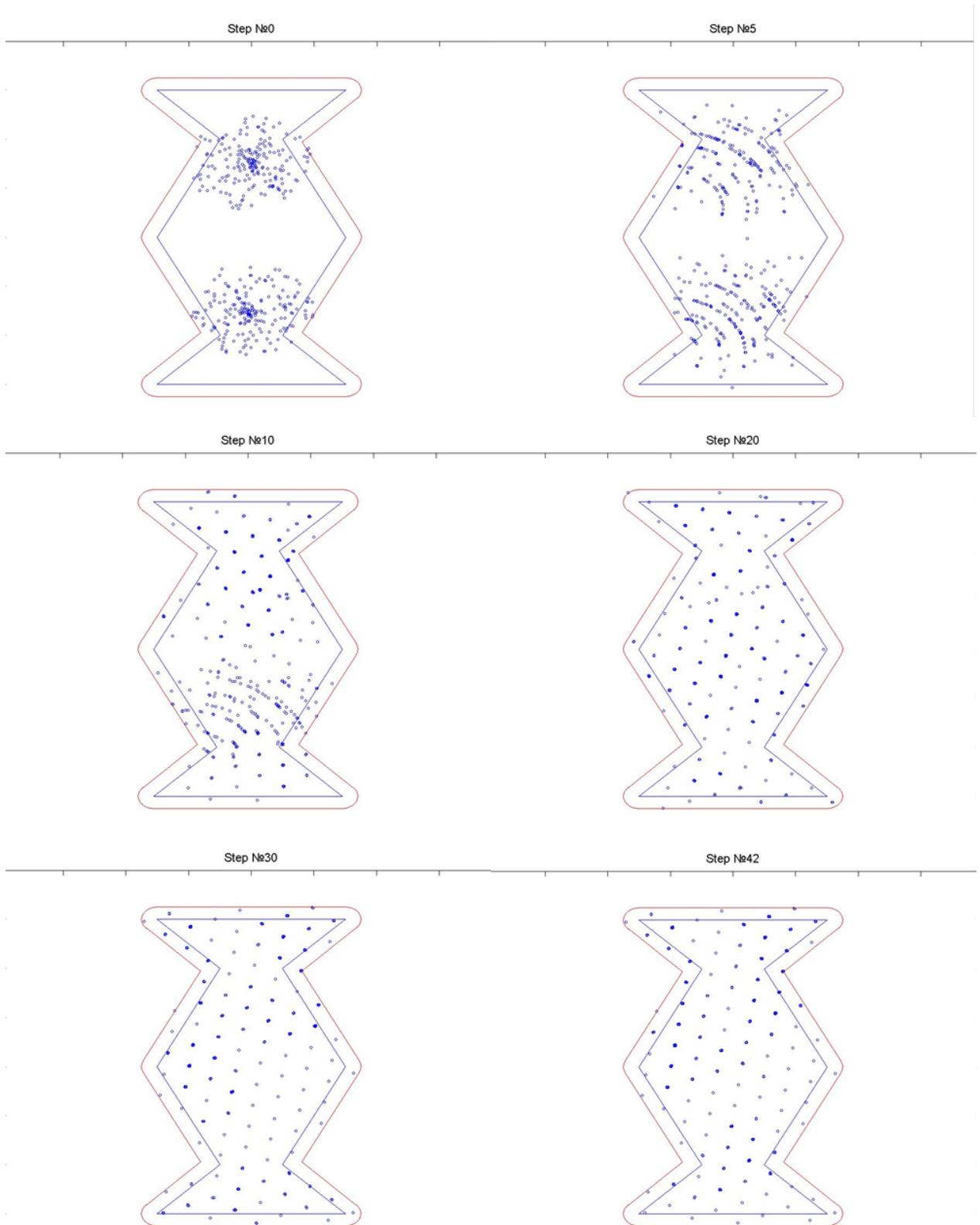


Рис. А.8. Динамика модифицированного алгоритма при $n = 400$ для области «конфета».

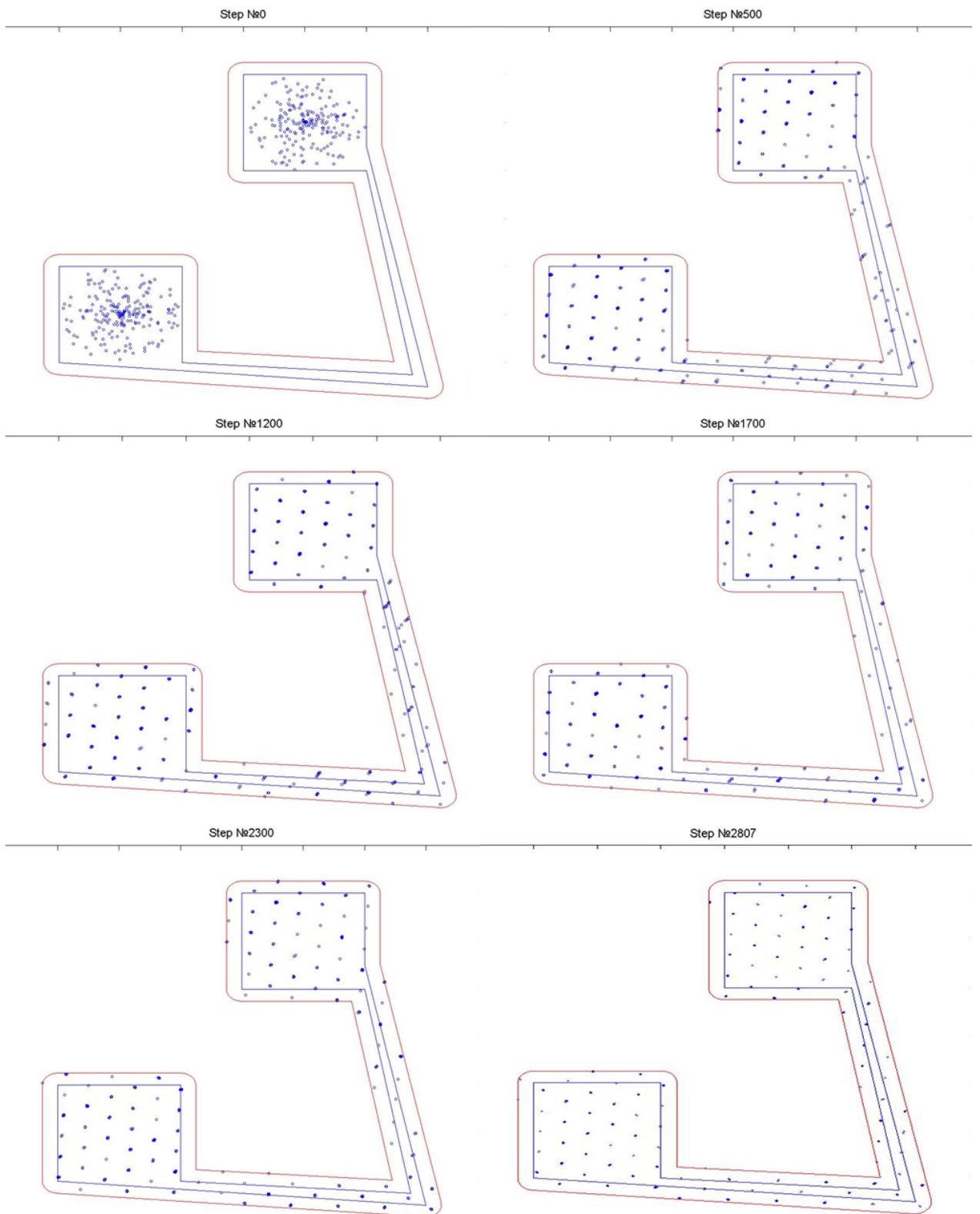


Рис. А.9. Динамика базового алгоритма при $n = 400$ для области с узким коридором.

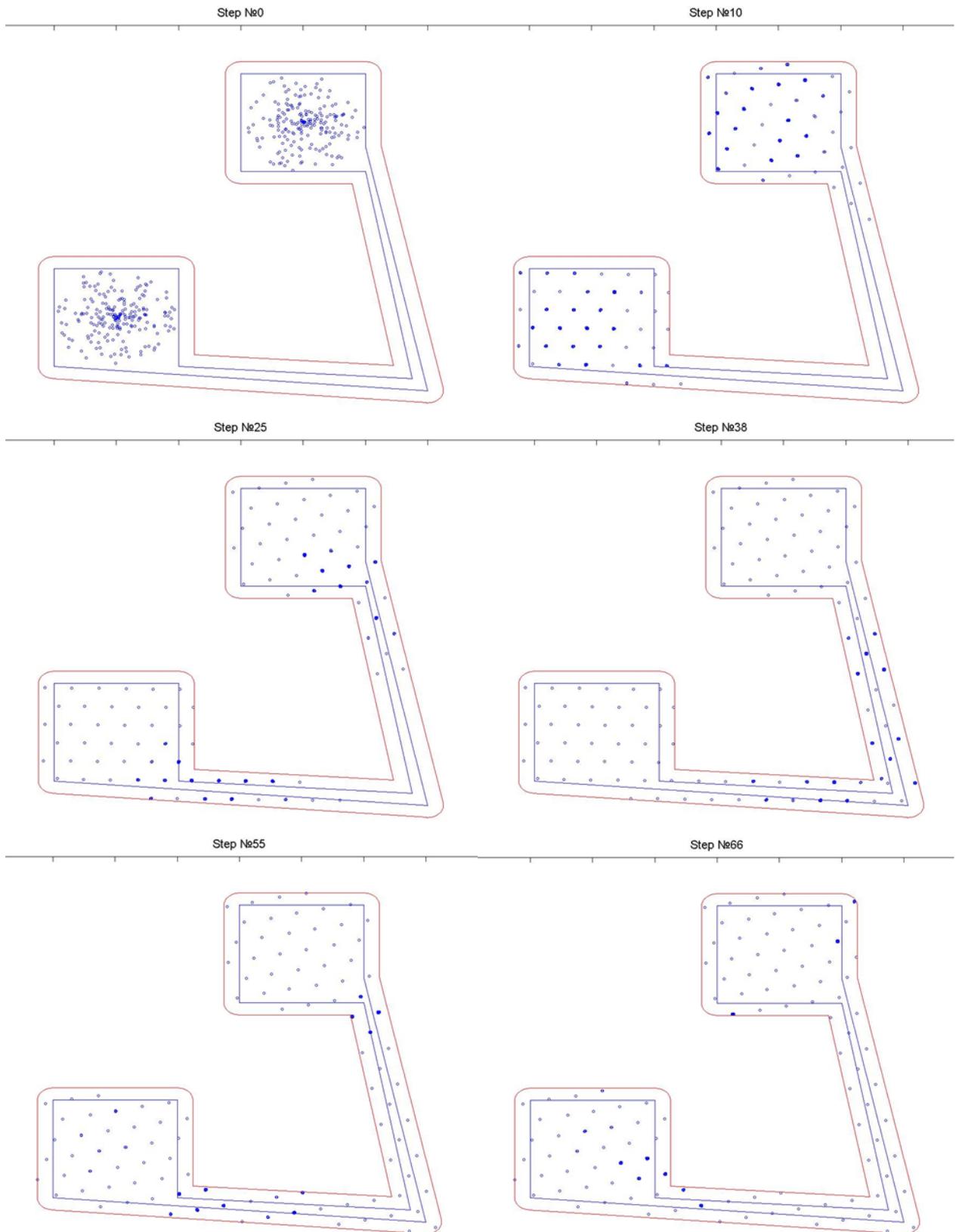


Рис. А.10. Динамика модифицированного алгоритма при $n = 400$ для области с узким коридором.

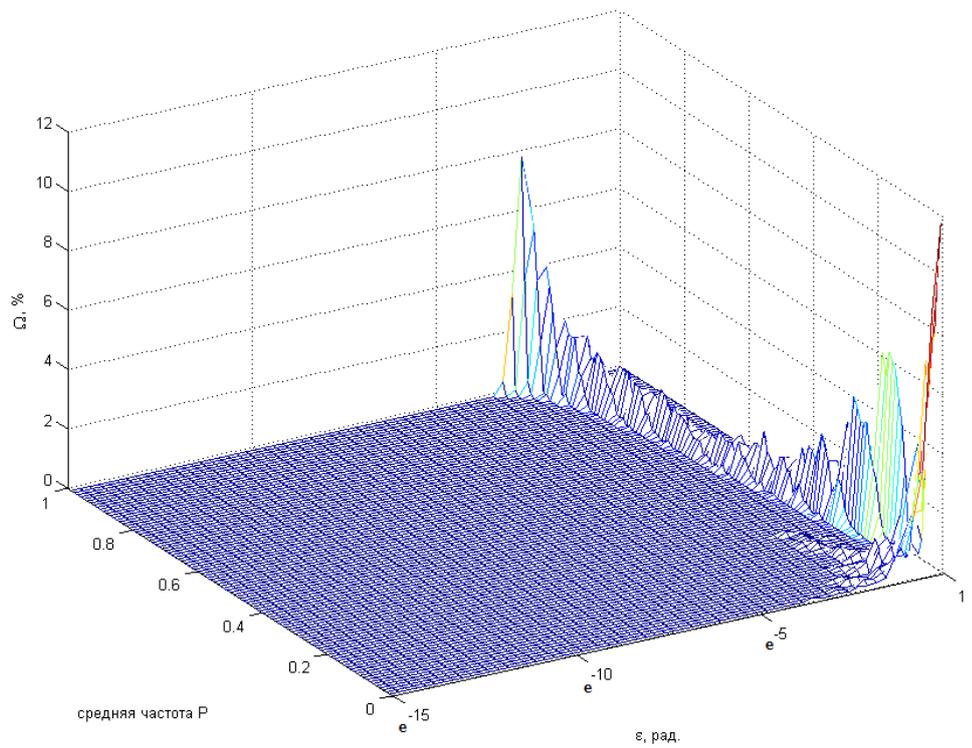
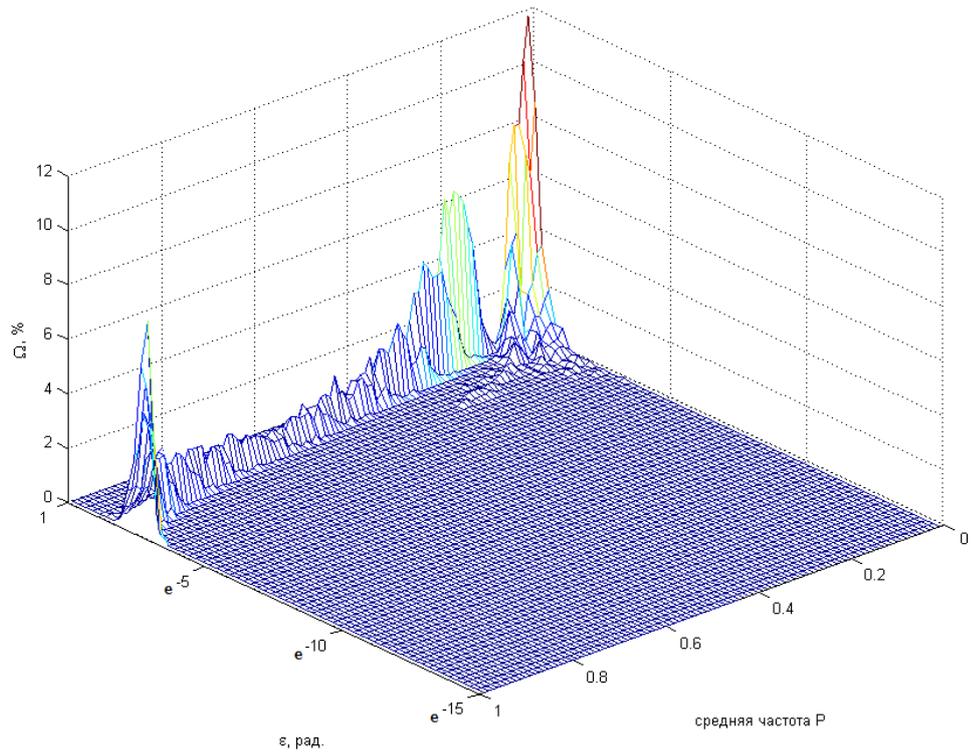


Рис. А.11. График зависимости вероятности ch. 1 из раздела 2.2 в зависимости от P и ϵ под двумя углами.

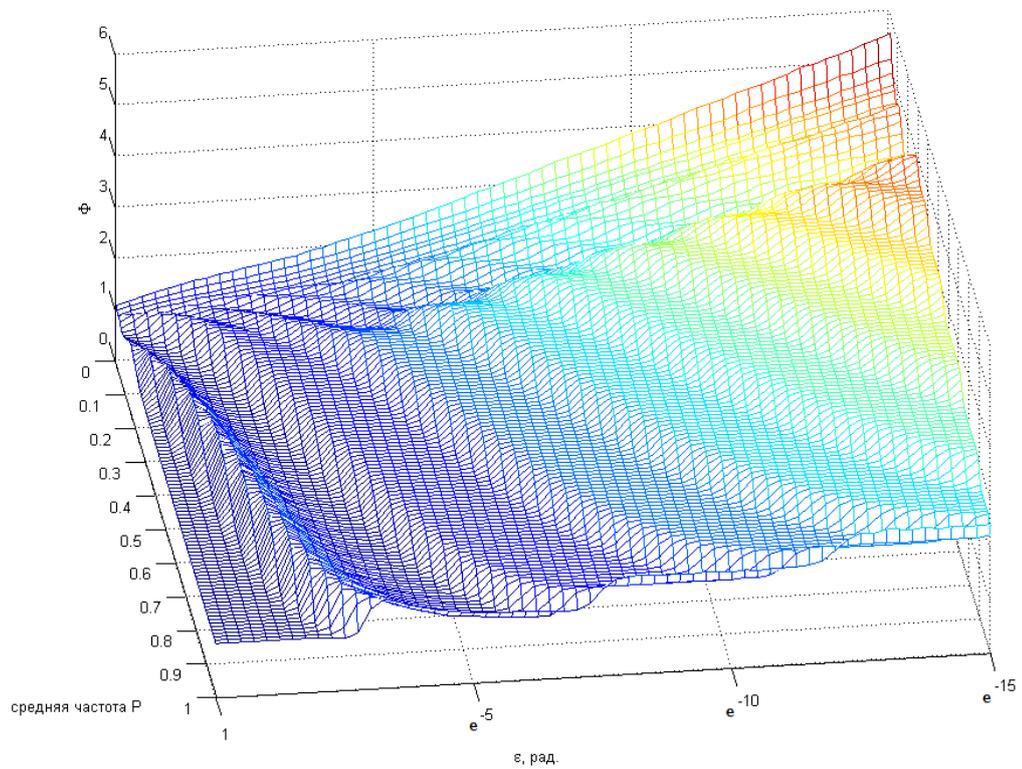
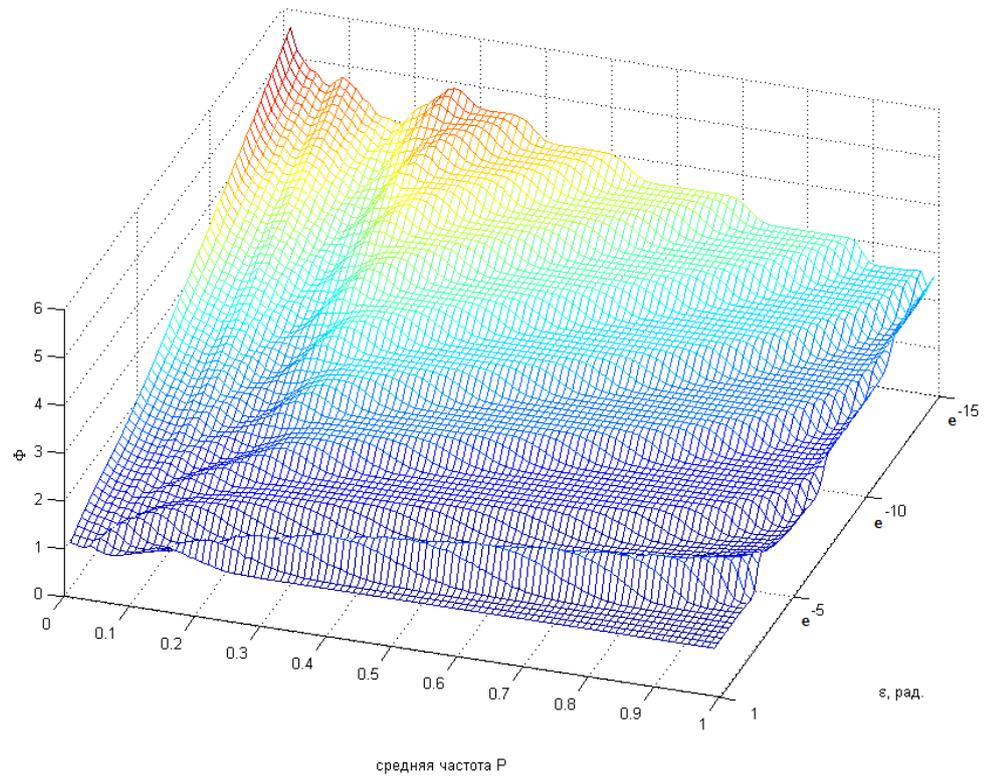


Рис. А.12. График зависимости характеристики ch. 2 из раздела 2.2 в зависимости от P и ϵ под двумя углами.

Приложение Б

Полное описание алгоритма

Б.1. Пояснение

Дробный момент времени дополнительно разбит на несколько блоков. Каждый занимает фиксированную временную часть шага и одинаков по продолжительности для всех шагов. Все правила формулируем для произвольного шага k , которые выполняются от лица произвольного агента с условным номером i .

Б.2. Целые моменты

Агенты согласуют параметры сеток и при необходимости переходят в окрестность узла новой сетки.

1. Передает значения параметров своей сетки $\theta_i(k)$, $d_i(k)$ сенсорам из $\mathcal{N}_i(k)$;
2. Принимает параметры сеток от сенсоров из $\mathcal{N}_i(k)$;
3. Определяет относительное положение и ориентацию сенсоров из $\mathcal{N}_i(k)$ в своей системе отсчета;
4. Строит правую декартову систему отсчета, ось абсцисс которой ориентирована по направляющей решетки данного сенсора в начале шага k ;
5. Для всех соседей j из $\mathcal{N}_i(k)$ производит пересчет полученных значений параметров $\theta_j(k)$, $d_j(k)$ в только что полученную систему координат, используя данные из пунктов 2 и 3, рассчитанные таким образом параметры обозначим за $\tilde{\theta}_j(k)$, $\tilde{d}_j(k)$;
6. Изменяет параметры своей сетки по правилам:
 - а. Для нахождения абсциссы x_i использует формулу

$$\tilde{x}_i(k+1) = \min_{j \in \mathcal{N}_i(k)} \tilde{x}_j(k);$$

и полученное значение $\tilde{x}_i(k+1)$ переносит в свою систему отсчета.

- б. Ордината $y_i(k+1)$ и угол $\theta_i(k+1)$ определяются аналогично (с точностью до замены x на y или θ);
- 7. Определяет расстояние до ближайшего узла обновленной сетки $d_i(k+1)$ и если оно не превосходит R_{tr} , то остается на месте и заканчивает целый момент;
- 8. Если указанное расстояние превосходит R_{tr} ,
 - а. Определяет множество $S_i(k)$ узлов обновленной сетки, для которых расстояние до сенсора не превосходит $2g + R_{tr}/2$;
 - б. Выбрасывает из $S_i(k)$ узлы, $(R_{tr}/2)$ -окрестность которых не пересекается с g -окрестностью области G ;
 - в. С вероятностью $\frac{1}{|S_i(k)|}$ выбирает один из узлов $S_i(k)$;
 - г. Перемещается в пересечение $(R_{tr}/2)$ -окрестность этого узла и g -окрестности области G ;

Б.3. Дробные моменты

Блок 1

Каждый агент решает, будет он «контролировать» ближайший узел или будет искать новый.

1. Определяет ближайший узел ζ своей решетки и множество $S_i(k)$ сенсоров, лежащих в R_{tr} -окрестности узла ζ ;
2. Если множеству $S_i(k)$ принадлежит только агент i , принимает решение о постановке узла ζ под собственный «контроль» и переходит к п. 4;
3. Иначе
 - а. Строит правую декартову систему отсчета, ось абсцисс которой ориентирована по направляющей решетки данного сенсора;
 - б. Определяет множество агентов из $S_i(k)$, у которых абсцисса превосходит его собственную абсциссу \bar{x}_i ;
 - і. Если оно непусто, принимает решение искать пустой узел и переходит к п. 4;

- ii. В противном случае определяет множество сенсоров из $S_i(k)$, у которых абсцисса равна \bar{x}_i ;
 - А. Если среди них есть сенсор, ордината которого превосходит ординату сенсора i , принимает решение искать пустой узел и переходит к п. 4;
 - Б. В противном случае принимает решение «контролировать» узел ζ ;
- 4. Ожидает окончание времени, отведенного для Блока 1.

Блок 2

Агенты ищут незанятые узлы решетки рядом с собой и при наличии определяют будущее значение своего параметра p или направление приоритетного движения.

- 1. Формирует множество $V_i(k)$ узлов своей текущей решетки из g -окрестности области G на расстоянии одного ребра (сетки) от ζ , определенному в Блоке 1;
- 2. В зависимости от принятого решения в Блоке 1:
 - а. Если агент i «контролирует» узел ζ , то принять $pLast_i(k) = 0$;
 - б. Если агент i не «контролирует» узел ζ , то принять $p_i(k) = 0$;
- 3. Определяет множество узлов из $V_i(k)$ в R_{tr} -окрестности которых нет агентов;
 - а. Если это множество не пусто, то выполняет одно из следующих действий и переходит к п. 4;
 - i. Если агент i «контролирует» узел ζ , то запомнить, что в Блоке 4 будет принят $p_i(k + 1) = 1$;
 - ii. Если агент i не «контролирует» узел ζ , то равновероятно выбирается один из этого множества и запоминает его как приоритетный;
 - б. Иначе
 - i. Если агент i «контролирует» узел ζ , то проверить текущее значение параметра $p_i(k)$ и если $p_i(k) = 1$, то запомнить, что в Блоке 4 будет принято $p_i(k + 1) = 0$;
- 4. Ожидает окончание времени, отведенного для Блока 2.

Блок 3

В этот момент все агенты пытаются строить новое значение p или наиболее приоритетный узел сетки. При этом учитываются решения агентов из Блока 2.

1. Передает свое текущее значение параметра $p_i(k)$ всем агентам из зоны коммуникации и если в Блоке 2 было определено новое значение для параметра $p_i(k+1)$ или выбран приоритетный узел, переходит к п. 3;
2. Иначе
 - а. Принимает параметры $p_j(k)$ от агентов из зоны коммуникации и формирует множество $W_i(k)$ сенсоров с $p_i(k) \neq 0$, лежащих в объединении R_{tr} -окрестностей узлов из $V_i(k)$ определенного в Блоке 2;
 - б. Если множество $W_i(k)$ пусто, то выполняет одно из действий и переходит к п. 3.;
 - i. Если агент i «контролирует» узел ζ , то запомнить, что в Блоке 4 будет принято $p_i(k+1) = 0$;
 - ii. Если агент i не «контролирует» узел ζ , то запомнить, что в Блоке 4 будет принято $pLast_i(k+1) = 0$;
 - в. Иначе
 - i. Если агент i «контролирует» узел ζ , то запомнить, что в Блоке 4 будет принято

$$p_i(k+1) = 1 + \min_{j \in W_i(k)} p_j(k);$$
 - ii. Если агент i не «контролирует» узел ζ , то определить

$$pLast_i(k+1) = \min_{j \in W_i(k)} p_j(k),$$
 и сравнить значения $pLast_i(k+1)$ и $pLast_i(k)$. Если они совпали, то
 - А. Находит все узлы из $V_i(k)$, в R_{tr} -окрестности которых расположен сенсор j с $p_j(k) = pLast_j(k+1)$;
 - Б. Равновероятно выбирает один из них и запоминает его как приоритетный;
3. Ожидает окончание времени, отведенного для Блока 3.

Блок 4

Агенты утверждают значения параметров p и $pLast$ для следующих шагов и если определены приоритетные узлы, то перемещаются в них.

1. В зависимости от принятого решения в Блоке 1:
 - а. Если агент i «контролирует» узел ζ и $p_i(k+1) > p_{max}$, то принимает $p_i(k+1) = 0$;
 - б. Если агент i не «контролирует» узел ζ и определил приоритетный узел в одном из Блоках 1-3, то перемещается в $(R_{tr}/2)$ -окрестность этого узла и g -окрестность области G ;
2. Утверждает для следующего шага значения параметров p по следующему принципу:
 - а. Если в течение выполнения Блоков 1–4 было определено значения $p_i(k+1)$, то использует его в дальнейшем;
 - б. Иначе принимает

$$p_i(k+1) = p_i(k);$$
3. Процесс утверждения для $pLast$ аналогичен п. 2 с точностью до замены p на $pLast$;
4. Ожидает окончание времени, отведенного для Блока 4.