

Имитационная модель выбора оптимального маршрута

Т. Н. Астахова, А. А. Бердникова, Д. А. Маслова, М. В. Денисова

Аннотация – Рассматривается имитационная модель, предназначенная для решения задач поиска оптимального пути в сенсорном поле.

В данной статье представлена разработка имитационной модели, направленной на решение задач поиска оптимального пути в сенсорных полях. Актуальность исследования обусловлена растущими требованиями к эффективности маршрутизации в современных сенсорных сетях, которые широко применяются в различных областях, таких как умные города, мониторинг окружающей среды и логистика.

В статье представлены основные принципы построения имитационной модели, включая алгоритмы поиска пути и методы оценки эффективности маршрутов. В результате работы были разработаны стратегии, позволяющие учитывать динамические изменения в сенсорном поле.

Экспериментальные результаты демонстрируют высокую точность и эффективность предложенной модели в различных сценариях. Исследование подтверждает возможность применения имитационного моделирования для оптимизации маршрутов в сложных и изменяющихся условиях, что открывает новые горизонты для повышения эффективности работы сенсорных систем.

Ключевые слова – алгоритм A*, алгоритм Дейкстры, алгоритмы маршрутизации, имитационная модель, оптимальный путь, сенсорное поле.

Статья получена 17 января 2025

Астахова Татьяна Николаевна, Нижегородский государственный инженерно-экономический университет (e-mail: ctn_af@mail).

Бердникова Анна Александровна, Нижегородский государственный инженерно-экономический университет (e-mail: anya-romanova-07@yandex.ru).

Маслова Дарья Александровна, Нижегородский государственный инженерно-экономический университет (e-mail: dasha.kirilova.96@bk.ru).

Денисова Марина Валерьевна, ООО «Протон», г. Нижний Новгород (email: kngraa07@gmail.com)

I. ВВЕДЕНИЕ

В настоящее время развитие технологий интернета вещей (IoT-технологий) [1] является одним из актуальных направлений развития в области промышленности, медицины, военных технологий и сельского хозяйства [2]. Одним из ключевых аспектов внедрения IoT-технологий является эффективное управление сетью связи и передачей данных [3]. Эффективное управление сетью включает в себя контроль качества обслуживания, управление пропускной способностью, минимизацию задержек и

потерю данных, а также мониторинг и адаптацию сети к изменяющимся условиям и требованиям [4]. При этом необходимо учитывать энергоэффективность и экономические аспекты, так как снижение затрат и оптимизация использования ресурсов являются важными пунктами успеха внедрения IoT-технологий [5].

Имитационные модели играют важную роль в развитии технологии интернета вещей, так как они позволяют анализировать и оптимизировать работу систем, управляемых данными устройствами [6-8]. Использование имитационных моделей позволяет предвидеть возможные эффекты внедрения новых технологий интернета вещей, а также исследовать влияние различных параметров и стратегий управления на производительность и надежность системы [2].

Таким образом, разработка имитационной модели [9], которая показывает работу алгоритма маршрутизации [10], актуальна в наше время, так как благодаря использованию имитационных моделей можно провести детальное исследование эффективности сетей интернета вещей, предугадать возможные сбои и недостатки выбранной модели и предоставлять наглядные примеры в обучающем процессе.

Разработанная имитационная модель станет отличной платформой для проведения экспериментов по поиску оптимального пути между устройствами и может служить в качестве наглядного материала при обучении.

Также с увеличением объемов транспортных потоков и усложнением логистических систем выбор оптимального маршрута становится одной из ключевых задач в области транспортной логистики и управления движением. Эффективное решение этой задачи может существенно сократить время в пути, снизить затраты на топливо и улучшить общую эффективность транспортных операций. В данной статье будет разработана имитационная модель, позволяющая анализировать и выбирать оптимальные маршруты в различных условиях.

Разработанная модель позволит оценивать различные маршруты на основе множества факторов, таких как расстояние, время в пути, заторы, погодные условия и другие переменные. Модель должна помочь в принятии обоснованных решений при выборе маршрутов.

Ожидается, что разработанная имитационная модель позволит: ускорить процесс выбора оптимального маршрута, снизить затраты на транспортировку, повысить уровень удовлетворенности клиентов за счет более точного соблюдения сроков доставки, предложить новые

подходы к управлению транспортными потоками в условиях неопределенности.

Данная статья будет полезна как для исследователей в области логистики и транспортных систем, так и для практиков, заинтересованных в оптимизации процессов выбора маршрутов. Результаты работы могут быть использованы для дальнейших исследований и разработок в этой области.

II. ПОСТАНОВКА ПРОБЛЕМЫ

С развитием Интернета вещей (IoT) и увеличением количества подключенных устройств, задача оптимизации маршрутов для передачи данных становится всё более актуальной. В условиях динамично меняющейся среды IoT, где устройства могут быть расположены в различных местах и взаимодействовать друг с другом, необходимо разработать эффективные методы выбора оптимальных маршрутов для передачи информации. Это позволит улучшить качество связи, повысить энергоэффективность и сократить задержки в передаче данных.

Имитационные модели представляют собой мощный инструмент для анализа и оптимизации маршрутов в системах IoT. Они позволяют смоделировать различные сценарии взаимодействия устройств, а также учитывать множество факторов, таких как изменение нагрузки на сеть, расстояние между устройствами и характеристики каналов связи. В рамках данной статьи рассматривается создание имитационной модели выбора оптимального маршрута, в которой будут реализованы два известных алгоритма: алгоритм Дейкстры и алгоритм A*.

Алгоритм Дейкстры, будучи классическим методом для поиска кратчайшего пути в графах, обеспечивает нахождение оптимального маршрута, однако его применение в условиях IoT может быть ограничено из-за высокой вычислительной сложности и необходимости постоянного обновления данных о сети. Алгоритм A*, в свою очередь, использует эвристические функции для ускорения поиска, что делает его более подходящим для динамичных сред, но требует точной настройки эвристик для достижения наилучших результатов.

Таким образом, перед нами стоит ряд ключевых вопросов: как эффективно интегрировать оба алгоритма в единую имитационную модель, как оценить их производительность в различных сценариях IoT и какие параметры влияют на выбор оптимального маршрута. Решение этих вопросов позволит не только улучшить качество передачи данных в системах IoT, но и разработать универсальные подходы к оптимизации маршрутов, которые могут быть применены в различных областях, таких как умные города, транспортные системы и промышленные сети.

III. МАТЕРИАЛЫ И МЕТОДЫ

В рамках данного исследования рассматривается протокол (или алгоритм) OSPF [11]. Выбор данного протокола заключается в том, что он широко используется в корпоративных сетях, провайдерских сетях и других масштабируемых сетях, где требуется эффективная маршрутизация и управление трафиком [10]. OSPF обеспечивает быструю сходимость и высокую производительность сети, что делает его одним из популярных протоколов маршрутизации в IP-сетях [11, 12].

Для реализации имитационной модели было принято использовать алгоритмы Дейкстры и A* [12]. Алгоритм Дейкстры – это алгоритм поиска кратчайшего пути от одной из вершин графа до всех остальных вершин во взвешенном графе с неотрицательными весами ребер [12, 13]. **Ошибка! Источник ссылки не найден.**

Математическая формула, используемая в алгоритме Дейкстры [13], для расчета кратчайшего пути от начальной вершины до всех остальных вершин выглядит следующим образом [12]:

$$d(v) = \min\{d(v), d(u) + w(u, v)\},$$

где: $d(v)$ – это длина кратчайшего пути от начальной вершины до вершины v ; $d(u)$ – это длина кратчайшего пути от начальной вершины до вершины u ; $w(u, v)$ – это вес ребра между вершинами u и v .

Эта формула означает, что для каждой вершины V алгоритм Дейкстры находит кратчайший путь от начальной вершины до вершины V , обновляя его, если найден новый путь с меньшей длиной через вершину U . Формула связывает длины кратчайших путей к текущей вершине с длинами кратчайших путей к соседним вершинам и весами ребер в графе. Он использует «жадный» метод, выбирая каждый раз ближайшую вершину с наименьшим путем [13].

Алгоритм A*. Данный алгоритм является расширенной версией алгоритма Дейкстры, так как для расчёта стоимости пути учитывает не только расстояние от текущего узла до начального, но и оценочную (или эвристическую) стоимость от текущего узла до конечного. В качестве оценочной стоимости может выступать [12-16]:

– евклидово расстояние: обычно используется, если движение в поле возможно во всех доступных направлениях и равно прямому расстоянию между текущей точкой и целью, рассчитанному по теореме Пифагора [17,18];

– манхэттенское расстояние [18]: используется, если поле представлено в виде сетки и двигаться можно только по осям X и Y, а по диагонали передвижение запрещено, т.е. движение возможно только в четырех направлениях: вверх, вниз, влево, вправо, рассчитывается как сумма абсолютных

разностей соответствующих координат начального и конечного узла;

– расстояние Чебышёва [19]: используется, если движение в поле возможно только по вертикали, горизонтали и диагонали, рассчитывается как максимум из абсолютных разностей координат.

Общая формула для расчета оценки стоимости пути $f(n)$ от начального узла до конечного имеет вид [16]:

$$f(n) = g(n) + h(n),$$

где: $g(n)$ – это длина пути от стартового узла до текущего (расстояние по ребру); $h(n)$ – это оценочная стоимость пути от текущего узла до цели (например, расстояние «напрямую» от текущего узла до конечного узла).

Каждый шаг алгоритм исследует соседние узлы, находит узел с наименьшей стоимостью пути и делает его текущим узлом. Этот процесс продолжается до тех пор, пока алгоритм не достигнет конечного узла.

В качестве используемого метода распределения устройств в сенсорном поле было принято выбрать равномерное и нормальное распределение [20]. Равномерное распределение обеспечит одинаковую плотность точек во всем пространстве, а нормальное распределение позволит обеспечить естественность, так как многие реальные феномены следуют нормальному распределению или имеют центральную кластеризацию.

Для реализации проекта по созданию имитационной модели работы алгоритма маршрутизации необходимо использование программных, языковых и технических средств. В качестве программного средства будет использоваться AnyLogic [21-23].

IV. Модель

Имитационная модель состоит из нескольких важных элементов: сенсорное поле, сенсорные устройства (узлы связи), ребра, соединяющие сенсорные устройства между собой (линии связи), матрица расстояний и мощностей, блок выбора алгоритма маршрутизации, блок выбора начального и конечного устройства, блок вывода полученного маршрута, блок с выводом полученного расстояния и затраченной мощности.

Так же дополнительно можно добавить имитацию загруженности линий связи (ребер) и самих устройств

с автоматической генерацией трафика между устройствами. Исходя из этого, добавить на модель информацию о трафике: время передачи, вероятность передачи и общее количество переданных, потерянных и не нашедших адресата сообщений

Имитационная модель состоит из 4 основных агентов:

1. Main. В нем объединена работа всех остальных агентов, а также представлен пользовательский интерфейс.

2. Matr. Данный агент отвечает за формирование матрицы расстояний и мощностей.

3. Node. В данном агенте реализована работа узлов связи: координаты, информация о линиях связи у узлов и загруженность узла. Так же внутри агента создана имитация трафика.

4. Road. Этот агент отвечает за визуализацию ребер между узлами (линий связи) и вывода на ребрах информации о расстоянии или мощности.

Имитационная модель располагается в непрерывном пространстве, имеющем заданную длину и ширину. Тип расположения агентов в данном пространстве задан в виде кода на языке Java [24, 25].

Первым шагом в модели было реализовано сенсорное поле. Поле располагается внутри агента «Main».

Сенсорное поле было создано при помощи геометрической фигуры «Прямоугольник» из палитры «Презентация». Размер фигуры составляет 500 на 500 пикселей и имеет заливку цветом для его выделения на пользовательском интерфейсе.

Для создания узлов связи (сенсорных устройств) в модели добавлено популяция агентов «nodes» и связанный с данной популяцией параметр «node_count», где пользователь указывает количество узлов. Визуальное представление агента «node». Внутри овала был добавлен элемент «Текст» для того, чтобы в дальнейшем выводить порядковый номер сенсорного устройства. Все элементы были объединены в одну группу.

Основные элементы, находящиеся внутри агента «node»: параметры x и y , в которых хранятся координаты устройства; коллекция «my_roads», в которой хранятся линии связи с другими устройствами; переменная «feromon», где хранится информация о загруженности устройства.

Также в агенте «node» реализована имитация трафика при помощи элементов библиотеки моделирования процессов (рис. 1).

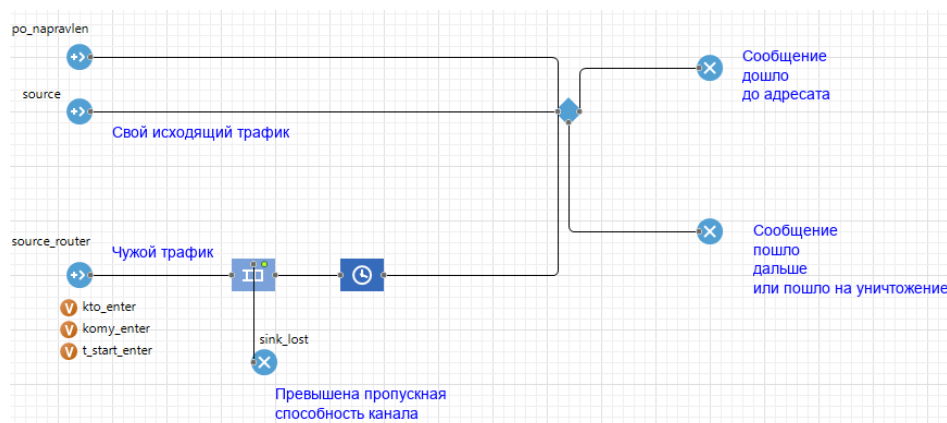


Рис. 1 – Процесс генерации и распределения трафика

Блок «source» создает трафик из текущего устройства и передает его в случайно выбранное устройство назначения. Блок «source_router» создает дополнительный трафик, как бы «ловит» трафик от других устройств. В блоке «po_napravlen» по таймеру «event» генерируется дополнительный трафик уже по тому направлению, по которому мы выбираем на агенте «Main» (направление, по которому идет поиск кратчайшего пути [26]). В ветке «source_router» расположен блок очереди обслуживания трафика «queue». Если превышена пропускная способность, то весь лишний трафик уходит в конечный блок «sink_lost», где происходит потеря/удаление трафика.

После блока очереди идет блок задержки трафика «delay», где происходит имитация обслуживания трафика за выделенное время. Время обслуживания задается в переменной «vremya_obrabotki».

После создания и отсеивания трафика идет его распределение при помощи блока «selectOutput» на трафик, который: успешно доходит до конечного устройства; передается в другие устройства, где заново проходит процесс распределения; уничтожается (например, если пути не существует).

В конце всех операций вся информация о трафике передается в агент «Main» для подсчета и обобщения, представленная на рисунке 2.

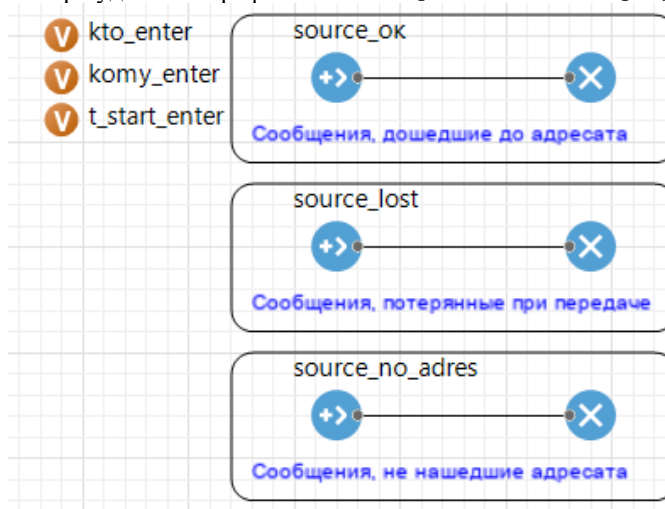


Рис. 2 – Ветки вывода информации о трафике

Переменные «kto_enter», «komy_enter» и «t_start_enter» – служебные, и хранят текущие значения по трафику о том, кто передал сообщение, кому передал и в какое время. Все полученные результаты из веток показываются при помощи элемента «Диаграмма с накоплением».

Агент «Road», в нем реализовано представление линий связи между устройствами. В агенте «Main» содержится популяция агентов «roads», в которой содержится заданное количество агентов «Road», изначально популяция пуста, так как нужное число линий связи напрямую зависит от количества и расположения сенсорных устройств. Агент «Road» визуальнo представляется на модели в виде линии. В

дальнейшем пользователь сможет при помощи радиокнопок выбирать выводимую информацию.

Каждый агент «Road» хранит в себе: параметры «nod1» (от какого устройства линия начинается) и «nod2» (до какого устройства линия идет); переменная «feromon», в которой так же, как и в устройстве хранится информация о загруженности; переменная «power», в которой содержится информация о мощности, затрачиваемой на прохождение этого пути.

Для вывода на имитационную модель информации о всех расстояниях и затрачиваемых мощностях между узлами был создан агент «Matr». Популяция агентов «matrs» созданная внутри агента «Main»

содержит в себе элементы матрицы. Каждый элемент матрицы выводится в виде квадрата, в котором сначала выводится значение расстояния, а ниже выводится затрачиваемая мощность при прохождении этого расстояния.

Затрачиваемая мощность рассчитывается в агенте «Main» при помощи формулы Фрииса [27-29]:

$$P_{пер} = \frac{P_{np}}{C_{np} \cdot C_{пер} \cdot \frac{\gamma}{(4 \cdot \pi \cdot r)^2}}.$$

После запуска модели матрица принимает вид, представленный на рис. 3.

	0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	0	0	0	0	0	0
1	0	0	99	0	80	0	0	0	0	0
2	0	0	98	0	64	0	0	0	0	0
3	0	99	0	115	67	0	0	0	0	0
4	0	98	0	132	46	0	0	0	0	0
5	0	0	115	0	0	106	0	0	0	0
6	0	0	132	0	0	113	0	0	0	0
7	0	80	67	0	0	0	0	0	96	0
8	0	64	46	0	0	0	0	0	93	0
9	0	0	0	106	0	0	0	86	0	75
0	0	0	0	113	0	0	0	74	0	57
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	86	0	0	0	61
4	0	0	0	0	0	74	0	0	0	37
5	0	0	0	0	96	0	0	0	0	0
6	0	0	0	0	93	0	0	0	0	0
7	0	0	0	0	0	75	0	61	0	0
8	0	0	0	0	0	57	0	37	0	0
9	0	0	0	0	0	0	0	0	0	0

Рис. 3 – Матрица расстояний и мощностей

Теперь рассмотрим главные функции, реализованным в агенте «Main». При запуске модели будет выполняться функция «init» (инициализация). Порядок действий, которые совершаются в данной функции:

1. Последовательно создается заданное количество сенсорных устройств. Координаты каждого устройства задаются при помощи выбранного метода распределения точек в поле. При генерации точек учитывается указанное допустимое минимальное расстояние между ними, чтобы каждое из новых устройств не было ближе к уже существующим устройствам на указанное расстояние.

2. Все устройства сравниваются попарно и, если расстояние между ними меньше, чем заданная максимальная дистанция, то создается линия связи (популяция «roads»). Без указания допустимой максимальной дистанции все устройства будут связаны со всеми. Визуально и логически это будет хаос и пропадет смысл поиска пути между ними, так как все устройства будут связаны друг с другом.

3. Информация о линиях связи вносится в множество устройств. Это нужно для того, чтобы каждое устройство знало с какими устройствами оно имеет связь для последующей маршрутизации трафика.

4. Выполняется функция «calc_priority», которая рассматривает все линии связи и устройства, сортируя их по трафику. Далее заполняются две коллекции с весами путей и весами устройств, где они отсортированы по степени их загруженности. Загруженность хранится в соответствующей переменной «feromon» как в устройствах, так и в линиях связи.

5. Устройства и линии связи на модели раскрашиваются от красного до черного цвета в соответствии с рассчитанными загруженностями. Чем загруженней, тем краснее линия и точка с устройством. Исключительно механическая процедура, которая добавляет понятную визуализацию и по существу ничего не меняет, хотя в последствии можно это использовать для альтернативных методов маршрутизации.

6. Обновляются загруженности устройств и линий связи. Эта функция взаимодействует с функциями, описанными в шагах 4 и 5. Все три функции, по сути, работают вместе, просто поделены для удобства.

7. Создаются элементы матрицы – популяция агентов «matrs». Для каждого элемента рассчитывается расстояние и мощность. Как идет расчёт было описано выше в пояснениях к популяции агентов.

8. Выполняется работа с Excel-файлом: записываются координаты каждого из устройств, записываются данные из матрицы (расстояния и мощности). Сам файл расположен в папке с проектом и его содержимое перезаписывается с каждым новым запуском модели.

На данном шаге выполнение функции «init» заканчивается и все работает самостоятельно.

После запуска модели пользователь должен выбрать начальное и конечное устройство, между которыми и будет происходить поиск кратчайшего пути [26]. Для ввода номеров начального и конечного устройств были добавлены текстовые поля. Затем пользователь нажимает на кнопку «Выбрать направление связи» и начинается выполнение

алгоритма поиска оптимального пути при помощи выбранного алгоритма (рис. 4).

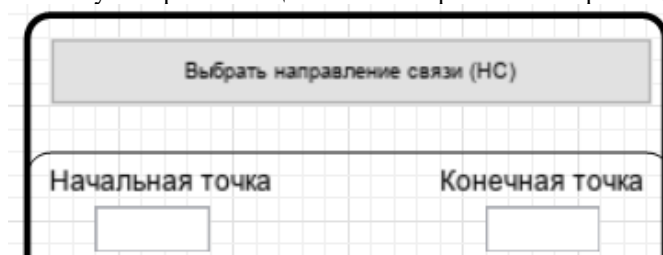


Рис. 4 – Текстовые поля и кнопка

Для выбора алгоритма в модели имеются радиокнопки «Дейкстра» и «А*».

После нажатия на кнопку «Выбор направления связи» выполняется несколько функций:

9. Выполняется сортировка линий связи по трафику и окрашивание линий связи и устройств в соответствии с загруженностью.

10. Рисуются найденный при помощи функции «get_Deijkstra_points» или «show_way_A» (в зависимости от выбранного алгоритма) кратчайший путь между начальным и конечным устройствами.

11. В текстовые поля на модель выводятся номера точек, через которые проходит найденный маршрут, среднее время передачи сообщения в выбранном направлении, расстояние и затраченная мощность.

Для удобства пользователя были добавлены радиокнопки для выбора значения, выводимого на

линиях связи. Можно выбрать или расстояние, или мощности. Был добавлен чек-бокс для того, чтобы пользователь мог вовсе скрыть все индексы на модели, если они мешают.

Основные элементы имитационной модели были описаны.

В имитационной модели были реализованы два алгоритма: Дейкстры и А*(A-star). Программный код алгоритма Дейкстры находится в функции «get_Deijkstra_points». Функция «show_way_A» в котором создана модифицированная версия алгоритма Дейкстры – алгоритм А*. Таким образом, в имитационной модели были реализованы два алгоритма поиска кратчайшего пути. Конечный вариант модели выглядит следующим образом:

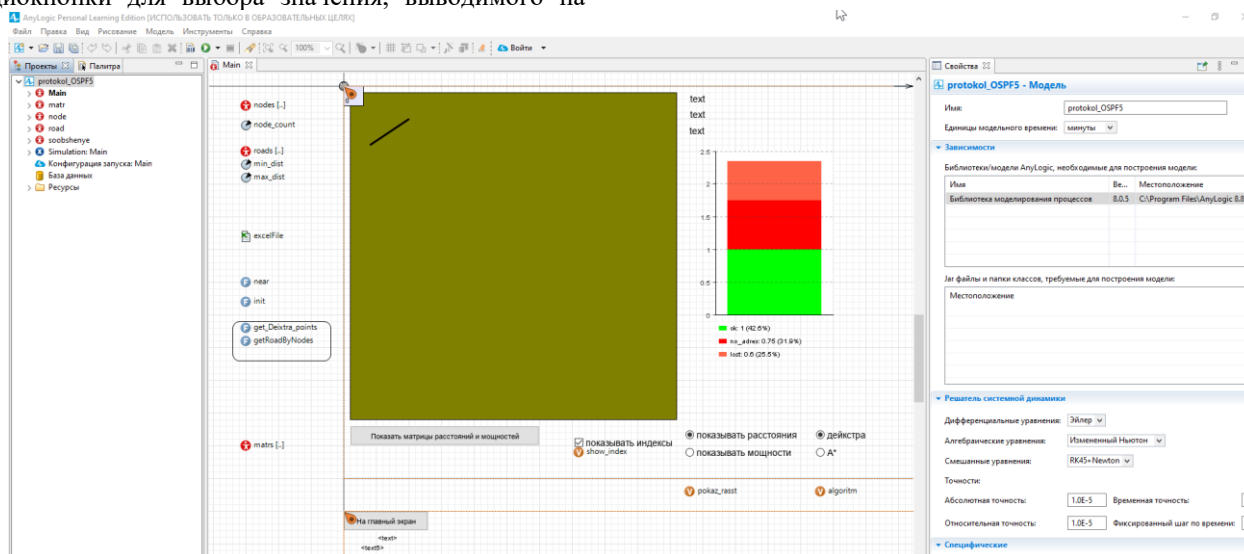


Рис. 5 – Открытая имитационная модель в программе AnyLogic PLE

Пример работы модели:

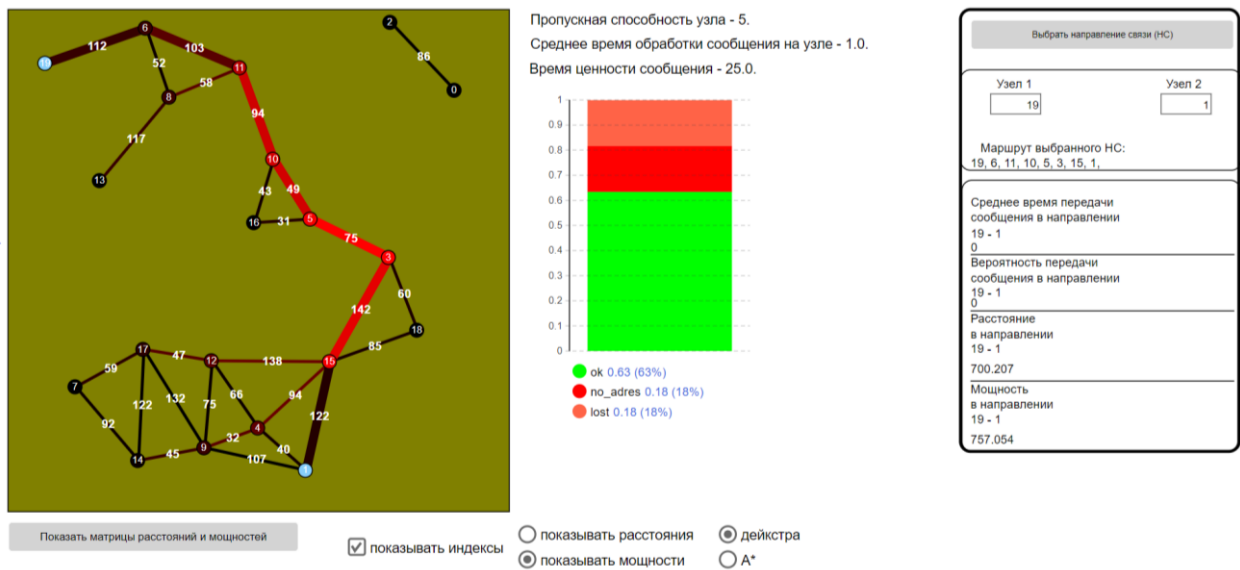


Рис. 6 – Поведение симуляции если между устройствами есть маршрут

Также в модели создается файл Microsoft Excel [30] с координатами устройств и расстояний между ними (рис. 7).

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	205,1015	422,2348	0	0	0	0	0	0	0	0	0	0	0	0
2	443,0666	129,6095	0	0	0	0	0	0	0	0	0	0	0	0
3	62,26252	288,6113	0	0	0	0	0	0	0	0	0	0	0	0
4	70,04871	357,385	0	0	0	0	0	0	0	0	0	0	0	0
5	198,9134	20,96864	0	0	0	0	0	0	0	0	0	0	0	0
6	154,8154	243,4313	0	0	0	0	0	0	0	0	0	0	0	0
7	375,7081	374,3669	0	0	0	0	0	0	0	0	0	0	0	0
8	262,8938	216,1655	0	0	0	0	0	0	0	0	0	0	0	0
9	401,9994	97,20098	0	0	0	0	0	0	0	0	0	0	0	0
10	238,4046	67,41801	0	0	0	0	0	0	0	0	0	0	0	0
11	44,13729	87,14565	0	0	0	0	0	0	0	0	0	0	0	0
12	159,1153	331,0895	0	0	0	0	0	0	0	0	0	0	0	0
13	228,3079	341,2145	0	0	0	0	0	0	0	0	0	0	0	0
14	174,8883	152,7685	0	0	0	0	0	0	0	0	0	0	0	0
15	424,2049	285,5595	0	0	0	0	0	0	0	0	0	0	0	0
16	479,358	321,576	0	0	0	0	0	0	0	0	0	0	0	0
17	344,6174	179,2326	0	0	0	0	0	0	0	0	0	0	0	0
18	152,8539	72,64131	0	0	0	0	0	0	0	0	0	0	0	0
19	498,5076	421,6605	0	0	0	0	0	0	0	0	0	0	0	0
20	288,7549	481,5812	0	0	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Рис. 7 – Файл Microsoft Excel

Меняя нужные параметры в модели и повторно запуская симуляции, можно проводить научные эксперименты.

V. ЭКСПЕРИМЕНТ МОДЕЛИ

Проведено несколько экспериментов при различных настройках модели. В каждом из экспериментов искали маршрут между первым и последним устройством.

Первый эксперимент был проведен с данными настройками: метод распределения: равномерное; количество точек: 20; минимально допустимое расстояние между точками: 20 единиц; максимально допустимое расстояние между точками: 120 единиц; алгоритм поиска кратчайшего пути: Дейкстра; количество запусков симуляции: 3.

Результат первого запуска симуляции представлен на рисунке 8.

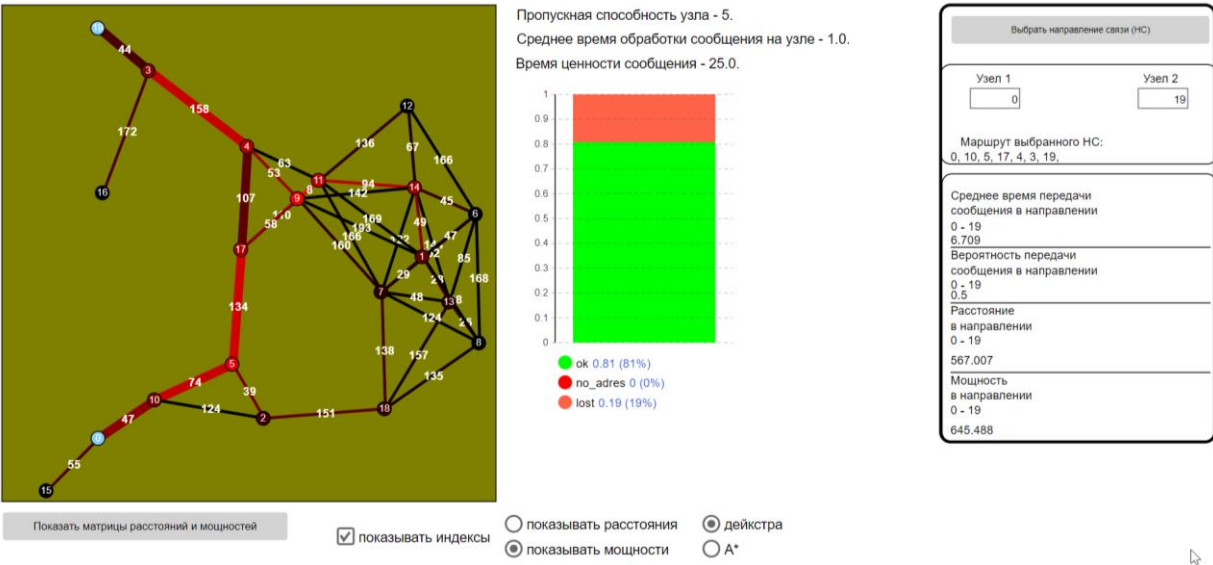


Рис. 8 – Эксперимент №1. Симуляция №1

Анализируя симуляцию, можно сказать, что между выбранными устройствами существует маршрут протяженностью 567 пикселей, проходящий через 5

устройств. Затрачиваемая мощность передатчиков составляет 645 у.е.Проведем второй запуск, результат представлен на рисунке 9.

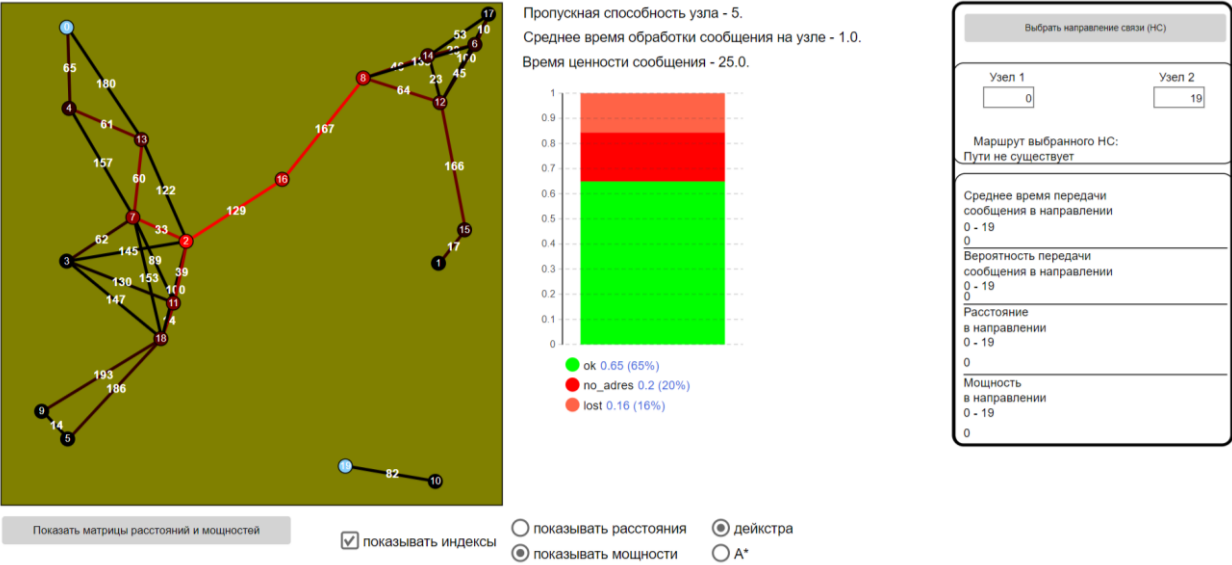


Рис. 9 – Эксперимент №1. Симуляция №2

Вторая попытка не увенчалась успехом, так как устройство под номером 19 отрезано от остальных устройств и имеет связь только с устройством номер 10.

Третий запуск имитации оказался успешным, между устройствами был найден оптимальный маршрут, проходящий через 4 других устройства.

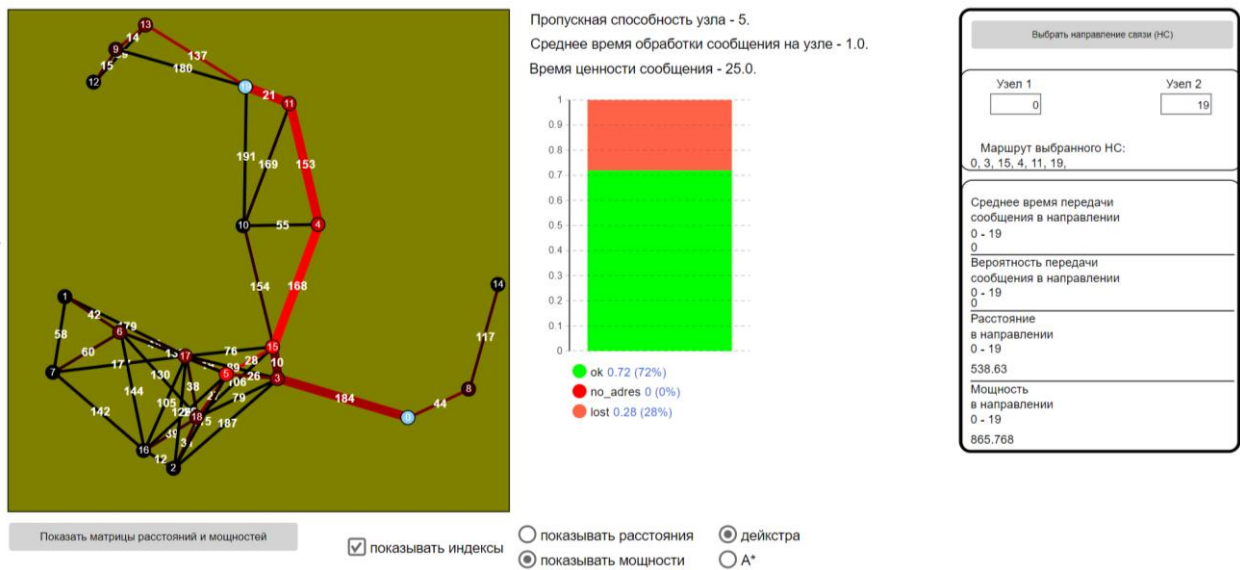


Рис. 10. Эксперимент №1. Симуляция №3

Анализируя рисунок 10, на котором представлена третья попытка можно сказать, что длина найденного маршрута составляет 538 пикселей, а затраченная мощность составляет 865 у.е.

По итогу первого эксперимента можно заметить, что устройства распределяются по полю не совсем равномерно, точнее даже дело возможно не в способе распределения, а в допустимом расстоянии, при котором образуется линия связи между устройствами. Далее остаются ограничения по расстояниям между устройствами и меняется способ распределения устройств, так как, изменяя расстояния, просто увеличивается количество линий связи между устройствами, которые уже, возможно, имеют связь через другие устройства.

Проведем второй эксперимент, немного изменив настройки прошлого эксперимента: метод распределения: нормальное; количество точек: 20; минимально допустимое расстояние между точками: 20 единиц; максимально допустимое расстояние между точками: 120 единиц; алгоритм поиска кратчайшего пути: Дейкстра; количество запусков симуляции: 3.

Для того чтобы использовалось нормальное распределение вместо равномерного, был изменен отрезок кода в функции «init». Встроенная библиотека «AnyLogic» имеет отдельную функцию «normal()», которую можно использовать для генерации значений по нормальному распределению.

После внесения изменений запустим первую симуляцию (рис. 11).

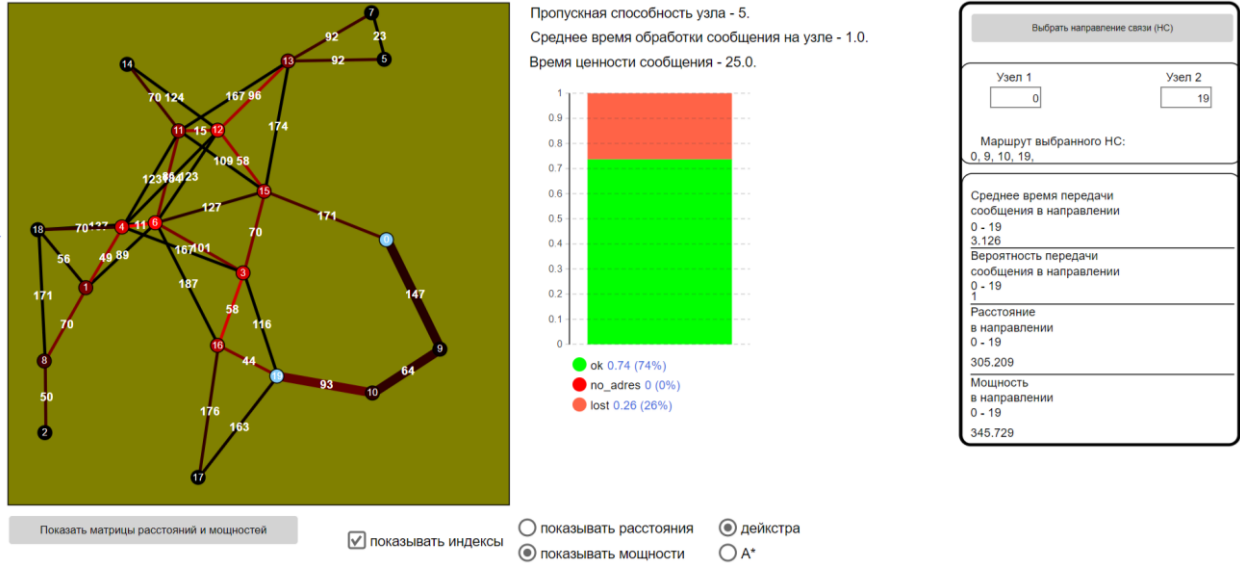


Рис. 11. Эксперимент №2. Симуляция №1

Маршрут между первым и последним устройстве существует и проходит через 2 других устройства. Длина маршрута составляет 305 пикселей, а

затрачиваемая мощность на передачу сообщения по данному маршруту составляет 345 у.е.

Запустим вторую симуляцию (Рис. 12).

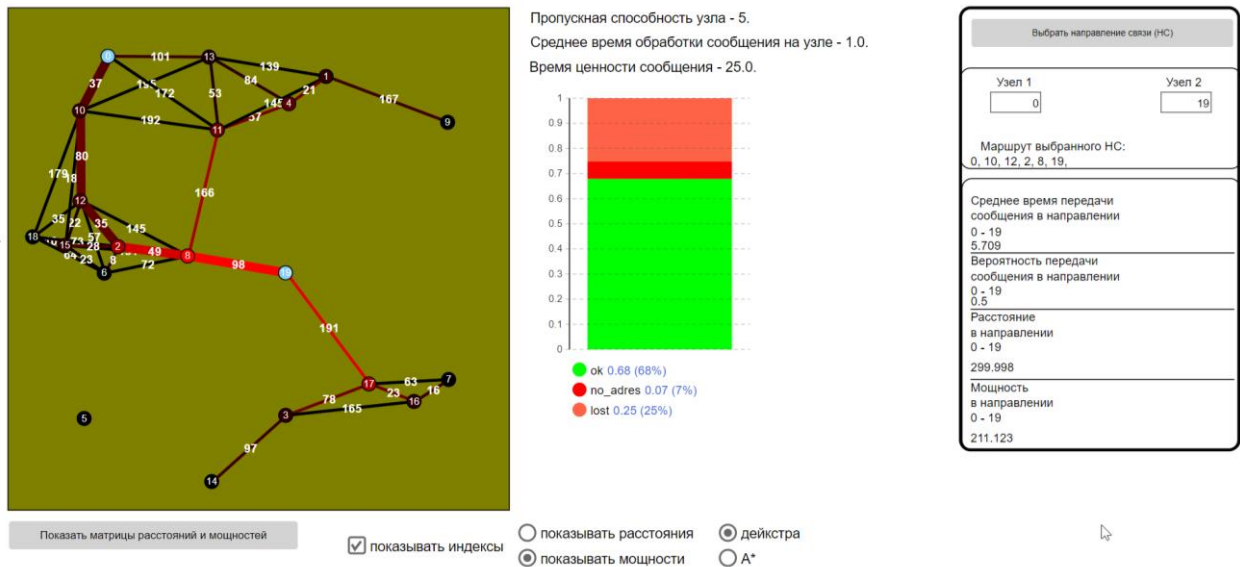


Рис. 12. Эксперимент №2. Симуляция №2

Полученный маршрут составляет 299 пикселя и проходит через 4 устройства. Затраченная мощность передатчиков составляет 211 у.е.

И наконец, запустим последнюю симуляцию в данном эксперименте (рис. 13).

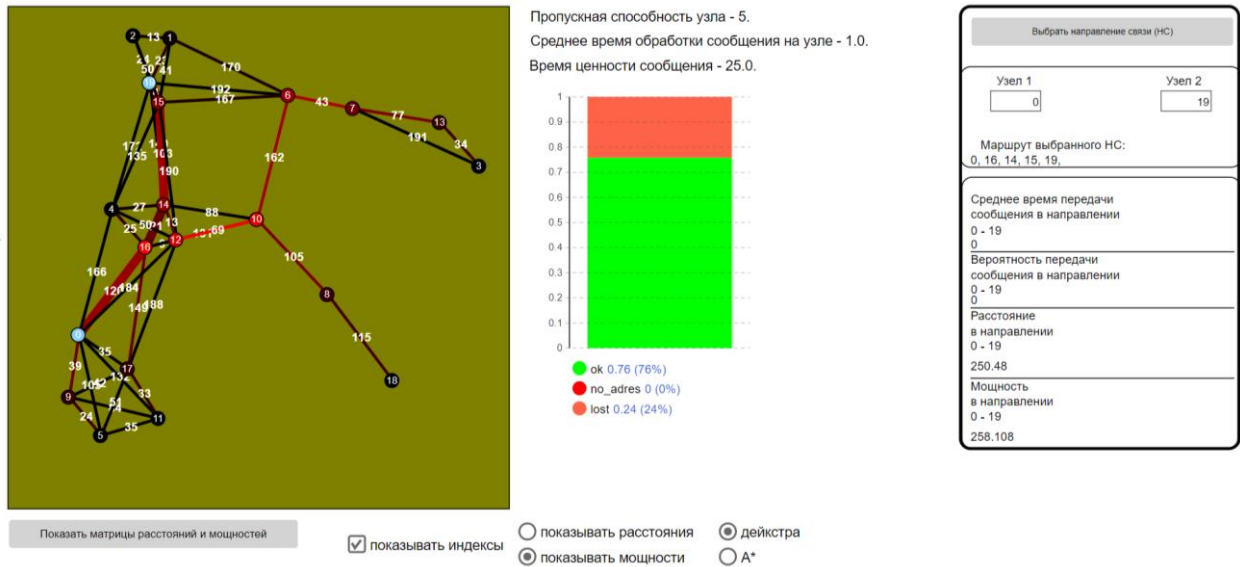


Рис. 13. Эксперимент №2. Симуляция №3

Найденный маршрут проходит через 3 устройства, длина маршрута составляет 250 пикселей, а затраченная мощность – 258 у.е.

По итогу трех проведенных симуляций видно, что нормальное распределение соизмеримо с равномерным, так как количество устройств, которые не имеют связи с другими устройствами практически такое же, как и при равномерном.

Подводя итог проведенных экспериментов, можно сказать, что созданная имитационная модель позволяет без проблем проводить эксперименты, изменяя необходимые параметры, тем самым подстраиваясь под требования пользователя и имеет в себе потенциал по усовершенствованию и

добавлению в него дополнительного функционала в зависимости от потребностей пользователя.

VI. ЗАКЛЮЧЕНИЕ

По итогам работы была достигнута основная цель: разработана имитационная модель для решения задач поиска оптимального пути в сенсорном поле. В результате анализа предметной области были выбраны алгоритмы и методы распределения, которые впоследствии были реализованы в модели.

Проведён хронометраж существующих средств по решению задач поиска маршрута. По итогам этого анализа выяснено, что создание собственного решения позволит значительно ускорить процесс

поиска кратчайшего пути, сократив его на 10-15 секунд для каждого эксперимента.

В ходе работы были определены требования к имитационной модели, которые включают в себя общие сведения, назначение и цели создания модели, функциональные и аппаратные требования, а также требования к документации.

Разработанная имитационная модель обеспечивает возможность проводить эксперименты с изменением необходимых параметров, что позволяет адаптироваться к требованиям пользователя. Модель также обладает потенциалом для улучшения и добавления новых функций в соответствии с потребностями пользователей.

В рамках исследования была разработана имитационная модель выбора оптимального маршрута, которая сочетает в себе традиционные алгоритмы Дейкстры и A^* , что позволяет значительно улучшить процесс принятия решений в области транспортной логистики. Основные аспекты новизны полученных результатов можно выделить следующим образом:

1. Комбинированный подход: в отличие от существующих решений, которые обычно используют один из алгоритмов, наша модель интегрирует алгоритмы Дейкстры и A^* для достижения более высокой точности и эффективности в выборе маршрутов. Это позволяет использовать преимущества обоих методов: точность поиска кратчайшего пути алгоритма Дейкстры и скорость работы алгоритма A^* за счет эвристического подхода.

2. Адаптивная система оценки: имитационная модель включает адаптивную систему, которая динамически настраивает параметры алгоритмов в зависимости от текущих условий (например, загруженности дорог, погодных условий и времени суток). Это новшество позволяет повысить актуальность и точность выбора маршрута в реальном времени.

3. Симуляция различных сценариев: разработанная имитационная модель предоставляет возможность симуляции различных сценариев, что позволяет пользователям оценивать эффективность различных маршрутов в условиях неопределенности. Это является важным преимуществом для логистических компаний, стремящихся оптимизировать свои операции.

4. Интуитивно понятный интерфейс: в отличие от многих существующих решений, наша модель предлагает интуитивно понятный интерфейс для ввода данных и получения результатов. Это делает ее доступной для широкого круга пользователей, включая тех, кто не имеет глубоких знаний в области программирования или теории графов.

5. Валидация и тестирование на реальных данных: полученные результаты были проверены на реальных данных, что подтверждает их практическую применимость и эффективность. Это обеспечивает дополнительную ценность для пользователей, так как

позволяет применять модель в реальных условиях транспортной логистики.

Таким образом, новизна разработанной имитационной модели выбора оптимального маршрута заключается в комплексном подходе, который сочетает в себе лучшие практики существующих алгоритмов и адаптируется к изменяющимся условиям, что делает ее более эффективной и применимой в современных условиях транспортной логистики.

БИБЛИОГРАФИЯ

- [1] Верзун Н. А., Воробьев А. И., Колбанев М. О., Цехановский В. В. Изучение технологий интернета вещей в рамках дисциплины «Инфокоммуникационные системы и сети» // Современное образование: содержание, технологии, качество. 2020. Т. 1. С. 71-74.
- [2] Утегенов Н. Б. Интернет вещей (IoT) и информационные системы // Universum: технические науки. 2023. Т. 7, вып. 112.
- [3] Laghari A. A., Wu K., Laghari R. A., Ali M., Khan, A. A. A review and state of art of Internet of Things (IoT) // Archives of Computational Methods in Engineering. 2021. pp. 1-19.
- [4] Верещагина Е. А., Капещкий И. О., Ярмонов А. С. Проблемы безопасности Интернета вещей: учеб. пособие // М.: Мир науки. 2021. Сетевое издание. URL: <https://izd-mn.com/PDF/20MNNPU21.pdf>.
- [5] Верзун Н. А., Воробьев А. И., Колбанев М. О. Технологии Интернета вещей // Санкт-Петербург: Санкт-Петербургский государственный экономический университет. 2020. 91 с.
- [6] Романова А. А. Имитационная модель информационного взаимодействия в интернете вещей // Известия СПбГЭТУ ЛЭТИ. 2022. № 8. С. 69-76. DOI 10.32603/2071-8985-2022-15-8-69-76.
- [7] Ефременков Я. А. Использование программного обеспечения Anylogic для создания имитационных моделей технических систем // Наукоемкие технологии и инновации (XXV научные чтения) : Сборник докладов Международной научно-практической конференции, Белгород, 23 ноября 2023 года. – Белгород: Белгородский государственный технологический университет им. В.Г. Шухова, 2023. С. 809-813.
- [8] Акопов А. С. Имитационное моделирование: учебник и практикум для вузов // Москва: Издательство Юрайт. 2023. 389 с.
- [9] Еремочкин С. Ю., Дорохов Д. В., Жуков А. А. Разработка имитационной модели однофазного асинхронного электропривода в среде динамического моделирования SimInTech // Вестник НГИЭИ. 2024. № 1(152). С. 59-71. DOI 10.24412/2227-9407-2024-1-59-71.
- [10] Астахова Т. Н., Кирилова Д. А., Колбанев М. О., Маслов Н. С. Разработка алгоритма маршрутизации для беспроводной сенсорной сети // Телекоммуникации. 2023. № 4. С. 30-38. DOI 10.31044/1684-2588-2023-0-4-30-38.
- [11] Бердникова, А. А., Верзун Н. А., Колбанев М. О. Протоколы сетевого уровня инфокоммуникационных систем и сетей // Санкт-Петербург : Санкт-Петербургский государственный экономический университет, 2024. 70 с. ISBN 978-5-7310-6454-5.
- [12] Олифер В. Г., Олифер Н. А. Компьютерные сети. Принципы, технологии, протоколы // СПб.: Питер. 2016. 992 с.
- [13] Калькулятор кратчайшего пути Дейкстры. [Электронный ресурс]. URL: <https://www.sas.com.ru/wp/ru/kalkuljator-kratchajshhego-puti-dejkstry/>.
- [14] Манаква В. А., Костин А. С. Анализ методов решения задачи коммивояжера при помощи алгоритмов Дейкстры и A^* // Системный анализ и логистика. 2023. № 3(37). С. 136-142. DOI 10.31799/2077-5687-2023-3-136-142.
- [15] Алгоритм Дейкстры – нахождения кратчайшего пути [Электронный ресурс]. URL: <https://www.matematicus.ru/diskretnaya-matematika/reshenie-algoritma-dejkstry> (Дата обращения: 09.09.2024).
- [16] Плотников О. А., Подвальный Е. С. Решение задачи поиска оптимального пути между двумя точками на графе с нерегулярным весом ребер // Вестник Воронежского государственного технического университета. 2012. Т. 8, № 6. С. 22-26.
- [17] Вторица К. С., Осипова М. В., Онофрейчук В. А. Теорема Пифагора // Способы, модели и алгоритмы управления модернизационными процессами. 2021. С. 157-159.

- [18] Половикова О. Н., Фокина В. В. Использование евклидова и манхэттенского расстояний в качестве меры близости для решения задачи классификации // Известия Алтайского государственного университета. – 2010. – № 1-1(65). С. 101-102.
- [19] Кривулин Н. К., Брюшинин М. А. Решение задачи о размещении двух объектов в пространстве с метрикой Чебышёва // Вестник Санкт-Петербургского университета. Математика. Механика. Астрономия. 2022. Т. 9. №. 4. С. 625-635
- [20] Литвин Д. Б., Таволжанская О. Н. Элементы теории вероятностей : учебное пособие // Ставрополь : СтГАУ, 2015. 91 с.
- [21] Бессонов А. С. Основы имитационного моделирования. Моделирование в среде Anylogic : методические указания // Москва : РТУ МИРЭА, 2024. 24 с.
- [22] Боев В. Д. Компьютерное моделирование в среде AnyLogic: учебное пособие для среднего профессионального образования // Москва: Издательство Юрайт. 2023. 298 с.
- [23] Боев В. Д. Моделирование в среде AnyLogic: учебное пособие для вузов // Москва: Издательство Юрайт. 2023. 298 с.
- [24] Пономарчук Ю. В., Кузнецов И. В. Программирование на языке Java : учебное пособие // Хабаровск : ДВГУПС, 2021. 103 с.
- [25] Java [Электронный ресурс] URL: <https://ru.wikipedia.org/?curid=2506&oldid=137993614>
- [26] Листопад Н. И., Карук И. А., Хайдер А. А. Алгоритмы поиска кратчайшего пути и их модификация // Цифровая трансформация. 2016. № 1. С. 48-63.
- [27] Верзун, Н. А., Колбанев М. О., Романова А. А. Показатели эффективности процесса информационного взаимодействия в сети интернета вещей // Известия СПбГЭТУ ЛЭТИ. 2022. № 3. С. 5-14. DOI 10.32603/2071-8985-2022-15-3-5-14.
- [28] Астахова Т. Н., Верзун Н. А., Колбанев М. О., Полянская Н. А., Шамин А. А. Вероятностно-энергетические характеристики взаимодействия умных вещей // Вестник НГИЭИ. 2019. № 4(95). С. 66-77.
- [29] Broyd  F., Clavelier E., Jelinek L., Capek M., Warnick K. Implementing Two Generalizations of the Friis Transmission Formula // Excem Research Papers in Electronics and Electromagnetics. 2024. №. 8.
- [30] Yusri R., Edriati S., Yuhendri R. Pelatihan Microsoft Office Excel Sebagai Upaya Peningkatan Kemampuan Mahasiswa Dalam Mengolah Data //Rangkiang: Jurnal Pengabdian Pada Masyarakat. 2020. Т. 2. №. 1. pp. 32-37....

Simulation Model of Choosing the Optimal Route

T. N. Astakhova, A. A. Berdnikova, D. A. Maslova, M. V. Denisova

Abstract – Sensor fields are considered. This article presents the development of a simulation model aimed at solving the problems of finding the optimal path in sensor fields. The relevance of the study is due to the growing requirements for routing efficiency in modern sensor networks, which are widely used in various fields, such as smart cities, environmental monitoring and logistics.

The article presents the basic principles of constructing a simulation model, including pathfinding algorithms and methods for evaluating the effectiveness of routes. As a result of the work, strategies were developed that take into account dynamic changes in the sensor field.

The experimental results demonstrate the high accuracy and efficiency of the proposed model in various scenarios. The study confirms the possibility of using simulation modeling to optimize routes in complex and changing conditions, which opens up new horizons for improving the efficiency of sensor systems.

Keywords – A* algorithm, Dijkstra's algorithm, routing algorithms, simulation model, optimal path, sensory field.

REFERENCES

- [1] Verzun N. A., Vorobyov A. I., Kolbanov M. O., Tsekhanovsky V. V. Study of the Internet of Things technologies within the framework of the discipline "Infocommunication systems and networks" // Modern education: content, technology, quality. 2020. Vol. 1. pp. 71-74. (in Russian).
- [2] Utegenov N. B. Internet of things (IoT) and information systems // Universum: technical sciences. 2023. Vol. 7, issue. 112. (in Russian).
- [3] Laghari A. A., Wu K., Laghari R. A., Ali M., Khan, A. A. A review and state of art of Internet of Things (IoT) // Archives of Computational Methods in Engineering. 2021. pp. 1-19.
- [4] Vereshchagina E. A., Kapetskiy I. O., Yarmonov A. S. Security issues of the Internet of Things: textbook // Moscow: Mir nauki. 2021. Online publication. URL: <https://izd-mn.com/PDF/20MNNPU21.pdf>. (in Russian).
- [5] Verzun N. A., Vorobyov A. I., Kolbanov M. O. Technologies of the Internet of Things // St. Petersburg: St. Petersburg State University of Economics. 2020. 91 p. (in Russian).
- [6] Romanova A. A. Simulation model of information interaction in the Internet of Things // Bulletin of ETU LETI. 2022. No. 8. pp. 69-76. DOI 10.32603/2071-8985-2022-15-8-69-76. (in Russian).
- [7] Efremkov Ya. A. Using Anylogic software to create simulation models of technical systems // Science-intensive technologies and innovations (XXV scientific readings): Collection of reports of the International scientific and practical conference, Belgorod, November 23, 2023. – Belgorod: Belgorod State Technological University named after V.G. Shukhov, 2023. pp. 809-813. (in Russian).
- [8] Akopov A. S. Simulation modeling: textbook and workshop for universities // Moscow: Yurait Publishing House. 2023. 389 p. (in Russian).
- [9] Eremochkin S. Yu., Dorokhov D. V., Zhukov A. A. Development of a simulation model of a single-phase asynchronous electric drive in the SimInTech dynamic modeling environment // Bulletin of NGIEI. 2024. No. 1 (152). pp. 59-71. DOI 10.24412/2227-9407-2024-1-59-71. (in Russian).
- [10] Astakhova T. N., Kirilova D. A., Kolbanov M. O., Maslov N. S. Development of a routing algorithm for a wireless sensor network // Telecommunications. 2023. No. 4. pp. 30-38. DOI 10.31044/1684-2588-2023-0-4-30-38. (in Russian).
- [11] Berdnikova, A. A., Verzun N. A., Kolbanov M. O. Network level protocols of infocommunication systems and networks // St. Petersburg: St. Petersburg State University of Economics, 2024. 70 p. ISBN 978-5-7310-6454-5. (in Russian).
- [12] Olifer V. G., Olifer N. A. Computer networks. Principles, technologies, protocols // St. Petersburg: Piter. 2016. 992 p. (in Russian).
- [13] Dijkstra's shortest path calculator. [Electronic resource]. URL: <https://www.sas.com.ru/wp/ru/kalkuljator-kratchajshego-puti-dejkstry/> (in Russian).
- [14] Manakova V. A., Kostin A. S. Analysis of methods for solving the traveling salesman problem using Dijkstra's and *A algorithms // Systems analysis and logistics. 2023. No. 3 (37). pp. 136-142. DOI 10.31799/2077-5687-2023-3-136-142. (in Russian).
- [15] Dijkstra's algorithm – finding the shortest path [Electronic resource]. URL: <https://www.matematicus.ru/diskretnaya-matematika/reshenie-algoritma-dejkstry> (Accessed: 09.09.2024) (in Russian).
- [16] Plotnikov O. A., Podvalny E. S. Solution to the problem of finding the optimal path between two points on a graph with irregular edge weight // Bulletin of the Voronezh State Technical University. 2012. Vol. 8, No. 6. pp. 22-26. (in Russian).
- [17] Vtyurina K. S., Osipova M. V., Onofreichuk V. A. Pythagoras' theorem // Methods, models and algorithms for managing modernization processes. 2021. pp. 157-159. (in Russian).
- [18] Polovnikov O. N., Fokina V. V. Using the Euclidean and Manhattan distances as a measure of proximity for solving a classification problem // Bulletin of the Altai State University. 2010. No. 1-1 (65). pp. 101-102. (in Russian).
- [19] Krivulin N. K., Bryushinin M. A. Solution of the problem of placing two objects in space with the Chebyshev metric // Bulletin of the St. Petersburg University. Mathematics. Mechanics. Astronomy. 2022. Vol. 9. No. 4. pp. 625-635 (in Russian).
- [20] Litvin D. B., Tavalzhanskaya O. N. Elements of probability theory: a tutorial // Stavropol: StGAU, 2015. 91 p. (in Russian).
- [21] Bessonov A. S. Fundamentals of simulation modeling. Modeling in the Anylogic environment: guidelines // Moscow: RTU MIREA, 2024. 24 p. (in Russian)
- [22] Boev V. D. Computer modeling in the AnyLogic environment: a tutorial for secondary vocational education // Moscow: Yurait Publishing House. 2023. 298 p. (in Russian).
- [23] Boev V. D. Modeling in the AnyLogic environment: a tutorial for universities // Moscow: Yurait Publishing House. 2023. 298 p. (in Russian).
- [24] Ponomarchuk Yu. V., Kuznetsov I. V. Programming in Java: a tutorial // Khabarovsk: DVGUPS, 2021. 103 p. (in Russian).
- [25] Java [Electronic resource] URL: <https://ru.wikipedia.org/?curid=2506&oldid=137993614>.
- [26] Listopad N. I., Karuk I. A., Khaider A. A. Algorithms for finding the shortest path and their modification // Digital transformation. 2016. No. 1. pp. 48-63. (in Russian).
- [27] Verzun, N. A., Kolbanov M. O., Romanova A. A. Indicators of the effectiveness of the information interaction process in the Internet of Things // Bulletin of ETU LETI. 2022. No. 3. pp. 5-14. DOI 10.32603/2071-8985-2022-15-3-5-14. (in Russian).
- [28] Astakhova T. N., Verzun N. A., Kolbanov M. O., Polyanskaya N. A., Shamin A. A. Probability-energy characteristics of the interaction of smart things // Bulletin of NGIEI. 2019. No. 4 (95). pp. 66-77. (in Russian).
- [29] Broyd F., Clavelier E., Jelinek L., Capek M., Warnick K. Implementing Two Generalizations of the Friis Transmission Formula // Excm Research Papers in Electronics and Electromagnetics. 2024. No. 8.
- [30] Yusri R., Edriati S., Yuhendri R. Pelatihan Microsoft Office Excel Sebagai Upaya Peningkatan Kemampuan Mahasiswa Dalam

Mengolah Data // Rangkang: Jurnal Pengabdian Pada Masyarakat.
2020. Vol. 2. No. 1. pp. 32-37/.