

## ПРОЕКТИРОВАНИЕ ОПТИМАЛЬНОЙ ТРАЕКТОРИИ ПРОВЕДЕНИЯ МОНИТОРИНГА ОБЪЕКТОВ

© 2024 М. Саллам, Т.А. Макаровских

*Южно-Уральский государственный университет*

*(454080 Челябинск, пр. Ленина, д. 76)*

*E-mail: mohamedslam2000@yahoo.com, Makarovskikh.T.A@susu.ru*

Поступила в редакцию: 25.04.2024

В статье рассматривается задача разработки системы для моделирования траекторий движения робота, проводящего мониторинг в замкнутой области с земли: осмотр помещений, съемка промышленных объектов, оценка состояния фруктовых деревьев и т.д. Во время мониторинга на пути робота могут возникнуть некоторые препятствия: временно возникшие и устранимые (люди, небольшая мебель, бытовая техника, зона чрезвычайной ситуации) или постоянные (стены, постоянно установленное оборудование, зафиксированная мебель). Управление роботом осуществляется с помощью разработанной авторами программы, в которой хранится база данных маршрутов, успешно преодоленных роботом ранее, для наиболее точного определения траектории движения робота с учетом препятствий, встречающихся вдоль построенного пути. В статье рассматривается алгоритм построения графовой модели области проведения мониторинга для последующего поиска кратчайшего маршрута робота, заключающийся в дискретизации исследуемой области, выявлении возможных путей перемещения робота, анализе имеющихся препятствий и задании расстояний между объектами исследования. Показано, что после построения такого графа возможно применить один из алгоритмов поиска гамильтонова пути. Он соединяет вершины графа, соответствующие точкам проведения мониторинга. Результатом применения алгоритма является кратчайший маршрут робота, либо сообщение о невозможности проведения мониторинга (частичной или полной неразрешимости задачи), если ряд возникших препятствий не позволяет проложить маршрут, проходящий через некоторые (либо через все) вершины.

*Ключевые слова: маршрутизация, плоский граф, алгоритм, оптимизация, ограничения, задача о дроне и грузовике, точное земледелие, мониторинг.*

### ОБРАЗЕЦ ЦИТИРОВАНИЯ

Саллам М., Макаровских Т.А. Проектирование оптимальной траектории проведения мониторинга объектов // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2024. Т. 13, № 3. С. 32–46. DOI: 10.14529/cmse240302.

### Введение

Мониторинг и анализ состояния различных наземных объектов является актуальной задачей во многих сферах деятельности — от промышленности до сельского хозяйства. Использование человеческих ресурсов в этом случае часто затруднено или невозможно, например, при обследовании нескольких гектаров посадок плодовых деревьев или сельскохозяйственных культур, при мониторинге большой территории в помещении или с вредными для человека условиями окружающей среды, при охране жилых помещений, обследовании инфекционных отделений больниц и т.д.

Во всех рассмотренных выше случаях в обследуемом пространстве имеется ряд препятствий, которые можно разделить на постоянные (деревья, сооружения, неподвижная мебель) и временные (люди, животные, грузы). Устройство (робот), осуществляющее мониторинг, должно учитывать эти препятствия при расчете оптимального способа проведения мониторинга и планировать маршрут своего передвижения с учетом этих препятствий. Использование беспилотных летательных аппаратов (БПЛА) могло бы позволить не учиты-

вать препятствия, однако при обследовании закрытых помещений невозможно проводить мониторинг с воздуха, в ряде случаев и в открытых пространствах (например, при обследовании плодовых деревьев) целесообразно проводить съемку с земли, а не с воздуха.

Известные подходы в целом не учитывают каких-либо препятствий. Классическая постановка задачи учитывает количество объектов, которые необходимо посетить. В [1] используется подход к решению проблемы покрытия как к известной  $\mathcal{NP}$ -трудной задаче коммивояжера (Travelling Salesman Problem, TSP) путем включения алгоритмов выбора дерева для повышения эффективности и точности. Этот подход представляет исследуемую область в виде графа, где каждая вершина представляет точку покрытия, а ребра — возможные переходы между ними. В этом случае граф является взвешенным, а веса ребер содержат расстояния между посещаемыми вершинами. С целью снижения сложности вычислений, авторы используют различные алгоритмы выбора дерева, чтобы найти наилучшую возможную отправную точку для алгоритма решения задачи коммивояжера. Результаты моделирования показывают, что по сравнению с классическими алгоритмами решения TSP предлагаемый метод значительно повышает качество покрытия при одновременном снижении вычислительных затрат. Этот подход довольно хорош, но он используется для решения задачи для беспилотника, поэтому наличие препятствий там не учитывается.

Некоторые идеи о том, как найти путь, проходящий через все объекты интереса, подробно изучены в [2] и [3], где авторы рассматривают так называемый мегаполис (непустое конечное множество) для выполнения некоторой работы. Эта работа, с точки зрения данного исследования, может и быть мониторингом. Выбор порядка выполнения задач зависит от условий предшествования, которые локализованы для некоторого набора задач. Функции затрат, используемые при формировании аддитивного критерия, могут включать в себя зависимость от перечня задач. Для построения решения предлагается двухэтапная процедура, основанная на методах динамического программирования. Тем не менее, эти подходы также не учитывают ни постоянные, ни движущиеся препятствия. Поэтому целесообразно разработать подход, учитывающий возможные препятствия.

Еще один подход для решения той же задачи рассмотрен в [4], но в этой статье также не учитываются препятствия. Рассматриваемая в этой статье задача имеет общие черты с так называемой задачей о дроне и грузовике [5], в которой также рассматриваются устройство для проведения мониторинга (беспилотник) и устройство для обслуживания беспилотника (грузовик). Отличия в рассматриваемой в нашей статье задаче заключаются в том, что робот должен уметь преодолевать препятствие, а в некоторых случаях грузовик не требуется.

В данной статье рассматривается алгоритм формализации задачи для построения кратчайшего маршрута для робота, осуществляющего мониторинг с земли. Управление роботом осуществляется с помощью разработанной авторами программы, в которой хранится база данных маршрутов, успешно преодоленных роботом, для наиболее точного определения траектории движения робота с учетом препятствий, встречающихся на его пути. Статья состоит из введения, трех разделов и заключения.

В первом разделе приводится постановка задачи, описание ограничений на построение маршрута. Во втором разделе обсуждается общая схема алгоритма проведения мониторинга, графовая постановка задачи, описание структуры базы данных препятствий, определяются условия разрешимости задачи для имеющихся условий. Третий раздел посвящен описанию разработанного авторами программного обеспечения для проведения мониторинга. В заключении приводятся выводы и определяются направления дальнейших исследований.

## 1. Постановка задачи и описание ограничений

Допустим, необходимо осмотреть группу объектов  $M_i$ ,  $i = 1, \dots, K$ , расположенных в некоторой ограниченной области  $F$ . Область  $F$  представляет собой многоугольник, как правило, невыпуклый. Координаты  $N$  углов многоугольника  $(x_j, y_j)$ ,  $j = 1, \dots, N$  заданы в виде вектора, координаты указаны по часовой стрелке, начиная с наименьшего значения первой координаты. Исследуемые объекты  $M_i$  полностью расположены в области  $F$ . Расстояния  $L(M_i, M_j)$ ,  $i, j = 1, \dots, K$ ,  $i \neq j$  между всеми парами объектов известны. Задача состоит в том, чтобы проводить мониторинг объектов  $M_i$ , используя минимум ресурсов устройства робота или беспилотника. В данном случае под ресурсами понимается: расход заряда батареи (как правило, во время мониторинга требуется подзарядка), размер карты памяти (как правило, объема памяти для мониторинга всех объектов недостаточно и требуется замена карты). Подзарядку аккумулятора или замену карты памяти можно выполнить как на месте, так и после возврата устройства мониторинга на стоянку. В первом случае потребуется так называемый «грузовик», для которого строится маршрут доставки аккумулятора или карты памяти. Во втором случае необходимо учитывать время возвращения на стоянку до того, как будет израсходован заряд аккумулятора, и выстраивать кратчайший маршрут возвращения. Что касается карты памяти, можно осуществлять ее замену в процессе проведения мониторинга либо осуществить алгоритм передачи полученной информации и автоматического удаления переданных файлов.

Таким образом, устройство мониторинга, в дальнейшем будем называть его «робот», находится на парковке с координатами  $(P_x, P_y)$ . Задача состоит в том, чтобы проложить оптимальный маршрут робота от точки  $(P_x, P_y)$ , называемой парковкой, до выполнения работ (мониторинга) вблизи объектов  $M_i$ ,  $i = 1, \dots, K$ . В общем случае объекты  $M_i$  определяются векторами координат ограничивающих их полигонов. Целью мониторинга является получение изображений с помощью робота для изучения состояния объектов  $M_i$  и проведения дальнейшего анализа.

В идеале мониторинг осуществляется без ограничений, но, очевидно, такой случай является исключительным. Чтобы решить поставленную задачу, можно рассмотреть один из частных случаев решения задачи коммивояжера (TSP) [6] и использовать один из известных алгоритмов (после проведения вычислительных экспериментов, можно определить наиболее подходящий алгоритм для рассматриваемой задачи). Однако при перемещении между объектами могут возникать различного рода препятствия, которые робот по определенным причинам не может преодолеть. Наличие этих препятствий может повлечь пересчет и пересмотр построенных ранее маршрутов и привести к частичной или полной неразрешимости задачи.

Препятствия  $S_i$ ,  $i = 1, \dots, s$  подразделим на временные и постоянные. К временным препятствиям относятся люди и животные (в т.ч. мертвые животные), а также техника (автомобили, тракторы, погрузчики и т.д.), оставленная на пути движения робота, временные хранилища (например, стог сена, контейнер с посылками или товарами). Это временные препятствия, информация о размерах которых хранится и используется для анализа при корректировке траектории в будущем, местоположение таких препятствий в базе не сохраняется. Остальные препятствия являются постоянными (стационарными или малоподвижными). К ним относятся новые стены и перегородки, конструкции, закрепленная мебель. Наличие и размеры этих препятствий также заносится в базу данных для анализа маршрута, поскольку их местоположение является постоянным или долгосрочным.

Чтобы определить траекторию движения робота, необходимо решить следующие задачи.

1. Создать базу данных маршрутов: идеальных (без каких-либо препятствий), учитывающих влияние внешних факторов (температура, сила ветра и т.д.), маршрутов с непреодолимыми препятствиями (доступны только маршруты со стационарными и подвижными препятствиями, добавленные ранее в базу данных). Под препятствием мы будем понимать любой объект, который стоит на пути движения робота и препятствует дальнейшему следованию по заданному ранее маршруту.
2. Определить разрешимость задачи маршрутизации (возможно ли построить маршрут для наблюдения за интересующими объектами при наличии существующих ограничений и препятствий). Разрешимость может быть полной, когда возможно провести мониторинг во всех указанных точках  $M_i$ ,  $i = 1, \dots, K$ ; частичной, когда существуют точки  $M_s$ , не достижимые при имеющемся наборе препятствий. Если все  $M_i$ ,  $i = 1, \dots, K$  не достижимы, такая задача называется неразрешимой.

В общем случае, существует  $K$  объектов для мониторинга, координаты каждого из них известны. Делается фото или видеозапись этого объекта, которая затем отправляется на сервер для дальнейшего анализа и принятия решения. Робот делает снимки по указанным координатам и переходит к следующему объекту. Когда съемка проведена по всем указанным координатам или была выявлена невозможность дальнейшего наблюдения, робот возвращается на парковку.

Если рабочий ресурс робота исчерпан, он либо возвращается на парковку (необходимо, чтобы было достаточно ресурса для возврата), либо отправляет оператору сообщение о необходимости устранения неполадки. В последнем случае необходимо решить проблему построения маршрута для служебного «грузовика», доставляющего пополненные ресурсы.

## 2. Алгоритм маршрутизации робота

### 2.1. Схема алгоритма

Схема алгоритма движения робота во время мониторинга приведена на рис. 1.

На рисунке предполагается, что маршрут  $P_i$  уже учитывает все внешние воздействия, и в нем учитываются все возможные подзарядки робота и замены карты памяти, если ресурса недостаточно для завершения полного цикла мониторинга. Перемещение вдоль маршрута  $P_i$  означает перемещение робота на одну позицию по маршруту. Считается, что все пространство, исследуемое роботом, делится на элементарные сегменты (плитки) одинаковой длины и ширины.

Возможностью продолжить движение считается возможность добраться до центральной точки следующего элементарного отрезка. Если на пути робота к этой точке обнаружено препятствие, дальнейшее продолжение движения по ранее выбранному пути считается невозможным и выполняется поиск обходного маршрута. Если на одном из этапов мониторинга не удастся найти обходной маршрут, робот отправляет оператору сообщение о невозможности дальнейшего проведения мониторинга и отключается (возвращается на стоянку для подзарядки). Маршрут по известной ранее области уже определен и может не рассчитываться заново, тем не менее, после обнаружения препятствия и размещения его на карте проведения мониторинга необходимо переопределить все длины путей, проходящих через область, перекрытую препятствием, а также обновить информацию о достижимости еще не обследованных объектов мониторинга. В большинстве случаев появление каждого

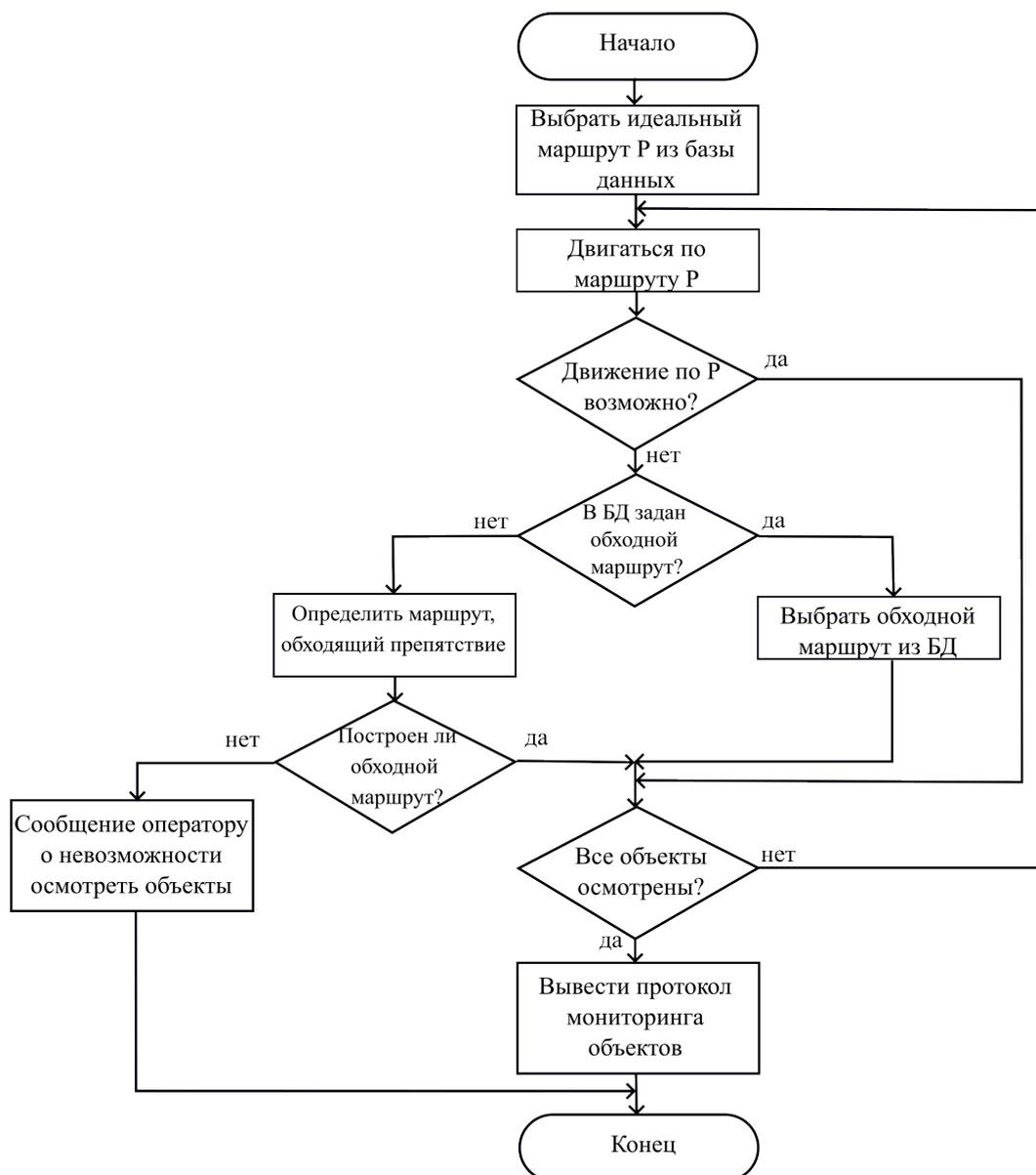


Рис. 1. Схема алгоритма перемещения робота в зоне мониторинга

препятствия требует заново решить задачу TSP для обновленной карты. Хранение ранее построенных маршрутов в базе данных позволит в ряде случаев избежать повторного решения TSP и выбрать уже имеющееся решение задачи. Например, если на пути робота закрыта дверь либо шлагбаум, это постоянное препятствие, решение для которого хранится в базе данных. По известным габаритам временных препятствий (например, припаркованного автомобиля или погрузчика) можно определить время для объезда и принять решение о целесообразности повторного решения задачи маршрутизации.

Препятствия на пути робота могут отслеживаться несколькими способами:

- ультразвук, который считается эффективным и быстрым, он позволяет рассчитать расстояния с точностью до миллиметра, вплоть до расстояния в три метра (которое может быть увеличено в зависимости от типа установленного ультразвукового датчика), чтобы определить расстояние между препятствием и роботом, а также габариты препятствия;

- программируемые интеллектуальные камеры для отслеживания новых препятствий на пути робота.

Таким образом, на основе построенной алгоритмической схемы могут быть поставлены следующие задачи.

1. Построение маршрута  $P_i$  с учетом внешних воздействий.
2. Определение возможности продолжения движения с помощью алгоритмов компьютерного зрения.
3. Поиск обходного маршрута с учетом внешних воздействий и обнаруженных препятствий.
4. Обновление информации в базе данных о препятствиях и обходных путях.

## 2.2. Графовая модель исследуемой области и алгоритм маршрутизации

Всю область, используемую для перемещения робота, разделим на небольшие квадратные плитки  $R_j$ ,  $j = 1, \dots, M$  с помощью алгоритма дискретизации [7]. Длина стороны каждой плитки равна расстоянию, проходимому роботом, между моментами передачи информации оператору. Обычно эта величина равна 30–50 см. Обозначим эту величину через  $l$ . Робот может перемещаться между углами плитки по горизонтали либо вертикали (тогда длина пути между парой моментов обмена информацией равна  $l$ ), либо по диагонали (в этом случае робот преодолеет расстояние  $\sqrt{2}l$  до следующего момента передачи информации).

Представим исследуемую область как граф  $G = (V, E)$ , множество вершин  $V$  которого являются точками, в которых робот обменивается информацией с прикладной программой, а ребра  $E$  — связями между ближайшими вершинами (как отмечалось выше, переходы осуществляются по горизонтали, вертикали либо диагонали). Поскольку при решении практической задачи точки дискретизации располагаются на расстоянии порядка 50 см, их количество становится довольно большим, поэтому  $|V|$  может содержать тысячи вершин. Следовательно, неразумно использовать все эти вершины для определения оптимального маршрута. Кроме того, граф является разреженным — степень каждой его вершины равна 2–8, в то время как количество вершин значительно выше. Веса ребер (длины переходов) будут равны либо  $l$ , либо  $\sqrt{2}l$ . Для хранения информации о графе будем использовать список смежных вершин.

Подразделим все вершины графа  $V = V_p \cup V_m \cup V_d \cup V_t$  на следующие подмножества:

- вершины «парковки» ( $p_i \in V_p$ ) — вершины, с которых робот может подзарядить аккумулятор либо получить сервисное обслуживание, стартовые вершины;
- вершины мониторинга ( $m_i \in V_m$ ) — вершины, в которых проводится мониторинг;
- вершины маршрута объезда ( $d_i \in V_d$ ) вершины, принадлежащие маршруту объезда препятствия;
- транзитные вершины  $v \in V_t$  — все остальные вершины графа, которые могут быть включены в маршрут.

Рассмотрим пример, приведенный на рис. 2. На рисунке приведена невыпуклая область, ограниченная жирной линией. Вершины, принадлежащие границе, в маршруте не используются;  $P$  — вершины для парковки  $V_p$ ;  $M_1 \dots, M_6$  — объекты мониторинга;  $S_1, S_2$  — некоторые препятствия. Вершины, отмеченные красным цветом, являются точками (вершинами) мониторинга  $V_m$ , вершины, обозначенные черным цветом, являются вершинами объезда  $V_d$ . Транзитные вершины  $V_t$  на этом рисунке не указаны, если их количество велико

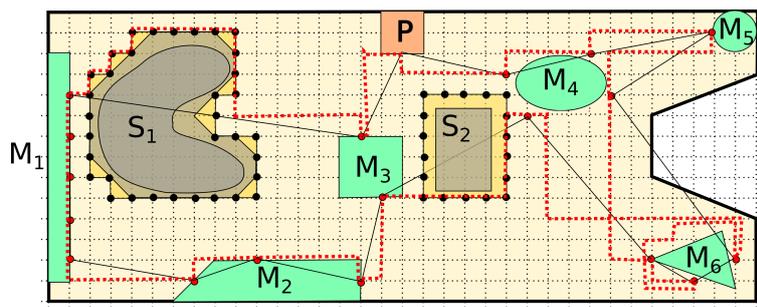


Рис. 2. Пример карты для проведения мониторинга

(это места пересечения прямых линий, используемые для дискретизации). Те же обозначения используются для всех последующих примеров.

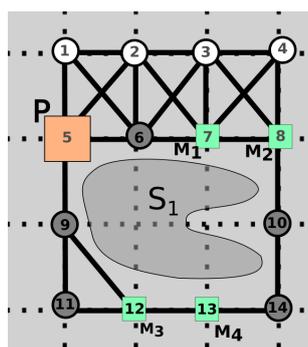
Информация о графе хранится в текстовом файле, имеющем следующий формат:

- Строка 1: количество вершин и ребер графа через пробел;
- Строка  $i$  ( $i = 2, \dots, |V|$ ): тип текущей вершины ( $P$  — парковка,  $M$  — мониторинг,  $D$  — объезд,  $T$  — транзитная), и далее список вершин, смежных  $v_i$ , каждое значение через пробел. Сначала указываются вершины, инцидентные ребрам с весом  $l$ , далее располагается служебный символ «|», после него следуют вершины, инцидентные ребрам с весом  $\sqrt{2}l$ . Если диагональных ребер нет, после символа «|» ничего не записывается.

Порядок следования вершин, инцидентных ребрам одного веса, не принципиален.

Если вершина одновременно является и вершиной объезда, и вершиной мониторинга (либо парковки), то при задании информации в файле она маркируется как вершина для мониторинга (либо парковки).

Например, для графа, приведенного на рис. 3а, файл, его задающий, имеет следующий вид (рис. 3б).



```

14 25
T 2 5 | 6
T 1 3 6 | 5 7
T 2 7 4 | 6 8
T 3 8 | 7
P 1 6 9 | 2
D 5 2 7 | 1 3
M 6 3 8 | 4 2
M 7 4 10 | 3
D 5 11 | 12
D 8 13 |
T 9 12 |
M 11 4 | 9
M 3 14 |
D 4 10 |
    
```

(а) Графическое представление (б) Представление в памяти компьютера

Рис. 3. Пример задания графовой модели исследуемой области

Пользуясь информацией для этого графа можно построить производный взвешенный граф  $G^* = (V_P \cup V_M, E^*)$ , вес каждого ребра которого равен длине кратчайшего пути между соответствующими парами вершин в графе  $G$ . Рассмотрим вершины  $v, w \in V_P \cup V_M$  в графе  $G$ . Если путь из вершины  $v$  в вершину  $w$  проходит через некоторую вершину  $v_k \in V_P \cup V_M$ , то вершины  $v$  и  $w$  не будут смежны в графе  $G^*$ . Очевидно, если для всех пар вершин  $v, w \in V_P \cup V_M$  пути, их соединяющие, не проходят ни через одну из вершин  $v_k \in V_P \cup V_M$ , то граф  $G^*$  является полным.

Так, для графа, представленного на рис. 3, граф  $G^*$  имеет вид, представленный на рис. 4.

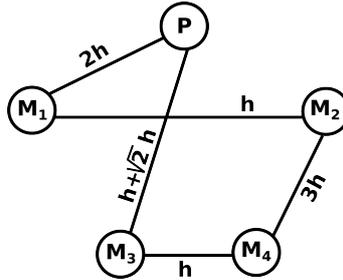


Рис. 4. Граф  $G^*$  для рассмотренного примера

Заметим, что в графе отсутствуют ребра  $\{P, M_2\}$  (поскольку путь  $P - 6 - M_1 - M_2$  содержит вершину  $M_1$ ),  $\{P, M_4\}$  (поскольку путь  $P - 9 - M_3 - M_4$  содержит вершину  $M_3$ ),  $\{M_1, M_4\}$  (поскольку путь  $M_1 - 6 - P - 9 - M_4$  содержит вершину  $P$ ),  $\{M_1, M_3\}$  (поскольку путь  $M_1 - 6 - P - 9 - M_3$  содержит вершину  $P$ ),  $\{M_2, M_3\}$  (поскольку путь  $M_2 - 10 - M_4 - M_3$  содержит вершину  $M_4$ ). Таким образом, имеем граф  $G^*$ , соответствующий метрической постановке задачи коммивояжера.

Приведем алгоритм поиска гамильтонова цикла  $T$  минимальной длины в графе  $G^*$ . Полученный в случае разрешимой задачи цикл  $T$  будет являться решением поставленной задачи о маршрутизации робота, проводящего мониторинг при условии отсутствия ограничений на время перемещения.

Задачу получения оптимального маршрута для проведения мониторинга можно разделить на следующие этапы.

### 2.2.1. Определение идеального маршрута

Введем определения разрешимости задачи, которыми будем пользоваться в дальнейших рассуждениях.

**Определение 1.** Задача маршрутизации является *разрешимой*, если граф  $G$ , задающий исследуемую область, является связным, либо все вершины из  $V_m$  принадлежат компонентам связности содержащим хотя бы одну вершину из  $V_p$ .

В примере, показанном на рис. 2, эта задача разрешима, и линией из красных точек показан маршрут мониторинга.

**Определение 2.** Задача является *частично разрешимой*, если существует  $b$  вершин из  $V_m$ ,  $b \neq |V_m|$ , принадлежащих компонентам связности графа  $G$ , не содержащим вершину из  $V_p$ .

На рис. 5а приведена частично разрешимая задача.

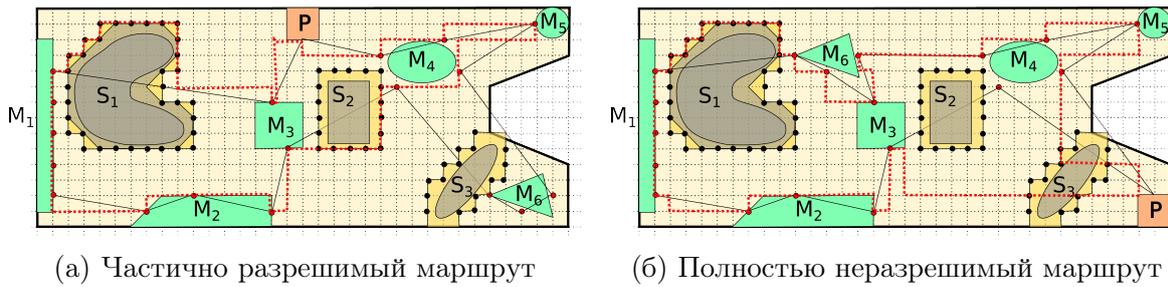


Рис. 5. Пример карты для проведения мониторинга

Объект  $M_6$  не может быть включен в маршрут проведения мониторинга, поскольку все пути к нему перекрыты препятствием  $S_3$ . Если  $S_3$  является стационарным препятствием, то  $M_6$  никогда не будет проверен. Если это временное препятствие, то необходимо принять решения для его устранения. Здесь маршрут, отмеченный линией из красных точек, проходит через все контролируемые объекты, кроме  $M_6$ .

**Определение 3.** Задача является *не разрешимой*, если ни одна вершина из  $V_m$  не принадлежит компоненте связности графа  $G$ , содержащей  $V_p$ .

На рис. 5б приведена неразрешимая задача.

В данном случае верно то же рассуждение о препятствии  $S_3$ , что и для предыдущего случая.

Для построения маршрута  $P_{ideal}$  наименьшей длины между вершинами мониторинга  $V_m$  необходимо выполнить следующие действия.

**Алгоритм МАРШРУТИЗАЦИЯ ( $G$ )**

**Исходные данные:** граф  $G$ ,  $l$  – размер плитки.

**Выходные данные:** – замкнутая гамильтонова цепь кратчайшей длины с началом в вершине  $v_0$ .

**Шаг 1. Определить разрешимость задачи маршрутизации.** Для этого, используя алгоритм просмотра вершин (поиск в ширину или глубину) определить количество компонент связности заданного графа  $G = G_1 \cup G_2 \cup \dots, \cup G_k$ . Если задача не разрешима, завершить работу алгоритма.

**Шаг 2. Построение графа для определения маршрута.** Для каждой компоненты связности  $G_i$  построить взвешенный полный граф  $G_{T_i} = G^*(|V_p| \cup |V_m|, E^*)$ . Каждое ребро  $e \in E^*$  этого графа будет иметь вес, равный  $\phi(e) = \min d(v_1(e), v_2(e))$ , определяемый как кратчайший путь между соответствующими вершинами в графе  $G_i$ , длина каждого ребра которого равна  $l$  либо  $\sqrt{2}l$ . Для решения этой задачи используется алгоритм Дейкстры. Если найденный путь  $P(v_1(e), v_2(e))$  содержит в качестве транзитной вершину  $v \in V_p \cup V_m$ , то вершины  $v_1(e)$  и  $v_2(e)$  в графе  $G^*$  не будут связаны отношением непосредственной достижимости.

**Шаг 3. Поиск гамильтонова цикла минимальной длины.** В каждом графе  $G_{T_i} = K_{|V_p| \cup |V_m|}$  определить гамильтонов цикл минимальной длины. Если длина цикла больше критической  $L_{critical}$  (например, для прохождения пути такой длины требуется подзарядка), необходимо учесть посещение произвольной вершины  $v \in V_p$ . Такая задача известна как метрическая задача VRP (Vehicle Routing Problem) [8]. Он и будет соответствовать маршруту робота для известной карты препятствий. Поскольку данная задача является  $\mathcal{NP}$ -трудной в сильном смысле, для ее решения можно использовать некоторую



2.2.4. Вычислительная сложность определения маршрута мониторинга

Говоря о вычислительной сложности представленного алгоритма, мы видим, что она определяется вычислительной сложностью алгоритма, используемого для решения задачи VRP на шаге 1 алгоритма. Шаг 2 может быть выполнен за полиномиальное время  $O(|V|^2)$ , потому что в худшем случае требуется просмотреть все вершины графа  $G$ . Даже в этом случае время, необходимое для решения этой задачи, может быть достаточно большим, поскольку  $|V|$  может содержать более  $10^3$  элементов. Для ускорения этого процесса могут быть использованы технологии параллельных вычислений.

Если препятствие представляет собой выпуклую область, то оно делится на две части идеальной траекторией, поэтому определение кратчайшего обходного пути заключается в проверке только двух путей. Если препятствие не является выпуклой областью, то количество путей, подлежащих проверке, является конечным множеством, поскольку  $|V_{d_i}|$  для  $S_i$  является конечным множеством.

### 3. Программное обеспечение для проведения мониторинга

Основная идея работы системы мониторинга по отслеживанию и досмотру объектов с помощью роботов и беспилотных летательных аппаратов показана на рис. 7.



Рис. 7. Технологический цикл работы системы мониторинга

Система функционирует циклически. Здесь на этапе «Смысл размещения» определяется пространство для проведения мониторинга, получается информация об известных системе с предыдущих этапов препятствиях: временных и постоянных. На этапе «Восприятие» робот изучает текущее состояние окружающей среды. Возможно, здесь вносятся изменения в информацию о препятствиях, если они были перемещены либо удалены из обследуемой области. На этапе «План» выполняется построение либо корректировка маршрута от текущей точки до оставшихся непройденными точек проведения мониторинга. Этап «Действие» посвящен переходу устройства мониторинга в новую точку исследуемой области, сохране-

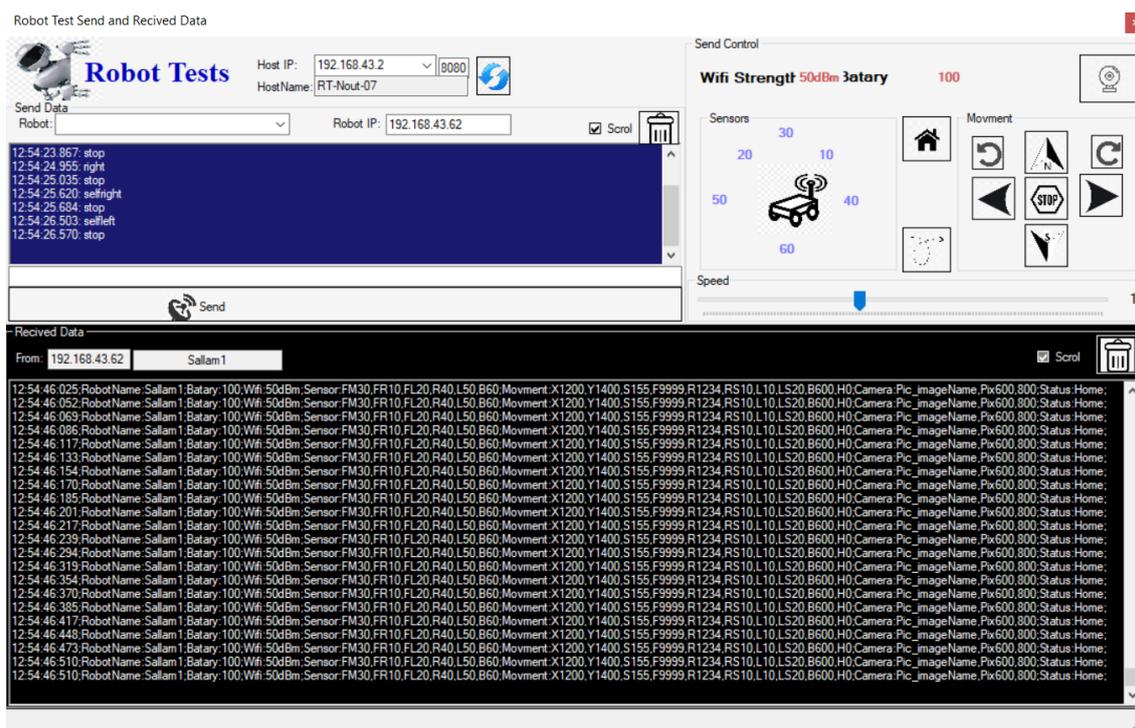


Рис. 8. Окно интерфейса программы для управления роботом при мониторинге объектов

нию полученной текущей информации о препятствиях. Эта информация становится новыми «данными для обучения» при построении дальнейших маршрутов и учитывается при формировании фактического маршрута, по которому движется устройство мониторинга.

Поэтому для управления устройством мониторинга было разработано настольное приложение для тестирования робота и получения данных с него (рис. 8), которое предназначено для дистанционного управления роботом (посредством панели управления) и передачи сообщений в консольном режиме, связи с роботом по Wi-Fi через IP-адрес устройства. Когда робот включен, данные передаются через ESP-карту (плата разработчика, на которой установлен микроконтроллер ESP32 с беспроводными интерфейсами WiFi/Bluetooth для передачи данных). Среди передаваемой информации:

- данные о расстоянии между роботом и ближайшим препятствием;
- данные о количестве шагов, необходимых для достижения этого препятствия по прямой;
- название файла с фотографией препятствия;
- координаты препятствия на карте исследуемой области;
- разрешение изображения;
- информация о состоянии робота (движется, стоит, припаркован).

Полученные данные записываются в CSV-файл. Управление роботом осуществляется на панели управления Send по аналогии с использованием пульта управления. При нажатии кнопки мыши на одной из кнопок управления робот перемещается по траектории, заданной оператором. Информация о командах, передаваемых роботу, передается на консольную панель. Роботом также можно управлять, отправляя команды на консольную панель.

## Заключение

В статье рассмотрена задача планирования траектории мониторинга с препятствиями на исследуемой территории. Приведен алгоритм, позволяющий улучшить заданную идеальную траекторию (то есть решение задачи TSP без ограничений) в соответствии с существующими стационарными и подвижными препятствиями на пути робота. Информация о стационарных препятствиях сохраняется в базе данных и может быть получена по запросу. Предполагается, что подвижные препятствия могут иметь не известное расположение на карте исследуемой области и идентифицированы только после старта проведения мониторинга. Этот подход можно использовать для наблюдения за различными объектами с земли, в частности, в помещении.

К задачам будущих исследований можно отнести следующие. Прежде всего, необходимо определить наиболее подходящий для данной задачи алгоритм определения идеального пути и пересчета оптимальной траектории при возникновении подвижных препятствий. Они могут возникнуть после определения начального маршрута, и в этом случае предлагается использовать модуль искусственного интеллекта для распознавания препятствия, определения его контуров, занесения информации о препятствии в базу данных и нанесении его на карту проведения мониторинга. Информация о наличии временных препятствий должна обновляться в соответствии с заданным расписанием. Если препятствие устранено, то оно помещается в архивную таблицу базы данных и извлекается из нее, как только оно будет распознано на карте при очередном мониторинге.

## Литература

1. Sundarrajan M., Jothi A., Prabakar D., Kadry S. The Smart Coverage Path Planner for Autonomous Drones Using TSP and Tree Selection // Mining Intelligence and Knowledge Exploration. MIKE 2023. Vol. 13924 / ed. by S. Kadry, R. Prasath. Springer, 2023. P. 161–172. Lecture Notes in Computer Science. DOI: 10.1007/978-3-031-44084-7\_16.
2. Ченцов А.Г. Задача маршрутизации на узкие места с системой первоочередных заданий // Известия Института математики и информатики Удмуртского государственного университета. 2023. Т. 61. С. 156–186. DOI: 10.35634/2226-3594-2023-61-09.
3. Ченцов А.Г., Ченцов П.А. Экстремальная двухэтапная задача маршрутизации и процедуры на основе динамического программирования // Труды Института математики и механики УрО РАН. 2022. Т. 28, № 2. С. 215–248. DOI: 10.21538/0134-4889-2022-28-2-215-248.
4. Wu J., Li M., Gao C., *et al.* Research on Deployment Scheme and Routing Optimization Algorithm of Distribution Cable Condition Monitoring Devices // Energies. 2023. Vol. 16. P. 852–879. DOI: 10.3390/en16196930.
5. Chung S.H., Sah B., Lee J. Optimization for drone and drone-truck combined operations: A review of the state of the art and future directions // Computers & Operations Research. 2020. Vol. 123. DOI: 10.1016/j.cor.2020.105004.
6. Петунин А.А., Ченцов А.Г., Ченцов П.А. Оптимальная маршрутизация в задачах последовательного обхода мегаполисов при наличии ограничений // Челябинский физико-математический журнал. 2022. Т. 7, № 2. С. 209–233. DOI: 10.47475/2500-0101-2022-17205.

7. Makarovskikh T., Panyukov A., Abotaleb M., *et al.* Optimal Route for Drone for Monitoring of Crop Yields // *Advances in Optimization and Applications. OPTIMA 2023*. Vol. 1913 / ed. by N. Olenev, Y. Evtushenko, M. Jaćimović, *et al.* Springer International Publishing, 2024. P. 228–240. *Communications in Computer and Information Science*. DOI: 10.1007/978-3-031-48751-4\_17.
8. Гимади Э.Х., Хачай М.Ю. Экстремальные задачи на множествах перестановок. Екатеринбург: Издательство УМЦ УПИ, 2016. 220 с.

Саллам Мохамед, аспирант, кафедра системного программирования, Южно-Уральский государственный университет (национальный исследовательский университет) (Челябинск, Российская Федерация)

Макаровских Татьяна Анатольевна, д.ф.-м.н., доцент, кафедра системного программирования, Южно-Уральский государственный университет (национальный исследовательский университет) (Челябинск, Российская Федерация)

---

DOI: 10.14529/cmse240302

## DESIGNING THE OPTIMAL TRAJECTORY FOR MONITORING OBJECTS

© 2024 M. Sallam, T.A. Makarovskikh

*South Ural State University (pr. Lenina 76, Chelyabinsk, 454080 Russia)*

*E-mail: mohamedslam2000@yahoo.com, Makarovskikh.T.A@susu.ru*

Received: 25.04.2024

The article considers the task of developing a system for modeling the trajectories of a robot monitoring in a closed area from the ground: inspection of premises, survey of industrial facilities, assessment of the condition of fruit trees, etc. During monitoring, some obstacles may arise in the robot's path: temporary and removable (people, small furniture, household appliances, emergency zone) or permanent (walls, permanently installed equipment, fixed furniture). The robot is controlled using a program developed by the authors, which stores a database of routes successfully overcome by the robot earlier, for the most accurate determination of the trajectory of the robot, taking into account obstacles encountered along the constructed path. The article considers an algorithm for constructing a graph model of the monitoring area for the subsequent search for the shortest route of the robot, which consists in discretization the area, identifying possible ways to move the robot, analyzing existing obstacles and setting distances between the objects of study. It is shown that after obtaining such a graph, it is possible to apply one of the algorithms for finding the Hamiltonian path in a graph. It connects the vertices of the graph corresponding to the monitoring points. The result of applying the algorithm is the shortest route of the robot, or a message about the impossibility of monitoring (partial or complete insolubility of the problem) if a number of obstacles do not allow you to plot a route passing through some (or all) vertices.

*Keywords: routing, plane graph, algorithm, optimization, constraints, drone and truck problem, precision farming, monitoring.*

### FOR CITATION

Sallam M., Makarovskikh T.A. Designing the Optimal Trajectory for Monitoring Objects. *Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering*. 2024. Vol. 13, no. 3. P. 32–46. (in Russian) DOI: 10.14529/cmse240302.

*This paper is distributed under the terms of the Creative Commons Attribution-Non Commercial 4.0 License which permits non-commercial use, reproduction and distribution of the work without further permission provided the original work is properly cited.*

---

## References

1. Sundarrajan M., Jothi A., Prabakar D., Kadry S. The Smart Coverage Path Planner for Autonomous Drones Using TSP and Tree Selection. Mining Intelligence and Knowledge Exploration. MIKE 2023. Vol. 13924 / ed. by K. S., P. R. Springer, 2023. P. 161–172. Lecture Notes in Computer Science. DOI: 10.1007/978-3-031-44084-7\_16.
2. Chentsov A.G. A bottleneck routing problem with a system of priority tasks. Izvestiya IMI UdGU. 2023. Vol. 61. P. 156–186. (in Russian) DOI: 10.35634/2226-3594-2023-61-09.
3. Chentsov A.G., Chentsov P.A. An extremal two-stage routing problem and procedures based on dynamic programming. Trudy Instituta Matematiki i Mekhaniki UrO RAN. 2022. Vol. 28, no. 2. P. 215–248. (in Russian) DOI: 10.21538/0134-4889-2022-28-2-215-248.
4. Wu J., Li M., Gao C., *et al.* Research on Deployment Scheme and Routing Optimization Algorithm of Distribution Cable Condition Monitoring Devices. Energies. 2023. Vol. 16. P. 852–879. DOI: 10.3390/en16196930.
5. Chung S.H., Sah B., Lee J. Optimization for drone and drone-truck combined operations: A review of the state of the art and future directions. Computers & Operations Research. 2020. Vol. 123. DOI: 10.1016/j.cor.2020.105004.
6. Petunin A.A., Chentsov A.G., Chentsov P.A. Optimal routing in problems of sequential traversal of megapolises in the presence of constraints. Chelyab. Phys.-Math. Journal. 2022. Vol. 7, no. 2. P. 209–233. (in Russian) DOI: 10.47475/2500-0101-2022-17205.
7. Makarovskikh T., Panyukov A., Abotaleb M., *et al.* Optimal Route for Drone for Monitoring of Crop Yields. Advances in Optimization and Applications. OPTIMA 2023. Vol. 1913 / ed. by N. Olenev, Y. Evtushenko, M. Jaćimović, *et al.* Springer International Publishing, 2024. P. 228–240. Communications in Computer and Information Science. DOI: 10.1007/978-3-031-48751-4\_17.
8. Gimadi E.K., Khachai M.Y. Extreme Tasks on Sets of Permutations. Ekaterinburg: Izdatel'stvo UMC UPI, 2016. 220 p. (in Russian).