УДК 519.87

ЗАДАЧА ПЛАНИРОВАНИЯ ПУТИ ПОКРЫТИЯ С ОГРАНИЧЕНИЯМИ НА ДЛИНУ ПУТИ И ВРЕМЯ

Кузнецов А.В. (Воронеж)

1. Постановка задачи

В статье мы обсудим ограниченную версию известной задачи планирования пути покрытия [1-3], известной также как задача коммивояжера (далее – ЗК). Пусть $D \subset \mathbb{R}^2$ – ограниченная область с границей в виде замкнутой ломаной ∂D. Далее мы будем называть такие области «многоугольниками» и обозначать множество таких областей как $\mathcal{P}$. Агент *ag* должен покрыть *D* попарно непересекающимися прямоугольниками размера $a \times b$ за минимально возможное время, выходя из точки $p_0 = (x_0, y_0) \in \partial D$. Агент *ag* имеет запас энергии *E* для того, чтоб двигаться по своему пути за время *T*, т. е. путь агента, движущегося с постоянной скоростью *v*, может иметь длину не более $l=v\ T>0$. Агент должен всегда возвращаться в точку p_0 по истечении времени *T*. Если *D* не может быть покрыта таким образом, то вместо p_0 агент может выбрать последовательно другие стартовые точки $p_1, p_2, \dots p_m \in \partial D$ для каждого выхода. Если *D* не может быть покрыто даже таким способом, агент может выбрать стартовую точку внутри *D*.

Такая постановка связана с тем, что агент-БПЛА может вылетать с мобильной платформы – зарядной станции, которая, в свою очередь, может перемещаться вдоль границ поля (но не заезжать вовнутрь, так как в сельскохозяйственных задачах это невозможно).

Агент должен задерживаться в специально указанных точках

$$POI = \{poi_1, poi_2, \dots, poi_n\} \in D.$$

Например, если агент – это сельскохозяйственный БПЛА, то в этих точках он должен осуществлять сброс ядохимикатов или удобрений. Решения ЗК в такой формулировке автору в доступных ему источниках найти не удалось.

Есть два варианта планирования пути. В первом варианте вся информация о местности известна заранее, оптимальный путь также рассчитывается заранее, а у агента есть список координат, по которым он должен двигаться. Во втором варианте агент вычисляет оптимальный путь непосредственно при прохождении через домен, и возможны изменения состояния домена. Далее мы будем исходить из первого варианта, который всегда следует учитывать, даже если мы используем второй, хотя бы для проверки качества эвристики. В случае реактивного планирования маршрута энергозатраты агента должны учитываться при расчете оптимального пути. Оптимальный маршрут можно рассчитать заранее, в этом случае энергозатраты на планирование маршрута равны нулю. В противном случае энергия агента уменьшается на величину E' при каждом акте планирования и максимальное время работы агента также уменьшается на величину T'. Более того, область *D* и множество *POI* могут меняться со временем, а заранее рассчитанный путь может оказаться неоптимальным.

Дополнительно, агент имеет предельную дальность связи *r* и не может отойти от точки p_0 на расстояние, большее чем *r*. Когда вышеупомянутая цель не может быть достигнута, агент может изменить положение начальной точки.

Рассмотрим следующие преобразования области:

1. Поворот $R_\alpha: \mathcal{P} \to \mathcal{P}$ области *D* на угол α, например, вокруг центра.
2. Дискретизация $G_{O,a,b}: \mathcal{P} \to RF$, т. е. построение сетки точек вида $O + (na, mb), (n, m) \in \mathbb{Z}^2$ с началом координат в *O*. Здесь $RF \subset \mathbb{R}^2$ – множество ко-

нечных подмножеств $\mathbb{R}^2$. Прямоугольники размером $a \times b$ с центрами в точках сетки должны покрывать всю область *D*.

3. Разбиение $S_{p,\beta}: 2^{\mathbb{R}^2} \to 2^{\mathbb{R}^2}$ множества (области *D* или сетки *G*) прямой, проходящей через точку *p* с направляющим вектором $(cos\alpha, sin\alpha)$
$$S_{p,\beta}(D) = \left\{(x,y) \in D: \det\begin{pmatrix} cos\alpha & sin\alpha \\ x-x_0 & y-y_0 \end{pmatrix} < 0\right\}.$$
4. Перспективное искажение $H: \mathbb{R}^2 \to \mathbb{R}^2$ множества (области *D* или сетки *G*). Это линейно-дробное преобразование определяется его матрицей гомографии. Данное преобразование соответствует наклонному положению камеры агента.
5. Покрытие *C*(*D*, *G*) области *D* прямоугольниками с центрами в точках сетки *G*.

Также пусть *L*(*G*) – решение ЗК для сетки *G*. Функцию длины для этого решения и веса для ЗК следует строить с учетом весов точек интереса. Чтобы решить проблему, мы должны постепенно разбить области на подобласти, где вышеупомянутая задача имеет решение. Алгоритм, вкратце, должен быть следующим:

0. Устанавливаем $Remain = \emptyset$.
1. Область *D* дискретизируется в сетку $G = G_{O,a,b}(R_\alpha D)$, где угол α таков, что $G_{O,a,b}(R_\alpha D) \to min$.
2. Решаем ЗК для сетки G и ищем путь *L*(*G*) с ограничением на длину length(*L*(*G*))≤*l*.
3. Если проблема неразрешима, мы можем выбрать **шаги 3.1** или **3.2**, иначе перейти к **шагу 4.2**.
3.1. Разбиваем *D* и устанавливаем $D' = S_{p_0,\beta}(R_\alpha D)$, $G = G_{O,a,b}(R_\alpha D'), Remain = D \setminus C(D', G)$.Угол β для отделенной области $S_{p_0,\beta}(R_\alpha D)$ должен максимизировать путь L(*G*) с ограничением length(*L*(*G*))≤*l*. Устанавливаем *D*=*D*'.
4.1. Добавляем (*D*, *G*, *L*(*G*)) в массив *Parts*.
5.1. Если $Remain! = \emptyset$, устанавливаем *G*=*Remain* и переходим к **шагу 2**, иначе переходим к **шагу 6**.
3.2. Разбиваем сетку и устанавливаем $G' = S_{p_0,\beta}(R_\alpha G), Remain = G \setminus G'$. Угол β для отделенной сетки $S_{p_0,\beta}(R_\alpha G)$ должен максимизировать путь L(*G*) с ограничением length(L(*G*))≤*l*. Устанавливаем *G*=*G*'.
4.2. Добавляем (*D*, *G*, L(*G*)) в массив *Parts*.
5.2. Если $Remain! = \emptyset$, то устанавливаем *G*=*Remain* и переходим к **шагу 2**, иначе переходим к **шагу 6**.
6. Конец.

Вариант 3.1 подходит, если в процессе мониторинга может измениться область определения или размер прямоугольников покрытия (т.е. поднимается или опускается агент). В этом случае следующая часть домена может иметь новую сетку с новым размером. Вариант 3.2 более эффективен в вычислительном отношении.

Реализация описанных в докладе алгоритмов в среде Wolfram Mathematica 12 свободно доступна в блоге Wolfram Community [4].

2. Генерация случайных доменов, построение и искажение сеток

Для отработки указанного в разделе 1 алгоритма создавались как выпуклые, так и невыпуклые многоугольники фиксированной площади с определенным количеством случайно расположенных вершин (рис. 1).

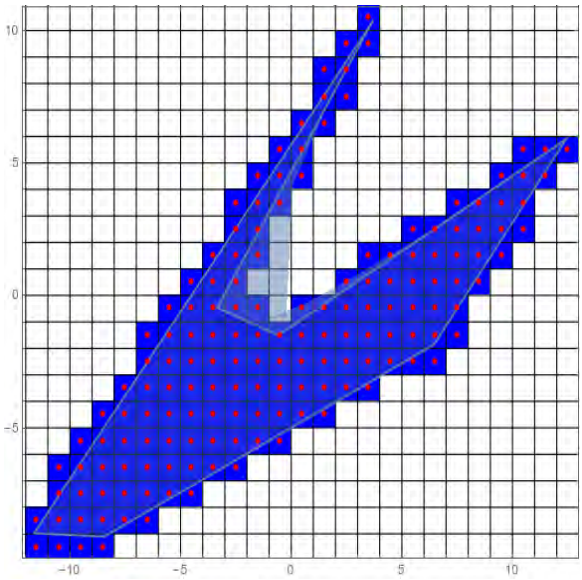

Рис. 1. Пример тестового многоугольника, сетки и покрытия квадратами

Размер ячейки, т. е. покрывающего прямоугольника размера $a \times b$, зависит от поля зрения камеры агента и от ее высоты над многоугольником. Если область довольно велика, то компенсировать недостаток энергии у агента можно за счет перспективного искажения области с матрицей гомографии [5]

$$H = \begin{pmatrix} Rot & tran \\ per & dis \end{pmatrix},$$

подняв камеру агента на достаточную высоту и наклонив ее. Здесь *Rot* – двумерная матрица поворота камеры, *tran* – вектор сдвига, вектор *per* отвечает за перспективу, а число *dis* – за подъем камеры. На рис. 2 показано искажение области с матрицей гомографии

$$H = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0.3 & 0.9 & 9 \end{pmatrix}.$$

Искаженный многоугольник покрывается меньшим числом прямоугольников, чем исходный, что упрощает задачу нахождения кратчайшего пути. При обратном преобразовании H^{-1} получается покрытие исходного многоугольника выпуклыми четырехугольниками.

При движении агента из точки r в точку r_0 матрица гомографии должна изменяться, чтобы сохранять возможность состыковать покрытие области в разных точках без пересечений и пробелов. Вид указанного изменения следует из выражения для дробно-линейного преобразования

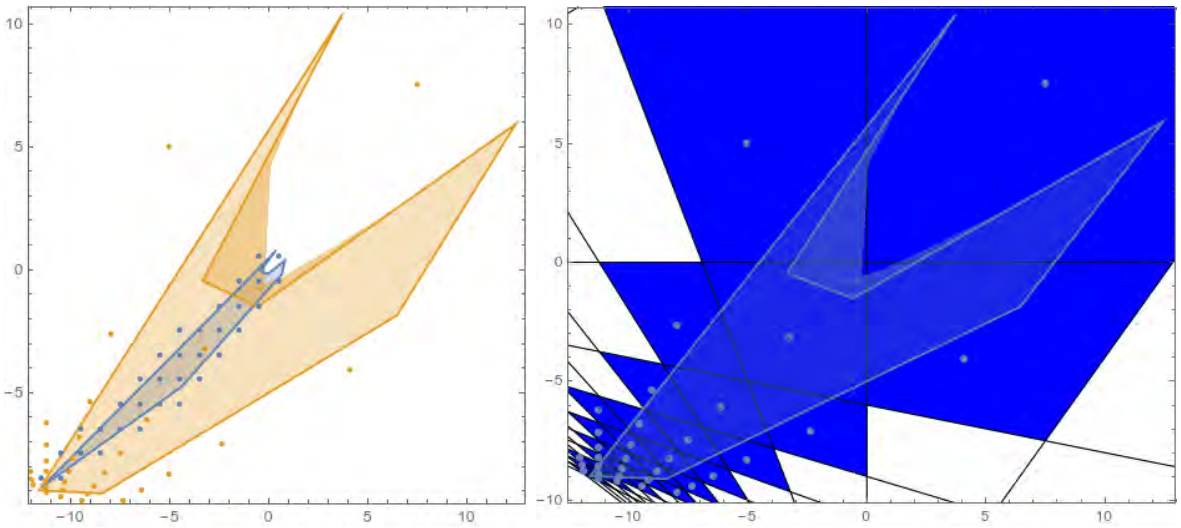

Рис. 2. Построение покрытия с перспективой

Видно, что элементов сетки существенно меньше, чем для покрытия без искажений

$$\frac{Rot\ (r + r_0) + tran}{per \cdot (r + r_0) + dis} = \frac{Rot\ r + (Rot\ r_0 + tran)}{per \cdot r + (per \cdot r_0 + dis)},$$

т. е. камеру агента надо поднимать выше по мере движения.

3. Вычисление и сглаживания кратчайшего пути в одной подобласти

Для вычисления решения (точного или приближенного) ЗК могут быть использованы свободно доступные библиотеки решателя Concorde [6]. В случае прямоугольной сетки задача о самом длинном пути имеет алгоритм решения с линейной сложностью [7]. Поскольку решение представляет собой ломаную линию, что может быть неестественно для некоторых агентов, можно сгладить найденный путь сплайнами того или иного вида.

На рис. 3 показаны примеры интерполяции пути сплайнами третьего порядка (красная линия) и B-сплайнами второго порядка (оранжевая линия).

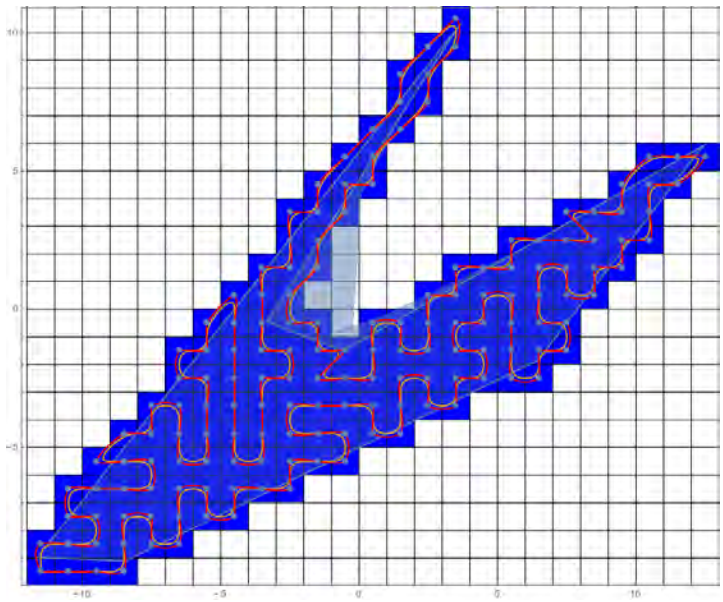

Рис. 3. Пример покрывающего пути и его сглаживаний сплайнами

4. Разбиение области на подобласти для выпуклых и невыпуклых многоугольников, поиск набора кратчайших путей для подобластей

Зададим прямую ℓ, отсекающую часть многоугольника, начальной точкой движения агента и направляющим вектором $(cos\beta, sin\beta)$. Алгоритм нахождения частей многоульника D с вершинами $d_1, \ldots, d_m$ таков:

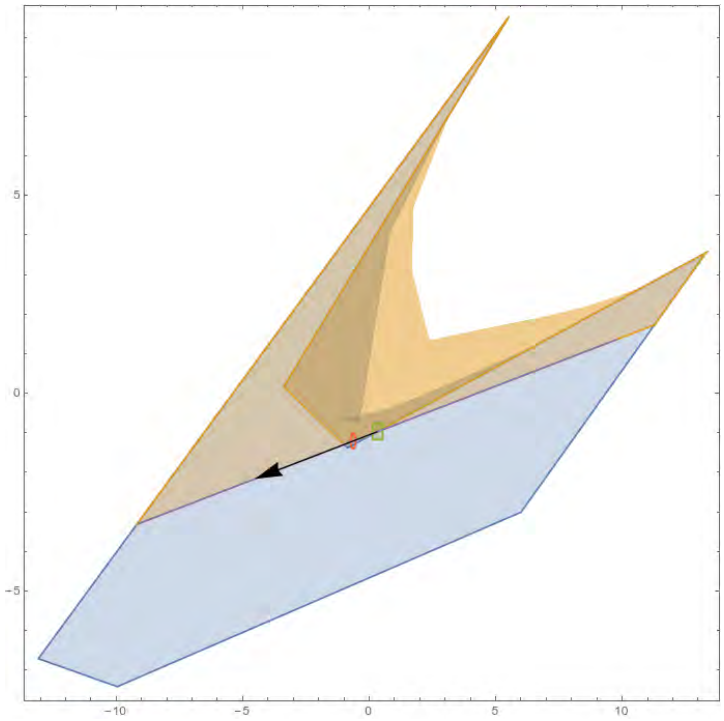

Рис. 4. Разбиение многоугольника.
Стрелкой обозначен направляющий вектор прямой

1. Для всех сторон многоугольника, заданных списком
$$\boldsymbol{d} = \{(d_1, d_2), (d_2, d_3), \ldots, (d_{m-1}, d_m)\},$$
находим точки пересечения $\boldsymbol{s}=\{s_1,\ldots, s_n\}$ с прямой ℓ. Если какая-то из сторон лежит на прямой ℓ, то переходим к следующей стороне.
2. Если какая-то из точек $\boldsymbol{s}$ совпадает с вершиной многоугольника D, то исключаем ее из $\boldsymbol{s}$.
3. Если $s_k \in \boldsymbol{s}$ – точка пересечения для $(d_{i-1}, d_i) \in \boldsymbol{d}$, и $\det\begin{pmatrix}(cos\beta, sin\beta)\\ d_i - p_0\end{pmatrix} \geq 0$, то заменяем в $\boldsymbol{d}$ сторону (d_{i-1},d_i) на (d_{i-1}, s_k).
4. Если $s_k \in \boldsymbol{s}$ – точка пересечения для $(d_{i-1}, d_i) \in \boldsymbol{d}$, и $\det\begin{pmatrix}(cos\beta, sin\beta)\\ d_i - p_0\end{pmatrix} < 0$, то заменяем в $\boldsymbol{d}$ сторону (d_{i-1},d_i) на $(s_k, d_i,)$.
5. Если $s_k \in \boldsymbol{s}$ – точка пересечения для $(d_{i-1}, d_i) \in \boldsymbol{d}$, s_{k+1} – точка пересечения для (d_{j-1}, d_j), то вставляем в $\boldsymbol{d}$ сторону (s_{k+1}, s_k) после стороны (d_{j-1}, d_j).

В результате мы получаем многоугольник $S_{p_0,\beta}(D)$, заданный своим списком сторон $\boldsymbol{d}$, полученным в результате применения алгоритма. Далее, для невыпуклых многоугольников необходимо получить список их частей, полученных в результате вращения отсекающей прямой от 0 до 2π с шагом $\Delta\beta$, зависящем от размера ячейки сетки $a \times b$ и от диаметра многоугольника. Каждой из полученных частей соответствует свое решение ЗК (см. рис. 5).

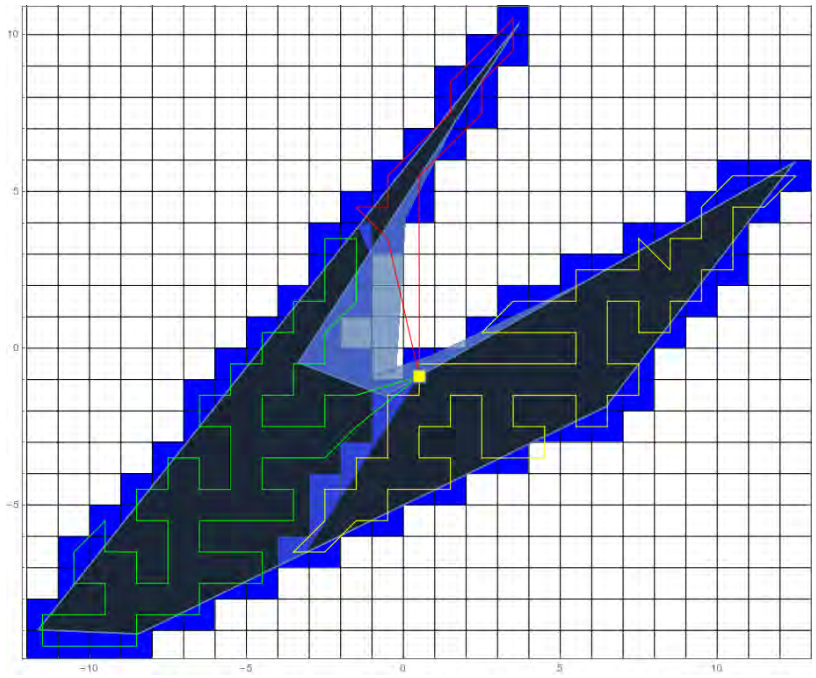

Рис. 5. Поиск покрывающего пути в разбитом многоугольнике.
Точка старта обозначена желтым квадратом

В случае выпуклого многоугольника (рис. 6) мы можем разбить его определенным образом, изменяя угол линии разбиения не от 0 до 2π, а в меньшем промежутке. Это может быть полезно для целей оптимизации. Мы рассмотрим два случая: начальная точка находится в вершине многоугольника и начальная точка находится на ребре многоугольника.

Если начальная точка – вершина d_i, то необходимо найти смежные вершины d_k, d_j и найти наименьший угол γ между ребрами (d_k, d_i), $(d_i, d_j) \in \boldsymbol{d}$. При начальном значении β отсекающая прямая проходит через (d_k, d_i), при конечном – через (d_i, d_j), при этом направляющий вектор прямой должен быть направлен к центру области D. Таким образом, вместо $\lceil 2\,\pi/\Delta\,\beta \rceil$ областей, надо будет сравнить лишь $\lceil \gamma/\Delta\,\beta \rceil$областей.

Если начальная точка не является вершиной и принадлежит стороне (d_i, d_{i+1}), то необходимо изменять угол β так, чтобы и в его начальном, и в его конечно значениях отсекающая прямая проходила через (d_i, d_{i+1}). Тогда придется строить решения ЗК для $\lceil \pi/\Delta\,\beta \rceil$ областей.

Если диаметр сетки, соответствующей $S_{p_0,\beta}(D)$, больше радиуса связи r или длина пути – решения ЗК больше l, то такое множество исключается из рассмотрения. Из оставшихся множеств выбираются такие, что для них длина пути – решения ЗК максимальна.

Таким образом, мы можем найти такое β, что решение задачи комивояжера внутри $S_{p_0,\beta}(D)$ будет удовлетворять ограничениям на длину пути l и на радиус связи r при этом длина пути будет максимально возможной при этом ограничении. Далее возникает проблема перекрывания путем оставшейся части D. Для исключения этого перекрывания, необходимо построить сетку для $S_{p_0,\beta}(D)$, построить по этой сетке покрытие прямоугольниками, а затем вычесть это покрытие из D. После этого упомянутый алгоритм можно повторять, пока все D не будет покрыто.

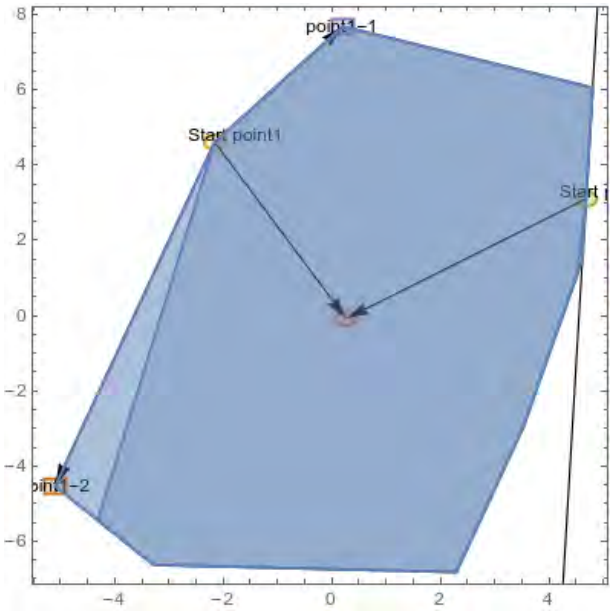

Рис. 6. Разбиение выпуклого многоугольника.
Эллипсами обозначены начальные точки и центр многоугольника

5. Смена стартовых точек

Иногда невозможно разделить многоугольник на меньшие многоугольники, чтобы агент мог покрывать их из одной стартовой позиции. Если у нас нет вариантов разделения, то мы можем изменить начальную точку. Из практических соображений новая начальная позиция должна предпочтительно находиться на границе домена. Причина в том, что в сельском хозяйстве агент не должен пересекать поля, чтобы не повредить ценные растения. Кроме того, мы можем триангулировать [8] многоугольник на меньшие многоугольники с площадью *maxArea*, внутри которых агент заведомо может построить покрывающий путь. Мы можем использовать любую из следующих точек на границе полигона в качестве начальной позиции агента.

В качестве начальных позиций следует выбрать точки, являющиеся общими вершинами наибольшего возможного числа треугольников (см. рис. 7). В этом случае мы можем покрыть множество треугольников, не меняя начальную точку. Но пути в треугольниках могут быть неоптимальными, и мы можем объединить треугольники, чтобы найти более оптимальные пути.

Каков критерий применения процесса триангуляции? Например, в равностороннем треугольнике изменение начальной точки может оказаться бесполезным. Будем предполагать, что мы триангулируем регион, если все его стороны длиннее, чем $l/2$, где l – максимально возможная длина пути агента.

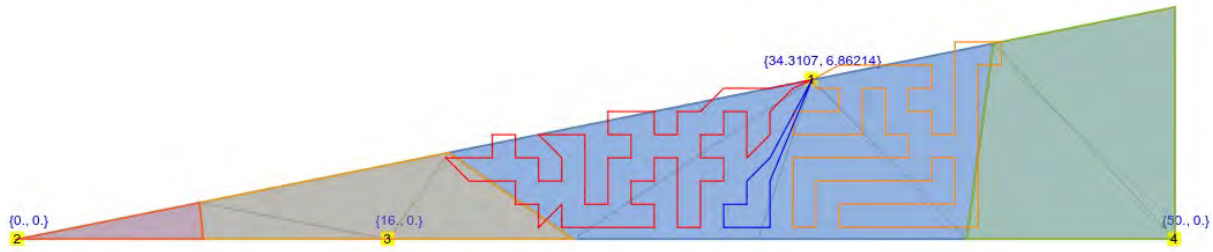

Рис. 7. Разбиение многоугольника со сменой стартовых позиций.
Желтые квадраты с цифрами 1, 2, 3, 4 обозначают стартовые позиции. Разными цветами выделены подобласти, покрываемые агентом из разных стартовых позиций

6. Случай динамической задачи коммивояжера

В процессе выполнения агентом задания могут измениться как параметры области, так и параметры самого агента. Мы рассмотрим следующие возможные события:

1. Изменение конфигурации домена.
2. Изменение четырехугольника покрытия.
3. Изменение скорости агента.
4. Изменяется максимальная длина пути агента (из-за изменения направления ветра или температуры).
5. Изменение списка важных для посещения агентом точек *POI*.

Событие 1 равносильно добавлению в матрицу весов (или удалению из матрицы) ЗК новых строк и столбцов, событие 2 может вызвать возникновение совершенно новой матрицы весов. Если мы покрываем области прямоугольниками, и длины соответствующих сторон нового и старого прямоугольников покрытия кратны, то событие 2 также вызывает лишь вставку новых строк и столбцов в матрицу весов (или удаление строк или столбцов из матрицы) для ЗК между уже существующими строками и столбцами. События 3 и 4 влияют на разбиение области на подобласти (количество подобластей уменьшается при увеличении максимальной длины пути агента и увеличивается при уменьшении максимальной длины пути), а событие 5 требует введения особой метрики, которую может оказаться невозможным представить в виде матрицы весов. Например, если агент должен посетить каждую точку из списка *POI* в точности определенное количество раз, то для каждой такой точки должен быть заведен счетчик количества посещений, значение которого может еще и изменяться независимо от действий агента. Внутри метрики должны быть использованы глобальные переменные – счетчики количества посещений заданных точек. В этом случае, ЗК превращается в сложную динамическую задачу коммивояжера. Concorde не предназначен для решения подобных задач и мы должны создать специальный решатель. Например, можно применить генетический алгоритм или алгоритм муравьиной колонии, как описано в [9, 10].

7. Вычислительная сложность алгоритма

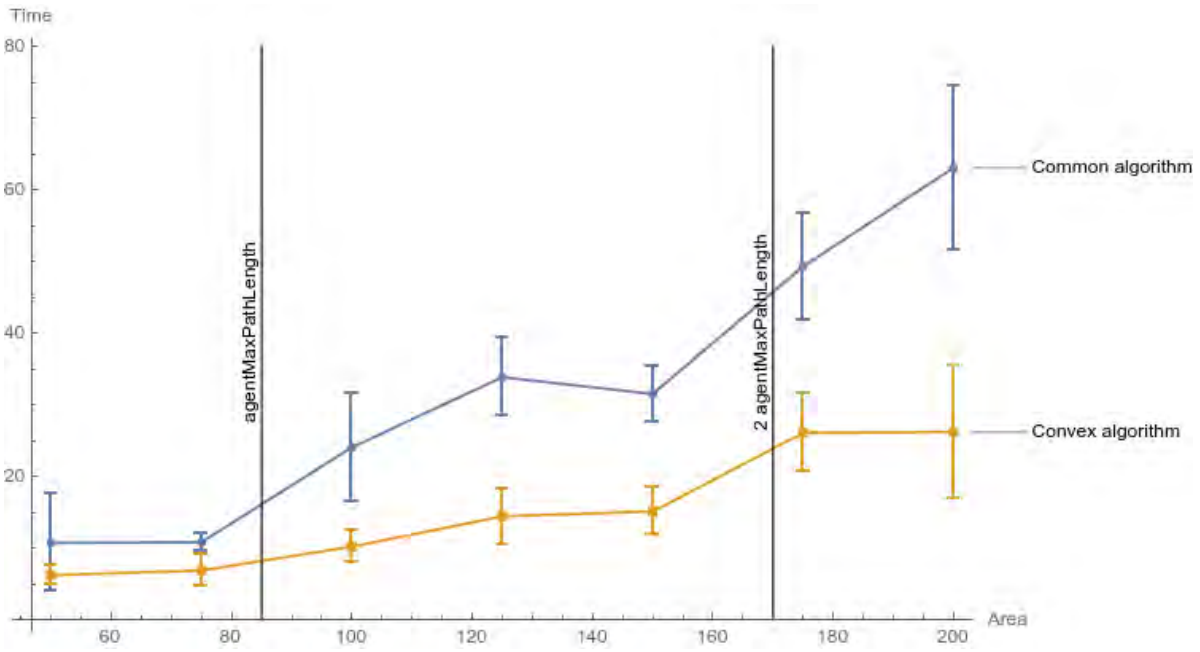

Рис. 8. Зависимость времени выполнения алгоритма от площади многоугольника. Максимальная длина пути $l = agentMaxPathLength$

Было проведено по 30 испытаний для восьмиугольников площадью от 50 до 200 с шагом 15 и максимальной длинной пути $l = 85$. Как и можно предполагать, время вычисления пути возрастает при значениях площади около 85 и $2 \cdot 85$, так как обычно добавляются новые подобласти при разбиении многоугольника (см. рис. 8). Также было произведено по 30 испытаний для многоугольников площадью 100 и количеством вершин от 3 до 10. В обоих случаях область покрывалась квадратами 1×1. Результаты представлены на рис. 9. Эксперименты проводились как для невыпуклых (синие маркеры), так и для выпуклых (оранжевые маркеры) многоугольников.

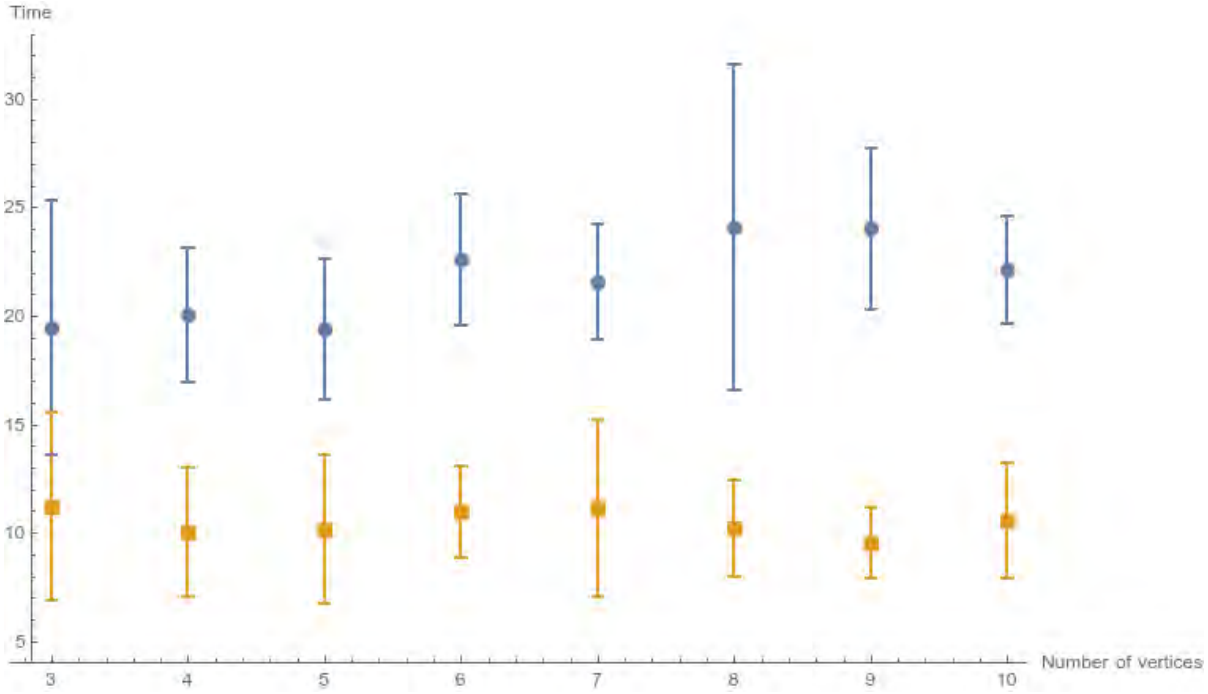

Рис. 9. Зависимость времени выполнения алгоритма от количества вершин многоугольника. Вертикальные линии обозначают среднеквадратическое отклонение

Заключение

В докладе были рассмотрены способы разбиения области с полигональной границей, так, чтобы в каждой части области существовало решение задачи коммивояжера с ограничениями на длину пути. Дальнейшим развитием идей этого доклада будет исследование трехмерных областей, нескольких агентов, одновременно строящих пути покрытия и избегающих коллизий и потери связи, и построение решения динамической задачи коммивояжера для изменяющихся в процессе движения агента ограничений. Помимо этого, практическое значение имеет моделирование системы зарядки, погрузки и разгрузки агента и явный учет энергетических расходов на планирование траектории в постановке задачи.

Работа выполнена при поддержке Минобрнауки России, государственное задание № FFGZ-2021-0001.

Литература

1. **Galceran Enric, Carreras Marc**. A survey on coverage path planning for robotics // Robotics and Autonomous Systems. – 2013. – Vol. 61, iss. 12. – P. 1258-1276. URL: https://doi.org/10.1016/j.robot.2013.09.004

2. **Fevgas G., Lagkas T., Argyriou V., Sarigiannidis P. Coverage** Path Planning Methods Focusing on Energy Efficient and Cooperative Strategies for Unmanned Aerial Vehicles // Sensors. – 2022. – Vol. 22. – P. 1235. URL: https://doi.org/10.3390/s22031235
3. **Acevedo Jose, Arrue Begoña, Maza Ivan, Ollero Anibal**. Distributed Approach for Coverage and Patrolling Missions with a Team of Heterogeneous Aerial Robots Under Communication Constraints // International Journal of Advanced Robotic Systems. – 2013. URL: https://doi.org/10.5772/52765
4. **Kuznetsov Alexander**. The coverage path planning problem with constraints on the path length and time // Wolfram Community, STAFF PICKS, August 23, 2023. URL: https://community.wolfram.com/groups/-/m/t/2997003
5. **Hartley R.I., Zisserman A.**, Multiple View Geometry in Computer Vision. – Cambridge, UK: Cambridge University Press, 2004. – P. 32. URL: http://www.cambridge.org/9780521540513
6. Concorde TSP Solver. URL: https://www.math.uwaterloo.ca/tsp/concorde.html
7. **Keshavarz-Kohjerdi Fatemeh, Bagheri Alireza, Asgharian-Sardroud Alireza.** A linear-time algorithm for the longest path problem in rectangular grid graphs // Discrete Applied Mathematics. – 2012. – Vol. 160, iss. 3. – P. 210-217. URL: https://doi.org/10.1016/j.dam.2011.08.010
8. **Weisstein Eric** W. Triangulation // MathWorld – A Wolfram Web Resource. URL: https://mathworld.wolfram.com/Triangulation.html
9. **Li C., Yang M., Kang L**. A New Approach to Solving Dynamic Traveling Salesman Problems // Simulated Evolution and Learning. SEAL 2006 // Lecture Notes in Computer Science. – 2006. – Vol. 4247. – Springer, Berlin, Heidelberg. URL: https://doi.org/10.1007/11903697_31
10. **DeJans Adam.** Solving the Dynamic Travelling Salesman Problem (DTSP) via Ant Colony Optimization. June 2018. URL: https://github.com/addejans/Dynamic-TSP

Math-Net.Ru https://www.mathnet.ru/rus/person42488
Google Академия https://scholar.google.com/citations?user=kHiFTDcAAAAJ&hl=ru
ZentralMATH Список публикаций на ZentralBlatt
eLIBRARY.RU https://elibrary.ru/author_items.asp?spin=9484-8071
ORCID https://orcid.org/0000-0002-0365-2077
RESEARCHERID https://www.webofscience.com/wos/author/record/H-5773-2017
Scopus https://www.scopus.com/authid/detail.url?authorId=14523105900
ResearchGate https://www.researchgate.net/profile/Alexandr_Kuznetsov3