

ПОСТРОЕНИЕ ОПТИМАЛЬНОГО МАРШРУТА ОБЛЕТА ТЕРРИТОРИИ С ПРЕПЯТСТВИЯМИ ДЛЯ СБОРА ДАННЫХ БЕСПИЛОТНЫМИ ЛЕТАТЕЛЬНЫМИ АППАРАТАМИ

Колчинский И.М.

магистрант 2-го года обучения,

Московский физико-технический институт

(национальный исследовательский университет), город Москва

Хачумов В.М.

профессор, доктор технических наук,

Институт системного анализа ФИЦ ИУ РАН, город Москва,

зав. лаб. интеллектуального управления,

ИПС им. А.К. Айламазяна РАН, г. Переславль-Залесский.

Аннотация. Повсеместное использование беспроводных сенсорных сетей (БСС) и необходимость сбора с них информации в условиях нахождения в труднодоступной местности привело к использованию беспилотных летательных аппаратов (БПЛА) в качестве элементов этих сетей. В статье предложены варианты эффективного кругового покрытия территории с включением непроходимых участков и решается задача оптимального маршрута облета территории с препятствиями с применением эвристических алгоритмов: метод ближайшего соседа, метод Литтла и метод отжига.

Ключевые слова: круговое покрытие плоскости, беспилотные летательные аппараты, оптимальный маршрут, кластеризация, сенсорная сеть

Введение

Использование БПЛА в качестве летающих базовых станций позволяет добиться большей эффективности покрытия и скорости работы беспроводных сетей для получения услуг связи в различных аварийных

ситуациях. Они могут легко перемещаться и обеспечивать быструю связь с наземными пользователями по необходимости. Актуальными задачами с назначением мобильных сетей на основе БПЛА являются: организация и обслуживание сетевых узлов в труднодоступных местах, увеличение срока жизненного цикла сенсорной сети, эффективный сбор информации за счет оптимизации траектории движения БПЛА.

Область мониторинга сенсоров представляется в виде круга определенного радиуса с центром в месте расположения сенсора. На первом месте стоит задача минимизации расходов энергии в БСС, т.к. позволяет увеличить длительность жизненного цикла сети. При большом числе сенсорных узлов для увеличения жизненного цикла целесообразно использовать кластерную организацию

Эффективное покрытие плоскости

В работе [1] представлены варианты эффективного регулярного покрытия плоскости правильными многоугольниками, в вершинах которых располагаются сенсоры одного или двух радиусов мониторинга. Наименее плотным покрытием первого типа является вариант пересечения кругов, расположенных в смежных вершинах, в центре многоугольника. Второй тип – касание кругов в смежных вершинах, при расположении дополнительного круга внутри многоугольника. В третьем типе круги радиуса R пересекаются, а дополнительный радиус r определяется из условия минимизации плотности покрытия. Радиус действия сенсора является регулируемым параметром, т.к. он соответствует мощности устройства и напрямую влияет на энергетические затраты. Подход к задаче построения наиболее эффективной сенсорной сети [2] заключается в том, что пропорциональное изменение радиусов кругов не меняет плотности покрытия, но позволяет минимизировать стоимость и энергетические затраты.

Исследования в [1] и [2] показывают, что плотность покрытия $D = \frac{N\pi R^2}{S}$,

где N – количество кругов, R – радиус, S – площадь территории, для первого и второго типа квадратной решетки одинаковы. На рисунке 1 представлен вариант территории с прямоугольным препятствием. Круги заполняют области вдоль одной оси по обе стороны от препятствия таким образом, чтобы граница препятствия совпадала с линией пересечения кругов, затем проводятся аналогичные действия для оставшейся незаполненной части по другой оси. Таким образом будет покрыта вся территория. На рисунке 2 показана зависимость плотности покрытия от радиуса и наблюдается локальный минимум при радиусе $R=7.5$, для которого D достигает значения 1.77 при количестве узлов $N = 52$.

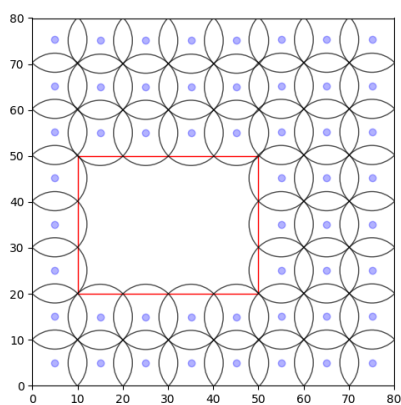


Рисунок 1. Эффективное покрытие для территории с прямоугольным препятствием.

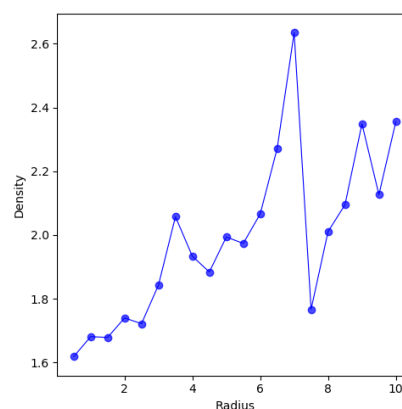


Рисунок 2. Зависимость плотности покрытия первого типа от радиуса для квадратной регулярной решетки.

При большом числе сенсоров целесообразно использовать кластерную организацию наземных пользователей для увеличения жизненного цикла БСС. Целью кластеризации методом k -средних [3] является разбиение данных на k групп, каждая из которых определяется центроидом (центром кластера) и элементами, присвоенными ближайшему центроиду. В процессе обучения алгоритм с каждым шагом обновляет координаты центра на основе полученных данных. Вначале обозначается требуемое количество кластеров и инициализируются центры кластеров. Результат, как правило, зависит от их

начального расположения, а оптимальный выбор неизвестен. Затем, образцы, находящиеся в непосредственной близости к центроиду, объединяются в одну группу, а новые координаты получаются усреднением по всем значениям в кластере. Итерации продолжаются, пока перемещения не прекратятся или не будут достаточно малы.

Важным дополнением к этому алгоритму будет перепривязка точек к новому центроиду, если между центром и пользователем находится препятствие. На этапе усреднения может возникнуть ситуация, когда обновленный центроид окажется внутри запрещенной области и исключится из дальнейшего рассмотрения. Вторая проблема - если рядом не окажется ни одного пользователя и усреднение даст деление 0 на 0. Решением для обеих ситуаций будет определение новых случайных координат за пределами препятствия. На рис. 3 представлен результат алгоритма для распределенных по Пуассону на территории пользователей при $k=13$.

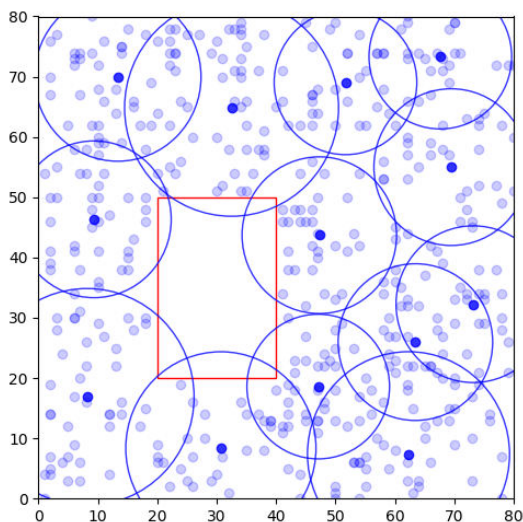


Рисунок 3. Кластеризация наземных пользователей на территории с препятствием.

Алгоритмы решения транспортных задач

При формировании динамических летающих сетей будет решаться известная симметричная задача коммивояжера по поиску оптимального маршрута: БПЛА, начиная свой путь в одном пункте, планирует посетить остальные $(n-1)$. Задача является NP-сложной и решается точно только

полным перебором всех $\frac{(n-1)!}{2}$ возможных туров. Маршрут между точками, на пути которых стоит препятствие, исключается из рассмотрения принятием расстояния достаточно большим, либо бесконечным. Существуют различные эвристические алгоритмы, которые не могут гарантировать получение точного решения, но позволяют получить оптимальный результат значительно быстрее.

Один из таких алгоритмов – **метод ближайшего соседа** (жадный алгоритм) [4]. На каждом этапе выбирается ближайший пункт назначения, который не был посещен ранее. Алгоритм отличается простотой реализации, но не дает оптимального решения, т.к. в конце можно получить большой прирост длины (рис.4)

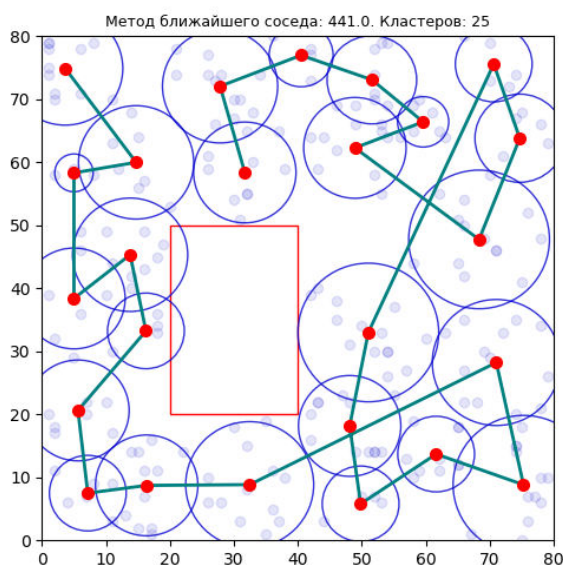


Рисунок 4. Метод ближайшего соседа для 25 кластеров в задаче с препятствиями

Метод Литгла является модификацией метода ветвей и границ и был предложен в 1963г. [5]. Построение маршрута формируется в виде бинарного дерева решений, каждой вершине которого соответствует некоторое подмножество всех туров. На каждом шаге идет поиск ребра, исключение которого позволит увеличить оценку снизу максимально. Множество разбивается на два непересекающихся подмножества, маршруты которых содержат/не содержат это ребро, и вычисляются их нижние границы. Если она

превышает найденное решение, то множество исключается из дальнейшего рассмотрения. Действия проводятся пока не останется матрица размером 2x2. Результат работы алгоритма представлен на рис.5.

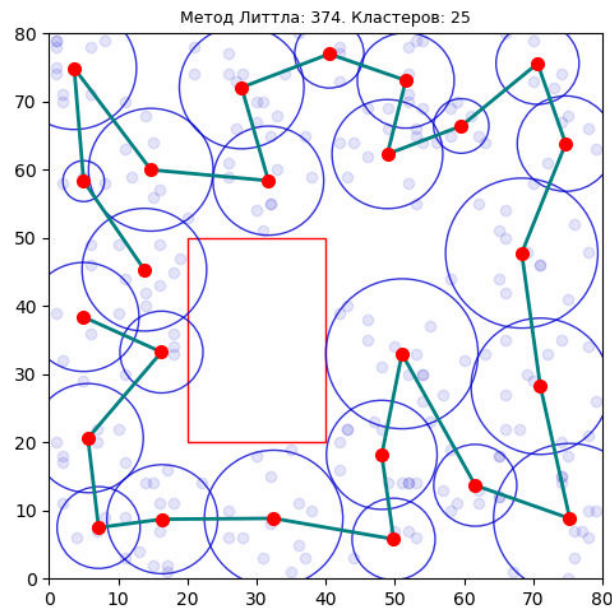


Рисунок 5. Метод Литтла для 25 кластеров в задаче с препятствиями

Метод имитации отжига [6] является вероятностным методом и основан на моделировании физического процесса – нагрева и управляемого охлаждения веществ с кристаллической решеткой, в результате чего получается прочная кристаллическая структура. Цель – привести систему к состоянию с наименьшей энергией. Задается случайный маршрут в качестве начального решения, начальная температура и коэффициент охлаждения. Случайно измененное решение (потенциальное) сравнивается с текущим. Если оно лучше, то становится основным, если нет, то вероятность того, что потенциальное станет текущим равна $P = \exp(-\frac{\delta E}{T})$, где δE – разница между оценками решений, а T – температура. Результат работы алгоритма представлен на рисунке 6.

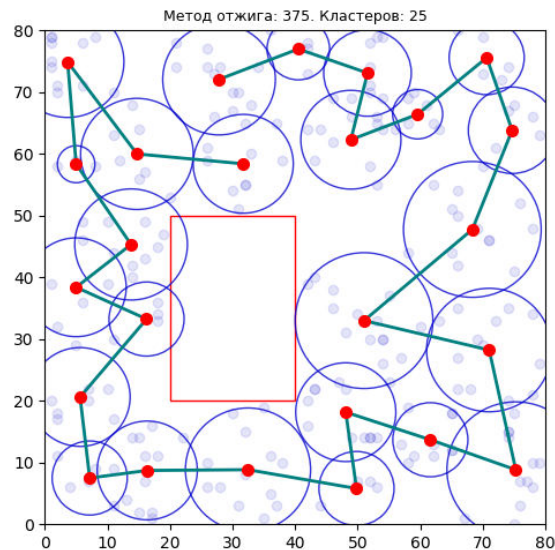


Рисунок 6. Метод имитации отжига для 25 кластеров в задаче с препятствиями.

Таблица 1

k	Метод ближайшего соседа	Метод Литтла	Метод имитации отжига
20	332	342	332
21	360	350	342
22	401	357	360
23	376	355	355
24	385	385	378
25	441	374	375

В таблице 1 представлены длины пройденных путей (в условных единицах) при использовании указанных алгоритмов.

Видно, что метод ближайшего соседа может дать большой прирост длины и в некоторых случаях не обеспечит оптимального результата. Метод Литтла в некоторых случаях уступает методу отжига, поскольку в процессе ветвления может отбросить «плохое» решение, которое в конце может оказаться оптимальным.

Благодарности

Исследование выполнено при частичной финансовой поддержке РФФИ (проекты № 18-07-00025-а, 18-29-03011 мк, 17-29-07003 офи_м) и Программы

фундаментальных исследований Президиума РАН (проект «Разработка и исследование методов и технологии высокопроизводительного сжатия целевой информации, передаваемой по каналам космической связи в интересах национальной безопасности Российской Федерации»).

Список литературы

1. Астраков С.Н., Ерзин А.И., В.В. Залюбовский Сенсорные сети и покрытие плоскости кругами // Дискретный анализ и исследование операций. 2009. Т. 16. №3. С. 3-19

2. Астраков С.Н., Квашнин А.Г., Короленко Л.А. Построение эффективных сенсорных сетей с учетом стоимостных затрат // Математические структуры и моделирование. 2017. №3 (43). С. 50-62

3. Открытая энциклопедия свойств алгоритмов // URL: [https://algowiki-project.org/ru/Алгоритм_k_средних_\(k-means\)](https://algowiki-project.org/ru/Алгоритм_k_средних_(k-means)) (дата обращения 26.04.2020)

4. Ejim S. Implementation of Greedy Algorithm in Travel Salesman Problem // ResearchGate. 2016. URL: <https://www.researchgate.net/publication/307856959>

5. Little J. D. C., Murty K. G., Sweeney D. W., Karel C. An algorithm for the traveling salesman problem // Operations Research. 1963. Vol. 11. No. 6. pp. 972-989.

6. Zhou A., Zhu L., B. Hu, S. Deng, Y. Song, H. Qiu Traveling-Salesman-Problem Algorithm Based on Simulated Annealing and Gene-Expression Programming // MDPI, November 2018.