

УДК 621.391

## ОПТИМИЗАЦИЯ ТРАЕКТОРИИ ДВИЖЕНИЯ БПЛА В ЛЕТАЮЩИХ СЕНСОРНЫХ СЕТЯХ

**К.С. Варельджян**, магистр СПбГУТ; karisha.var@gmail.com

**А.И. Пармонов**, доцент кафедры сетей связи и передачи данных СПбГУТ, к.т.н.; alex-in-spb@yandex.ru

**Р.В. Киричек**, доцент кафедры сетей связи и передачи данных СПбГУТ, к.т.н.; kirichek@sut.ru

Предлагается метод взаимодействия с наземной беспроводной сенсорной сетью с использованием беспилотных летательных аппаратов (БПЛА). Предложенный метод обеспечивает оптимизацию траектории движения с целью минимизации времени доставки информации пользователю. Приводятся сравнительные оценки различных алгоритмов выбора траектории движения для сбора информации с узлов наземной беспроводной сенсорной сети.

**Ключевые слова:** беспроводная сенсорная сеть USN, беспилотный летательный аппарат, сеть толерантная к задержкам DTN, кластеризация, задача коммивояжера, алгоритм штрафования вершин.

**Введение.** Интенсивное развитие технологий построения всепроникающих сенсорных сетей (USN — Ubiquitous sensor Networks) [1, 2], являющихся технологической основой реализации концепции Интернета Вещей [3], приводит к росту их проникновения в различные области деятельности человека. Во многих случаях задачей такой сети является мониторинг некоторых параметров внешней среды или определенных объектов. Самоорганизация USN [4, 5], простота и дешевизна их построения позволяет существенно повысить эффективность решения этих задач. Широкое разнообразие областей применения сенсорных сетей [6] также выражается в широком диапазоне технических решений по их реализации [7].

В ряде случаев сети датчиков могут располагаться на обширной территории, в труднодоступных местах, организовываться на ограниченное время и т.д. Иногда построение инфраструктуры, позволяющей применять проводные или беспроводные технологии для обеспечения постоянных каналов связи с USN, экономически неоправданно или невозможно. В таких случаях требуется использовать некоторые компромиссные решения, гарантирующие необходимые параметры функционирования сети при минимальном использовании технических и других средств.

Один из вариантов такого решения — применение в качестве элементов связи и обслуживания БПЛА, которые благодаря своей доступности и разнообразию позволяют получать достаточно эффективные решения. Такие сети были предложены в [8] и названы летающими сенсорными сетями (ЛСС). ЛСС включают наземный и летающий сегменты. В статье рассматривается один из вариантов взаимодействия наземной USN с БПЛА для решения задач передачи данных между узлами сети и их потребителем.

**Постановка задачи.** Будем полагать, что USN размещена на некоторой плоской поверхности. Информация о топологии сети (координатах ее узлов) может быть получена при построении сети, если при ее размещении фиксируются координаты каждого из узлов, либо путем исследования сети с помощью БПЛА.

При наличии полной информации о топологии сети взаимодействие БПЛА с узлами USN, в первую очередь,

предполагает снятие полученной (накопленной) информации. В системе USN-БПЛА-Шлюз роль БПЛА может быть различна в зависимости от способа организации доставки данных (рис.1).

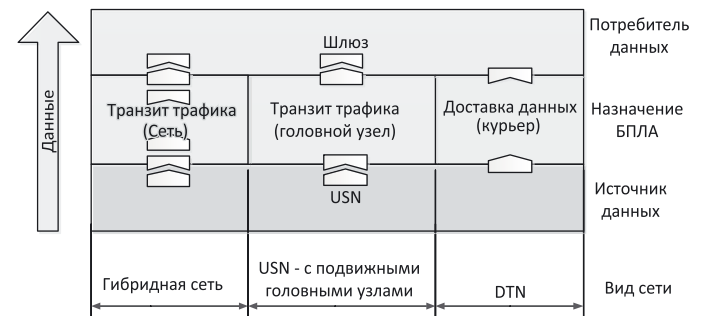


Рис. 1. Роль БПЛА в системе USN-БПЛА-Шлюз

Возможны следующие варианты:

- транзит трафика через *сеть БПЛА*, оборудованную узлом USN, выступающим в роли транзитного (возможно, головного) узла [9, 10]. В БПЛА входят средства связи БПЛА-БПЛА, образующие сеть между подвижными объектами, обеспечивающую маршрут трафика к шлюзу;
- транзит трафика через *узел БПЛА*, оборудованный узлом USN, выступающим в роли транзитного и/или головного узла, и средствами связи со шлюзом. В этом случае узел, размещенный на БПЛА находится в зоне связи шлюза;
- доставка данных. БПЛА оборудован узлом USN, выступающим в роли головного узла или шлюза. Получаемые узлом данные хранятся в его запоминающем устройстве в течение времени, необходимого для перемещения БПЛА к точке базирования, где производится считывание собранных данных. В этом случае имеет место задержка доставки данных, равная времени движения БПЛА.

Рассмотрим последний из вариантов использования БПЛА. Систему можно рассматривать как сеть толерантную к задержкам (DTN) [11]. В сети имеют место два основных способа передачи (транспортировки) данных: первый — с помощью технологией беспроводной связи (USN-БПЛА) и второй (механический) — перемещение БПЛА в пространстве. Вероятно, что второй способ требует существенно большего времени. Одним из основных показателей качества функционирования DTN является время доставки данных.

**Метод минимизации времени доставки данных.** Доставка (транспортировка) данных предполагает сбор и доставку данных в точку пространства, где возможна их передача в шлюз. Сбор данных производится с помощью технологии связи между узлами USN, имеющими ограниченный радиус связи. Поэтому, в зависимости от масштабов сети и требований ко времени доставки данных, сбор данных

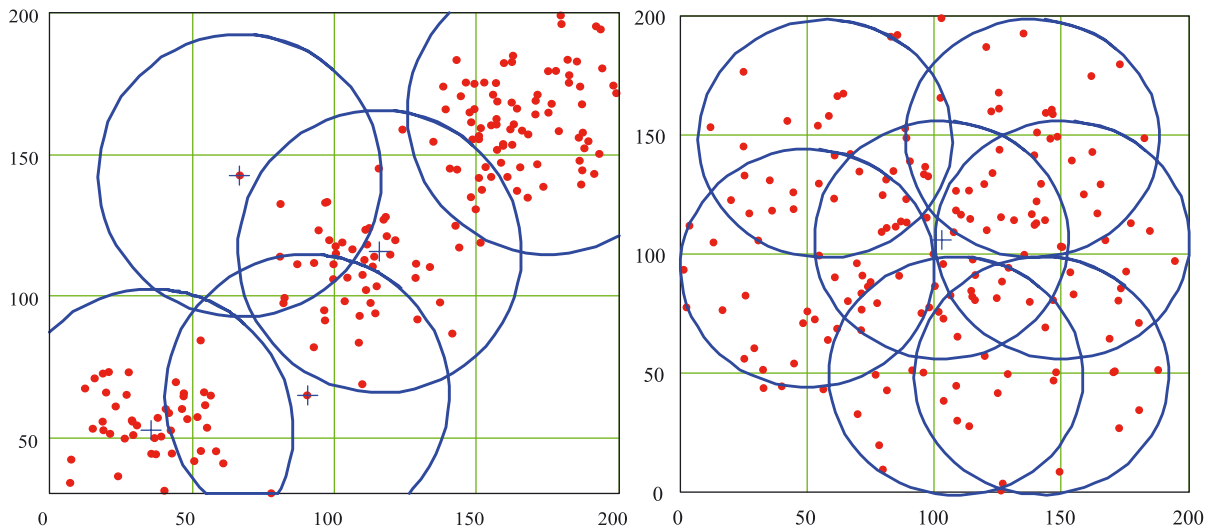


Рис. 2. Различный характер размещения узлов на территории обслуживания

представляет собой перемещение БПЛА между узлами или группами узлов USN и считывание данных.

В этом случае траектория движения БПЛА зависит от характера размещения узлов сети на обслуживаемой территории. Если радиус связи БПЛА может обеспечить взаимодействие с некоторым числом узлов USN, то целесообразно найти точки, в которых может быть обслужено максимальное число узлов. Тогда траектория движения БПЛА будет линией, соединяющей данные точки. От выбора траектории существенно зависит длина маршрута БПЛА, а следовательно, и время доставки данных. На рис. 2 приведены примеры покрытия узлов USN кругами с радиусом, равным радиусу связи БПЛА при различном характере размещения узлов по территории.

Для нахождения траектории, при движении по которой БПЛА имеет возможность считать информацию со всех узлов сети, требуется найти покрытие всех узлов USN. При этом следует исходить из минимума затрат времени. Полагая, что скорость движения БПЛА постоянна, найдем траекторию наименьшей длины. Предположим, что зона действия БПЛА представляет собой круг радиуса  $R$ , а считывание информации с узлов USN может производиться в произвольных точках. Разделим данную задачу на две составляющие:

- нахождение минимального числа точек, в которых считывается информация (минимизация);
- определение оптимального маршрута движения БПЛА между этими точками.

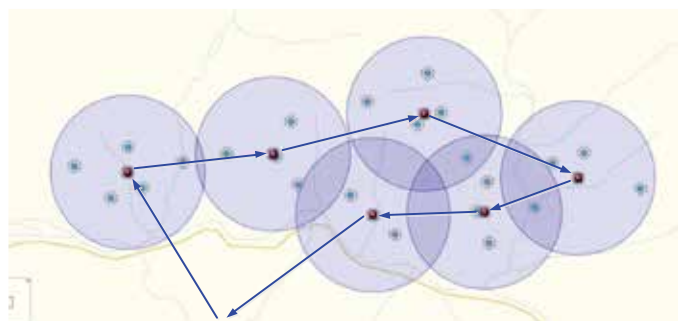


Рис. 3. Пример траектории движения БПЛА при известной топологии сети

Нахождение минимального числа точек считывания информации заключается в покрытии исходного множества точек (узлов) сети минимальным числом кругов радиуса  $R$ . Для решения задачи может быть применен метод кластерного анализа, в частности, FOREL [12–14].

Алгоритм FOREL решает задачу кластеризации, минимизируя суммарное квадратичное расстояние элементов кластеров (узлов USN) от центров масс этих кластеров. Минимизируемая функция может быть записана как:

$$M = \sum_{i=1}^k \sum_{x_j \in S_i} (x_j - \mu_i)^2, \tag{1}$$

где  $k$  — число кластеров;  $S_i$  — множество элементов  $i$ -го кластера;  $\mu_i$  — координаты центра масс  $i$ -го кластера;  $x_j$  — координаты центра масс  $j$ -го элемента кластера. Значение  $(x_j - \mu_i)$  представляет собой евклидово расстояние между элементом кластера и центром масс кластера.

Рассмотрим алгоритм для двумерного пространства (плоскости). В данном случае каждый элемент рассматривается как точка на плоскости и характеризуется своими координатами  $(x_j, y_j)$ .

Координаты центра масс  $i$ -го кластера:

$$x_i^{(\mu)} = \frac{1}{n_i} \sum_{j=1}^{n_i} x_j; \quad y_i^{(\mu)} = \frac{1}{n_i} \sum_{j=1}^{n_i} y_j. \tag{2}$$

Каждый объект (точка) может характеризоваться некоторым параметром (массой)  $m_j$ . Тогда центр масс (плоской фигуры) будет определяться как

$$x_i^{(\mu)} = \frac{1}{m_i^{(\Sigma)}} \sum_{j=1}^{n_i} m_j x_j; \quad y_i^{(\mu)} = \frac{1}{m_i^{(\Sigma)}} \sum_{j=1}^{n_i} m_j y_j; \tag{3}$$

$$m_i^{(\Sigma)} = \sum_{j \in S_i} m_j.$$

В FOREL задается размер кластера  $R$ . В двумерной задаче на геометрической плоскости под  $R$  понимается максимальное расстояние от элемента кластера до его центра масс (радиус). Каждый элемент также рассматривается как точка на плоскости и характеризуется своими координатами  $(x_j, y_j)$ . Координаты центра масс  $i$ -го кластера определяются согласно (2) или (3).

Алгоритм кластеризации включает следующие основные шаги.

1. Задаются границы области, координаты объектов (точек)  $I = \{I_1, I_2, \dots, I_n\}$  и максимальный размер кластера ( $R$ ), номер кластера  $i = 1$ .

2. Выбирается случайная точка  $m_i$  в заданной области. Эту точку на начальном этапе принимаем за центр масс кластера.

3. Все объекты, находящиеся на расстоянии не более  $R$ , приписываются к данному кластеру.

4. Для полученного кластера вычисляется центр масс  $\hat{m}_i$ , согласно (2) или (3). Если вычисленные координаты центра масс совпадают с точкой  $m_i$ , то полагаем, что кластер  $i$  определен. Все приписанные данному кластеру точки помечаются номером кластера и исключаются из дальнейшего рассмотрения. Переход к п. 5 (поиск следующего кластера). Если вычисленные координаты центра масс не совпадают с точкой, то процесс поиска продолжается, т.е. принимаем  $m_i = \hat{m}_i$  (переход к п. 3).

5. Проверяем, остались ли объекты, не отнесенные ни к одному кластеру. Если нет, то все кластеры определены, завершение поиска и переход к п. 6. Если объекты остались, то переход к п. 2 (поиск очередного кластера).

6. Выводим данные о принадлежности объектов к кластерам и данные о центрах масс полученных кластеров. Конец поиска.

*Примечание.* Во время поиска очередного кластера может оказаться, что на удалении менее  $R$  от выбранного центра масс нет ни одного объекта. В таком случае следует выбрать новый центр масс (случайный выбор). Общий алгоритм FOREL приведен на рис. 4.

Следует отметить, что если элементы сети расположены не на плоской поверхности, то аналогично кластеризации в двумерном пространстве такую задачу можно решать и для трехмерного.

После выбора точек снятия данных необходимо найти траекторию движения БПЛА. Вероятно, можно предъявлять различные требования к свойствам искомой траектории: протяженность (длина), число посещаемых точек снятия данных, угол изменения направления движения, высота полета и т.д. При решении задачи будем принимать в качестве критерия только длину пути. Найденные точки снятия данных представим вершинами полного графа.

Предположим, что БПЛА может начинать и заканчивать облет точек сбора в любых точках над обслуживаемой территорией. Эти точки могут находиться как в вершинах полученного графа, так и вне его. В последнем случае граф должен быть дополнен двумя вершинами, соответствующими этим точкам. Можно показать, что если требуется начинать и заканчивать облет в одной и той же точке, то ее можно представить двумя точками, расстояние между которыми равно нулю. В общем случае искомая траектория представляет собой маршрут, при движении по которому при обходе всех вершин графа длина пройденного пути минимальна.

Если протяженность маршрута больше максимально допустимой (возможной дальности полета БПЛА или времени доставки данных), то следует отыскать несколько маршрутов, обеспечивающих параллельное снятие данных. В общем случае решение, состоящее в том, чтобы «стянуть» заданное число вершин исходного графа ребрами с минимальным суммарным весом, связано с нахождением минимального дерева Штейнера (SMT).

Известно, что эта задача не имеет общего решения [15, 16]. Если потребовать, чтобы маршрут проходил через каждую из вершин только один раз, то задача сводится к поиску минимального гамильтонова цикла или цепи (задача коммивояжера). Такая задача относится к классу NP сложных задач [15]. При относительно малом числе вершин графа она может быть решена полным перебором возможных вариантов. Однако при росте числа вершин (размера сети) ее сложность возрастает экспоненциально.

Определим исходные данные задачи. Примем за вершины полного графа  $G$  найденные на предшествующем этапе точки сбора данных (центры масс кластеров в решении задачи кластеризации)  $X = \{x_i\}$ ,  $i = 1..k$ , где  $k$  — число найденных точек. Причем, будем полагать, что граф неориентированный. Такое предположение допустимо, когда нет ограничений на направление движения БПЛА над территорией обслуживания. Для задания точек начала и конца маршрута дополним исходный граф двумя вершинами  $x_s$  и  $x_f$  соответственно.

Предположим, что координаты точек  $x_s$  и  $x_f$  известны, следовательно, могут быть вычислены и расстояния от них до всех остальных вершин графа. Пусть решением задачи является маршрут между заданными вершинами  $x_s$  и  $x_f$ , проходящий через все вершины графа. Полагаем, что маршрут должен начинаться в вершине  $x_s$ , заканчиваться в  $x_f$  и проходить через каждую из вершин графа только один раз, длина маршрута должна быть минимальна. При данных ограничениях решением задачи будет кратчайшая гамильтонова цепь (маршрут коммивояжера) [15]. Для ее решения могут быть использованы различные методы: точные — полного перебора, или приближенные, например, алгоритм, использующий дерево решений, или алгоритм штрафования вершин [15].

Точные методы требуют слишком большого объема вычислений, поэтому отдадим предпочтение алгоритму штрафования вершин (алгоритму Кристофидеса) [15]. Этот алгоритм имеет относительно простую реализацию и обеспечивает сходимость к оптимальному решению в большинстве случаев. Как показано в [15], при не сходимости алгоритма к оптимальному решению, может быть получено решение близкое к оптимальному. Метод, реализуемый алгоритмом штрафования вершин, основан на последовательном поиске кратчайших остовов графа (SST) и «штрафования» вершин, степень которых превышает 2. Результат выполнения алгоритма — кратчайший остов графа, все вершины которого

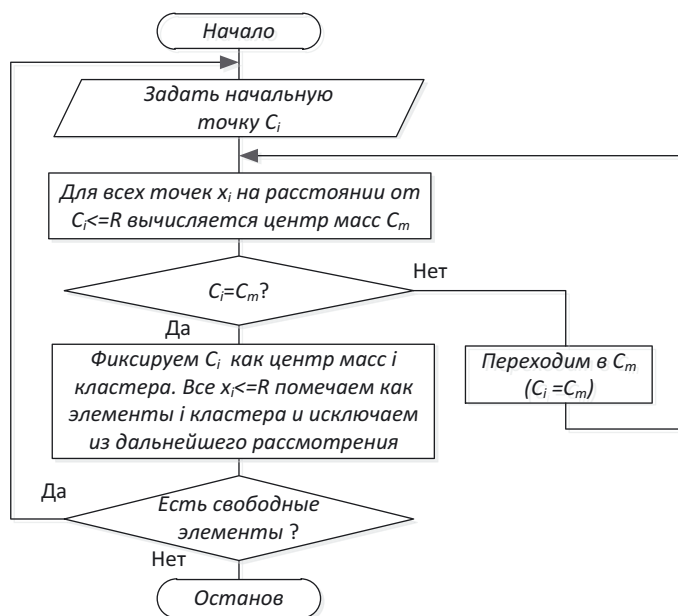


Рис. 4. Алгоритм кластеризации FOREL

(кроме вершин  $x_s$  и  $x_t$ ) имеют степень 2, т.е. искомый маршрут, длина которого близка к длине гамильтоновой цепи.

Если в задаче требуется построить несколько маршрутов, то исходный полный граф может быть разделен на необходимое число полных подграфов, для каждого из которых должна быть решена задача поиска оптимальных маршрутов движения БПЛА.

**Поиск маршрута алгоритмом штрафования вершин.** Полагаем, что граф задан матрицей длин ребер

$$C_0 = \|c_{ij}\|; \quad c_{ij} < \infty; \quad i, j = 1 \dots k,$$

где  $k$  — число вершин графа.

Заменим значения в строках или столбцах  $i=s$  и  $i=t$ , или  $j=s$  и  $j=t$  значениями  $c_{sj} = c_{sj} + N$ ,  $j = 1 \dots k$ , или  $c_{is} = c_{is} + N$ ,  $i = 1 \dots k$ , где  $N$  — достаточно большое число, т.е.  $N \gg c_{ij}$ ;  $i, j = 1 \dots k$ . Получим результирующую матрицу

$$C = \|c_{ij}\|; \quad i, j = 1 \dots k.$$

Далее находим кратчайший остов по матрице  $C$ . Если он является гамильтоновой цепью, т.е. каждая из вершин графа, кроме  $s$  и  $t$ , имеет степень равную 2, то задача решена. Если нет, то «оштрафуем» те вершины графа, степени которых превышают 2. Если вершина  $r$  имеет степень больше 2, то

$$c_{ir} = c_{ir} + \eta(c_{ir} - 2); \quad i, r = 1 \dots k,$$

где  $\eta$  — величина штрафа.

Для обновленной матрицы  $C$  вновь находится кратчайший остов и производится проверка на гамильтонову цепь. Итерации повторяются пока очередной кратчайший остов не будет представлять собой гамильтонову цепь, последняя и является решением задачи. В общем случае, если число итераций слишком велико, а кратчайшая цепь еще не найдена, то итерационный процесс может быть остановлен, а для поиска использован иной метод. Следует отметить, что отличная от нуля вероятность того, что поиск не сойдется к оптимальному решению, характерна для многих известных методов.

Проблема требует дальнейшего изучения, однако в качестве практического решения можно предложить как модифицировать исходный граф, например, добавлением новой вершины со случайными координатами, или изменением координат вершин, степень которых превышает 2 на случайную величину, лежащую в границах точности их оценки. Последнее фактически эквивалентно «штрафованию» вершины на некоторую случайную величину. При не сходимости алгоритма, т.е. для гарантированного получения маршрута, можно использовать эвристический метод «ближайшего соседа», дающий менее точное решение, но гарантированное решение.

Возможность решения задачи зависит от сходимости алгоритма и существования искомого гамильтонового цикла (или цепи), необходимое условие которого следует из его определения и состоит в том, что все вершины исходного графа должны иметь степень не менее 2. В качестве достаточного условия можно использовать условие Поша [17]: если число вершин графа  $n > 2$  и если для любого  $k$ ,  $0 < k < \frac{n-1}{2}$ , число вершин со степенями, не превосходящими  $k$ , не превосходит  $k$ , то граф содержит гамильтонову цепь. Особенность данной задачи состоит в том, что исходный граф является полным, т.е. каждая вершина связана с каждой, следовательно, каждая вершина имеет степень  $n - 1$ . Условие Поша выполняется, следовательно, любой возможный граф является гамильтоновым графом. В этом случае возможность получения решения зависит только от сходимости алгоритма поиска маршрута (гамильтоновой цепи).

Сходимость и время выполнения алгоритма штрафования вершин существенно зависят от штрафа  $\eta$ . В [15] показано, что при слишком малом штрафе требуется большее число итераций, а при большом — алгоритм может не сойтись. В той же работе предложено несколько подходов к выбору величины штрафа  $\eta$ .

Для анализа сходимости алгоритма и времени его выполнения были проведены испытания на достаточно большом наборе случайных полных графов различной размерности. При этом использовались постоянные положительные штрафы. Значение  $\eta$  выбиралось исходя из распределения длин ребер исходного графа. Ниже приведены результаты тестирования алгоритма штрафования вершин, его модификации со «случайными штрафами» и метода «ближайшего соседа». Тестирование произведено на графах различной размерности от 10 до 60, для каждого из размеров проведено 20 тестирований на случайных графах. Вершины случайного графа представляли собой пуассоновское поле точек на плоскости в области, ограниченной квадратом, а длины ребер — расстояния между ними. На рис. 5 приведен пример найденного маршрута в графе из 60 вершин.

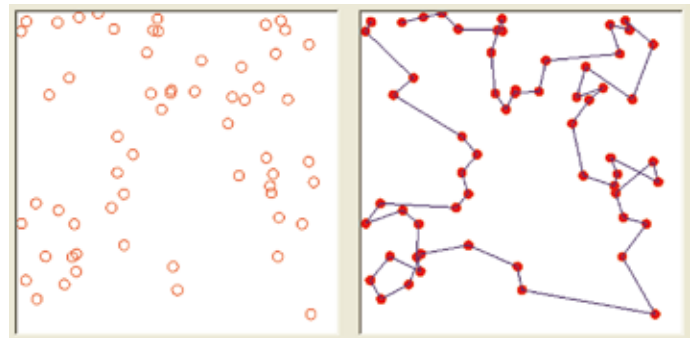


Рис. 5. Пример найденной гамильтоновой цепи в графе из 60 вершин

Число вершин	Постоянные штрафы		Случайные штрафы		Метод «ближайшего соседа»
	Вероятность сходимости	Среднее число итераций	Среднее число итераций	Относительная ошибка длины пути, %	Относительная ошибка длины пути, %
10	1	95	94	0,91	12,92
20	1	610	147	3,72	17,05
30	1	4106	235	7,80	21,52
40	0,90	10426	1168	20,87	26,14
50	0,45	13475	1861	34,22	22,68
60	0,15	15573	2426	38,33	21,57

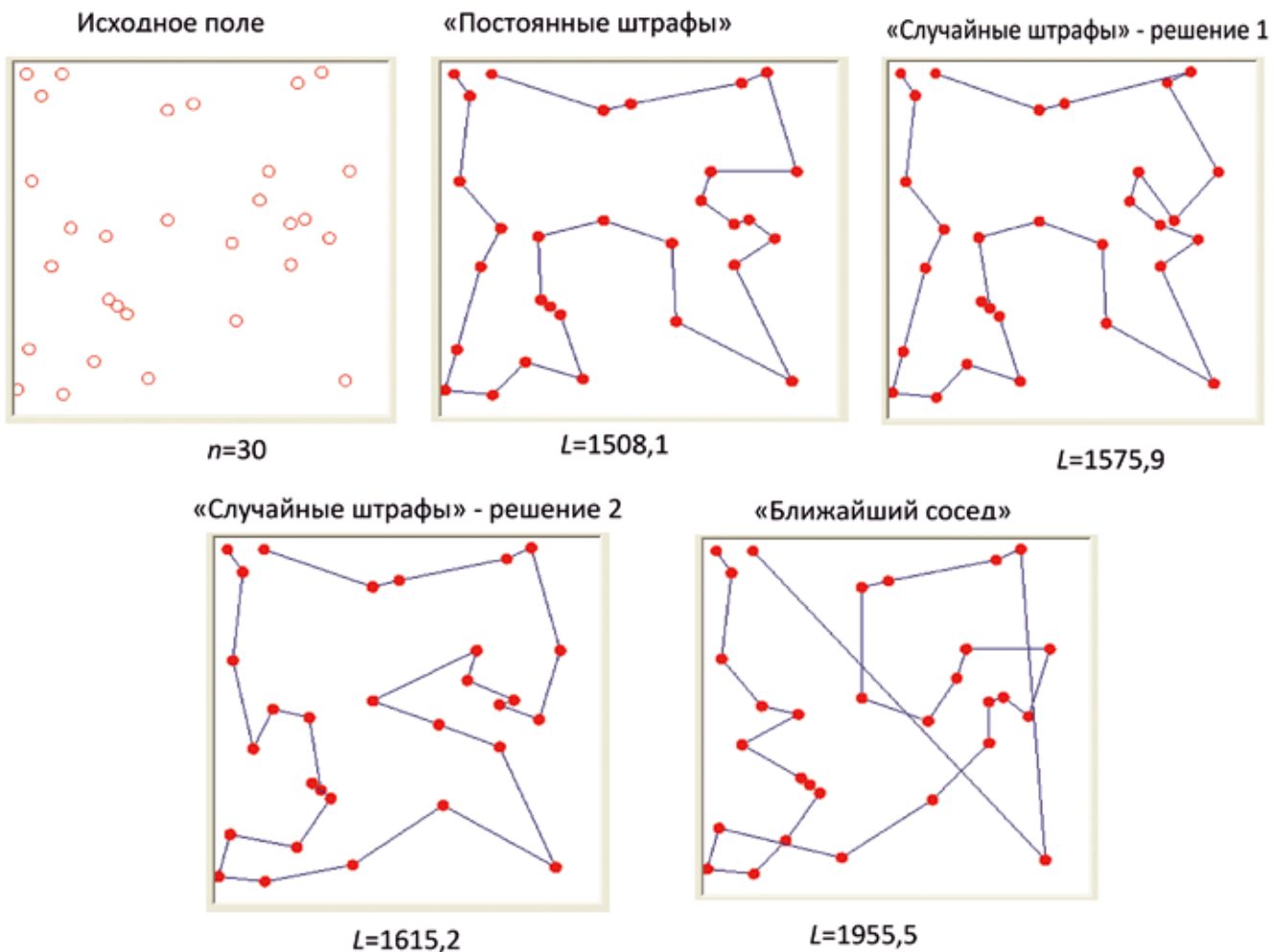


Рис. 6. Примеры результатов поиска пути различными методами

Результаты, полученные различными методами поиска, приведены на рис. 6. В примере использовано исходное поле точек и маршрут, найденный методом «постоянных штрафов». Для метода «случайных штрафов» приведены два варианта решения.

По результатам тестирования (см. таблицу) получены оценки вероятности сходимости алгоритмов и погрешность длины пути по отношению к методу «штрафования вершин» с постоянными штрафами.

На рис. 7 показана зависимость вероятности сходимости для метода «постоянных штрафов» от числа вершин графа. Как было получено в эксперименте, вероятность сходимости начинает заметно снижаться при числе вершин графа больше 30. Когда их более 40 наблюдается резкое снижение вероятности сходимости алгоритма. Следует отметить, что алгоритмы «случайных штрафов» и «ближайшего соседа» обеспечивали 100% сходимость во всем диапазоне размеров графа.

На рис. 8 приведена зависимость относительной ошибки длины найденного маршрута для методов «случайных штрафов» и «ближайшего соседа» по отношению к методу с постоянными штрафами.

Как видно из приведенных графиков, при малом числе вершин относительная ошибка метода «случайных штрафов» ниже, чем метода «ближайшего соседа». Однако при увеличении числа вершин до 45–50 наблюдается рост ошибки данного метода по сравнению с методом «ближайшего соседа». Следует отметить, что вычислительная слож-

ность этих методов существенно отличается в пользу метода «ближайшего соседа».

На рис. 9 показана зависимость числа итераций для методов «постоянных штрафов» и «случайных штрафов». Здесь под итерацией понимается однократное выполнение алгоритма поиска минимального остоного дерева. Из рисунка видно, что число итераций, необходимых для получения решения при методе случайных штрафов, значительно меньше, хотя и соизмеримо с числом итераций для метода «постоянных штрафов».

На рис. 9 не приведены значения числа итераций для метода «ближайшего соседа», так как он использует иной способ поиска маршрута. Его можно рассматривать как одну единственную итерацию (независимо от числа вершин графа), соизмеримую по сложности с поиском остоного дерева.

Следует отметить, что можно улучшить результаты работы алгоритма, именуемого методом «постоянных штрафов», выбрав более эффективную стратегию «штрафования» вершин [15]. Упомянутые модификации метода здесь не рассматривались. Вероятно, они улучшат его сходимость, но существенно не уменьшат вычислительную сложность, по сравнению с эвристическим методом «ближайшего соседа».

**Выводы.** 1. Поиск маршрута БПЛА целесообразно разделить на две задачи: нахождение точек взаимодействия с узлами сети и прокладка через эти точки маршрута кратчайшей длины. Для решения первой задачи может быть ис-

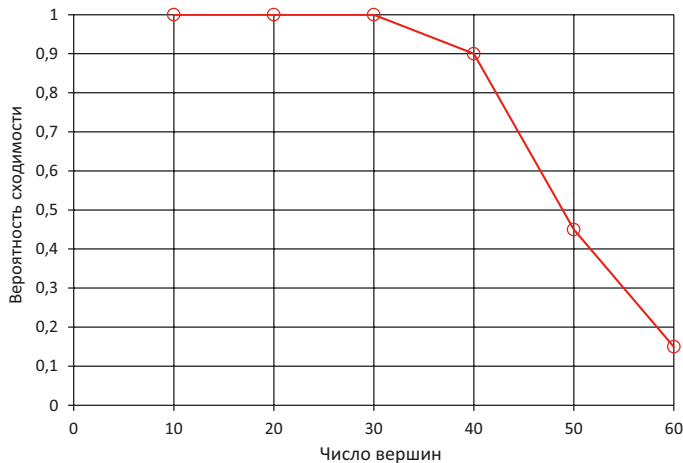


Рис. 7. Зависимость вероятности сходимости для метода «постоянных штрафов» от числа вершин графа

пользован метод кластеризации FOREL, а для второй — методы решения задачи коммивояжера.

2. Задача коммивояжера представляет собой NP-сложную задачу, поэтому для ее решения могут быть применены как методы поиска точного (оптимального) решения, так и эвристические методы поиска решения близкого к оптимальному.

3. Точность результатов, получаемых рассмотренными методами, зависит от размерности графа (числа точек считывания данных). Относительная ошибка предложенного эвристического метода «штрафования вершин случайными штрафами» и эвристического метода «ближайшего соседа» уменьшается с ростом числа вершин.

4. Вычислительная сложность метода «штрафования вершин случайными штрафами» существенно ниже сложности метода «штрафования вершин», однако она имеет аналогичную тенденцию роста с увеличением числа вершин графа.

5. Проведенный анализ методов «штрафования вершин», а также предложенной его модификации «штрафования вершин случайными штрафами» показал, что первый из них может быть использован при относительно малом числе точек считывания данных (до 30). Метод «штрафования вершин случайными штрафами» целесообразно применять при числе точек от 30 до 50. При большем их числе следует пользоваться эвристическим методом «ближайшего соседа».

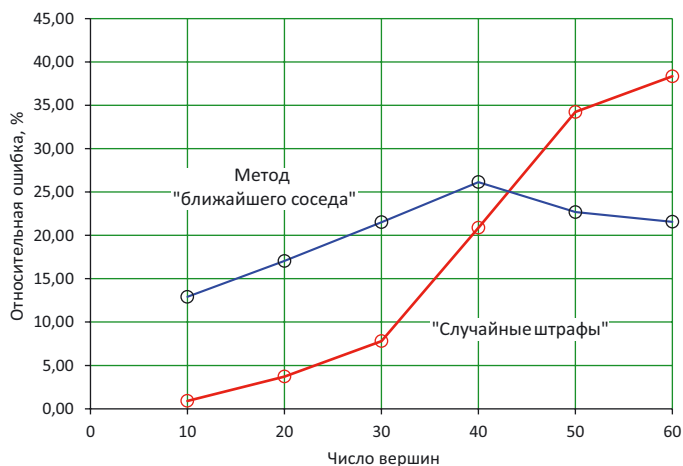


Рис. 8. Зависимость относительной ошибки длины найденного маршрута для методов «случайных штрафов» и «ближайшего соседа»

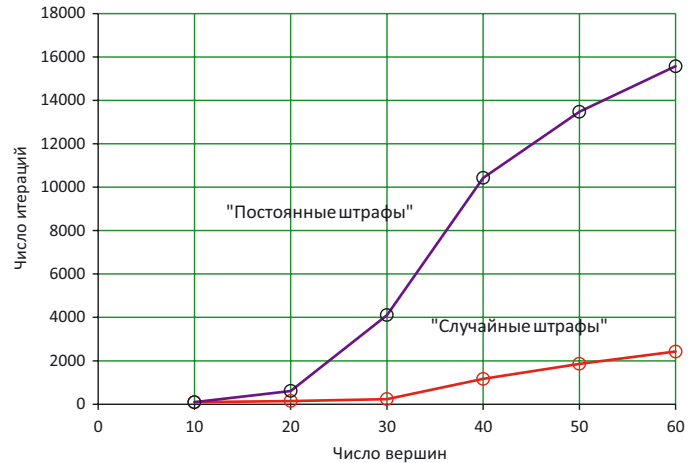


Рис. 9. Зависимость числа итераций для методов «постоянных штрафов» и «случайных штрафов»

Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта N15-07-09431а «Разработка принципов построения и методов самоорганизации для летающих сенсорных сетей».

#### ЛИТЕРАТУРА

1. Кучерявый А.Е., Парамонов А.И., Кучерявый Е.А. Сети связи общего пользования. Тенденции развития и методы расчета. — М.: ГУП ЦНИИС, 2008.
2. Кучерявый А.Е., Кучерявый Е.А. От e-России к u-России: тенденции развития электросвязи // Электросвязь. — 2005. — № 5.
3. Кучерявый А.Е. Интернет Вещей // Электросвязь. — 2013. — № 1.
4. Кучерявый А.Е., Прокопьев А.В., Кучерявый Е.А. // Самоорганизующиеся сети. — СПб.: Любавич, 2011.
5. Кучерявый А.Е. Самоорганизующиеся сети и новые услуги // Электросвязь. — 2009. — № 1.
6. Кучерявый А.Е., Парамонов А.И. Модели трафика для сенсорных сетей в u-России // Электросвязь. — 2006. — № 6.
7. Футахи А., Парамонов А.И., Прокопьев А.В., Кучерявый А.Е. Сенсорные сети в гетерогенной зоне системы длительной эволюции. // Электросвязь. — 2015. — № 3.
8. Кучерявый А.Е., Владыко А.Г., Киричек Р.В. Летающие сенсорные сети // Электросвязь. — 2014. — № 9.
9. Koucheryavy A., Salim A. Cluster head selection for homogeneous Wireless Sensor Networks / Proceedings. International Conference on Advanced Communication Technology (ICTACT 2009). Phoenix Park, Korea. — 2009.
10. Koucheryavy A., Salim A. Prediction-based Clustering Algorithm for Mobile Wireless Sensor Networks / Proceedings. International Conference on Advanced Communication Technology (ICTACT 2010). Phoenix Park, Korea. — 2010.
11. Akyildiz I.F., Akan O.B., Chen C. et al. InterPlaNetary Internet: state-of-the-art and research challenges // Computer Networks. — 2003. — № 43.
12. Парамонов А.И. Разработка и исследование комплекса моделей трафика для сетей связи общего пользования / Автореферат на соискание ученой степени доктора технических наук. — СПб.: СПбГУТ, 2014.
13. Мандель И.Д. Кластерный анализ. — М.: Финансы и статистика, 1988. — 176 с.
14. Ким Дж.О., Мьюллер Ч.У., Клекка У.Р. и др. Факторный, дискриминантный и кластерный анализ. — М.: Финансы и статистика, 1989. — 215 с.
15. Кристофидес Н. Теория графов. Алгоритмический подход. — М.: Мир, 1978. — 432 с.
16. Берж К. Теория графов и ее применения. — М.: Изд-во иностр. лит., 1962. — 311 с.
17. Оре О. Теория графов. — М.: Наука, 1982.