

ISSN 2949-5598

# ДИСКРЕТНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ

Том 32 № 2 2025

Новосибирск  
Издательство Института математики

## РАЗМЕЩЕНИЕ ДРОНОВ ДЛЯ ОПТИМАЛЬНОГО ПОКРЫТИЯ БАРЬЕРА

А. И. Ерзин<sup>1, a</sup>, А. В. Шадрина<sup>2, b</sup>

<sup>1</sup> Институт математики им. С. Л. Соболева,  
пр. Акад. Коптюга, 4, 630090 Новосибирск, Россия

<sup>2</sup> Новосибирский гос. университет,  
ул. Пирогова, 2, 630090 Новосибирск, Россия

E-mail: <sup>a</sup>adilerzin@math.nsc.ru, <sup>b</sup>shadrinanzhela@gmail.com

**Аннотация.** На плоскости задан отрезок прямой линии (*барьер*), а также координаты депо, в которых нужно разместить мобильные устройства (сенсоры или дроны). Каждый дрон стартует из своего депо к некоторой точке барьера, движется вдоль барьера и возвращается в своё депо, пройдя путь ограниченной длины. Часть барьера, вдоль которой двигался дрон, считается *покрытой* этим дроном. Барьер покрыт, если каждая его точка покрыта хотя бы одним дроном. Требуется разместить ограниченное число дронов в заданном множестве депо и определить траектории дронов таким образом, чтобы покрыть весь барьер и при этом число дронов было минимальным (MinNum), или общая длина путей дронов была минимальной (MinSum), или длина самого протяжённого пути среди всех дронов была минимальной (MinMax).

Ранее авторы исследовали аналогичную задачу с неограниченным числом дронов и предложили псевдополиномиальный алгоритм для её решения, зависящий от длины барьера  $L$ . В этой статье рассматривается обобщённая задача, в которой число дронов ограничено, и для построения её оптимального решения предложен алгоритм такой же трудоёмкости. Вместе с тем, в случае неограниченного числа дронов новый алгоритм имеет трудоёмкость в  $L$  раз меньше трудоёмкости ранее разработанного алгоритма. Ил. 2, библиогр. 20.

**Ключевые слова:** покрытие барьера, линейная маршрутизация, мобильное устройство (дрон), ограниченная энергия, трудоёмкость.

### Введение

Многие протяжённые объекты (дороги, трубопроводы, ЛЭП и др.) можно представить в виде отрезка прямой на плоскости («барьера»).

Для эффективного функционирования этих объектов необходимо регулярно осуществлять осмотр (мониторинг) их состояния. Для этого не всегда возможно/целесообразно привлекать человека («обходчика»). Часто для этой цели, особенно в удалённых и труднодоступных местах, используются автономные беспилотные мобильные устройства, которые для краткости назовём *дронами*. Для выполнения мониторинга дрон оснащён необходимым оборудованием (видеокамерой, сенсорами, системой позиционирования и т. п.). В подавляющем большинстве публикаций, относящихся к теме, дрон или сенсор собирает информацию в пределах некоторой области, часто круга [1–8]. При этом предполагается, что дрон покрывает эту область. Если требуется осуществить мониторинг протяжённого объекта, то это называют *покрытием барьера*, понимая под ним отрезок прямой линии. Дрон, как правило, имеет ограниченный запас энергии, что влечёт ограничение на длину пути, который он может пройти без восстановления энергии (подзарядки или заправки). В научной литературе рассматриваются задачи оптимального покрытия барьера с использованием трёх критериев: минимизация числа дронов (задача MinNum), минимизация максимальной длины пройденного дроном пути (задача MinMax) и минимизация общей длины путей дронов (задача MinSum).

В настоящей работе рассматривается задача MinSum, в то время как задачи MinNum и MinMax решены нами ранее в [9]. Во многих статьях [2, 4–6, 8, 10–12] рассматривалась задача MinSum в «традиционной» постановке, когда каждый дрон находится в некоторой начальной точке на плоскости (депо) и покрывает круг некоторого радиуса с центром в точке расположения дрона. Для каждого дрона требуется найти такое конечное положение, что барьер покрыт (каждая его точка принадлежит хотя бы одному кругу покрытия) и общая длина путей, пройденных дронами, минимальна [1–3, 5, 7, 8]. Известно, что такая задача NP-трудна, когда круги покрытия дронов разные [2, 3]. Если же круги одинаковые, то сложностной статус задачи не известен. Для последнего случая в [5] предложен алгоритм, который строит  $\sqrt{2}$ -приближённое решение задачи с трудоёмкостью  $O(m^4)$ , где  $m$  — число дронов. Позже в [8] трудоёмкость уменьшена до  $O(m^2)$ .

Ниже рассмотрим «нетрадиционную» постановку задачи MinSum, когда дрон покрывает только точку, в которой он находится. Следовательно, для покрытия части барьера дрон должен стартовать из своего депо, пройти вдоль этой части, после чего вернуться в своё депо. В результате, в евклидовой метрике траектория дрона — это треугольник, одна сторона которого находится на барьере и одна вершина — в депо. Более того, длина траектории ограничена сверху некоторой величиной, зависящей от запаса энергии.

В последнее время дроны находят применение во многих областях, при этом задача маршрутизации дронов ещё недостаточно исследована. В [13] для дронов определена задача маршрутизации и оптимизации траектории (routing and trajectory optimization problem, RTOP). Авторы ввели таксономию и осуществили обзор результатов оптимизации траекторий и статей, посвящённых маршрутизации дронов. В [14] поставлена задача маршрутизации дрона на графе (drone arc routing problem, DARP) и изучена её связь с известной задачей маршрутизации почтальона на графе (postman arc routing problems, PARP). В статье [15] предложены две задачи маршрутизации с ограничением на длину пути и изучены потенциальные преимущества от гибкого назначения начального и конечного депо для каждого дрона.

Настоящая работа продолжает исследование, начатое в [16], где доказаны некоторые свойства решения в случае, когда в каждом депо находится один дрон. Авторы на основе динамического программирования предложили эффективную реализацию для построения допустимого решения в прямоугольной метрике с трудоёмкостью  $O(m^2)$ , а также указали частные случаи, в которых построенное покрытие оптимально.

**Новизна полученных результатов.** В отличие от [16], в этой статье число дронов в каждом депо — искомая величина. В [9] число дронов в каждом депо также подлежит определению, вместе с тем общее их число неограниченно, при этом предложены алгоритмы для решения задач MinNum, MinMax и MinSum. Если предположить, что каждый дрон покрывает отрезок барьера с целочисленными границами длины не менее 1, то для задачи MinSum в [9] предложен алгоритм трудоёмкости  $O(mL^3)$ , где  $m$  — число депо, а  $L$  — длина барьера. В этом случае наиболее трудоёмкой процедурой оказывается препроцессинг, в ходе которого вычисляется минимальная суммарная длина траекторий для дронов каждого депо, покрывающих произвольный отрезок  $[a, b] \subseteq [0, L]$ .

В постановке ниже рассматриваем более общую задачу, когда число дронов ограничено сверху некоторым натуральным  $n$ . Для решения этой задачи разработаны два подхода на основе динамического программирования. Без учёта препроцессинга первый алгоритм ищет наилучшее покрытие *оптимальным* числом дронов  $n^*$  с трудоёмкостью  $O(L^2)$  — это линейная трудоёмкость от длины входной информации. Если  $n^* \leq n$ , то построенное покрытие оптимально и для случая ограниченного числа дронов. Иначе второй алгоритм с трудоёмкостью  $O(mnL^2)$  строит оптимальное покрытие. Более быстрый алгоритм удалось получить благодаря более эффективному препроцессингу трудоёмкости  $O(mnL)$ . Если каждый дрон покрывает отрезок барьера длины не менее 1, то  $n \leq L$ . В результате при неограниченном числе дронов трудоёмкость решения

задачи уменьшена в  $L$  раз, тем самым для более общего случая получен алгоритм той же трудоёмкости, что и для постановки из [9], в которой имеется ограничение на число используемых дронов.

### 1. Постановка задачи

Пусть барьер — это отрезок  $[0, L]$  на оси  $OX$ . На плоскости заданы координаты депо  $d_i = (x_i, y_i)$ ,  $y_i \geq 0$ ,  $i \in D$ ,  $|D| = m$ , в которых нужно разместить не более  $n$  дронов. Пронумеруем депо слева направо согласно их абсциссам, т. е.  $i < j$  тогда и только тогда, когда  $x_i < x_j$ . Очевидно, что при равенстве абсцисс депо с большей ординатой можно исключить. Пусть длина пути каждого дрона ограничена величиной  $q > 0$ . Для покрытия барьера траектория каждого дрона начинается в своём депо, проходит по отрезку барьера и заканчивается в своём депо. Часть барьера, вдоль которой прошёл дрон, считается *покрытой* этим дроном. Будем говорить, что дроны из депо  $i$  *покрывают* отрезок  $[a_i, b_i] \subseteq [0, L]$ , если каждый из них стартует из депо  $i$ , перемещается вдоль барьера и возвращается в своё депо, проходя путь не длиннее  $q$ , а совокупно дроны покрывают отрезок  $[a_i, b_i]$ .

**Определение 1.** *Покрывание  $S$  барьера  $[0, L]$  — это множество таких отрезков  $[a_i, b_i] \subseteq [0, L]$ , что каждый из них покрывается дронами одного депо  $i \in D$ , выполняется условие  $\bigcup_{i \in D} [a_i, b_i] \supseteq [0, L]$  и траектория каждого дрона не длиннее  $q$ .*

Для задания покрытия  $S$  необходимо определить число дронов в каждом депо, отрезок барьера  $[a_i, b_i]$ , покрываемый дронами каждого депо  $i \in D$ , а также траектории всех дронов.

**Задача MinSum.** *Требуется разместить не более  $n$  дронов в  $m$  депо так, чтобы покрыть весь барьер  $[0, L]$  и общая длина траекторий дронов была минимальной.*

Очевидно, что в евклидовой метрике оптимальная траектория каждого дрона — это треугольник с одной вершиной в депо, с одной стороной на барьере и с периметром не более  $q$ . Более того, в оптимальном покрытии, естественно, отрезки, покрываемые разными дронами, не пересекаются.

### 2. Алгоритмы

В этом разделе представим алгоритмы для решения задачи MinSum. Для проверки существования решения достаточно запустить алгоритм  $A_{\text{MinNum}}$  из [9], который определяет минимальное число дронов  $n_{\min}$ , покрывающих барьер. Решение задачи MinSum существует, если  $n_{\min} \leq n$ .

Далее нам понадобятся следующие определения и утверждения.

**Определение 2.** Покрытие, в котором для всех  $i, j \in D$  неравенство  $b_i \leq a_j$  выполняется тогда и только тогда, когда  $i < j$ , называется *покрытием, сохраняющим порядок*.

**Лемма 1** [9]. Существует оптимальное покрытие, в котором дроны из одного депо покрывают один отрезок барьера.

**Лемма 2** [9]. Для задачи MinSum существует оптимальное покрытие, сохраняющее порядок.

Пусть  $g(k, a, b)$  — минимальная суммарная длина траекторий  $k$  дронов из одного депо, покрывающих отрезок  $[a, b]$ .

**Лемма 3** [9]. Для любого фиксированного отрезка  $[a, b] \subseteq [0, L]$  функция  $g(k, a, b)$  монотонно возрастает с ростом  $k$ .

Из приведённых выше утверждений следует, что в оптимальном покрытии любой отрезок барьера покрывается *минимальным* количеством дронов *одного* депо. Для определения минимального числа дронов  $n_i(a, b)$  из депо  $i$ , покрывающих отрезок  $[a, b]$ , достаточно учесть, что все дроны, кроме, возможно, одного, проходят путь максимальной длины  $q$ .

В [9] для  $i \in D$  введена функция  $f_i(a, b) \geq 0$ , равная минимальной суммарной длине траекторий  $n_i(a, b)$  дронов из депо  $i$ , покрывающих отрезок  $[a, b]$  (при этом по определению  $f_i(a, a) = 0$ ). Предположим, что нам известны значения функций  $f_i(a, b)$  и  $n_i(a, b)$  для всех  $i = 1, \dots, t$  и целых  $a = 0, 1, \dots, L-1$  и  $b = a+1, \dots, L$ . Введём новую функцию

$$f(a, b) = \min_{i \in D} f_i(a, b) = f_{k(a,b)}(a, b).$$

Существует депо с номером  $k(a, b) \in D$  такое, что  $n_{k(a,b)}(a, b)$  дронов из этого депо покрывают отрезок  $[a, b]$ , проходя совместно минимальное расстояние. Таким образом, любое разбиение барьера на отрезки однозначно определяет покрытие. Следовательно, необходимо решить следующую задачу оптимального разбиения отрезка на подотрезки:

$$\sum_{i=1}^p f(z_{i-1}, z_i) \rightarrow \min_{z, p}, \quad (1)$$

$$0 = z_0 < z_1 < \dots < z_p = L, \quad (2)$$

где неизвестное  $p$  — это число подотрезков, т. е. требуется оптимально разбить барьер на оптимальное число подотрезков.

Задача (1), (2) хорошо исследована. Метод динамического программирования для её решения стал известен не позже 1970-х гг. (см., например, монографию [17]). Позже алгоритм был несколько раз переоткрыт [18–20]. Для описания алгоритма — назовём его  $A_1$  — предположим,

что  $L$  и параметр  $l \in [0, L]$  целые. Обозначим через  $S(l)$  минимальную длину траекторий дронов, покрывающих отрезок  $[0, l]$ ,  $l = 0, 1, \dots, L$ .

**2.1. Алгоритм  $A_1$ .** Приведём схему алгоритма динамического программирования с одним прямым и одним обратным ходом. Во время выполнения прямого хода для каждого значения параметра  $l = 0, 1, \dots, L$  найдём такую самую правую точку разбиения  $z(l) < l$ , что общая длина траекторий дронов, покрывающих отрезок  $[0, l]$ , минимальна.

Прямой ход. Воспользуемся рекуррентными соотношениями

$$\begin{aligned} S(0) &= 0, \\ S(l) &= \min_{0 \leq z < l} \{f(z, l) + S(z)\} = f(z(l), l) + S(z(l)), \quad l = 1, \dots, L, \end{aligned} \quad (3)$$

запоминая условно оптимальную точку разбиения  $z(l) < l$  для каждого  $l = 1, \dots, L$ .

ОБРАТНЫЙ ХОД. На этом этапе определим число точек разбиения  $p^*$  в оптимальном решении, а также само оптимальное разбиение. Положим сначала оптимальное число дронов, участвующих в покрытии, равным  $n^* = 0$ , а число точек разбиения равным  $p^* = 0$ . После завершения работы прямого хода при  $l = L$  были найдены оптимальное значение целевой функции  $S^* = S(L)$  (1), а также последняя (правая) точка разбиения  $z(L)$ . Значит, отрезок  $[z(L), L]$  покрывается  $n_{k(z(L), L)}(z(L), L)$  дронами из депо  $k(z(L), L)$ , и мы полагаем  $p^* = p^* + 1$  и  $n^* = n^* + n_{k(z(L), L)}(z(L), L)$ . Теперь правый конец барьера равен  $l = z(L)$ . Так как во время выполнения прямого хода запомнили  $z(l)$  для всех  $l = 1, \dots, L$ , мы знаем последнюю (правую) точку разбиения отрезка  $[0, l]$ , которая является предпоследней точкой разбиения барьера  $[0, L]$ . Если  $z(l) > 0$ , то полагаем  $p^* = p^* + 1$ ,  $n^* = n^* + n_{k(z(l), l)}(z(l), l)$ ,  $l = z(l)$  и продолжаем построение оптимального разбиения, пока не получим  $z(l) = 0$ . В результате будет найдено оптимальное разбиение барьера  $z^*$  на  $p^*$  частей, а также оптимальное число дронов  $n^*$ , участвующих в покрытии.

**Лемма 4.** Трудоемкость алгоритма  $A_1$  равна  $O(L^2)$ .

ДОКАЗАТЕЛЬСТВО. Очевидно, что соотношения (3) справедливы также и для действительных значений переменных. Однако в этом случае получим континуум значений функций. Если же параметры и переменные целые, то при выполнении прямого хода для каждого значения  $l$  находится последняя точка разбиения  $z(l) < l$  отрезка  $[0, l]$ . Так как  $l = 0, 1, \dots, L$ , трудоемкость прямого хода равна  $O(L^2)$ . Обратный ход менее трудоёмкий. Поскольку условно оптимальные точки разбиения  $z(l) < l$  хранятся для всех  $l = 1, \dots, L$ , для восстановления оптимального разбиения достаточно проделать  $O(L)$  операций. Лемма 4 доказана.

**Замечание 1.** Трудоёмкость алгоритма  $A_1$  линейно зависит от длины входной информации  $O(L^2)$ , которая определяется числом значений функции  $f(a, b)$ . Однако в терминах входа исходной задачи алгоритм псевдополиномиальный. В любом случае трудоёмкость алгоритма  $A_1$  в  $m$  раз меньше трудоёмкости алгоритма  $A_{\text{MinSum}}$ , предложенного в [9] для решения такой же задачи.

Пока ничего не сказано о трудоёмкости вычисления значений функций  $f_i(a, b)$ . Менее трудоёмкий способ по сравнению с процедурой, описанной в [9], будет предложен ниже в соответствующем разделе.

Если  $n^* \leq n$ , то задача MinSum решена. Иначе необходимо применить следующий алгоритм  $A_2$ .

**2.2. Алгоритм  $A_2$ .** Описываемый ниже алгоритм применяется в случае, когда оптимальное число дронов  $n^*$ , полученное в результате работы алгоритма  $A_1$ , окажется больше  $n$ . Для построения решения в этом случае предлагается следующий алгоритм двухпараметрического динамического программирования  $A_2$ . Пусть функция  $S_i(l, N)$  равна минимальной длине траекторий  $N$  дронов из депо  $1, \dots, i$ , которые покрывают отрезок  $[0, l]$ . Из леммы 3 следует, что если отрезок может быть покрыт дронами из одного депо, то он покрывается минимальным числом дронов. Следовательно, если разбиение барьера известно, то число дронов, участвующих в покрытии, определяется однозначно. Заметим, что число дронов, покрывающих барьер, может оказаться меньше  $n$ . Обозначим через  $N_i(l, N) \leq N$  количество дронов в первых  $i$  депо, где параметр  $N = 0, 1, \dots, n$  — это максимальное число дронов, которые можно разместить в депо  $1, \dots, i$ .

В решении, построенном алгоритмом  $A_2$ , барьер разбивается на  $m$  частей. При этом некоторые точки разбиения  $0 = z_0 \leq z_1 \leq \dots \leq z_m = L$  могут совпадать, т. е. в некоторых депо может не быть ни одного дрона. Это значит, что дроны такого депо покрывают пустой отрезок.

**ПРЯМОЙ ХОД.** Для всех  $i = 1, \dots, m$ ,  $l = 0, 1, \dots, L$  и  $N = 0, 1, \dots, n$  вычислим

$$S_1(l, N) = \begin{cases} f_1(0, l), & \text{если } N_1(l, N) = n_1(0, l) \leq N, \\ +\infty & \text{иначе,} \end{cases} \quad (4)$$

$$\begin{aligned} S_i(l, N) &= \min_{z=0, \dots, l} \{f_i(z, l) + S_{i-1}(z, N - n_i(z, l))\} = \\ &= f_i(z_i(l, N), l) + S_{i-1}(z_i(l, N), N - n_i(z_i(l, N), l)), \end{aligned} \quad (5)$$

если  $N_i(l, N) = n_i(z_i(l, N), l) + N_{i-1}(z_i(l, N), N - n_i(z_i(l, N), l)) \leq N$ , и положим  $S_i(l, N) = +\infty$  в противном случае.

Запомним  $z_i(l, N)$ ,  $i = 1, \dots, m$ ,  $l = 0, 1, \dots, L$ ,  $N = 1, \dots, n$ .

**ОБРАТНЫЙ ХОД.** По завершении прямого хода оказывается найденным оптимальное значение функционала  $S^* = S_m(L, n)$ , количество дронов  $N^* = N_m(L, n)$  и последняя (правая) точка разбиения  $z_{m-1}^* = z(l)$ . В результате найдём отрезок  $[z_{m-1}^*, L]$  (который может быть пустым), покрытый  $n_m(z_{m-1}^*, L)$  дронами депо  $m$ . Это значит, что отрезок  $[0, z_{m-1}^*]$  покрывается дронами депо  $1, \dots, m-1$ . Положим  $l = z_{m-1}^*$ . При выполнении прямого хода найдены условно оптимальные точки разбиения  $z_i(l, N)$  для всех целочисленных значений параметров. Следовательно, следующую справа точку оптимального разбиения можем определить равенством  $z_{m-2}^* = z(l, N^* - n_m(z_{m-1}^*, L))$ . Продолжая процесс, найдём оптимальное разбиение всего барьера на  $m$  частей.

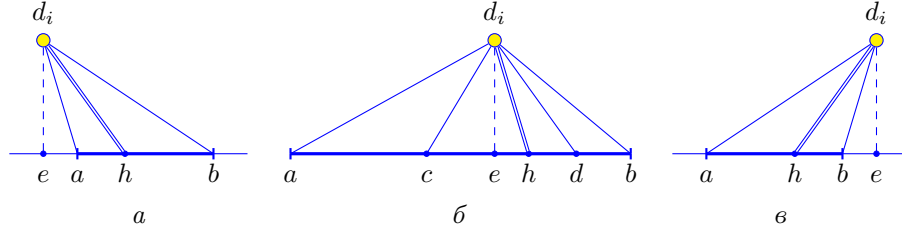
**Лемма 5.** Трудоёмкость алгоритма  $A_2$  равна  $O(mnL^2)$ .

**ДОКАЗАТЕЛЬСТВО.** Рекуррентные соотношения (4), (5) справедливы для произвольных значений параметров. Однако для реализуемости алгоритма полагаем их целыми. Тогда при выполнении прямого хода для любых  $i = 1, \dots, m$ ,  $l = 0, 1, \dots, L$  и  $N \leq n$  значение  $S_i(l, N)$  находится путём перебора целых  $l$  и  $N$ . Следовательно, трудоёмкость прямого хода равна  $O(mnL^2)$ . Для восстановления оптимального решения на обратном ходе выполняется  $m$  шагов. На каждом шаге определяется одна точка разбиения. В результате трудоёмкость обратного хода равна  $O(m)$ , а следовательно, трудоёмкость алгоритма  $A_2$  равна  $O(mnL^2)$ . Лемма 5 доказана.

**Замечание 2.** На вход алгоритма  $A_2$  подаются значения функций  $f_i(a, b)$ , которых  $O(mL^2)$ . Трудоёмкость алгоритма в  $n$  раз больше длины входа, а  $n$  — это число. Следовательно, даже без учёта трудоёмкости вычисления значений функций  $f_i(a, b)$ , алгоритм псевдополиномиальный, но с учётом неравенства  $n \leq L$  трудоёмкость алгоритма  $A_2$  не превосходит трудоёмкости алгоритма из [9] для решения более общей задачи.

### 3. Вычисление функций $n_i(a, b)$ и $f_i(a, b)$

Для реализации описанных выше алгоритмов необходимо знать значения функций  $f_i(a, b)$  и  $n_i(a, b)$  для всех  $i = 1, \dots, m$ ,  $a = 0, 1, \dots, L-1$  и  $b = a+1, \dots, L$ , т. е. в общей сложности  $O(mL^2)$  значений. Нахождение этих значений — препроцессинг, который необходимо выполнить до запуска алгоритмов  $A_1$  и  $A_2$  и который во многом определяет трудоёмкость решения задачи. В [9] предложен алгоритм для нахождения значений  $f_i(a, b)$  с трудоёмкостью  $O(mnL^2)$ . Покажем, что вычисление значений функций  $f_i(a, b)$  и  $n_i(a, b)$  для каждого депо  $i \in D$  можно осуществить с трудоёмкостью  $O(nL)$ .

Рис. 1. Иллюстрация вычисления  $f_i(a, b)$ 

Сначала вычислим  $n_i(a, b)$ . Так как  $O(n)$  дронов могут участвовать в покрытии отрезка, для каждой пары значений  $a$  и  $b$  нужно найти  $n_i(a, b)$ , для чего потребуется  $O(nL^2)$  операций. Для уменьшения трудоёмкости мы предлагаем вычислять  $n_i(a, b)$  по-другому. Для каждого значения  $a \in \{0, 1, \dots, L-1\}$  будем искать отрезок максимальной длины  $[a, b(n_i)]$ , который покрывается  $n_i \in \{1, \dots, n\}$  дронами. Поскольку в оптимальном решении для покрытия отрезка  $[a, b]$  используется минимальное число дронов из одного депо, можно определить  $n_i(a, b)$  для всех  $b > a$ . Для этого положим  $n_i(a, b) = n_i$ ,  $b \in (b(n_i - 1), b(n_i)]$ .

Первым найдём отрезок  $[a, b(1)]$ , который покрывает один дрон с максимальной длиной траектории  $q$ . Покрытый отрезок содержит некоторое число целых  $b \in (a, b(1)]$ . Зная  $b(1)$ , найдём максимальный отрезок  $[b(1), b(2)]$ , который покрывает второй дрон. Добавляя дроны, определим минимальное число дронов, покрывающих отрезок  $[a, b]$  для всех целых  $b = a + 1, \dots, L$ . Следовательно, для вычисления  $n_i(a, b)$  для всех  $i = 1, \dots, m$ ,  $a = 0, \dots, L-1$  и  $b = 1, \dots, L$  потребуется  $O(Lnm)$  операций.

Зная  $n_i(a, b)$ , можно вычислить значения  $f_i(a, b)$ . Для каждого депо  $d_i = (x, y)$  существует три варианта расположения относительно отрезка  $[a, b]$ : (а)  $x < a$ , (б)  $x \in [a, b]$  и (в)  $x > b$  (рис. 1). В случаях (а) и (в) каждый из  $n_i(a, b) - 1$  дронов проходит путь максимальной длины  $q$ , покрывая вместе отрезок  $[h, b]$  в случае (а) и отрезок  $[a, h]$  в случае (в), а один дрон покрывает отрезок  $[a, h]$ , двигаясь по ребру треугольника  $ad_ih$  в случае (а) (рис. 1а), и отрезок  $[h, b]$ , двигаясь по ребру треугольника  $hd_ib$ , в случае (в) (рис. 1в).

В случае (б) (рис. 1б) отрезки  $[a, c]$  и  $[d, b]$  покрываются  $n_i(a, c) + n_i(d, b)$  дронами, каждый из которых проходит путь длины  $q$ . Если отрезок  $[c, d]$  может покрыть один дрон, то он его покрывает. Иначе отрезок  $[c, d]$  будет покрыт двумя дронами так, что линия  $hd_i$  имеет минимальный наклон (т. е. она находится максимально близко к перпендикуляру  $d_ie$ ). Точку  $h$  легко найти. Если один дрон может покрыть отрезок  $[c, e]$ , а другой — отрезок  $[e, d]$ , то  $d_ih$  — это перпендикуляр к барьеру, т. е.  $h = e$ . Иначе один из этих отрезков не может быть покрыт одним дроном.

Пусть это отрезок  $[e, d]$ . Тогда один дрон покрывает отрезок максимальной длины  $[h, d]$ , а другой — оставшийся отрезок  $[c, h]$ .

В результате получаем, что трудоёмкость вычисления значений функций  $n_i(a, b)$  и  $f_i(a, b)$  для всех  $i \in D$ ,  $a$  и  $b$  равна  $O(Lnm)$ . Если число дронов неограниченно, то при естественном предположении, что каждый дрон покрывает отрезок длины не менее 1, трудоёмкость предложенного алгоритма равна  $O(mL^2)$ .

**3.1. Пример.** Рассмотрим пример на рис. 2, в котором число депо равно  $m = 3$ , число дронов не превосходит  $n = 3$ , барьер имеет длину  $L = 156$ , длина траектории каждого дрона не превосходит  $q = 140$  и местоположение депо  $d_1 = (18, 10)$ ,  $d_2 = (78, 10\sqrt{35})$  ( $|dd_2| = |d_2h| = 60$ ),  $d_3 = (138, 10)$ .

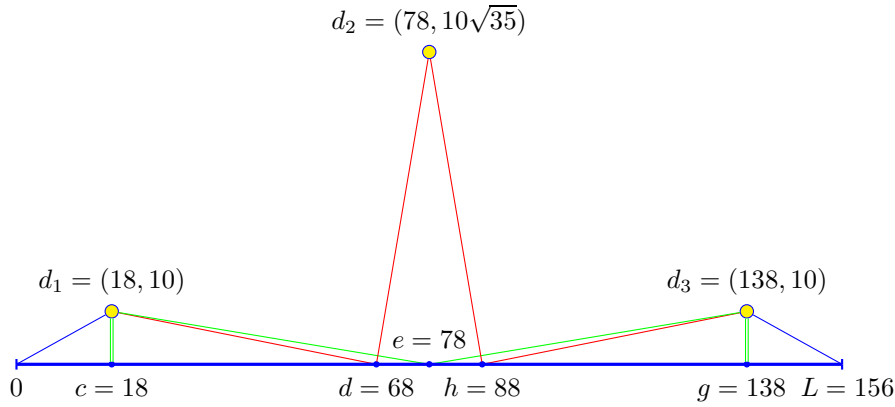


Рис. 2. Иллюстрация к примеру

Так как всего три депо, барьер будет разбит на 3 или менее частей, и каждая часть будет покрыта дронами из одного депо. Поскольку мы ищем покрытие, сохраняющее порядок, дроны из депо 1 должны покрыть левый отрезок  $[0, x]$ , а дроны из депо 3 должны покрыть правый отрезок  $[y, L]$ . При этом, конечно, может получиться, что в каких-то депо не будет размещено ни одного дрона.

Сначала найдём минимальное число дронов  $n_i(a, b) \leq n = 3$  в каждом депо  $i = 1, 2, 3$ , покрывающих отрезок  $[a, b]$ ,  $a = 0, 1, \dots, 155$ ,  $b = a + 1, \dots, 156$ . Напомним, что один дрон покрывает отрезок длины не менее 1. Дроны из депо 1 максимально могут покрыть отрезок  $[0, e]$  при  $e = 78$ . В то же время один дрон из депо 1 покрывает отрезок  $[0, b]$ , если  $b \leq d = 68$ , а для покрытия отрезка  $[0, b]$  при  $b \in [d + 1, e]$  необходимо два дрона в депо 1. Значит,  $n_1(0, d) = 1$ ,  $n_1(0, b) = 2$ ,  $b = d + 1, \dots, e$ , и  $n_1(0, b) = +\infty$ , когда  $b > e$ . Далее определим  $n_2(a, b)$  для  $a \in [41, 114]$

и всех допустимых целых  $b > a$ . В частности, имеем  $n_2(d, h) = 1$ . Так как в покрытии, сохраняющем порядок, дроны из депо 3 должны покрывать отрезок с правой границей  $L$ , нам требуются значения  $n_3(a, L)$ . Депо 3 расположено симметрично относительно депо 1, поэтому  $n_3(a, L) = +\infty$  для  $a < e$ ,  $n_3(a, L) = 2$  для  $a \in [e + 1, h - 1]$  и  $n_3(a, L) = 1$  для  $a \geq h$ .

Затем найдём минимальную длину траекторий  $f_i(a, b)$  дронов из каждого депо  $i = 1, 2, 3$ , покрывающих отрезок  $[a, b]$ . Из депо 1 будет покрыт некоторый отрезок  $[0, b]$ . Пока  $n_1(0, b) = 1$ , длина пути одного дрона  $f_1(0, b)$  равна периметру треугольника  $0d_1b$ . Если  $n_1(0, b) = 2$ , то для минимизации суммарной длины путей двух дронов нужно, чтобы траекториями обоих дронов были прямоугольные треугольники с общей стороной  $d_1c$  — перпендикуляром к  $[0, L]$ . Тогда длина путей двух дронов равна

$$f_1(0, b) = |0d_1| + 2|d_1c| + |d_1b| + |0b|.$$

Так как отрезок с правым концом  $b > d$  дроны из депо 1 покрыть не могут, положим  $f_1(0, b) = +\infty$ ,  $b > d$ . Далее вычислим  $f_2(a, b)$  для всех допустимых  $a \in [41, 114]$  и  $b > a$ . В частности, получим  $f_2(d, h) = q = 140$ . Вместе с тем для  $a \in [e, h - 1]$  находим

$$f_3(a, L) = |ad_3| + 2|d_3g| + |d_3L| + |aL|,$$

а для  $a \in [h, L - 1]$  —

$$f_3(a, L) = |hd_3| + 2|d_3g| + |d_3L| + |aL|.$$

Наконец, найдём значения функции  $f(a, b) = \min_{i=1,2,3} f_i(a, b)$  для всех целых  $a \in [0, L - 1]$  и  $b \in [a + 1, L]$ . Получим  $f(a, b) = f_1(a, b)$  для  $b \leq e$  и  $f(a, b) = f_3(a, b)$  для  $a \geq e$ .

Применим алгоритм  $A_1$ . Получим решение, в котором в депо 1 и 3 размещено по 2 дрона. Дроны из депо 1 покрывают отрезок  $[0, e]$ , а дроны депо 2 — отрезок  $[e, L]$ , а общая длина траекторий дронов равна 340. Однако количество использованных дронов больше максимального их числа  $n = 3$ . Следовательно, построенное алгоритмом  $A_1$  покрытие недопустимо, и нужно применить алгоритм  $A_2$ .

Алгоритм  $A_2$  учитывает ограничение на общее число дронов. В нашем примере ни одно депо не может иметь более 1 дрона. В противном случае барьер не будет покрыт тремя дронами. Из-за симметрии крайних депо один дрон из депо 1 покроет отрезок  $[0, d]$ , один дрон из депо 3 покроет отрезок  $[h, L]$ , а дрон из депо 2 покроет отрезок  $[d, h]$ . При этом общая длина траекторий дронов равна 420.

### Заключение

В статье рассмотрена задача покрытия барьера, представляющего собой отрезок  $[0, L]$  на плоскости. Покрытие осуществляется одинаковыми мобильными устройствами (дронами) с ограниченным запасом энергии, который определяет максимальную длину  $q$  траектории дрона. Каждый дрон стартует из своего депо, движется вдоль отрезка барьера длины не менее 1 и возвращается в своё депо, проделав путь длины не более  $q$ . Отрезок барьера, вдоль которого двигался дрон, считается покрытым этим дроном. В задаче MinSum требуется разместить не более  $n$  дронов в  $m$  депо и определить траекторию каждого дрона так, чтобы покрыть весь барьер, а общая длина траекторий дронов была минимальна.

Ранее в [16] рассматривалась задача MinSum, в которой в каждом депо находился один дрон и оставалось определить траекторию каждого дрона. Для такой постановки задачи не всегда существует оптимальное покрытие, сохраняющее порядок, поэтому динамическое программирование не всегда строит оптимальное решение. Если число дронов в каждом депо — это искомая величина (переменная), то всегда существует оптимальное покрытие, сохраняющее порядок [9]. В [9] предложен алгоритм динамического программирования, который решает задачу с трудоёмкостью  $O(mL^3)$ , если общее число дронов неограниченно.

В настоящей статье предложены два новых алгоритма  $A_1$  и  $A_2$ . Алгоритм  $A_1$  находит оптимальное покрытие оптимальным числом дронов  $n^*$  с трудоёмкостью  $O(L^2)$ . Если  $n^* \leq n$ , то это также решение для рассматриваемой задачи MinSum. Если  $n^* > n$ , то с трудоёмкостью  $O(nmL^2)$  алгоритм  $A_2$  строит оптимальное решение. Так как каждый дрон покрывает отрезок длины не менее 1, то  $n \leq L$ . Следовательно, трудоёмкость алгоритма  $A_2$  не превосходит трудоёмкости алгоритма, предложенного в [9], при этом алгоритм  $A_2$  решает более общую задачу.

Для описанных алгоритмов входом являются функции  $n_i(a, b)$  — минимальное число дронов и  $f_i(a, b)$  — минимальная суммарная длина траекторий дронов из депо  $i$ , покрывающих отрезок  $[a, b]$ . Также для алгоритма  $A_1$  вычисляется функция  $f(a, b) = \min_{i=1, \dots, m} f_i(a, b)$ . В результате

длина входа алгоритма  $A_1$  равна  $O(L^2)$ , следовательно, алгоритм имеет линейную трудоёмкость от длины входа. Однако вычисление значений функций  $n_i(a, b)$  и  $f_i(a, b)$ , т. е. препроцессинг достаточно трудоёмкий. В [9] не рассматривались функции  $n_i(a, b)$  и все  $O(mL^2)$  значений функций  $f_i(a, b)$  были найдены с трудоёмкостью  $O(mL^3)$ . В этой статье нам удалось уменьшить трудоёмкость вычисления значений функций  $n_i(a, b)$  и  $f_i(a, b)$  до  $O(nmL) \leq O(mL^2)$ .

В этой статье, как и в [9], для определения трудоёмкости рассмотрен случай, когда дроны покрывают отрезок  $[a, b]$  при целых  $a$  и  $b$ , а один

дрон покрывает отрезок длины не менее 1. Конечно, длина отрезка, покрываемого одним дроном, может быть произвольным числом  $\delta > 0$ . Если  $\delta$  рациональное, то умножение на знаменатель приведёт к нужным свойствам. Однако в оптимальном решении дроны одного депо могут покрыть отрезок с нецелыми концевыми точками, и тогда алгоритмы  $A_1$  и  $A_2$  построят приближённое решение. В дальнейшем планируется оценить точность решения, построенного с использованием шага  $\delta$ , и разработать FPTAS.

### Финансирование работы

Исследование выполнено в рамках государственного задания Института математики им. С. Л. Соболева (проект № FWNF-2022-0019). Дополнительных грантов на проведение или руководство этим исследованием получено не было.

### Конфликт интересов

Авторы заявляют, что у них нет конфликта интересов.

### Литература

1. **Bhattacharya B., Burmester M., Hu Y., Kranakis E.** Optimal movement of mobile sensors for barrier coverage of a planar region // *Theor. Comput. Sci.* 2009. V. 410, No. 52. P. 5515–5528. DOI: 10.1007/978-3-540-85097-7\_10.
2. **Czyzowicz J., Kranakis E., Krizanc D.** [et al.]. On minimizing the sum of sensor movements for barrier coverage of a line segment // *Ad-hoc, mobile and wireless networks. Proc. 9th Int. Conf. (Edmonton, Canada, Aug. 20–22, 2010)*. Heidelberg: Springer, 2010. P. 29–42. (Lect. Notes Comput. Sci.; V. 6288). DOI: 10.1007/978-3-642-14785-2\_3.
3. **Dobrev S., Durocher S., Eftekhari M.** [et al.]. Complexity of barrier coverage with relocatable sensors in the plane // *Theor. Comput. Sci.* 2015. V. 579. P. 64–73. DOI: 10.1016/j.tcs.2015.02.006.
4. **Wang H., Zhang X.** Minimizing the maximum moving cost of interval coverage // *Algorithms and computation. Proc. 26th Int. Symp. (Nagoya, Japan, Dec. 9–11, 2015)*. Heidelberg: Springer, 2015. P. 188–198. (Lect. Notes Comput. Sci.; V. 9472). DOI: 10.1007/978-3-662-48971-0\_17.
5. **Cherry A., Gudmundsson J., Mestre J.** Barrier coverage with uniform radii in 2D // *Algorithms for sensor systems. Rev. Sel. Pap. 13th Int. Symp. Algorithms and Experiments for Wireless Sensor Networks (Vienna, Austria, Sept. 4–8, 2017)*. Cham: Springer, 2017. P. 57–69. (Lect. Notes Comput. Sci.; V. 10718). DOI: 10.1007/978-3-319-72751-6\_5.
6. **Erzin A. I., Lagutkina N. A.** FPTAS for barrier covering problem with equal circles in 2D // *Optim. Lett.* 2021. V. 15, No. 4. P. 1397–1406. DOI: 10.1007/s11590-020-01650-8.

7. **Erzin A. I., Lagutkina N. A., Ioramishvili N.** Barrier covering in 2D using mobile sensors with circular coverage areas // Learning and intelligent optimization. Rev. Sel. Pap. 13th Int. Conf. (Chania, Greece, May 27–31, 2019). Cham: Springer, 2020. P. 342–354. (Lect. Notes Comput. Sci.; V. 11968). DOI: 10.1007/978-3-030-38629-0\_28.
8. **Erzin A. I., Lagutkina N. A.** Barrier coverage problem in 2D // Algorithms for sensor systems. Rev. Sel. Pap. 14th Int. Symp. Algorithms and Experiments for Wireless Sensor Networks (Helsinki, Finland, Aug. 23–24, 2018). Cham: Springer, 2019. P. 118–130. (Lect. Notes Comput. Sci.; V. 11410). DOI: 10.1007/978-3-030-14094-6\_8.
9. **Erzin A. I., Shadrina A. V.** Optimal placement of mobile sensors for the distance-constrained line routing problem // Mathematical optimization theory and operations research: Recent trends. Rev. Sel. Pap. 23th Int. Conf. (Omsk, Russia, June 30–July 6, 2024). Cham: Springer, 2024. P. 172–184. (Commun. Comput. Inf. Sci.; V. 2239). DOI: 10.1007/978-3-031-73365-9\_12.
10. **Bar-Noy A., Rawitz D., Terlecky P.** «Green» barrier coverage with mobile sensors // Algorithms and complexity. Proc. 9th Int. Conf. (Paris, France, May 20–22, 2015). Cham: Springer, 2015. P. 33–46. (Lect. Notes Comput. Sci.; V. 9079). DOI: 10.1007/978-3-319-18173-8\_2.
11. **Benkoczi R., Friggstad Z., Gaur D., Thom M.** Minimizing total sensor movement for barrier coverage by non-uniform sensors on a line // Algorithms for sensor systems. Rev. Sel. Pap. 11th Int. Symp. Algorithms and Experiments for Wireless Sensor Networks (Patras, Greece, Sept. 17–18, 2015). Cham: Springer, 2015. P. 98–111. (Lect. Notes Comput. Sci.; V. 9536). DOI: 10.1007/978-3-319-28472-9\_8.
12. **Andrews A. M., Wang H.** Minimizing the aggregate movements for interval coverage // Algorithmica. 2017. V. 78, No. 1. P. 47–85. DOI: 10.1007/s00453-016-0153-8.
13. **Coutinho W. P., Battarra M., Fliege J.** The unmanned aerial vehicle routing and trajectory optimisation problem, a taxonomic review // Comput. Ind. Eng. 2018. V. 120. P. 116–128. DOI: 10.1016/j.cie.2018.04.037.
14. **Campbell J. F., Corberan A., Plana I., Sanchis J. M.** Drone arc routing problems // Networks. 2018. V. 72, No. 4. P. 543–559. DOI: 10.1002/net.21858.
15. **Kek A. G. H., Cheu R. L., Meng Q.** Distance-constrained capacitated vehicle routing problems with flexible assignment of start and end depots // Math. Comput. Model. 2008. V. 47. P. 140–152. DOI: 10.1016/j.mcm.2007.02.007.
16. **Erzin A. I., Plotnikov R. V.** Distance-constrained line routing problem // Optimization and applications. Rev. Sel. Pap. 10th Int. Conf. (Petrovac, Montenegro, Sep. 30–Oct. 4, 2019). Cham: Springer, 2020. P. 43–55. (Commun. Comput. Inf. Sci.; V. 1145). DOI: 10.1007/978-3-030-38603-0\_4.
17. **Береснев В. Л., Гимади Э. Х., Дементьев В. Т.** Экстремальные задачи стандартизации. Новосибирск: Наука, 1978. 334 с.

18. **Vidal R. V. V.** Optimal partition of an interval—The discrete version // Applied simulated annealing. Heidelberg: Springer, 1993. P. 291–312. (Lect. Notes Econ. Math. Syst.; V. 396). DOI: 10.1007/978-3-642-46787-5\_15.
19. **Jackson B., Scargle J. D., Barnes D.** [et al.]. An algorithm for optimal partitioning of data on an interval // IEEE Signal Proces. Lett. 2005. V. 12, No. 2. P. 105–108. DOI: 10.1109/LSP.2001.838216.
20. **Tilli P., Zucco D.** Optimal partitioning of an interval and applications to Sturm–Liouville eigenvalues // J. Differ. Equ. 2019. V. 268, No. 8. P. 4900–4919. DOI: 10.1016/j.jde.2019.12.026.

*Ерзин Адиль Ильясович*  
*Шадрина Анжела Викторовна*

Статья поступила  
10 февраля 2025 г.  
После доработки —  
11 марта 2025 г.  
Принята к публикации  
22 апреля 2025 г.

## DRONE PLACEMENT FOR OPTIMAL BARRIER COVERAGE

*A. I. Erzin*<sup>1, a</sup> and *A. V. Shadrina*<sup>2, b</sup><sup>1</sup> Sobolev Institute of Mathematics,  
4 Acad. Koptug Avenue, 630090 Novosibirsk, Russia<sup>2</sup> Novosibirsk State University,  
2 Pirogov Street, 630090 Novosibirsk, RussiaE-mail: <sup>a</sup>adilerzin@math.nsc.ru, <sup>b</sup>shadrinanzhela@gmail.com

**Abstract.** On the plane, a straight line segment (*barrier*) is specified, as well as the location of depots in which mobile devices (sensors or drones) are to be placed. Each drone starts from and returns to its depot and is able to travel along the barrier passing through a limited-length path. The part of the barrier along which a drone moved is *covered* by this drone. The barrier is covered if each point of it is covered by at least one drone. It is necessary to place some number of drones in each depot in order to cover the entire barrier and minimize the number of drones used (MinNum), or to minimize the total length of paths travelled by drones (MinSum), or to minimize the maximum distance traveled by a drone (MinMax).

Previously, the authors investigated a similar problem with an unlimited number of drones and, for its solution, proposed a pseudo-polynomial algorithm depending on the length of the barrier  $L$ . In this paper, a generalized problem with a limited number of drones is considered and, to construct an optimal solution, we propose an algorithm with the same complexity. However, in the case of an unlimited number of drones, the new algorithm has complexity  $L$  times less than the previous one. Illustr. 2, bibliogr. 20.

**Keywords:** barrier covering, linear routing, mobile device (drone), limited energy, optimization.

## References

1. **B. Bhattacharya, M. Burmester, Y. Hu, and E. Kranakis**, Optimal movement of mobile sensors for barrier coverage of a planar region, *Theor. Comput. Sci.* **410** (52), 5515–5528 (2009), DOI: 10.1007/978-3-540-85097-7\_10.

---

English transl.: *Journal of Applied and Industrial Mathematics* **19** (2) (2025).

2. **J. Czyzowicz, E. Kranakis, D. Krizanc**, [et al.], On minimizing the sum of sensor movements for barrier coverage of a line segment, in *Ad-Hoc, Mobile and Wireless Networks* (Proc. 9th Int. Conf., Edmonton, Canada, Aug. 20–22, 2010) (Springer, Heidelberg, 2010), pp. 29–42 (Lect. Notes Comput. Sci., Vol. 6288), DOI: 10.1007/978-3-642-14785-2\_3.
3. **S. Dobrev, S. Durocher, M. Eftekhari**, [et al.], Complexity of barrier coverage with relocatable sensors in the plane, *Theor. Comput. Sci.* **579**, 64–73 (2015), DOI: 10.1016/j.tcs.2015.02.006.
4. **H. Wang and X. Zhang**, Minimizing the maximum moving cost of interval coverage, in *Algorithms and Computation* (Proc. 26th Int. Symp., Nagoya, Japan, Dec. 9–11, 2015) (Springer, Heidelberg, 2015), pp. 188–198 (Lect. Notes Comput. Sci., Vol. 9472), DOI: 10.1007/978-3-662-48971-0\_17.
5. **A. Cherry, J. Gudmundsson, and J. Mestre**, Barrier coverage with uniform radii in 2D, in *Algorithms for Sensor Systems* (Rev. Sel. Pap. 13th Int. Symp. Algorithms and Experiments for Wireless Sensor Networks, Vienna, Austria, Sept. 4–8, 2017) (Springer, Cham, 2017), pp. 57–69 (Lect. Notes Comput. Sci., Vol. 10718), DOI: 10.1007/978-3-319-72751-6\_5.
6. **A. I. Erzin and N. A. Lagutkina**, FPTAS for barrier covering problem with equal circles in 2D, *Optim. Lett.* **15** (4), 1397–1406 (2021), DOI: 10.1007/s11590-020-01650-8.
7. **A. I. Erzin, N. A. Lagutkina, and N. Ioramishvili**, Barrier covering in 2D using mobile sensors with circular coverage areas, in *Learning and Intelligent Optimization* (Rev. Sel. Pap. 13th Int. Conf., Chania, Greece, May 27–31, 2019) (Springer, Cham, 2020), pp. 342–354 (Lect. Notes Comput. Sci., Vol. 11968), DOI: 10.1007/978-3-030-38629-0\_28.
8. **A. I. Erzin and N. A. Lagutkina**, Barrier coverage problem in 2D, in *Algorithms for Sensor Systems* (Rev. Sel. Pap. 14th Int. Symp. Algorithms and Experiments for Wireless Sensor Networks, Helsinki, Finland, Aug. 23–24, 2018) (Springer, Cham, 2019), pp. 118–130 (Lect. Notes Comput. Sci., Vol. 11410), DOI: 10.1007/978-3-030-14094-6\_8.
9. **A. I. Erzin and A. V. Shadrina**, Optimal placement of mobile sensors for the distance-constrained line routing problem, in *Mathematical Optimization Theory and Operations Research: Recent Trends* (Rev. Sel. Pap. 23th Int. Conf., Omsk, Russia, June 30–July 6, 2024) (Springer, Cham, 2024), pp. 172–184 (Commun. Comput. Inf. Sci., Vol. 2239), DOI: 10.1007/978-3-031-73365-9.
10. **A. Bar-Noy, D. Rawitz, and P. Terlecky**, “Green” barrier coverage with mobile sensors, in *Algorithms and Complexity* (Proc. 9th Int. Conf., Paris, France, May 20–22, 2015) (Springer, Cham, 2015), pp. 33–46 (Lect. Notes Comput. Sci., Vol. 9079), DOI: 10.1007/978-3-319-18173-8\_2.
11. **R. Benkoczi, Z. Friggstad, D. Gaur, and M. Thom**, Minimizing total sensor movement for barrier coverage by non-uniform sensors on a line, in *Algorithms for Sensor Systems* (Rev. Sel. Pap. 11th Int. Symp. Algorithms and Experiments for Wireless Sensor Networks, Patras, Greece, Sept. 17–18, 2015) (Springer, Cham, 2015), pp. 98–111 (Lect. Notes Comput. Sci., Vol. 9536), DOI: 10.1007/978-3-319-28472-9\_8.

12. **A. M. Andrews** and **H. Wang**, Minimizing the aggregate movements for interval coverage, *Algorithmica* **78** (1), 47–85 (2017), DOI: 10.1007/s00453-016-0153-8.
13. **W. P. Coutinho**, **M. Battarra**, and **J. Fliege**, The unmanned aerial vehicle routing and trajectory optimisation problem, a taxonomic review, *Comput. Ind. Eng.* **120**, 116–128 (2018), DOI: 10.1016/j.cie.2018.04.037.
14. **J. F. Campbell**, **A. Corberan**, **I. Plana**, and **J. M. Sanchis**, Drone arc routing problems, *Networks* **72** (4), 543–559 (2018), DOI: 10.1002/net.21858.
15. **A. G. H. Kek**, **R. L. Cheu**, and **Q. Meng**, Distance-constrained capacitated vehicle routing problems with flexible assignment of start and end depots, *Math. Comput. Model.* **47**, 140–152 (2008), DOI: 10.1016/j.mcm.2007.02.007.
16. **A. I. Erzin** and **R. V. Plotnikov**, Distance-constrained line routing problem, in *Optimization and Applications* (Rev. Sel. Pap. 10th Int. Conf., Petrovac, Montenegro, Sep. 30–Oct. 4, 2019) (Springer, Cham, 2020), pp. 43–55 (Commun. Comput. Inf. Sci., Vol. 1145), DOI: 10.1007/978-3-030-38603-0\_4.
17. **V. L. Beresnev**, **Eh. Kh. Gimadi**, and **V. T. Dementyev**, *Extremal Problems of Standardization* (Nauka, Novosibirsk, 1978) [Russian].
18. **R. V. V. Vidal**, Optimal partition of an interval—The discrete version, in *Applied Simulated Annealing*, (Springer, Heidelberg, 1993), pp. 291–312 (Lect. Notes Econ. Math. Syst., Vol. 396), DOI: 10.1007/978-3-642-46787-5\_15.
19. **B. Jackson**, **J. D. Scargle**, **D. Barnes**, [et al.], An algorithm for optimal partitioning of data on an interval, *IEEE Signal Proces. Lett.* **12** (2), 105–108 (2005), DOI: 10.1109/LSP.2001.838216.
20. **P. Tilli** and **D. Zucco**, Optimal partitioning of an interval and applications to Sturm–Liouville eigenvalues, *J. Differ. Equ.* **268** (8), 4900–4919 (2019), DOI: 10.1016/j.jde.2019.12.026.

Adil I. Erzin  
Anzhela V. Shadrina

Received February 10, 2025  
Revised March 11, 2025  
Accepted April 22, 2025