

Оптимизация, системный анализ и исследование операций

© 2023 г. С.Н. МЕДВЕДЕВ, канд. физ.-мат. наук
(s_n_medvedev@mail.ru)
(Воронежский государственный университет)

ЖАДНЫЕ И АДАПТИВНЫЙ АЛГОРИТМЫ РЕШЕНИЯ ЗАДАЧИ МАРШРУТИЗАЦИИ ТРАНСПОРТНЫХ СРЕДСТВ С НЕСКОЛЬКИМИ ЦЕНТРАМИ С ЧЕРЕДОВАНИЕМ ОБЪЕКТОВ

В статье рассматривается задача маршрутизации транспортных средств с несколькими центрами с чередованием объектов. Предлагаются формальная постановка задачи с двумя типами объектов и математическая модель с двумя блоками булевых переменных. Вначале рассматривается модель без единого места сбора транспортных средств (мобильных объектов), а после вводится дополнительный специальный объект — место сбора. Показаны дополнительные ограничения, которые добавляются в математическую модель с учетом нового объекта. Отдельное внимание уделено условию отсутствия подциклов. Данное условие учитывается на основе матрицы смежности. Для предложенной задачи предлагаются жадные алгоритмы решения. Всего представлено пять алгоритмов, два из которых являются итеративными. На основе одного из рассмотренных жадных алгоритмов строится его вероятностная модификация на основе рандомизации переменных — адаптивный алгоритм. В завершение приводятся результаты вычислительного эксперимента по сравнению предложенных алгоритмов с точки зрения среднего значения целевой функции, а также времени работы. Также приводятся результаты эксперимента по настройке параметров адаптивного алгоритма.

Ключевые слова: задача маршрутизации транспортных средств, адаптивный алгоритм, жадный алгоритм.

DOI: 10.31857/S0005231023030078, EDN: ZZRSPP

1. Введение

В исследованиях, посвященных задачам маршрутизации транспортных средств (ЗМТС), можно выделить два глобальных направления: моделирование различных модификаций и дополнений задачи и разработка эффективных алгоритмов решения.

В [1] приводится обзор возможных постановок ЗМТС. Среди них автор выделяет графовую постановку, постановку в виде задачи математического целочисленного программирования, двух- и трехиндексные постановки задачи, а также в завершение приводится постановка в виде задачи теории

расписания. В ходе анализа последней постановки автор приходит к выводу о возможности применений к решению предложенной задачи эвристик, связанных с задачей коммивояжера, для определения верхней оценки числа транспортных средств. В [2] исследовалась модель зонирования для ЗМТС за счет ввода дополнительных ограничений. Данная модель позволяет определить разбиение клиентов на зоны и найти оптимальные маршруты в каждой из них. Исследование [3] посвящено непрерывной модели ЗМТС. В нем особое место уделено анализу скорости движения мобильных объектов. Статья [4] посвящена конструированию математической модели, включающей в себя ограничения многих известных постановок ЗМТС, для решения которой разработан алгоритм на основе муравьиных колоний с эволюционной стратегией.

В [5] предложен муравьиный алгоритм решения ЗМТС с адаптивным механизмом обучения с использованием градиентного спуска. Основная идея метода состоит из двух частей: классическая состоит в вычислении и отложении феромона на маршруте лучшего муравья в популяции, а вторая, модифицированная — в адаптивном обновлении матрицы феромонов с помощью градиентов. Представлены результаты расчетов, в которых разработанный авторами алгоритм на всех предложенных тестовых данных дает лучшее решение по сравнению с другими алгоритмами. В [6] предлагается генетический алгоритм решения ЗМТС с несколькими депо с временными окнами со специально разработанной техникой инициализации популяции. В [7] представлен подход, объединяющий в себе генетические алгоритмы и алгоритмы муравьиной колонии.

В [8, 9] авторами приводятся различные варианты математических моделей для ЗМТС с чередованием объектов и единым местом сбора, а также приводятся эвристические алгоритмы для ее решения, основанные на алгоритмах муравьиных колоний и генетических алгоритмах. В [10] предлагается точный алгоритм решения рассмотренной задачи, основанный на методе ветвей и границ.

Множество исследований ЗМТС имеют прикладной характер, так как область применения подобного рода задача крайне широка. Различные вариации ЗМТС могут использоваться, например, при группировке географических объектов в разнообразные кластеры [2] для грамотного планирования процессов продаж и перевозок, обслуживания, сбора мусора, предоставления медицинской помощи и т.п. Или же для грамотного составления маршрутов пилотируемых и беспилотных мобильных объектов в городских и межгородских транспортных сетях, при сборке урожая на полях, при тушении пожаров воздушными и наземными средствами [1, 5, 8]. Также подобного рода задачи встречаются при планировании вычислений, когда нескольким устройствам необходимо назначить определенные процессы [11]. Таким образом, актуальность в исследовании моделей и алгоритмов различных модификаций ЗМТС подтверждается не только теоретической необходимостью, но и практической значимостью.

В данной статье предложена математическая модель в виде задачи дискретной оптимизации для ЗМТС с чередованием объектов и единым местом сбора с условием отсутствия подциклов. В настоящем исследовании акцент сделан на построении математической модели с двумя блоками двухиндексных переменных и построении альтернативного условия отсутствия подциклов. Также в постановке не рассматривается фиксированное число мобильных объектов (транспортных средств). Разработаны итеративные и неитеративные жадные алгоритмы решения задачи, а также разработан адаптивный алгоритм, являющийся вероятностной модификацией одного из предложенных жадных алгоритмов. Два блока двухиндексных переменных в математической модели позволяют выделить в адаптивном алгоритме блок, не требующий обновления вероятностей на каждом шаге.

2. Постановка задачи

Приведем постановку ЗМТС с несколькими центрами с чередованием объектов [10].

Есть совокупность неподвижных объектов (вершин) двух типов: целевые объекты (тип А) и центры (тип В). Известны затраты, необходимые для перемещения между объектами. Несколько мобильных объектов (транспортных средств) должны в совокупности посетить все неподвижные объекты типа А таким образом, чтобы суммарные затраты были минимальными. Причем целевые объекты и центры должны чередоваться в маршрутах транспортных средств. Кроме того, каждый целевой объект в совокупности можно посетить только один раз, а любой центр можно посетить неограниченное число раз. При этом все мобильные объекты начинают и заканчивают свой маршрут в некоторой фиксированной точке сбора.

Под затратами можно понимать расстояние, временные затраты, стоимость и т.п.

Заметим, что количество мобильных агентов в постановке не оговаривается. Их может быть несколько, а может быть и только один.

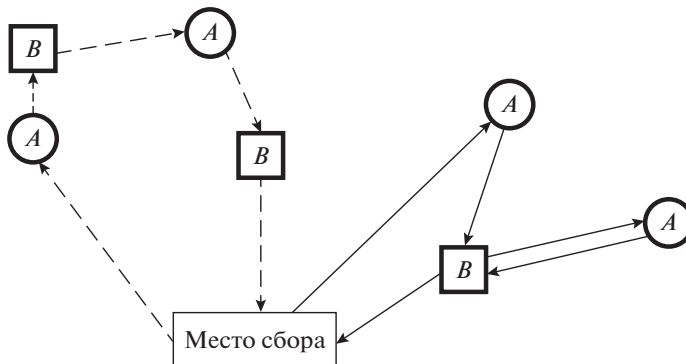


Рис. 1. ЗМТС с несколькими центрами с чередованием и единым местом сбора.

На рис. 1 представлена графическая интерпретация предложенной задачи, где целевые объекты обозначены кружками, центры — квадратами, а место сбора — прямоугольником. Дуги (пунктирные и сплошные) соответствуют некоторому маршруту.

На рис. 1 представлен маршрут с двумя посещениями места сбора. Классическая интерпретация ЗМТС говорит о наличии двух мобильных объектов, один из которых движется по пунктирным дугам, а второй — по сплошным. Другая интерпретация говорит о том, что присутствует только один мобильный объект, но он посещает место сбора два раза. В данной статье сделан акцент на количестве посещений единого места сбора (одним или несколькими транспортными средствами), а не на количестве мобильных объектов.

Заметим, что учет в постановке точного количества имеющихся мобильных объектов приведет к другой математической модели и другим алгоритмам решения.

3. Математическая модель

Вначале построим математическую модель задачи без требования того, чтобы все мобильные объекты начинали и заканчивали свой маршрут в некоторой фиксированной точке сбора (рис. 2).

Для составления математической модели введем следующие обозначения:

n — количество целевых объектов (неподвижных объектов типа А);

m — количество центров (неподвижных объектов типа В);

$(c_{ij}^1)_{m \times n}$ — матрица, задающая затраты для перемещения между i -м центром (тип В) и j -м целевым объектом (тип А);

$(c_{ji}^2)_{n \times m}$ — матрица, задающая затраты для перемещения между j -м целевым объектом (тип А) и i -м центром (тип В).

Введем два блока переменных:

$$x_{ij}, y_{ji} = \{0, 1\}, \quad i = 1, \dots, m, \quad j = 1, \dots, n.$$

$x_{ij} = 1$, если мобильный объект от i -го центра приезжает к j -му целевому объекту, и $x_{ij} = 0$ в противном случае;

$y_{ji} = 1$, если мобильный объект от j -го целевого объекта приезжает к i -му центру, и $y_{ji} = 0$ в противном случае.

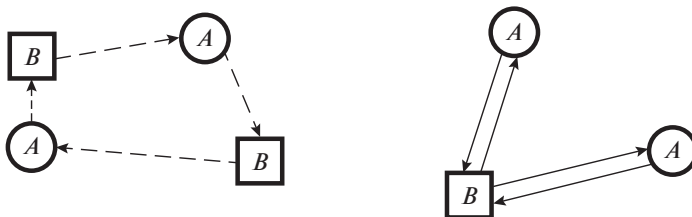


Рис. 2. ЗМТС с несколькими центрами с чередованием без единого места сбора.

Введем целевую функцию на основе условия, что совокупные затраты должны быть минимальны. Она будет иметь следующий вид:

$$(1) \quad L(X, Y) = \sum_{i=1}^m \sum_{j=1}^n (c_{ij}^1 x_{ij} + c_{ij}^2 y_{ij}) \rightarrow \min.$$

Здесь $X = (x_{ij})_{m \times n}$ и $Y = (y_{ji})_{n \times m}$ — матрицы соответствующих переменных.

Перейдем к составлению ограничений задачи, формирующих допустимое множество решений.

Из постановки следует, что к каждому целевому объекту должен приехать только один мобильный объект и от каждого целевого объекта должен уехать только один мобильный объект. Это можно записать одним ограничением вида

$$\left(\sum_{i=1}^m x_{ij} \right) \left(\sum_{i=1}^m y_{ji} \right) = 1, \quad j = 1, \dots, n,$$

которое является нелинейным. Вместо него можно записать эквивалентные ограничения:

$$(2) \quad \sum_{i=1}^m x_{ij} = 1, \quad j = 1, \dots, n,$$

$$(3) \quad \sum_{i=1}^m y_{ji} = 1, \quad j = 1, \dots, n,$$

которые относятся к типу ограничений классической задачи коммивояжера, и при этом математическая модель задачи останется линейной.

Для того чтобы маршрут каждого мобильного объекта был неразрывным, необходимо учитывать следующие условия: если к целевому объекту приезжает мобильный объект, то он же должен уехать от этого целевого объекта, и если от целевого объекта отъезжает мобильный объект, то он же должен был приехать к этому целевому объекту. Такие условия могут быть записаны нелинейными ограничениями вида

$$\begin{aligned} \left(\sum_{i=1}^m x_{ij} \right) \left(1 - \sum_{i=1}^m y_{ji} \right) &= 0, \quad j = 1, \dots, n, \\ \left(\sum_{i=1}^m y_{ji} \right) \left(1 - \sum_{i=1}^m x_{ij} \right) &= 0, \quad j = 1, \dots, n, \end{aligned}$$

которые эквивалентно преобразуются в линейное ограничение

$$(4) \quad \sum_{i=1}^m x_{ij} = \sum_{i=1}^m y_{ji}, \quad j = 1, \dots, n.$$

Такое ограничение является избыточным из-за (2) и (3).

Аналогично для неразрывности маршрута необходимо учитывать следующие условия: если к центру приезжает мобильный объект, то он же должен уехать от этого центра, и если от центра уезжает мобильный объект, то он же должен приехать к этому центру. Такие условия в свою очередь будут записаны следующим образом:

$$\begin{aligned} \left(\sum_{j=1}^n y_{ji} \right) \left(\sum_{j=1}^n y_{ji} - \sum_{j=1}^n x_{ij} \right) &= 0, \quad i = 1, \dots, m, \\ \left(\sum_{j=1}^n x_{ij} \right) \left(\sum_{j=1}^n x_{ij} - \sum_{j=1}^n y_{ji} \right) &= 0, \quad i = 1, \dots, m. \end{aligned}$$

Данные нелинейные ограничения также можно эквивалентно переписать одним линейным ограничением вида

$$(5) \quad \sum_{j=1}^n x_{ij} = \sum_{j=1}^n y_{ji}, \quad i = 1, \dots, m.$$

Заметим, что такое ограничение можно интерпретировать следующим образом: количество въездов в центр совпадает с количеством выездов из него.

Ограничения (2), (3), (5) при введенных переменных $x_{ij}, y_{ji} = \{0, 1\}$, $i = 1, \dots, m, j = 1, \dots, n$ формируют допустимое множество решений Ω .

Таким образом, построена математическая модель (1)–(3), (5) ЗМТС с несколькими центрами с чередованием.

Теперь рассмотрим случай, когда все мобильные объекты начинают и заканчивают свой маршрут в некоторой фиксированной точке сбора.

В качестве места сбора могут выступать: фиктивный целевой объект (объект типа А), фиктивный центр (объект типа В), пара фиктивных объектов с направлением прохода «целевой объект – центр» (А-В), пара фиктивных объектов с направлением прохода «центр – целевой объект» (В-А), причем для пар фиктивных объектов расстояние между ними равно нулю.

Каждый из этих случаев может быть продиктован конкретной практической постановкой. Например, при уборке стогов с поля удобно использовать пару «целевой объект – центр» (рис. 3), при работе грузовиков в песчаном карьере – пару «центр – целевой объект», при тушении ландшафтных пожаров самолетами – фиктивный объект типа В, при работе курьеров с обязательной доставкой документов – фиктивный объект типа А.

В данном исследовании будет использоваться пара «целевой объект – центр», которую, как уже было сказано, можно применить к практической задаче уборки стогов с поля. Если определить целевые объекты как стога, центры как места или машины, в которые загружаются стога, то очевидно, что от места сбора мобильные объекты (трактора) должны направиться к стогам, а не к машинам. Аналогично, после погрузки последнего стога в машину трактор от нее должен направиться на место сбора. Тогда для со-

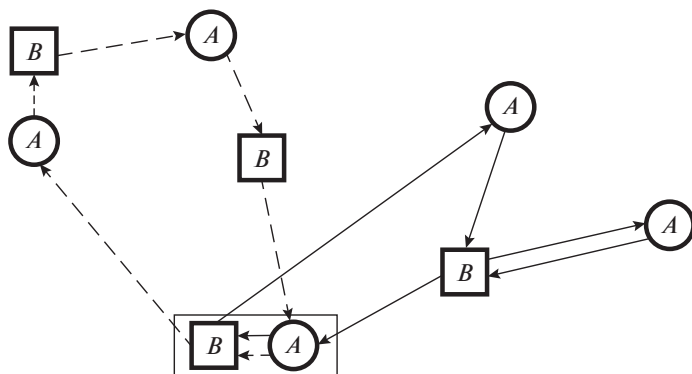


Рис. 3. ЗМТС с несколькими центрами с чередованием и единым местом сбора, заданным парой «целевой объект – центр».

хранения чередования в маршруте он должен быть следующим: «фиктивный стог – фиктивная машина – стог – машина – ... – стог – машина – фиктивный стог».

Итак, рассмотрим задачу, в которой место сбора задано парой «целевой объект – центр» (рис. 3).

Водятся фиктивный целевой объект и фиктивный центр. Пусть им отвечают индексы $j = 0$ и $i = 0$ соответственно. При этом на данную пару накладываются следующие условия:

- 1) мобильный объект из фиктивного целевого объекта может приехать только в фиктивный центр;
- 2) мобильный объект не может приехать от фиктивного центра в фиктивный целевой объект;
- 3) мобильный объект не может приехать в фиктивный центр из обычного целевого объекта;
- 4) мобильный объект не может приехать от фиктивного целевого объекта к обычному центру.

Необходимо ввести новые переменные x_{0j} , x_{i0} , y_{j0} , y_{0i} , $j = 0, \dots, n$, $i = 0, \dots, m$. При этом $x_{0j} \in \{0, 1\}$, $j = 0, \dots, n$ и $x_{0j} = 1$, если мобильный объект от фиктивного центра приезжает к j -му целевому объекту, и $x_{0j} = 0$ в противном случае. Переменные $x_{i0} \in \mathbb{N} \cup \{0\}$, $i = 0, \dots, m$ показывают количество посещений фиктивного целевого объекта из i -го центра. Заметим, что $x_{00} = 0$ согласно условию 2. Переменные $y_{j0} = 0$, $y_{0i} = 0$, $j = 1, \dots, n$, $i = 1, \dots, m$ согласно условиям 3 и 4, а $y_{00} \in \mathbb{N}$.

Использование целочисленных переменных x_{i0} и y_{00} вместо булевых объясняется наличием ограничения (5), расширенного для $i = 0$ и $j = 0$. Записав это ограничение для $i = 0$, получим

$$\sum_{j=0}^n x_{0j} = \sum_{j=0}^n y_{j0}.$$

Если y_{00} при этом будет булевой переменной (автоматически $y_{00} = 1$, так как маршрут должен быть замкнутым), то с учетом условия 3 получим

$$\sum_{j=0}^n x_{0j} = \sum_{j=0}^n y_{j0} = y_{00} = 1,$$

т.е.

$$\sum_{j=0}^n x_{0j} = 1.$$

Таким образом, получается, что из фиктивного центра может выехать только один мобильный объект, что не отражает сути места сбора мобильных объектов и в общем случае не так. Поэтому вводится положительная целочисленная переменная y_{00} .

Аналогично рассуждая, получим, что и $x_{i0} \in \mathbb{N} \cup \{0\}$, $i = 0, \dots, m$. Здесь добавление нулевого элемента обусловлено тем, что не обязательно из каждого центра мобильный объект приедет в фиктивный целевой объект.

Рассмотрим дополнительные ограничения, которые накладываются на фиктивные объекты.

Заметим, что количество выездов с места сбора и количество приездов на место сбора из постановки задачи не известны. Поэтому невозможно записать аналоги ограничений (2) и (3) для $j = 0$. Однако возможно записать аналог ограничения (4)

$$\sum_{i=0}^m x_{i0} = \sum_{i=0}^m y_{0i},$$

которое интерпретируется следующим образом: количество въездов в фиктивный целевой объект равно количеству выездов из него.

Расширение ограничения (5) уже было рассмотрено ранее и заключается в равенстве сумм:

$$\sum_{j=0}^n x_{0j} = \sum_{j=0}^n y_{j0},$$

и означает, что количество въездов в фиктивный центр равно количеству выездов из него.

Как было сказано ранее, переменные $y_{j0} = 0$, $y_{0i} = 0$, $j = 1, \dots, n$, $i = 1, \dots, m$, поэтому два последних равенства представляются в виде:

$$\begin{aligned} \sum_{i=0}^m x_{i0} &= y_{00}, \\ \sum_{j=0}^n x_{0j} &= y_{00}. \end{aligned}$$

Это дает возможность записать еще одно ограничение

$$\sum_{i=0}^m x_{i0} = \sum_{j=0}^n x_{0j}.$$

Очевидно, что оно избыточное, но оно отражает важное свойство места сбора: количество въездов и выездов для него совпадает.

Итак, на данном этапе получена следующая математическая модель:

$$(6) \quad L(X, Y) = \sum_{i=0}^m \sum_{j=0}^n (c_{ij}^1 x_{ij} + c_{ij}^2 y_{ij}) \rightarrow \min,$$

$$(7) \quad \sum_{i=0}^m x_{ij} = 1, \quad j = 1, \dots, n,$$

$$(8) \quad \sum_{i=0}^m x_{i0} = y_{00},$$

$$(9) \quad \sum_{i=0}^m y_{ji} = 1, \quad j = 1, \dots, n,$$

$$(10) \quad \sum_{j=0}^n x_{0j} = y_{00},$$

$$(11) \quad \sum_{j=0}^n x_{ij} = \sum_{j=0}^n y_{ji}, \quad i = 0, \dots, m,$$

$$(12) \quad x_{00} = 0,$$

$$(13) \quad y_{j0} = 0, \quad j = 1, \dots, n,$$

$$(14) \quad y_{0i} = 0, \quad i = 1, \dots, m,$$

$$(15) \quad x_{ij}, y_{ij} \in \{0, 1\}, \quad i = 1, \dots, m, \quad j = 1, \dots, n,$$

$$(16) \quad x_{0j} \in \{0, 1\}, \quad j = 0, \dots, n,$$

$$(17) \quad x_{i0} \in \mathbb{N} \cup \{0\}, \quad i = 0, \dots, m,$$

$$(18) \quad y_{00} \in \mathbb{N}.$$

В дальнейшем для обозначения количества посещений места сбора (въездов или выездов) вместо суммы $\sum_{j=0}^n x_{0j}$ будет использоваться обозначение K .

Стоит заметить, что полученная модель пока не учитывает требования того, что маршрут какого-либо мобильного объекта не содержит подциклы. Поэтому далее будет рассмотрено условие отсутствия подциклов.

4. Условие отсутствия подциклов

Сейчас модель допускает наличие неразрывного маршрута для некоторого мобильного объекта, который не проходит через фиктивную пару — место

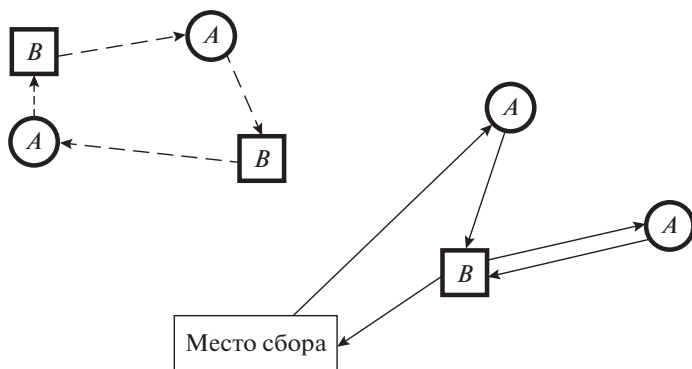


Рис. 4. Наличие неразрывного маршрута, который не проходит через место сбора.

сбора. Возможен случай, как на рис. 4, который не противоречит математической модели (6)–(18).

Такое решение содержит подциклы, выделенные пунктиром на рисунке. Таким образом, в математическую модель необходимо ввести дополнительное условие отсутствия подциклов. Оно может быть записано с использованием матрицы смежности.

Определение 1. Назовем два центра l и h *смежными*, если существует целевой объект j такой, что $x_{lj} = 1$ и $y_{jh} = 1$.

Будем говорить, что такие два центра соединены маршрутом длины 1 при условии $l \neq h$.

Будем считать, что если центр входит в маршрут, то он является смежным сам с собой.

Рассмотрим процесс построения матрицы смежности $A = (a_{lh})_{(m+1) \times (m+1)}$, $l, h = 0, \dots, m$. Диагональные элементы a_{ll} равны единице, если существует индекс j такой, что $x_{lj} = 1$, т.е. $\sum_{j=0}^n x_{lj} \geq 1$, и равны нулю в противном случае. Остальные элементы a_{lh} , $l \neq h$, равны единице, если существует индекс j такой, что $x_{lj} = 1$ и $y_{jh} = 1$, т.е. $\sum_{j=0}^n x_{lj}y_{jh} \geq 1$, и равны нулю в противном случае.

Используя скобку Айверсона, данные условия можно записать в виде

$$a_{ll} = \left[\sum_{j=0}^n x_{lj} \geq 1 \right], \quad a_{lh} = \left[\sum_{j=0}^n x_{lj}y_{jh} \geq 1 \right], \quad l, h = 0, \dots, m.$$

Пусть по найденному маршруту построена матрица смежности центров $A = (a_{lh})_{(m+1) \times (m+1)}$ по следующему правилу: если существует индекс j такой, что $x_{lj} = 1$ и $y_{jh} = 1$, то $a_{lh} = 1$. Элементы матрицы $\underbrace{A \cdot A \cdot \dots \cdot A}_k = A^k = (a_{lh}^k)_{(m+1) \times (m+1)}$ определяют количество маршрутов длины k между соответствующими центрами [12]. Если всего есть $(m+1)$ центр, то наибольшая

длина пути между ними равна m . Таким образом, если в матрице A^m нет нулевых элементов, то это означает, что из каждого центра можно попасть в каждый, т.е. все они связаны между собой и в маршруте нет подциклов.

Рассмотрим матрицу $A_M = (a_{lh})_{M \times M}$, полученную из матрицы A удалением нулевых строк и столбцов, $M \leq m$. Таким образом, матрица A_M — матрица смежности, в которой учтены только задействованные в маршруте центры. Если в матрице $(A_M)^k$, где k — любое из чисел $k \geq M$, нет нулевых элементов, то это означает, что все центры, входящие в маршрут, связаны между собой.

Таким образом, условие отсутствия подциклов может быть записано следующим образом:

$$(19) \quad a_{lh}^{(m)} \geq 1, l, \quad h = 0, \dots, m,$$

где $a_{lh}^{(m)}$ — элементы матрицы $(A_M)^m$, построенной по переменным x_{ij} и y_{ji} .

Ограничения (7)–(19) формируют допустимое множество решений Ω .

Таким образом, целевая функция (6) и ограничения (7)–(19) составляют математическую модель ЗМТС с несколькими центрами с чередованием и единым местом сбора.

5. Жадные алгоритмы решения

Рассмотрим один из вариантов жадного алгоритма решения поставленной задачи, общая идея которых состоит в принятии локально оптимальных решений на каждом этапе, допуская, что конечное решение также окажется оптимальным. Алгоритм заключается в построении связного маршрута с выбором ближайшего объекта нужного типа на каждом этапе [13].

Далее множество целевых объектов будем обозначать через J .

Алгоритм 1. Прямой жадный алгоритм.

1. Задать $L = 0$, $J = \{0, \dots, n\}$, $K = 0$, $i = 0$.

2. Если $i = 0$, то найти $j' = \arg \min_{j \in J \setminus \{0\}} \{c_{ij}^1\}$,

иначе найти $j' = \arg \min_{j \in J} \{c_{ij}^1\}$.

3. Положить $x_{ij'} = 1$, $L = L + c_{ij'}^1$.

4. Если $j' = 0$, то $i' = 0$, положить $y_{00} = y_{00} + 1$, $K = K + 1$,

иначе найти $i' = \arg \min_{i=1, \dots, m} \{c_{ji}^2\}$, положить $y_{j'i'} = 1$, $L = L + c_{j'i'}^2$.

5. Если $j' \neq 0$, то обновить J : $J = J \setminus \{j'\}$.

6. Проверить: $J = \{0\}$?

Если нет, то положить $i = i'$ и перейти к шагу 2,

если да, то положить $x_{i'0} = x_{i'0} + 1$, $L = L + c_{i'0}^1$, и $y_{00} = y_{00} + 1$, $K = K + 1$. Ответ получен.

На шаге 5 алгоритма не может быть ситуации, когда $J = \{0\}$ и $j' = 0$, так как при $j' = 0$ не происходит обновление множества J , и если оно уже

содержит только нулевой элемент, то окончание работы алгоритма должно было произойти на предыдущем шаге с предыдущим $j' \neq 0$. Таким образом, ветка «да» на шаге 6 не будет дублировать увеличение y_{00} на шаге 4 и не даст значение $x_{00} = 1$. Также на шаге 6 происходит выполнение ограничения (11) для $i = 0$.

Под ответом понимается маршрут, заданный в матричном виде, его длина L и количество посещений места сбора K .

Заметим, что данный алгоритм гарантирует хотя бы один выезд и один въезд в место сбора. Это обеспечивается шагами 1 и 6 на ветке «да». Также по построению маршрут получается замкнутым. Таким образом, выполняются ограничения, накладываемые на маршрут.

Алгоритм 1 является «интуитивным», прямым решением поставленной задачи. Каждый его шаг может являться командой мобильному объекту, находящемуся изначально в месте сбора.

Другой вариант жадного алгоритма не подразумевает обязательное начало построения маршрута из места сбора, но обязательно учитывает для него хотя бы один заезд и один выезд. То есть он также отвечает ограничениям. Данный алгоритм начинает работу в произвольном центре. Такая схема основывается на том, что жадные алгоритмы не обязаны начинать свою работу с первого объекта (первой строки матрицы затрат, первого столбца, первого города и т.п.), а могут это делать с произвольного объекта. С учетом того, что в рассматриваемой задаче присутствует два типа объектов, для каждого из них разработаны соответствующие схемы (*алгоритм 2* и *алгоритм 4*).

Алгоритм 2. Жадный алгоритм центров.

1. Задать $L = 0$, $J = \{0, \dots, n\}$, $K = 0$, задать произвольный индекс $i_{\text{нач}} = \{0, \dots, m\}$.

2. Если $i_{\text{нач}} = 0$, то выполнить *алгоритм 1* и завершить *алгоритм 2*, иначе перейти к шагу 3.

3. Положить $x_{i_{\text{нач}}0} = 1$, $L = L + c_{i_{\text{нач}}0}^1$, $y_{00} = 1$, $K = K + 1$, $i = 0$.

4. Если $i = 0$, то найти $j' = \arg \min_{j \in J \setminus \{0\}} \{c_{ij}^1\}$,

иначе найти $j' = \arg \min_{j \in J} \{c_{ij}^1\}$.

5. Положить $x_{ij'} = 1$, $L = L + c_{ij'}^1$.

6. Если $j' \neq 0$, то обновить J : $J = J \setminus \{j'\}$.

7. Проверить: $J = \{0\}$?

Если да, то перейти к шагу 10,

если нет, то перейти к шагу 8.

8. Если $j' = 0$, то положить $i' = 0$, $y_{00} = y_{00} + 1$, $K = K + 1$,

иначе положить $i' = \arg \min_{i=1, \dots, m} \{c_{ji}^2\}$, положить $y_{j'i'} = 1$, $L = L + c_{j'i'}^2$.

9. Положить $i = i'$ и перейти к шагу 4.

10. Положить $y_{j'i_{\text{нач}}} = 1$, $L = L + c_{j'i_{\text{нач}}}^2$. Ответ получен.

Заметим, что на шаге 3, т.е. в начале работы алгоритма, происходит обязательный проход через место сбора. Данный шаг необходим, иначе алгоритм не гарантирует посещение фиктивных объектов. На шаге 10 происходит замыкание маршрута, т.е. выполнение ограничения (11) для $i_{\text{нач}}$.

Данный алгоритм подразумевает итеративную модификацию, в которой на каждой итерации построение маршрута начинается с нового центра. Здесь используется классическая схема итеративной модификации жадных алгоритмов, основанная на переборе всех возможных начальных точек. С учетом наличия двух типов объектов предложены два итеративных жадных алгоритма (*алгоритм 3* и *алгоритм 5*).

Алгоритм 3. Итеративный жадный алгоритм центров.

1. Задать $i_{\text{тек}} = 0$, $L^* = \infty$.

2. Выполнить *алгоритм 2*, считая для него $i_{\text{нач}} = i_{\text{тек}}$. В результате получено решение $L^{i_{\text{тек}}}$, $(x_{ij})^{i_{\text{тек}}}$, $(y_{ji})^{i_{\text{тек}}}$, $K^{i_{\text{тек}}}$.

3. Проверить: $L^{i_{\text{тек}}} < L^*$?

Если да, то обновить L^* : $L^* = L^{i_{\text{тек}}}$, положить $(x_{ij})^* = (x_{ij})^{i_{\text{тек}}}$, $(y_{ji})^* = (y_{ji})^{i_{\text{тек}}}$, $K^* = K^{i_{\text{тек}}}$ и перейти к шагу 4,

иначе перейти к шагу 4.

4. Проверить: $i_{\text{тек}} = m$?

Если да, то перейти к шагу 6,

Если нет, то перейти к шагу 5.

5. Положить $i_{\text{тек}} = i_{\text{тек}} + 1$ и перейти к шагу 2.

6. Ответ L^* , $(x_{ij})^*$, $(y_{ji})^*$, K^* получен.

Другие варианты жадного алгоритма и его итеративной модификации заключаются в начале построения маршрута в некотором целевом объекте. В этом случае обязательный проход через место сбора происходит в конце работы алгоритма.

Алгоритм 4. Жадный алгоритм целевых объектов.

1. Задать $L = 0$, $J = \{0, \dots, n\}$, $K = 0$, задать произвольный индекс $j_{\text{нач}} = \{0, \dots, n\}$.

2. Если $j_{\text{нач}} = 0$, то положить $i = 0$ и выполнить *алгоритм 1* и завершить *алгоритм 4*,

иначе перейти к шагу 3.

3. Положить $j = j_{\text{нач}}$.

4. Если $j = 0$, то положить $i' = 0$, $y_{00} = y_{00} + 1$, $K = K + 1$, иначе найти $i' = \arg \min_{i=1, \dots, m} \{c_{ji}^2\}$, положить $y_{ji'} = 1$, $L = L + c_{ji'}^2$.

5. Если $j \neq 0$, то обновить J : $J = J \setminus \{j\}$.

6. Проверить: $J = \{0\}$?

Если да, то перейти к шагу 9,

если нет, то перейти к шагу 7.

7. Если $i' = 0$, то найти $j' = \arg \min_{j \in J \setminus \{0\}} \{c_{i'j}^1\}$,

иначе найти $j' = \arg \min_{j \in J} \{c_{i'j}^1\}$.

8. Положить $x_{i'j'} = 1$, $L = L + c_{i'j'}^1$, $j = j'$ и перейти к шагу 4.

9. Положить $x_{i'0} = x_{i'0} + 1$, $L = L + c_{i'0}^1$, и $y_{00} = y_{00} + 1$, $K = K + 1$, $x_{0j_{\text{нач}}} = 1$, $L = L + c_{0j_{\text{нач}}}^1$. Ответ получен.

В данном алгоритме замыкание маршрута в $j_{\text{нач}}$ и обязательное посещение фиктивных объектов (выполнение ограничений (7) и (9)) происходит на шаге 9.

Также рассмотрим итеративную модификацию этого алгоритма.

Алгоритм 5. Итеративный жадный алгоритм целевых объектов.

1. Задать $j_{\text{тек}} = 0$, $L^* = \infty$.

2. Выполнить *алгоритм 4*, считая для него $j_{\text{нач}} = j_{\text{тек}}$. В результате получено решение $L^{i_{\text{тек}}}$, $(x_{ij})^{i_{\text{тек}}}$, $(y_{ji})^{i_{\text{тек}}}$, $K^{i_{\text{тек}}}$.

3. Проверить: $L^{i_{\text{тек}}} < L^*$?

Если да, то обновить L^* : $L^* = L^{i_{\text{тек}}}$, положить $(x_{ij})^* = (x_{ij})^{i_{\text{тек}}}$, $(y_{ji})^* = (y_{ji})^{i_{\text{тек}}}$, $K^* = K^{i_{\text{тек}}}$ и перейти к шагу 4,

иначе перейти к шагу 4.

4. Проверить: $j_{\text{тек}} = n$?

Если да, то перейти к шагу 6,

Если нет, то перейти к шагу 5.

5. Положить $j_{\text{тек}} = j_{\text{тек}} + 1$ и перейти к шагу 2.

6. Ответ L^* , $(x_{ij})^*$, $(y_{ji})^*$, K^* получен.

6. Адаптивный алгоритм

Другой возможной модификацией жадных алгоритмов являются адаптивные алгоритмы [14], которые основываются на переходе к вероятностной постановке задачи.

Задача

$$L(X, Y) \rightarrow \min_{X, Y \in \Omega}$$

заменяется задачей вида

$$M(L(X, Y)) \rightarrow \min_{\{X\}: X \in \Omega, \{Y\}: Y \in \Omega},$$

где $\{X\}$ и $\{Y\}$ – множества случайных величин X и Y с реализациями X и Y из множества Ω .

Здесь $X = [X^1, \dots, X^n]$ – матрица случайных величин размера $(m+1) \times (n+1)$, $X^i = (X_{i0}, X_{i1}, \dots, X_{in})$ – строка матрицы X , X_{ij} – случайная величина, $i = 0, \dots, m$, $j = 0, \dots, n$; $Y = [Y^1, \dots, Y^n]$ – матрица случайных величин размера $(n+1) \times (m+1)$, $Y^j = (Y_{j0}, Y_{j1}, \dots, Y_{jm})$ – строка матрицы Y , Y_{ji} – случайная величина, $i = 0, \dots, m$, $j = 0, \dots, n$.

Таблица 1. Ряды распределений

X_{ij}			Y_{ji}		
Возможные значения	1	0	Возможные значения	1	0
Вероятность	p_{ij}^1	$q_{ij}^1 = 1 - p_{ij}^1$	Вероятность	p_{ij}^2	$q_{ij}^2 = 1 - p_{ij}^2$

Замечание 1. Для дальнейшего конструирования адаптивного алгоритма необходим только факт выезда мобильного объекта (или же его отсутствие) из центра в фиктивный целевой объект, а также факт выезда из фиктивного целевого объекта в фиктивный центр, а не количество этих выездов. Таким образом, будем считать, что случайные величины X_{i0} , $i = 0, \dots, m$ и Y_{00} принимаю только два возможных значения: 0 и 1.

Итак, с учетом замечания случайные величины заданы рядами распределений, представленными в табл. 1.

Для каждого $j = 0, \dots, n$ вводятся полные группы событий A_{ij} и B_{ij} : событие A_{ij} заключается в том, что x_{ij} примет значение 1 для какого-либо $i = 0, \dots, m$, $B_{ji} - y_{ji}$ примет значение 1 для какого-либо $i = 0, \dots, m$, т.е.

$$(20) \quad \sum_{i=0}^m p_{ij}^1 = 1, \quad j = 0, \dots, n,$$

$$(21) \quad \sum_{i=0}^m p_{ji}^2 = 1, \quad j = 0, \dots, n.$$

С учетом того, что изначально нет никаких предположений о возможном маршруте, и мобильный объект к целевому объекту может приехать от любого центра с равной вероятностью, а также с учетом множества Ω и условий (20), (21), распределения могут быть заданы, например, табл. 2.

Таблица 2. Ряды распределений

X_{00}			Y_{00}		
Возможные значения	1	0	Возможные значения	1	0
Вероятность	$p_{00}^1 = 0$	$q_{00}^1 = 1$	Вероятность	$p_{00}^2 = 1$	$q_{00}^2 = 0$
$X_{i0}, i = 1, \dots, m$			$Y_{j0}, j = 1, \dots, n, Y_{0i}, i = 1, \dots, m$		
Возможные значения	1	0	Возможные значения	1	0
Вероятность	$p_{i0}^1 = \frac{1}{m}$	$q_{i0}^1 = \frac{m-1}{m}$	Вероятность	$p_{j0}^2 = 0,$ $p_{0i}^2 = 0$	$q_{j0}^2 = 1,$ $q_{0i}^2 = 1$
$X_{ij}, i = 0, \dots, m, j = 1, \dots, n$			$Y_{ji}, j = 1, \dots, n, i = 1, \dots, m$		
Возможные значения	1	0	Возможные значения	1	0
Вероятность	$p_{ij}^1 = \frac{1}{m+1}$	$q_{ij}^1 = \frac{m}{m+1}$	Вероятность	$p_{ji}^2 = \frac{1}{m}$	$q_{ji}^2 = \frac{m-1}{m}$

Другими словами матрицы вероятностей P^1, P^2 для возможных положительных значений будут выглядеть следующим образом

$$P^1 = (p_{ij}^1)_{(m+1) \times (n+1)} = \begin{pmatrix} 0 & \frac{1}{m+1} & \cdots & \frac{1}{m+1} \\ \frac{1}{m} & \frac{1}{m+1} & \cdots & \frac{1}{m+1} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{m} & \frac{1}{m+1} & \cdots & \frac{1}{m+1} \end{pmatrix},$$

$$P^2 = (p_{ij}^2)_{(n+1) \times (m+1)} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & \frac{1}{m} & \cdots & \frac{1}{m} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \frac{1}{m} & \cdots & \frac{1}{m} \end{pmatrix}.$$

В основе адаптивного алгоритма лежат этапы I–III [13]:

I. Задание движения во множестве случайных величин X_{ij}, Y_{ji} .

II. Решение неравенства — условия локального улучшения (УЛУ)

$$M \left[L((X_{ij})^{N+1}, (Y_{ji})^{N+1}) - L((X_{ij})^N, (Y_{ji})^N) \right] \leq 0.$$

III. Пересчет вероятностей $p_{ij}^1, p_{ji}^2, i = 0, \dots, m, j = 0, \dots, n$ в соответствии с результатом УЛУ.

Данный алгоритм является итеративным, и величины $(X_{ij})^{N+1}, (Y_{ji})^{N+1}$ на $(N + 1)$ -й итерации по УЛУ должны выбираться так, чтобы значение целевой функции в них было лучше, чем в предыдущих точках на предыдущей итерации.

Необходимое теоретическое описание каждого этапа представлено в [13]. Здесь приведем лишь необходимые выкладки и формулы.

Движение во множестве случайных величин на этапе I зададим следующим образом

$$(X)^{N+1} = \bar{u}(X)^N + u(\chi)^{N+1},$$

$$(Y)^{N+1} = \bar{v}(Y)^N + v(\gamma)^{N+1}.$$

Здесь $(\chi)^{N+1}$ и $(\gamma)^{N+1}$ — неизвестные случайные величины, задающие направление движения на $(N + 1)$ -м шаге. При этом $(\chi)^{N+1} \in X, (\gamma)^{N+1} \in Y$ и χ_{ij} и γ_{ji} имеют ряды распределения, заданные табл. 3.

Таблица 3. Ряды распределения

$(\chi)_{ij}$			$(\gamma)_{ji}$		
Возможные значения	1	0	Возможные значения	1	0
Вероятность	π_{ij}^1	$1 - \pi_{ij}^1$	Вероятность	π_{ji}^2	$1 - \pi_{ji}^2$

Таблица 4. Ряды распределения

u			v		
Возможные значения	1	0	Возможные значения	1	0
Вероятность	p_u	$q_u = 1 - p_u$	Вероятность	p_v	$q_v = 1 - p_v$

Для случайных величин u и v ряды распределения представлены табл. 4.

Формулы пересчета вероятностей на этапе III с учетом формул движения будут выглядеть следующим образом:

$$\begin{aligned} (p_{ij}^1)^{N+1} &= q_u(p_{ij}^1)^N + p_u(\pi_{ij}^1)^{N+1}, \quad i = 0, \dots, m, \quad j = 0, \dots, n, \\ (p_{ji}^2)^{N+1} &= q_v(p_{ji}^2)^N + p_v(\pi_{ji}^2)^{N+1}, \quad i = 0, \dots, m, \quad j = 0, \dots, n. \end{aligned}$$

Вероятности π_{ij}^1 и π_{ji}^2 находятся из УЛУ на этапе II.

Так как целевая функция (6) представима в виде суммы двух функций по переменным x и y , запишем УЛУ этапа II в следующем виде

$$M\left[L_X((X_{ij})^{N+1}) - L_X((X_{ij})^N)\right] + M\left[L_Y((Y_{ji})^{N+1}) - L_Y((Y_{ji})^N)\right] \leq 0.$$

Как показано в [13], УЛУ может быть переписано в виде

$$\begin{aligned} M\left[L_X((\chi_{ij})^{N+1}) - L_X((X_{ij})^N)\right] + M\left[L_Y((\gamma_{ji})^{N+1}) - L_Y((Y_{ji})^N)\right] = \\ = M\left[L_X((\chi_{ij})^{N+1})\right] - M\left[L_X((X_{ij})^N)\right] + \\ + M\left[L_Y((\gamma_{ji})^{N+1})\right] - M\left[L_Y((Y_{ji})^N)\right] \leq 0. \end{aligned}$$

В качестве базового алгоритма для вероятностной модификации будем использовать *алгоритм 4*. То есть в ходе работы алгоритма необходимо сначала найти переменную $y_{j'j'} = 1$, а затем – переменную $x_{i'i'} = 1$.

Неравенство УЛУ имеет бесконечное множество решений. Воспользуемся идеей алгоритма покоординатного спуска.

Предположим, что на текущей $(N + 1)$ -й итерации алгоритма на текущем шаге рассматривается l -я строка матриц (c_{ji}^2) и γ , т.е. рассматривается неизвестная случайная величина γ^l . Будем считать, что $(\pi_{ji}^2)^{N+1} = (p_{ji}^2)^N$, т.е. случайные величины $(\gamma_{ji})^{N+1}$ и $(Y_{ji})^N$ имеют одинаковое распределение, $i = 0, \dots, m, j \in J, j \neq l$. Будем решать неравенство относительно неизвестной величины γ^l .

$$\begin{aligned} M\left[L_X((\chi_{ij})^{N+1})\right] - M\left[L_X((X_{ij})^N)\right] + M\left[L_Y((\gamma_{ji})^{N+1})\right] - M\left[L_Y((Y_{ji})^N)\right] = \\ = M\left[L_X((\chi_{ij})^{N+1})\right] - M\left[L_X((X_{ij})^N)\right] + \\ + M\left[\sum_{i=0}^m c_{li}^2(\gamma_{li})^{N+1} + \sum_{i=0}^m \sum_{j \in J, j \neq l} c_{ji}^2(\gamma_{ji})^{N+1}\right] - \\ - M\left[\sum_{i=0}^m c_{li}^2(Y_{li})^N + \sum_{i=0}^m \sum_{j \in J, j \neq l} c_{ji}^2(Y_{ji})^N\right] \leq 0. \end{aligned}$$

Учитывая, что $\sum_{i=0}^m p_{li}^2 = 1$, зафиксируем реализацию случайной величины Y^l . Пусть $Y^l = e_{i_1}$, где e_{i_1} — единичный вектор размерности $(n+1)$ с единицей на i_1 -м месте, т.е. $y_{li_1} = 1$, $y_{lj} = 0$, $i \neq i_1$, $j \neq l$. Тогда выражение $M \left[\sum_{i=0}^m c_{li}^2 (Y_{li})^N + \sum_{i=0}^m \sum_{j \in J, j \neq l} c_{ji}^2 (Y_{ji})^N \right]$ с учетом того, что $Y : Y \in \Omega$, примет вид

$$c_{li_1}^2 + M_{Y|Y^l=e_{i_1}} \left[\sum_{i=0}^m \sum_{j \in J, j \neq l} c_{ji}^2 (Y_{ji})^N \right] = c_{li_1}^2 + \sum_{i=0}^m \sum_{j \in J, j \neq l} c_{ji}^2 (p_{ji}^2)^N.$$

Аналогично, при какой-либо реализации γ^l , равной e_{i_2} , с учетом $\gamma \in Y : Y \in \Omega$ и $(\pi_{ji}^2)^{N+1} = (p_{ji}^2)^N$, $i = 0, \dots, m$, $j \in J$, $j \neq l$ выражение $M \left[\sum_{i=0}^m c_{li}^2 (\gamma_{li})^{N+1} + \sum_{i=0}^m \sum_{j \in J, j \neq l} c_{ji}^2 (\gamma_{ji})^{N+1} \right]$ примет вид

$$c_{li_2}^2 + \sum_{i=0}^m \sum_{j \in J, j \neq l} c_{ji}^2 (p_{ji}^2)^N.$$

Тогда в неравенстве УЛУ выражение $M \left[L_Y((\gamma_{ji})^{N+1}) \right] - M \left[L_Y((Y_{ji})^N) \right]$ примет вид

$$(22) \quad \left(c_{li_2}^2 + \sum_{i=0}^m \sum_{j \in J, j \neq l} c_{ji}^2 (p_{ji}^2)^N \right) - \left(c_{li_1}^2 + \sum_{i=0}^m \sum_{j \in J, j \neq l} c_{ji}^2 (p_{ji}^2)^N \right) = c_{li_2}^2 - c_{li_1}^2.$$

Далее рассмотрим оставшуюся часть неравенства УЛУ.

Итак, пусть теперь рассматривается i_2 -я строка матриц (c_{ij}^1) и χ , т.е. рассматривается неизвестная случайная величина χ^{i_2} . Будем считать, что $(\pi_{ij}^1)^{N+1} = (p_{ij}^1)^N$, т.е. случайные величины $(\chi_{ij})^{N+1}$ и $(X_{ij})^N$ имеют одинаковое распределение, $i = 0, \dots, m$, $i \neq i_2$, $j \in J$. Будем решать неравенство относительно неизвестной величины χ^{i_2} .

Пусть реализация случайной строки X^{i_2} зафиксирована, например, на предыдущей итерации алгоритма, т.е. $X^{i_2} = e_{l_1, \dots, l_L}$, где e_{l_1, \dots, l_L} — вектор нулей размерности $(n+1)$ с единицами на местах l_1, \dots, l_L . Тогда $x_{i_2 l_1} = 1, \dots, x_{i_2 l_L} = 1$. Тогда математическое ожидание $M \left[L_X((X_{ij})^N) \right] =$

$$= M \left[\sum_{j \in J} c_{i_2 j}^1 (X_{i_2 j})^N + \sum_{\substack{i=0 \\ i \neq i_2}}^m \sum_{j \in J} c_{ij}^1 (X_{ij})^N \right] \text{ с учетом того, что } X : X \in \Omega, \text{ а}$$

именно с учетом ограничения (7), примет вид

$$\begin{aligned}
& c_{i_2 l_1}^1 + \dots + c_{i_2 l_L}^1 + M_{X|X^{i_2}=e_{l_1, \dots, l_L}} \left[\sum_{\substack{i=0 \\ i \neq i_2}}^m \sum_{j \in J, j \neq l_1, \dots, j \neq l_L} c_{ij}^1 (X_{ij})^N \right] = \\
& = c_{i_2 l_1}^1 + \dots + c_{i_2 l_L}^1 + \sum_{\substack{i=0 \\ i \neq i_2}}^m \sum_{j \in J, j \neq l_1, \dots, j \neq l_L} c_{ij}^1 (p_{ij}^1)^N = \\
& = c_{i_2 l_1}^1 + \dots + c_{i_2 l_L}^1 + \sum_{\substack{i=0 \\ i \neq i_2}}^m \sum_{j \in J} c_{ij}^1 (p_{ij}^1)^N - \\
& - \left(\sum_{\substack{i=0 \\ i \neq i_2}}^m c_{i l_1}^1 (p_{i l_1}^1)^N + \dots + c_{i l_L}^1 (p_{i l_L}^1)^N \right).
\end{aligned}$$

Заметим, что если некоторый индекс $l_k = 0$, то его можно не исключать из суммирования в математическом ожидании.

Заметим, что на одной итерации может быть выбрано несколько переменных x_{ij} , $j \in J$, равных единице. Но на одном шаге алгоритма такая переменная может быть только одна. Итак, на текущем шаге алгоритма выбирается одно значение $x_{i_2 q} = 1$, т.е. фиксируется промежуточная реализация χ^{i_2} , равная e_q . Тогда с учетом $\chi \in X: X \in \Omega$ и $(\pi_{ij}^1)^{N+1} = (p_{ij}^1)^N$, $i = 0, \dots, m$, $i \neq i_2$,

$$j \in J, \text{ выражение } M[L_X((\chi_{ij})^{N+1})] = M \left[\sum_{j \in J} c_{i_2 j}^1 (\chi_{hj})^{N+1} + \sum_{\substack{i=0 \\ i \neq i_2}}^m \sum_{j \in J} c_{ij}^1 (\chi_{ij})^{N+1} \right]$$

примет вид

$$c_{i_2 q}^1 + \sum_{\substack{i=0 \\ i \neq i_2}}^m \sum_{j \in J, j \neq q} c_{ij}^1 (p_{ij}^1)^N = c_{i_2 q}^1 + \sum_{\substack{i=0 \\ i \neq i_2}}^m \sum_{j \in J} c_{ij}^1 (p_{ij}^1)^N - \left(\sum_{\substack{i=0 \\ i \neq i_2}}^m c_{i q}^1 (p_{i q}^1)^N \right).$$

Тогда в неравенстве УЛУ выражение $M[L_X((\chi_{ij})^{N+1})] - M[L_X((X_{ij})^N)]$ примет вид

$$\begin{aligned}
& \left[c_{i_2 q}^1 + \sum_{\substack{i=0 \\ i \neq i_2}}^m \sum_{j \in J} c_{ij}^1 (p_{ij}^1)^N - \left(\sum_{\substack{i=0 \\ i \neq i_2}}^m c_{i q}^1 (p_{i q}^1)^N \right) \right] - \\
& - \left[c_{i_2 l_1}^1 + \dots + c_{i_2 l_L}^1 + \sum_{\substack{i=0 \\ i \neq i_2}}^m \sum_{j \in J} c_{ij}^1 (p_{ij}^1)^N - \left(\sum_{\substack{i=0 \\ i \neq i_2}}^m c_{i l_1}^1 (p_{i l_1}^1)^N + \dots + c_{i l_L}^1 (p_{i l_L}^1)^N \right) \right] =
\end{aligned}$$

$$= \left[c_{i_2q}^1 - \left(\sum_{\substack{i=0 \\ i \neq i_2}}^m c_{i_2q}^1 (p_{i_2q}^1)^N \right) \right] - \\ - \left[c_{i_2l_1}^1 + \dots + c_{i_2l_L}^1 - \left(\sum_{\substack{i=0 \\ i \neq i_2}}^m c_{il_1}^1 (p_{il_1}^1)^N + \dots + c_{il_L}^1 (p_{il_L}^1)^N \right) \right] \leq 0.$$

Таким образом, с учетом (22) УЛУ примет вид

$$(23) \quad c_{i_2}^2 - c_{l_{i_1}}^2 + \left[c_{i_2q}^1 - \sum_{\substack{i=0 \\ i \neq i_2}}^m c_{i_2q}^1 (p_{i_2q}^1)^N \right] - \\ - \left[c_{i_2l_1}^1 + \dots + c_{i_2l_L}^1 + \left(\sum_{\substack{i=0 \\ i \neq i_2}}^m c_{il_1}^1 (p_{il_1}^1)^N + \dots + \sum_{\substack{i=0 \\ i \neq i_2}}^m c_{il_L}^1 (p_{il_L}^1)^N \right) \right] \leq 0$$

или

$$(24) \quad \left[c_{i_2}^2 + c_{i_2q}^1 - \sum_{\substack{i=0 \\ i \neq i_2}}^m c_{i_2q}^1 (p_{i_2q}^1)^N \right] - \\ - \left[c_{l_{i_1}}^2 + c_{i_2l_1}^1 + \dots + c_{i_2l_L}^1 + \left(\sum_{\substack{i=0 \\ i \neq i_2}}^m c_{il_1}^1 (p_{il_1}^1)^N + \dots + \sum_{\substack{i=0 \\ i \neq i_2}}^m c_{il_L}^1 (p_{il_L}^1)^N \right) \right] \leq 0.$$

Итак, если реализация случайной величины Y^l зафиксирована на предыдущей итерации алгоритма: $Y^l = e_{i_1}$, то для гарантированного выполнения неравенств УЛУ (23) необходимо выбирать индексы i_2 (с учетом того, что на предыдущей итерации X^{i_2} зафиксирована: $X^{i_2} = e_{l_1, \dots, l_L}$) и q так, чтобы

величины $c_{i_2}^2$ и $\left[c_{i_2q}^1 - \sum_{\substack{i=0 \\ i \neq i_2}}^m c_{i_2q}^1 (p_{i_2q}^1)^N \right]$ были минимальны.

В результате неизвестные величины $(\gamma)^{N+1}$ находится следующим образом:

$$(\gamma_{l\mu})^{N+1} : \pi_{l\mu}^2 = 1, \text{ если } \min_{i=0, \dots, m} \{c_{li}^2\} = c_{l\mu}^2, \\ (\gamma_{li})^{N+1} : \pi_{li}^2 = 0, \quad i = 0, \dots, m, i \neq \mu, \\ (\gamma^j)^{N+1} = (Y^j)^{N+1}.$$

Пересчет вероятностей при этом на этапе III происходит по формулам

$$(25) \quad (p_{l\mu}^2)^{N+1} = q_\nu (p_{l\mu}^2)^N + p_\nu,$$

$$(26) \quad (p_{li}^2)^{N+1} = q_\nu (p_{li}^2)^N, \quad i = 0, \dots, m, \quad i \neq \mu,$$

$$(27) \quad (p_{ji}^2)^{N+1} = (p_{ji}^2)^N, \quad i = 0, \dots, m, \quad i \in J, \quad j \neq l.$$

Величина $(\chi)^{N+1}$ находится следующим образом:

$$\begin{aligned} (\chi_{\mu\nu})^{N+1} : \pi_{\mu\nu}^1 = 1, \text{ если } \min_{j \in J} \left\{ c_{\mu j}^1 - \left(\sum_{\substack{i=0 \\ i \neq \mu}}^m c_{ij}^1 (p_{ij}^1)^N \right) \right\} = \\ = c_{\mu\nu}^1 - \left(\sum_{\substack{i=0 \\ i \neq \mu}}^m c_{i\nu}^1 (p_{i\nu}^1)^N \right), \end{aligned}$$

$$(\chi_{i\nu})^{N+1} : \pi_{i\nu}^1 = 0, \quad i = 0, \dots, m, \quad i \neq \mu,$$

$$(\chi_{ij})^{N+1} = (X^{ij})^{N+1}, \quad i = 0, \dots, m, \quad j \in J, \quad j \neq \nu.$$

Формулы для пересчета вероятностей

$$(28) \quad (p_{\mu\nu}^1)^{N+1} = q_u (p_{\mu\nu}^1)^N + p_u,$$

$$(29) \quad (p_{i\nu}^1)^{N+1} = q_u (p_{i\nu}^1)^N, \quad i = 0, \dots, m, \quad i \neq \mu,$$

$$(30) \quad (p_{ij}^1)^{N+1} = (p_{ij}^1)^N, \quad i = 0, \dots, m, \quad i \in J, \quad j \neq \nu.$$

В выборе минимального элемента по матрице (c_{ij}^2) не участвуют вероятности, поэтому пересчет матрицы вероятностей (p_{ij}^2) по формулам (25)–(27) не требуется для работы алгоритма.

Полученные вероятности (28)–(30) необходимо пересчитать по формуле безусловных вероятностей, исходя из предположения, что все гипотезы изначально распределены равномерно

$$(31) \quad (\bar{p})^{N+1} = \frac{N}{N+1} \left((\bar{p})^N + \frac{1}{N} (p)^{N+1} \right).$$

Здесь $(\bar{p})^{N+1}$ – безусловная вероятность, $(p)^{N+1}$ – условная вероятность, полученная по формулам (28)–(30).

Таким образом, на каждом этапе происходит подстройка или адаптация вероятностей на основе предыдущих итераций алгоритма.

Таким образом, предложенное решение неравенства УЛУ позволяет модифицировать шаг 6 алгоритма 4 следующим образом:

при заданном i' найти

$$j' = \arg \min_{j \in J} \left\{ c_{i'j}^1 - \left(\sum_{\substack{i=0 \\ i \neq i'}}^m c_{ij}^1 (p_{ij}^1)^N \right) \right\}.$$

Далее представлен адаптивный алгоритм решения поставленной задачи.

Алгоритм 6. Адаптивный алгоритм.

1. Задать рекордное значение $L^* = \infty$. Задать максимальное число итераций N_{\max} . Задать текущий номер итерации $N = 1$. Задать начальное распределение вероятностей $(p_{ij}^1)^N$ по табл. 2. Задать p_u .

2. Задать $L = 0, J = \{0, \dots, n\}, K = 0, j = 0$.

3. Если $j = 0$, то положить $i' = 0, y_{00} = y_{00} + 1, K = K + 1$,
иначе найти $i' = \arg \min_{i=1, \dots, m} \{c_{ji}^2\}$, положить $y_{ji'} = 1, L = L + c_{ji'}^2$.

4. Если $j \neq 0$, то обновить $J: J = J \setminus \{j\}$.

5. Проверить: $J = \{0\}$?

Если да, то перейти к шагу 10,

если нет, то перейти к шагу 9.

6. Если $i' = 0$, то найти $j' = \arg \min_{j \in J \setminus \{0\}} \left\{ c_{i'j}^1 - \left(\sum_{\substack{i=0 \\ i \neq i'}}^m c_{ij}^1 (p_{ij}^1)^N \right) \right\}$,

иначе найти $j' = \arg \min_{j \in J} \left\{ c_{i'j}^1 - \left(\sum_{\substack{i=0 \\ i \neq i'}}^m c_{ij}^1 (p_{ij}^1)^N \right) \right\}$.

7. Обновить вероятности по формулам (28)–(30).

8. Обновить безусловные вероятности по формуле (31).

9. Положить $x_{i'j'} = 1, L = L + c_{i'j'}^1, j = j'$ и перейти к шагу 3.

10. Положить $x_{i'0} = x_{i'0} + 1, L = L + c_{i'0}^1, y_{00} = y_{00} + 1, K = K + 1, x_{0j_{\text{нач}}} = 1, L = L + c_{0j_{\text{нач}}}^1$. В результате получено решение на текущей итерации $L^N, (x_{ij})^N, (y_{ji})^N, K^N$.

11. Проверить: $L^N < L^*$?

Если да, то обновить $L^*: L^* = L^N$, положить $(x_{ij})^* = x_{ij}^N, (y_{ji})^* = y_{ji}^N, K^* = K^N$ и перейти к шагу 12,

иначе перейти к шагу 12.

12. Проверить $N = N_{\max}$?

Если да, то перейти к шагу 14,

иначе перейти к шагу 13.

13. Положить $N = N + 1$ и перейти к шагу 2.

14. Ответ $L^*, (x_{ij})^*, (y_{ji})^*, K^*$ получен.

7. Вычислительный эксперимент

Все предложенные алгоритмы были программно реализованы в среде разработки Visual Studio 2019 на языке C# на ЭВМ с процессором Intel Pentium N4200, частотой 1,1 ГГц, оперативной памятью 4 Гб, 64-разрядной операци-

онной системой. С использованием программной реализации был проведен вычислительный эксперимент для сравнения результатов работы алгоритмов.

Для эксперимента были сгенерированы матрицы затрат различных размерностей по 500 шт. для каждого случая. В качестве затрат были взяты расстояния между объектами. Размерности задач выбраны исходя из практических соображений о том, что количество центров зачастую намного меньше количества целевых объектов. В последующих таблицах, отражающих результаты, приведены средние значения целевой функции и количества посещений места сбора — K . Заметим, что в таблицах представлены размерности без учета фиктивных объектов типов А и В.

Так как *алгоритмы 2 и 4* выполняются как *алгоритм 1* в случае нулевых начальных индексов, то в эксперименте для них выбирались начальные индексы, равные 1. Так же отметим, что результаты для самого *алгоритма 1* в таблицах не приведены, так как они схожи с результатами *алгоритма 2*.

В табл. 5 представлены значения целевой функции L^* , а также полученное значение K^* . Для адаптивного *алгоритма 6* выбраны следующие параметры: $N_{\max} = 100$, $p_i = 0, 1$.

В табл. 5 полужирным шрифтом выделены лучшие (или два близких) значения целевой функции для каждой размерности. Заметим, что для большинства тестовых данных лучшие результаты получены адаптивным алгоритмом, также хорошие результаты показывает *алгоритм 5* — итеративный алгоритм по целевым объектам. Стоит отметить, что при увеличении количества целевых объектов (тип А) значение целевой функции увеличивается. Это очевидно, так как увеличивается количество мест посещения для мобильных объектов. При увеличении количества центров (тип В) значение целевой функции уменьшается, что объясняется большей вариативностью при возврате от целевых объектов при построении маршрутов мобильными объектами. С точки зрения сравнения неитеративных *алгоритмов 2 и 4* лучшие результаты показывает *алгоритм 2*. Напротив, его итеративная модификация (*алгоритм 3*) работает хуже итеративного *алгоритма 5*. Это связано с большей вариативностью выбора в переборе целевых объектов, чем в переборе центров.

Заметим, что в данной статье ставится цель сравнения жадных алгоритмов с адаптивным и проведения соответствующего анализа. Однако, опираясь на результаты статьи [10], в которой использовались схожие данные для тестирования точного алгоритма, можно сделать вывод, что адаптивный алгоритм дает значения от 2 до 15% хуже оптимального в зависимости от размерности задачи.

В [15] доказано, что в случае, когда затраты заданы как расстояния, оптимальное значение K равно 1. Это подтверждается проведенным экспериментом. В табл. 5 наименьшие значения K соответствуют наименьшим значениям целевой функции. Также был проведен дополнительный регрессионный

Таблица 5. Значение целевой функции и количество мобильных объектов

тип В × тип А	Алгоритм 2		Алгоритм 3 (итер.)		Алгоритм 4		Алгоритм 5 (итер.)		Алгоритм 6 (адапт.)	
	L^*	K^*	L^*	K^*	L^*	K^*	L^*	K^*	L^*	K^*
1 × 10	1106	1,932	1106	1,932	1155	1,87	1071	1,298	1086	1
1 × 30	3188	3,67	3188	3,67	3255	3,582	3159	2,716	3052	1,002
1 × 50	5261	5,478	5261	5,478	5334	5,398	5237	4,502	4989	1,006
1 × 100	10513	10,012	10513	10,012	10591	9,932	10496	9,038	9891	1,012
2 × 10	723	1,004	711	1,004	778	1,006	709	1,001	768	1,034
2 × 30	1960	1,005	1956	1,002	2043	1,004	1954	1,001	2033	1,116
2 × 50	3211	1,004	3207	1,002	3303	1,01	3206	1,001	3298	1,258
2 × 100	6356	1,007	6354	1,003	6460	1,02	6353	1,003	6461	1,526
5 × 10	677	2,092	652	2,092	701	1,884	618	1,444	612	1,078
5 × 30	1516	1,798	1488	1,798	1592	2,334	1476	1,484	1447	1,11
5 × 50	2802	3,88	2770	3,88	2854	3,8	2745	3,21	2657	1,37
5 × 100	5447	2,02	5439	2,02	5547	2,024	5426	1,384	5428	1,816
8 × 10	598	1,512	583	1,512	644	1,648	564	1,162	553	1,042
8 × 30	1554	3,508	1550	3,508	1643	3,522	1529	2,626	1425	1,202
8 × 50	2383	2,726	2368	2,726	2439	2,878	2336	2,146	2281	1,28
8 × 100	5097	5,268	5041	5,268	5117	5,236	4981	4,13	4740	1,636
10 × 10	564	1,644	545	1,644	598	1,678	513	1,17	510	1,04
10 × 30	1282	3,08	1277	3,08	1349	3,022	1233	2,08	1168	1,126
10 × 50	2287	2,498	2283	2,498	2362	2,62	2259	1,804	2215	1,214
10 × 100	3465	6,13	3384	6,13	3443	5,512	3331	4,898	3255	1,566

анализ с целью выявления линейной зависимости между K и значениями целевой функции. Оценивалась модель по трем наборам данных: первая модель строилась по значениям всех алгоритмов для одной размерности, вторая модель строилась для фиксированного количества объектов типа В по результатам адаптивного алгоритма, третья модель — для фиксированного количества объектов типа А по результатам адаптивного алгоритма. В табл. 6 приведены коэффициенты детерминации полученных моделей.

Отметим, что для первого набора данных можно увидеть сильно выраженную линейную зависимость в большинстве случаев (коэффициент детерминации близок к 1), однако есть размерность, для которой линейная зависимость не подтверждена (эти случаи выделены в таблице полужирным шрифтом). Второй набор подтвердил линейную зависимость K от значений целевой функции. В третьем случае коэффициент детерминации варьируется от 0,56 до 0,66, что нельзя считать подтверждением линейной зависимости.

Исходя из данной таблицы, можно сделать предположение о линейной (или близкой к линейной) зависимости значения K от значений целевой функции. Однако для детального изучения данного вопроса требуется дополнительное исследование не в рамках представленной статьи.

Таблица 6. Коэффициенты детерминации

тип В × тип А	Коэффициент детерминации модели 1	тип В × тип А	Коэффициент детерминации модели 2	тип В × тип А	Коэффициент детерминации модели 3	
1 × 10	0,429364	1 × 10	0,98947	1 × 10	0,561744	
1 × 30	0,840595	1 × 30		2 × 10		
1 × 50	0,938962	1 × 50		5 × 10		
1 × 100	0,984294	1 × 100		8 × 10		
2 × 10	0,359795	2 × 10	0,995703	10 × 10	0,684219	
2 × 30	0,311298	2 × 30		1 × 30		
2 × 50	0,369441	2 × 50		2 × 30		
2 × 100	0,997976	2 × 100		5 × 30		
5 × 10	0,599609	5 × 10	0,984412	8 × 30	0,666925	
5 × 30	0,91009	5 × 30		10 × 30		
5 × 50	0,788263	5 × 50		1 × 50		
5 × 100	0,219113	5 × 100		2 × 50		
8 × 10	0,789282	8 × 10	5 × 50	8 × 50	0,653224	
8 × 30	0,775139	8 × 30	0,994379			10 × 50
8 × 50	0,854995	8 × 50	0,904579			1 × 100
8 × 100	0,964218	8 × 100				2 × 100
10 × 10	0,759837	10 × 10	0,904579	5 × 100	0,653224	
10 × 30	0,797575	10 × 30		8 × 100		
10 × 50	0,729778	10 × 50		10 × 100		
10 × 100	0,751988	10 × 100				

Таблица 7. Сравнение итеративных алгоритмов 3, 5 и 6

тип В × тип А	Алгоритм 3 (итер.)	Алгоритм 5 (итер.)	Алгоритм 6 (адапт.)	Алгоритм 6 (адапт.) 100 ит.
10 × 10	545	513	523	510
30 × 30	936	892	856	844
50 × 50	1300	1223	1159	1152
100 × 100	1647	1547	1501	1501

В табл. 5 для адаптивного алгоритма было выбрано фиксированное значение $N_{\max} = 100$ для всех размерностей. Однако в жадных итеративных алгоритмах количество итераций меняется в зависимости от размерности задачи. Было проведено сравнение трех итеративных алгоритмов при равном числе итераций в каждом из них. Для этого были сгенерированы дополнительные матрицы затрат размерностей 30×30 , 50×50 , 100×100 . Результаты представлены в табл. 7, в которой есть два столбца с адаптивным алгоритмом 6: в первом случае число итераций выбиралось равным размерности задачи, т.е. совпадало с количеством итераций алгоритмов 3 и 5, во втором — $N_{\max} = 100$.

Видно, что алгоритм 3 (итеративный алгоритм центров) работает хуже алгоритма 5 (итеративный алгоритм целевых объектов). Адаптивный ал-

Таблица 8. Сравнение итеративных *алгоритмов 5 и 6*

тип В × тип А	<i>Алгоритм 5</i> (итер.)	<i>Алгоритм 6</i> (итер.)	<i>Алгоритм 6</i> (адапт.) 100 ит.
1 × 10	1071	1086	1086
1 × 30	3159	3052	3052
1 × 50	5237	4989	4989
1 × 100	10496	9891	9891
2 × 10	709	768	768
2 × 30	1954	2033	2003
2 × 50	3206	3298	3298
2 × 100	6353	6461	6461
5 × 10	618	616	612
5 × 30	1476	1451	1447
5 × 50	2745	2658	2657
5 × 100	5426	5428	5428
8 × 10	564	564	553
8 × 30	1529	1431	1425
8 × 50	2336	2283	2281
8 × 100	4981	4740	4740
10 × 10	513	523	510
10 × 30	1233	1175	1168
10 × 50	2259	2217	2215
10 × 100	3331	3255	3255

алгоритм 6 дает лучший или схожий результат при одинаковом с другими алгоритмами числе итераций, а при их увеличении улучшает собственный результат.

Аналогичное сравнение представлено в табл. 8. Здесь адаптивный алгоритм сравнивается с *алгоритмом 5* при равном числе итераций в каждом из них. Также здесь представлены данные из табл. 5 с $N_{\max} = 100$.

Только в задачах с количеством объектов типа В, равным 2, *алгоритм 5* дает результаты значительно лучше, чем адаптивный *алгоритм 6*.

В табл. 9 представлены результаты по времени работы алгоритмов в миллисекундах.

Очевидно, что *алгоритмы 2 и 4* работают очень быстро. Итеративный *алгоритм 3* перебора центров также работает быстро, при этом время его работы в большей степени зависит от количества центров. Время работы итеративного *алгоритма 5* зависит от количества целевых объектов, поэтому оно существенно больше времени работы *алгоритма 3*, так как количество целевых объектов превосходит количество центров. Адаптивный *алгоритм 6* работает намного дольше всех остальных алгоритмов. Это связано и с заданным количеством итераций, и с пересчетом вероятностей и, главное, с необходимостью дополнительных вычислений сумм внутри поиска минимума на шаге 6. Дополнительные модификации алгоритмов подсчета и пересчета данных сумм могут существенно улучшить время его работы.

Таблица 9. Время работы алгоритмов

тип В × тип А	Алгоритм 2	Алгоритм 3 (итер.)	Алгоритм 4	Алгоритм 5 (итер.)	Алгоритм 6 (адапт.)
1 × 10	< 0,001	< 0,001	< 0,001	0,002	23,19
1 × 30	< 0,001	< 0,001	< 0,001	1,006	51,738
1 × 50	0,012	0,001	0,002	4,37	74,648
1 × 100	< 0,001	0,002	< 0,001	28,63	95,238
2 × 10	< 0,001	< 0,001	< 0,001	0,003	22,236
2 × 30	< 0,001	< 0,001	< 0,001	1,002	44,26
2 × 50	< 0,001	0,002	0,001	4,324	66,134
2 × 100	0,004	0,014	0,002	27,744	119,342
5 × 10	< 0,001	< 0,001	< 0,001	0,004	23,758
5 × 30	< 0,001	< 0,001	0,001	1,006	49,384
5 × 50	< 0,001	0,004	< 0,001	4,67	81,254
5 × 100	< 0,001	1	< 0,001	30,418	185,048
8 × 10	< 0,001	0,001	< 0,001	0,002	25,652
8 × 30	< 0,001	0,001	< 0,001	1,036	63,858
8 × 50	< 0,001	0,01	< 0,001	4,766	95,93
8 × 100	< 0,001	2,002	< 0,001	29,062	217,754
10 × 10	< 0,001	< 0,001	< 0,001	0,004	25,658
10 × 30	< 0,001	0,001	< 0,001	1,038	56,93
10 × 50	< 0,001	0,144	< 0,001	4,876	97,696
10 × 100	< 0,001	2,83	< 0,001	30,914	238,14

Таблица 10. Влияние параметра p_u

p_u	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1
5 × 30	1447	1457	1466	1473	1479	1485	1490	1494	1498	1503
p_u	0,01	0,02	0,03	0,04	0,05	0,06	0,07	0,08	0,09	0,1
5 × 30	1438	1439	1439	1441	1441	1442	1444	1445	1446	1447
p_u	0,001	0,002	0,003	0,004	0,005	0,006	0,007	0,008	0,009	0,01
5 × 30	1485	1460	1450	1445	1443	1441	1439	1438	1439	1438

Таблица 11. Влияние количества итераций

N_{\max}	10	20	30	40	50	60	70	80	90	100
5 × 30	1495	1466	1454	1448	1445	1442	1440	1439	1438	1438
N_{\max}	100	200	300	400	500	600	700	800	900	1000
5 × 30	1438	1437	1436	1436	1436	1436	1436	1436	1436	1436

Важным преимуществом адаптивного алгоритма 6 по сравнению с алгоритмами 2–5 является наличие настраиваемых параметров: p_u и N_{\max} . Их грамотный выбор может улучшить поисковые возможности алгоритма.

В табл. 10 и 11 представлены результаты по настройке указанных параметров. В табл. 10 $N_{\max} = 100$.

В ходе первого эксперимента с параметром p_u его значения варьировались от 0,1 до 1. Было выявлено, что результаты улучшаются при уменьшении значений параметра. Поэтому в следующем эксперименте были взяты значения от 0,01 до 0,1. В нем также получилось, что результаты лучше при малых значениях. Однако в третьем эксперименте, со значениями от 0,001 до 0,01, данная тенденция не сохранилась, и с последующим уменьшением значения параметра p_u значения целевой функции стали расти. Таким образом, было выявлено оптимальное значения параметра p_u , равное 0,01. Стоит сделать оговорку, что данное значение параметра является оптимальным именно для размерности 5×30 .

Далее был проведен эксперимент по подбору максимального значения для количества итераций N_{\max} при заданном значении $p_u = 0,01$.

Так как алгоритм зависит от значений вероятности, которые изменяются и настраиваются от итерации к итерации, то логично предположить, что при увеличении количества итераций, результаты будут улучшаться. Это предположение подтвердилось в двух проведенных экспериментах. Заметим, что при увеличении максимального числа итераций от 10 до 100 происходит заметное улучшение результатов работы алгоритма, но при дальнейшем увеличении этого числа от 100 до 1000 результаты изменились на 2 единицы. При дальнейшем увеличении до 5000 значение целевой функции уменьшилось еще на 1, т.е. стало 1435. Таким образом, можно сделать вывод, что увеличение числа итераций более 100 не приводит к существенному улучшению результатов работы алгоритма.

Итак, по результатам проведенных экспериментов получены следующие обобщенные выводы:

1. Неитеративные жадные алгоритмы работают быстро, но их результаты хуже остальных алгоритмов.

2. Итеративные жадные алгоритмы улучшают результаты неитеративных алгоритмов, но выполняются дольше. Итеративный алгоритм целевых объектов дает среднее значение целевой функции меньше, чем итеративный алгоритм центров. Причем это подтверждается и когда целевых объектов больше центров (т.е. количество итераций алгоритмов разное), и когда число тех и других объектов совпадает (т.е. количество итераций одинаковое).

3. Адаптивный алгоритм дает значение целевой функции меньше других алгоритмов, однако время его работы значительно больше. Это опять же подтверждается для разного числа итераций в адаптивном алгоритме по сравнению с итеративными жадными.

4. Адаптивный алгоритм зависит от параметров, их грамотная настройка позволяет улучшить его работу.

5. Сделано предположение о линейной зависимости значения K от значений целевой функции.

8. Заключение

В данной статье представлена задача маршрутизации транспортных средств с несколькими центрами и с чередованием объектов. Построены математические модели с двумя блоками переменных для задач с единым местом сбора и без. Для задачи с единым местом сбора предложены жадные, итеративные жадные и адаптивный алгоритмы решения. На основе программной реализации алгоритмов проведен вычислительный эксперимент. Главные его результаты приведены в соответствующем разделе. В дальнейшем подробное исследование следует посвятить зависимости значения K от значений целевой функции. Так как адаптивный алгоритм сильно зависит от параметров, то дальнейшие исследования должны быть направлены на поиск их оптимальных значений в зависимости от размерностей задачи. Также необходимо рассмотреть функциональное изменение параметра p_u в зависимости от номера текущей итерации или от расстояния Хэмминга между двумя решениями.

СПИСОК ЛИТЕРАТУРЫ

1. Демиденко В.М. Модели маршрутизации транспортных средств в товаропроводящих сетях // Экономика, моделирование, прогнозирование. 2012. № 6. С. 94–106.
2. Веларде М., Литвинчев И.С., Цедильо Г. Интегрированная модель маршрутизации транспортных средств и построения зон обслуживания // Известия РАН. Теория и системы управления. 2017. № 6. С. 74–79.
3. Косоногова Л.Г., Королёва А.А., Дубасов А.В. Анализ оптимального распределения транспортного потока при маршрутизации количества транспортных средств // Вестник: научный журнал. 2021. № 6 (48). С. 81–85.
4. Юсупова Н.И., Валеев Р.С. Об одной задаче маршрутизации для доставки однородного продукта различным клиентам автомобильными транспортными средствами // Современные наукоемкие технологии. 2020. № 4. С. 84–88.
5. Yi Zhoua, Weidong Lib, Xiaomao Wanga, Yimin Qiua, Weiming Shen Adaptive gradient descent enabled ant colony optimization for routing problems // Swarm and evolutionary computation. 2022. Vol. 70 (3). <https://doi.org/10.1016/j.swevo.2022.101046>
6. Ramalingam A., Vivekanandan K. Genetic algorithm based solution model for multi-depot vehicle routing problem with time windows // International journal of advanced research in computer and communication engineering. 2014. Vol. 3. Issue 11. pp. 8433–8439.
7. Mazidi A., Fakhrahmad M., Sadreddini M. A Meta-heuristic approach to CVRP problem: local search optimization based on GA and ant colony // Journal of advances in computer research. 2016. Vol. 7. No. 1. pp. 1–22.
8. Medvedev S.N., Medvedeva O.A., Zueva Y.R., Chernyshova G.D. Formulation and algorithmization of the interleaved vehicle routing problem // Journal of Physics: Conference Series 1203 012053. 2019. <https://doi.org/10.1088/1742-6596/1203/1/012053>

9. *Medvedev S., Sorokina A., Medvedeva O.* The vehicle routing problem for several agents among the objects of two types // 2019 XXI International Conference Complex Systems: Control and Modeling Problems (CSCMP), Samara, Russia. 2019. pp. 535–540.
10. *Медведев С.Н.* Математическая модель и алгоритм решения задачи маршрутизации транспортных средств с несколькими центрами с чередованием и единым местом сбора // Вестник Воронежского государственного университета. Серия: Системный анализ и информационные технологии. 2021. № 1. С. 21–32.
11. *Кензин М.Ю., Бычков И.В., Максимкин Н.Н.* Комплексный многоцелевой мониторинг группой автономных транспортных средств // Известия ЮФУ. Технические науки. 2019. № 7 С. 82–92.
12. *Кристофидес Н.* Теория графов. Алгоритмический подход. М.: Мир, 1978. 432 с.
13. *Гольштейн Е.Г., Юдин Д.Б.* Задачи линейного программирования транспортного типа. М.: Наука, 1969. 382 с.
14. *Медведев С.Н., Медведева О.А.* Адаптивный алгоритм решения аксиальной трехиндексной задачи о назначениях // АиТ. 2019. № 4. С. 156–172.
Medvedev S.N., Medvedeva O.A. An adaptive algorithm for solving the axial three-index assignment problem // Autom. Remote Control. 2019. V. 80. No. 4. P. 718–732.
15. *Медведев С.Н.* Об оптимальном решении задачи маршрутизации транспорта с чередованием с единым местом сбора // Современные методы прикладной математики, теории управления и компьютерных технологий (ПМТУКТ-2021): сборник трудов Всероссийской научной конференции, Воронеж. 2021. С. 97–101.

Статья представлена к публикации членом редколлегии А.А. Лазаревым.

Поступила в редакцию 22.07.2022

После доработки 28.09.2022

Принята к публикации 26.10.2022