УДК 519.85

# *КЛАСТЕРИЗАЦИОННЫЙ ПОДХОД К РЕШЕНИЮ ЗАДАЧИ НЕСКОЛЬКИХ КОММИВОЯЖЁРОВ В MINIMAX-ПОСТАНОВКЕ*

***Расторгуев А.В.***

*студент,*

*Российский университет транспорта (МИИТ),*

*Москва, Россия*

***Иванова А.П.***

*к.ф.-м.н., доцент,*

*Российский университет транспорта (МИИТ),*

*Москва, Россия*

**Аннотация:**

В работе рассматривается задача нескольких коммивояжёров в minimax-постановке. Сравниваются три способа построения решения: случайное распределение городов между коммивояжёрами, разбиение городов методом k-средних (KMeans) и подход на основе специализированного решателя OR-Tools. Для каждого экземпляра задачи вычисляется J(y) – максимальная длина маршрута среди всех коммивояжёров. Вычислительный эксперимент проводится на случайно сгенерированных экземплярах задачи размером до 8 коммивояжёров и 30 городов. Показано, что KMeans заметно улучшает результат по сравнению со случайным распределением, но уступает OR-Tools, который учитывает маршрутизационную структуру задачи более полно.

**Ключевые слова:** задача нескольких коммивояжёров, MTSP, TSP, KMeans, OR-Tools, маршрутизация, кластеризация.

# *A CLUSTERING-BASED APPROACH TO THE MINIMAX MULTIPLE TRAVELING SALESMAN PROBLEM*

***Rastorguev A.V.***
*student,*
*Russian University of Transport (MIIT),*
*Moscow, Russia*
***Ivanova A.P.***
*Ph.D., Associate Professor,*
*Russian University of Transport (MIIT),*
*Moscow, Russia*

**Abstract:** The paper studies the multiple traveling salesman problem in the minimax formulation. A clustering-based approach is considered: cities are divided between salesmen using the k-means method, and local routes are then constructed for the obtained groups. The approach is compared with random assignment and OR-Tools. The quality of a solution is measured by J(y), the length of the longest route among all salesmen. Computational experiments on randomly generated instances up to 8 salesmen and 30 cities show that k-means significantly improves upon random assignment, while OR-Tools provides the strongest practical reference result.

**Keywords**: multiple traveling salesman problem, MTSP, k-means, OR-Tools, routing, clustering.

## Введение

Задачи маршрутизации широко применяются в логистике, распределении заказов, транспортном планировании и сервисных системах. В классической задаче коммивояжёра требуется построить один маршрут, проходящий через все города.

Задача нескольких коммивояжёров, или Multiple Traveling Salesman Problem (MTSP), является её обобщением: в ней используется несколько коммивояжёров, а города необходимо не только упорядочить внутри маршрутов, но и распределить между исполнителями. Такая постановка относится к задачам дискретной оптимизации и маршрутизации на графах [1-5].

В данной работе рассматривается minimax-постановка MTSP. В этом случае качество решения определяется не суммарной длиной маршрутов, а длиной самого длинного маршрута среди всех коммивояжёров. Такой критерий отражает задачу балансировки нагрузки, поскольку время завершения всей работы определяется наиболее загруженным маршрутом.

Целью работы является экспериментальное сравнение трёх методов решения minimax-MTSP: случайного распределения городов, разбиения методом k-средних (KMeans) и решения с использованием OR-Tools. Основной акцент сделан на том, насколько полезна простая геометрическая кластеризация как приближённый способ распределения городов между коммивояжёрами.

**1. Постановка задачи**

Пусть задано множество точек $X = \{x_0, x_1, \dots, x_n\}$, где $x_0$ является общей начальной и конечной точкой маршрутов, а $x_1, \dots, x_n$ – города. Все точки расположены на плоскости. Расстояние между двумя точками определяется стандартной евклидовой метрикой:

$$d(x_i, x_j) = \sqrt{(x_i^1 - x_j^1)^2 + (x_i^2 - x_j^2)^2}.$$

Требуется распределить $n$ городов между $m$ коммивояжёрами. Каждый город должен быть назначен ровно одному коммивояжёру. Вектор распределения обозначим через $y$, где $y_i$ показывает номер коммивояжёра, которому назначен $i$-й город. Для каждого коммивояжёра a формируется подмножество городов $S_a(y)$,

после чего строится замкнутый маршрут, начинающийся и заканчивающийся в депо:

$$J(y) = max\, L_a \;\rightarrow\; min, \qquad a = 1, \dots, m,$$

где $L_a$ — длина маршрута коммивояжёра $a$.

Значение $J(y)$ показывает длину самого длинного маршрута и далее используется как основная метрика качества решения. Задача состоит в поиске такого распределения городов, при котором $J(y)$ минимально. Уже на уровне распределения существует $m^n$ вариантов, поэтому полный перебор быстро становится непрактичным.

**2. Описание методов**

В дальнейшем под методом Random понимается случайное распределение городов между коммивояжёрами. Этот метод используется как нижняя контрольная планка: он показывает качество решения без учёта координат, расстояний и положения депо. Для каждого города независимо выбирается один из $m$ коммивояжёров с одинаковой вероятностью:

$$P(y_\text{i} = a) = \frac{1}{m},\ a = 1, \dots, m.$$

Такой способ не предназначен для построения качественного маршрута, но нужен для проверки того, насколько остальные методы используют геометрическую структуру задачи.

Метод k-средних, далее KMeans, использует координаты городов. В кластеризацию передаются только города, а депо не участвует в вычислении центроидов. Города разбиваются на $m$ кластеров, после чего номер кластера интерпретируется как номер коммивояжёра. Стандартная целевая функция метода имеет вид [6, 7]:

$$\sum_{i=1}^{n} \min_{a=1,\dots,m} \|x_i - \mu_\text{a}\|^2 \rightarrow \min.$$

После разбиения каждый кластер становится подмножеством городов одного коммивояжёра. KMeans не решает MTSP напрямую: он минимизирует внутрикластерную сумму квадратов расстояний до центроидов, а не значение $J(y)$. Тем не менее компактные геометрические группы обычно приводят к более коротким локальным маршрутам, чем случайное распределение.

OR-Tools рассматривается как сильный метод сравнения на основе специализированного решателя. В отличие от Random и KMeans, он не ограничивается предварительным разбиением городов, а строит маршрутизационную модель для нескольких коммивояжёров. OR-Tools является открытой библиотекой Google для задач оптимизации и маршрутизации [8]. В настоящей работе его результат используется как практическая опорная планка при заданном лимите времени, а не как доказательство глобального оптимума.

Для Random и KMeans применяется двухэтапная схема: сначала строится распределение городов между коммивояжёрами, затем для каждого коммивояжёра решается локальная задача коммивояжёра на множестве, состоящем из депо и назначенных ему городов. Такая декомпозиция соответствует общей логике рассмотрения маршрутизационных задач как задач на графах и позволяет сравнивать разные способы распределения городов на единой метрике $J(y)$ [1-4].

Маршрут внутри каждого кластера строился отдельно. Для соответствующего коммивояжёра формировалась локальная TSP-подзадача, включающая депо и все города данного кластера. В вычислительном эксперименте эта подзадача решалась TSP-решателем OR-Tools для одного коммивояжёра [8]; найденная длина замкнутого маршрута затем использовалась при вычислении $J(y)$. Такой шаг необходим, поскольку Random и KMeans задают только распределение городов, но не порядок их обхода; локальные TSP-подзадачи при этом относятся к

классическому классу задач коммивояжёра, для которых широко применяются эвристические процедуры построения маршрутов [9, 10].

**3. Вычислительный эксперимент**

Вычислительный эксперимент был реализован на языке Python с использованием стандартных средств обработки данных и численных вычислений [11].

Экземпляры задачи генерировались случайно. Координаты городов выбирались независимо и равномерно в единичном квадрате, а депо фиксировалось в центре квадрата:

$$x_i \sim U([0,1]^2), \qquad x_0 = (0.5,0.5).$$

Рассматривались пять размеров задач: 2×12, 4×20, 6×24, 8×24 и 8×30. Запись $m \times n$ означает, что в экземпляре задачи используется $m$ коммивояжёров и $n$ городов. Для каждого размера было использовано 128 независимых экземпляров задачи, всего 640 экземпляров. Все методы применялись к одним и тем же экземплярам задачи, поэтому сравнение является парным. Для каждого метода фиксировалось значение $J(y)$.

Таблица 1. – Параметры вычислительного эксперимента. Авторская разработка

| Параметр | Значение |
|---|---|
| Область генерации городов | $x_i \sim U([0,1]^2)$ |
| Координаты депо | (0.5, 0.5) |
| Размеры задач | 2×12, 4×20, 6×24, 8×24, 8×30 |
| Экземпляров задачи на размер | 128 |
| Всего экземпляров задачи | 640 |
| Методы | Random, KMeans, OR-Tools |
| Ограничение времени OR-Tools | 1 секунда на экземпляр задачи |
| Основная метрика | $J(y)$ – максимальная длина маршрута |

## 4. Результаты

В таблице 2 приведена общая статистика по всем 640 экземплярам задачи. Меньшее значение $J(y)$ соответствует более качественному решению.

Таблица 2. – Общая статистика качества методов. Авторская разработка

| Метод | $n$ | Среднее значение $J(y)$ | Медианное значение | 95% доверительный интервал для среднего значения $J(y)$ |
|---|---|---|---|---|
| Random | 640 | 2.693 | 2.669 | [2.669; 2.717] |
| KMeans | 640 | 1.663 | 1.568 | [1.639; 1.687] |
| OR-Tools | 640 | 1.477 | 1.377 | [1.457; 1.499] |

На рис. 1 видно, что Random остаётся худшим методом на всех размерах задач. KMeans снижает среднее значение $J(y)$ за счёт геометрически осмысленного разбиения городов. Наименьшие значения $J(y)$ показывает OR-Tools, поэтому он используется как наиболее сильная практическая планка сравнения.

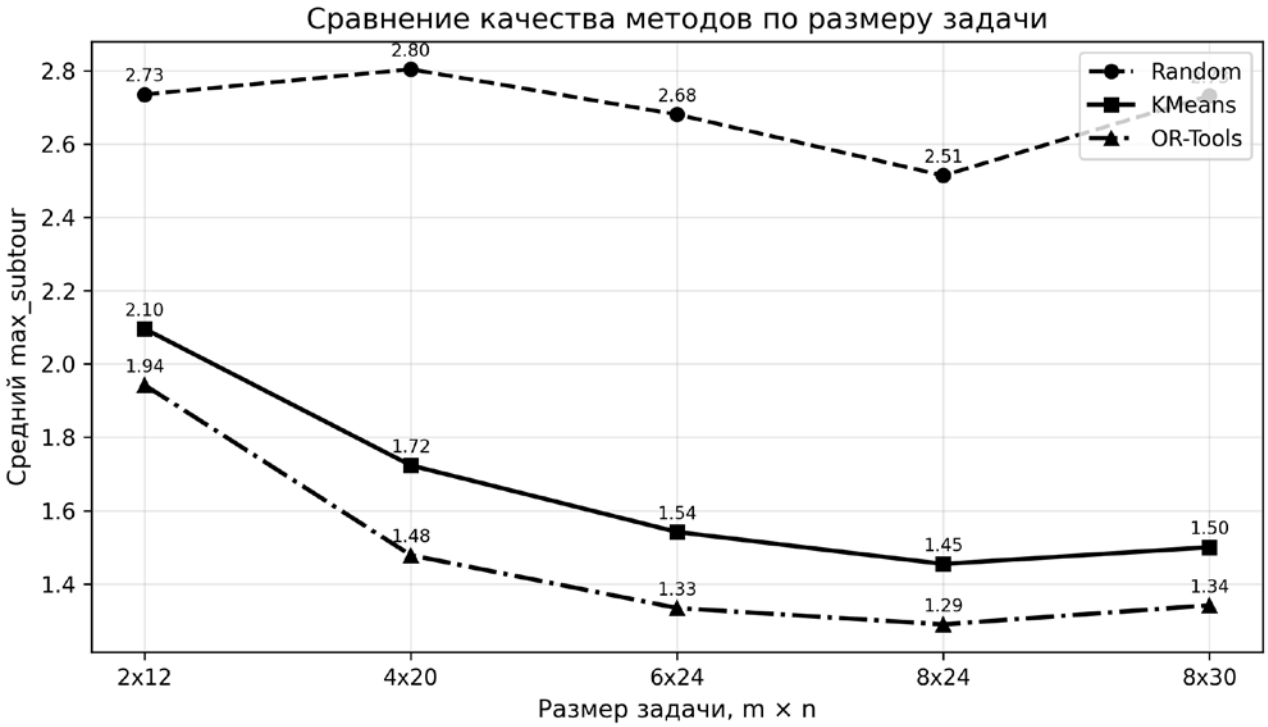


Рис. 1. – Сравнение среднего значения $J(y)$ по размерам задач. Авторская разработка.

Распределения на рис. 2 подтверждают тот же порядок качества. Для крупнейшего случая 8×30 метод KMeans заметно устойчивее случайного распределения, однако OR-Tools имеет более низкую медиану и меньший разброс значений $J(y)$.

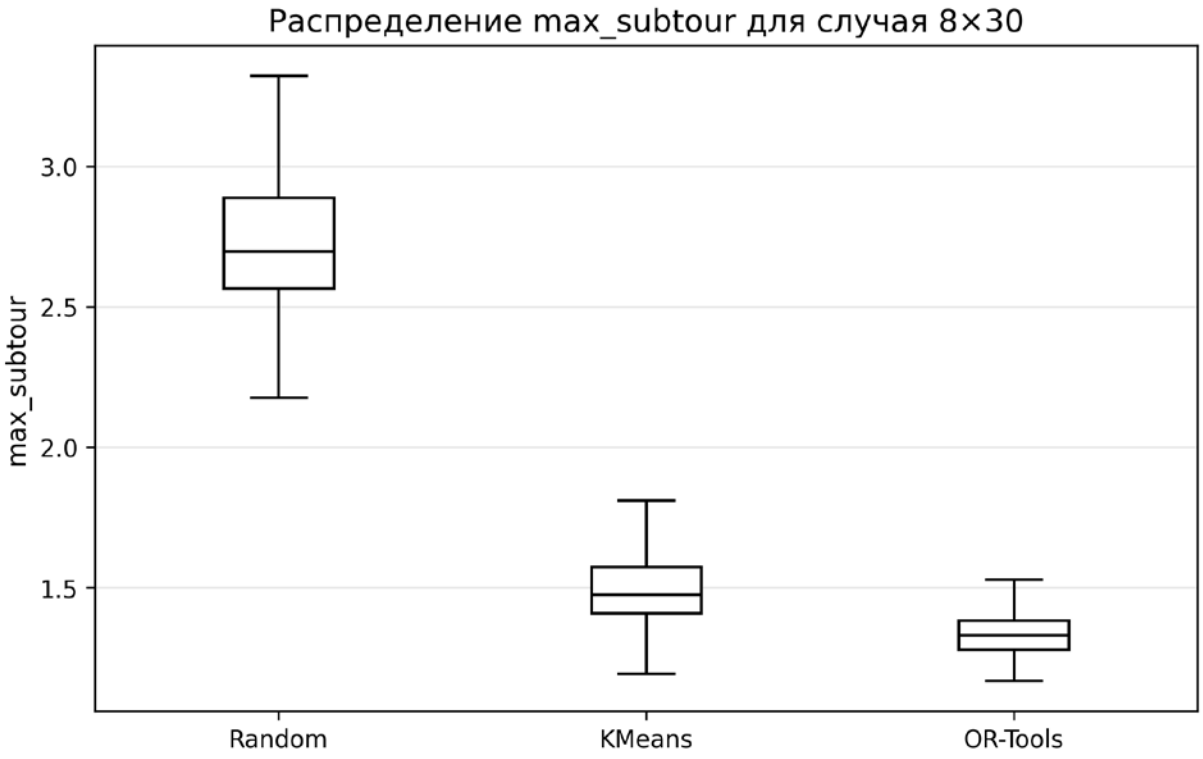


Рис. 2. – Распределение $J(y)$ для случая 8×30. Авторская разработка

В таблице 3 и на рис. 3 показаны результаты парного сравнения методов.

Таблица 3. – Парное сравнение методов. Авторская разработка

| Сравнение | $n$ | Среднее улучшение $J(y)$, % | Доля выигрышей, % | Вывод |
|---|---|---|---|---|
| KMeans vs Random | 640 | 37.71 | 99.4 | KMeans лучше Random |
| OR-Tools vs Random | 640 | 44.69 | 100.0 | OR-Tools лучше Random |
| OR-Tools vs KMeans | 640 | 10.55 | 86.7 | OR-Tools лучше KMeans |

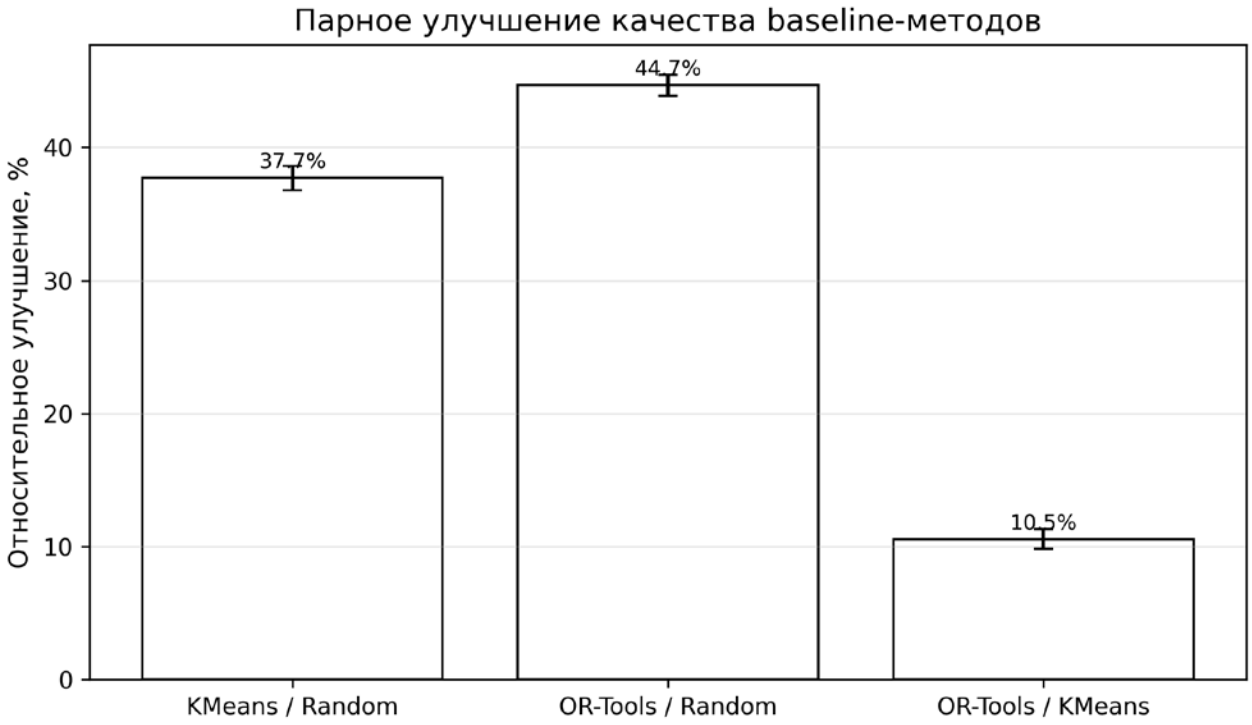


Рис. 3. – Парное относительное улучшение качества методов. Авторская разработка

Числовые показатели полезно дополнить визуальным анализом маршрутов. На рис. 4 и рис. 5 показаны решения для одного и того же набора городов в случае 8×30. Медианным примером называется экземпляр задачи, для которого значение $J(y)$ близко к медианному результату метода на выбранной группе задач. Это позволяет сопоставить не только численную метрику, но и геометрическую структуру построенных маршрутов.

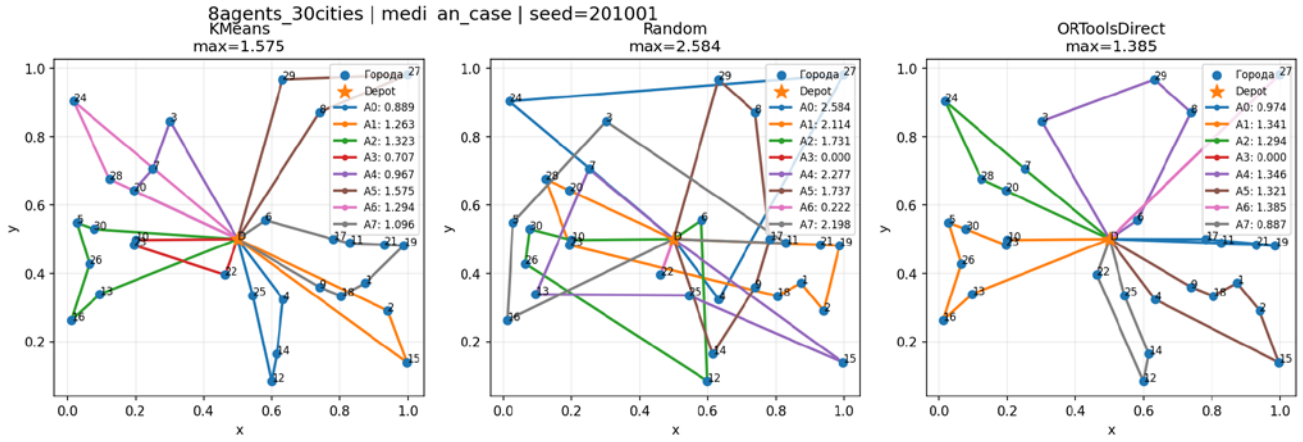


Рис. 4. – Визуальное сравнение KMeans, Random и OR-Tools на медианном примере 8×30. Авторская разработка

Случайное распределение формирует пространственно разорванные маршруты: города одного коммивояжёра могут находиться в разных частях плоскости. KMeans строит более компактные зоны обслуживания, поэтому его маршруты выглядят естественнее. OR-Tools дополнительно учитывает порядок обхода городов и обычно получает более сбалансированное решение.

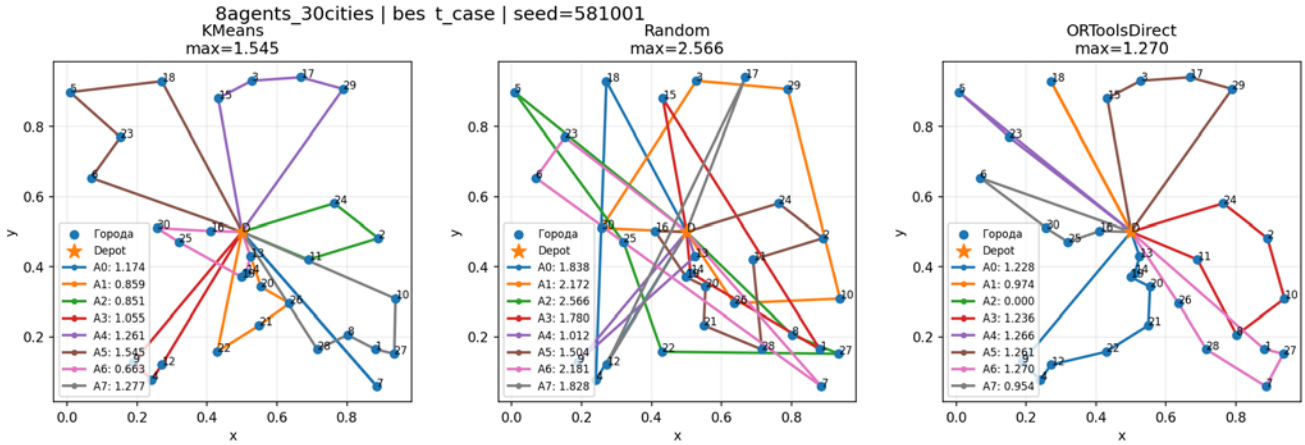


Рис. 5. – Дополнительный пример маршрутов для случая 8×30. Авторская разработка

**5. Обсуждение результатов**

Проведённый вычислительный эксперимент показывает устойчивый порядок качества: Random уступает KMeans, а KMeans уступает OR-Tools. Среднее значение $J(y)$ для Random составило 2.693, для KMeans – 1.663, для OR-Tools – 1.477. Переход от случайного распределения к KMeans даёт среднее относительное улучшение 37.71%, а переход от KMeans к OR-Tools – ещё 10.55%.

Преимущество KMeans связано с использованием геометрии задачи. Близко расположенные города объединяются в кластеры, поэтому локальные маршруты обычно становятся компактнее. Ограничение метода состоит в том, что он не оптимизирует $J(y)$ напрямую и не учитывает порядок обхода городов во время кластеризации.

OR-Tools показывает более сильный результат, поскольку работает не только с разбиением точек, но и с маршрутизационной моделью. Однако его применение требует задания параметров поиска и учёта вычислительного бюджета. Поэтому KMeans можно рассматривать как быстрый и интерпретируемый приближённый метод, а OR-Tools – как более сильный метод сравнения на основе специализированного решателя.

Сравнение времени в данной работе не выносилось в основной результат. Random и KMeans сначала распределяют города, а затем оцениваются через локальные TSP-подзадачи, тогда как OR-Tools решает задачу маршрутизации для нескольких коммивояжёров напрямую. Поэтому время полного вычислительного контура не является чистым временем работы отдельного метода.

**Заключение**

В работе выполнено сравнение трёх методов решения задачи нескольких коммивояжёров в minimax-постановке: Random, KMeans и OR-Tools. Метод Random использован как нижняя контрольная планка. Метод KMeans применён как

простая геометрическая эвристика распределения городов между коммивояжёрами. OR-Tools использован как сильный метод сравнения на основе специализированного решателя.

На основании проведённого вычислительного эксперимента можно сделать следующие выводы:

1. Случайное назначение городов даёт худшее качество, поскольку не использует геометрию задачи.

2. KMeans существенно улучшает решение относительно Random за счёт формирования компактных кластеров городов.

3. OR-Tools показывает наилучшее качество среди рассмотренных методов.

4. KMeans может использоваться как быстрый, простой и интерпретируемый метод начального разбиения городов для minimax-MTSP.

**Библиографический список:**

1. Сигал, И. Х. Введение в прикладное дискретное программирование: модели и вычислительные алгоритмы: учебное пособие / И. Х. Сигал, А. П. Иванова. – 2-е изд., испр. и доп. – М.: ФИЗМАТЛИТ, 2007. – 304 с.

2. Лекции по теории графов / Емеличев В.А., Мельников О.И., Сарванов В.И., Тышкевич Р.И. – М.: Наука, 1990. – 384 с.

3. Кристофидес Н. Теория графов. Алгоритмический подход. – М.: Мир, 1978. – 432 с.

4. Иванова А.П. Теория графов. Часть 2: учебное пособие по дисциплине «Теория графов». – М.: РУТ (МИИТ), Янус-К, 2026. – 99 с.

5. Bektas T. The multiple traveling salesman problem: an overview of formulations and solution procedures // Omega. 2006. – Vol. 34. – № 3. – P. 209-219.

6. MacQueen J. Some methods for classification and analysis of multivariate observations // Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability. – 1967. – Vol. 1. – P. 281-297.

7. Lloyd S. Least squares quantization in PCM // IEEE Transactions on Information Theory. – 1982. – Vol. 28. – № 2. – P. 129-137.

8. Perron L., Furnon V. OR-Tools. Google. URL: https://developers.google.com/optimization/ (дата обращения: 14.05.2026).

9. Rosenkrantz D. J., Stearns R. E., Lewis P. M. An analysis of several heuristics for the traveling salesman problem // SIAM Journal on Computing. 1977. – Vol. 6. – № 3. – P. 563-581.

10. Helsgaun K. An effective implementation of the Lin–Kernighan traveling salesman heuristic // European Journal of Operational Research. 2000. – Vol. 126. – № 1. – P. 106-130.

11. Python Software Foundation. Документация Python 3. URL: https://docs.python.org/ru/3/ (дата обращения: 14.05.2026).