Государственное бюджетное общеобразовательное учреждение города Москвы "Школа № 1532"

**Клиент-серверное решение задачи коммивояжёра с использованием роботизированных систем**

10 класс, ГБОУ Школа №1532,

Саватеев Дмитрий Витальевич

Руководитель: учитель информатики, ГБОУ Школа №1532,

Сергиенко Антон Борисович

Москва 2023

# Содержание

[Содержание 2](#_Toc128493080)

[Введение 3](#_Toc128493081)

[Основная часть 4](#_Toc128493082)

[Точные методы 4](#_Toc128493083)

[Полный перебор 4](#_Toc128493084)

[Метод ветвей и границ 6](#_Toc128493085)

[Приближённые и эвристические методы 7](#_Toc128493086)

[Жадный алгоритм 7](#_Toc128493087)

[Генетический алгоритм 8](#_Toc128493088)

[Метод Монте-Карло 11](#_Toc128493089)

[Модификация генетического алгоритма 12](#_Toc128493090)

[Сравнение эффективности алгоритмов 14](#_Toc128493091)

[Взаимодействие с пользователем 19](#_Toc128493092)

[Робототехническая часть 21](#_Toc128493093)

[Заключение 23](#_Toc128493094)

[Список использованных источников 24](#_Toc128493095)

# Введение

Задача коммивояжёра – задача комбинаторной оптимизации, заключающаяся в поиске самого короткого маршрута, проходящего через указанные города по одному разу с возвратом в исходный город. Задача является актуальной, так как имеет множество практических применений, таких как маршрутизация транспорта и служит основой транспортной логистики - отрасли, занимающейся планированием транспортных перевозок. Автором предложено усложнение стандартного решения проблемы коммивояжёра путём добавления приоритетных городов. Задача относится к NP полным это означает, что не существует полных решений за полиномиальное время. В данном проекте, помимо возможных алгоритмов решения, будет рассмотрено использование роботехнических систем для решения задачи коммивояжёра.

Решение задачи коммивояжёра с использованием роботехнических систем может применяться в различных областях, таких как автоматизированная логистика и транспорт, системы управления производством, проведение спасательных операций, работа в сельскохозяйственных угодиях и другие.

Цель работы заключается в разработке алгоритмов решения стандартной задачи коммивояжёра и предложенной автором задачи коммивояжёра с приоритетными городами с использованием управляемых робототехнических систем с возможностью автономной работы в условиях отсутствия управления со стороны человека, а также анализ эффективности и возможности практического применения данного решения. А поскольку микропроцессор робота не позволяет решать сложные комбинаторные задачи, необходима разработка клиент-серверной архитектуры.

# Основная часть

Изначально имеются координаты городов в декартовой системе координат, цель задачи заключается в том, чтобы найти наикратчайший маршрут, который проходит через все города и возвращается в исходный город. В процессе работы были рассмотрены следующие алгоритмы: полный перебор, жадный алгоритм, метод ветвей и границ и др. Для решений задач с использованием предложенных алгоритмов предусматривается построение интерактивного графика. На графике синие точки обозначают города, через которые проходит найденный путь, красная – начало пути, разноцветные прямые соединяют города-вершины между собой, образуя неориентированный граф, являющийся кратчайшим маршрутом.

Для каждого алгоритма автором было **предложено усложнение в виде добавления приоритетных городов в условии.** Такие города на графике отмечаются большими оранжевыми точками. В процессе нахождения оптимального решения в первую очередь (но не обязательно) учитываются города с приоритетом, а после них находится кратчайший путь между остальными городами.

## Точные методы

Точные методы осуществляют полный перебор всех вариантов и находят точный маршрут, существуют алгоритмы, оптимизирующие эту задачу, поэтому перебор может занять не слишком много времени.

**Полный перебор** – точный алгоритм, находящий решение путём рассмотрения всех возможных вариантов. Уровень сложности при таком алгоритме напрямую зависит от количества городов в условии задачи и равен *,* где n – количество рассматриваемых городов.

Если городов будет 5, то у нас будет 120 вариантов, которые надо перебрать, с этим компьютер сможет быстро справится, но, если вариантов будет всего на 10 больше, количество возможных путей возрастёт до 1307674368000 вариантов. Для перебора такого количества возможных путей потребуются колоссальные временные и ресурсные затраты, поэтому данный метод является неактуальным, особенно при практическом применении решения задачи коммивояжёра.

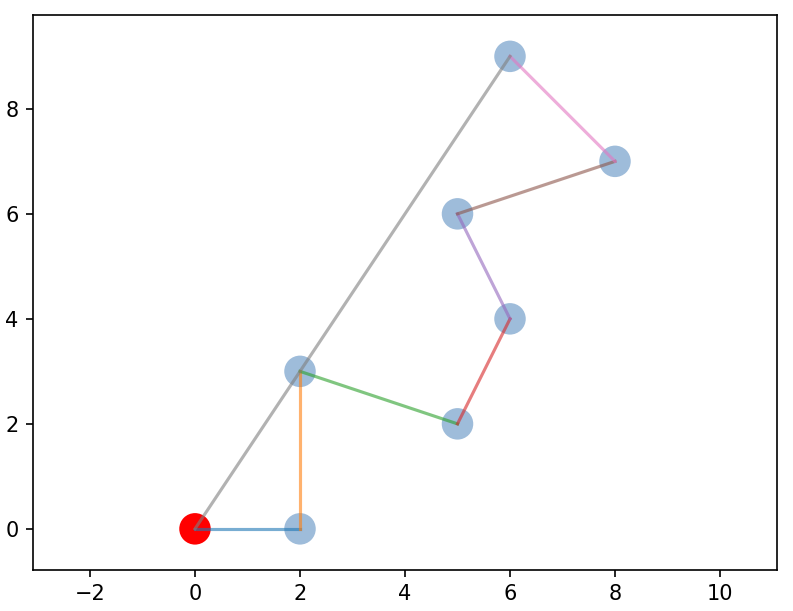
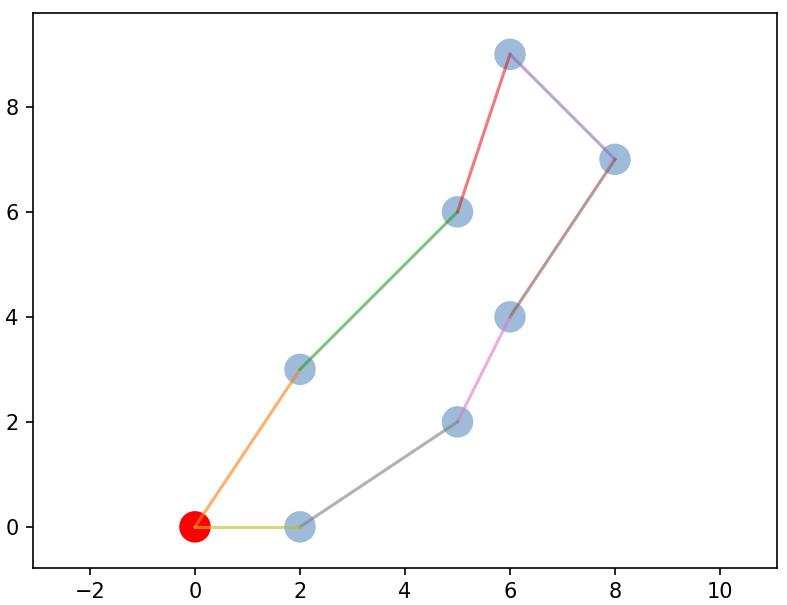
Теоретический предел для компьютера составляет 66 городов, уже при 67 городах, количество вариантов превысит предел Бремерманна [1] и будет равно 36\*1093, а значит перебрать их все будет невозможно, так как потребуется в несколько раз больше времени, чем существует наша планета.

Рисунок 3 — Решение задачи коммивояжёра полным перебором c учётом приоритетных городов, при n (кол-во городов) = 8

Рисунок 2 — Пример неоптимального решения задачи коммивояжёра

Рисунок 1 — Решение задачи коммивояжёра полным перебором без учёта приоритетных городов, при n (кол-во городов) = 8

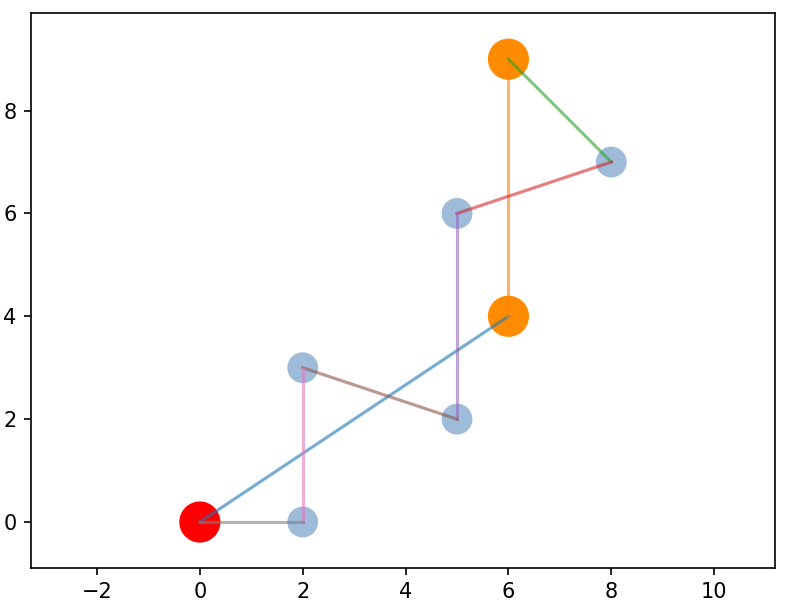
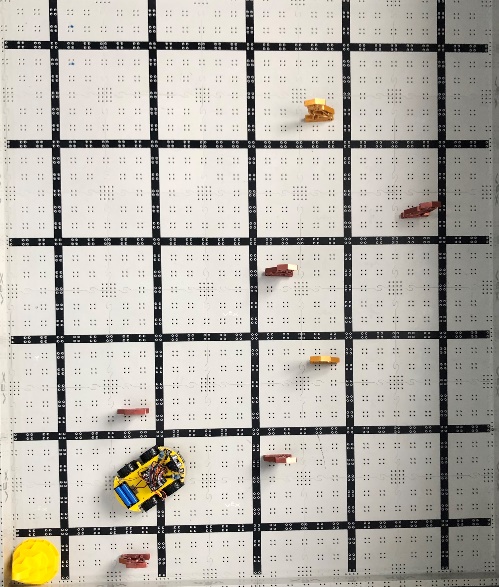


Рисунок 4 — Маршрут движения робота по расставленными городами, где золотые фигуры – это приоритетные города



**Метод ветвей и границ** – метод упорядоченного перебора всех возможных вариантов решения. Сложность данного алгоритма представлена формулой

(1)

*,*

где *n* — число городов.

Смысл метода ветвей и границ заключается в том, что необходимо последовательно разбивать множества допустимых решений на подмножества. На каждом шаге разбившиеся элементы проверяются на содержание в данном подмножестве оптимального решения. Проверка происходит путём вычисления оценки снизу для целевой функции на данном подмножестве, если оценка больше рекорда, наилучшего из найденных решений, то подмножество отбрасывается. Также подмножество может быть отброшено, если в нём удается найти наилучшее решение. Когда значение целевой функции на найденном решении меньше наилучшего из найденных решений, происходит смена рекорда. Если все разбившиеся элементы будут отброшены, то рекорд станет оптимальным решением задачи. В противном случае, из не отброшенных подмножеств выбирается наилучшее, например, с наименьшим значением нижней оценки, и подвергаются разбиению. Новые подмножества проверяются по такому же алгоритму.

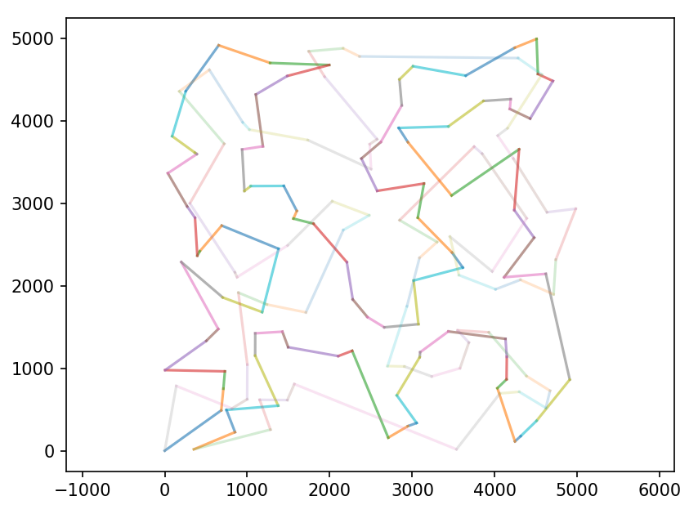
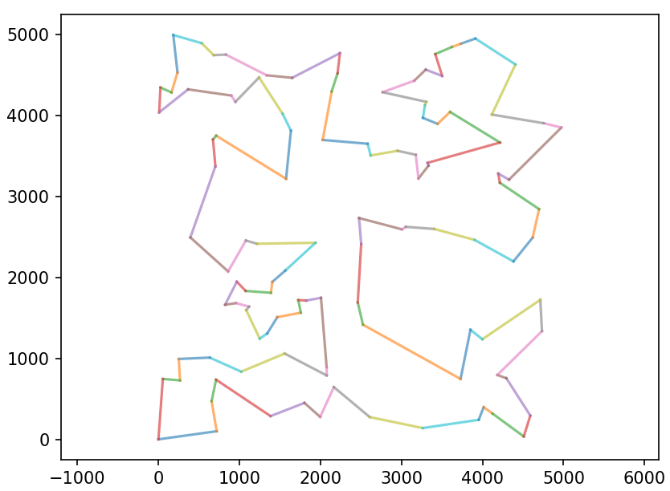


Рисунок 5 — Решение задачи коммивояжёра методом ветвей и границ без приоритетных городов, при n (кол-во городов) = 114

Рисунок 6 — Решение задачи коммивояжёра методом ветвей и границ с приоритетными городами, при n (кол-во городов) = 114

## Приближённые и эвристические методы

Приближённые и эвристические методы не дают точного результата, зато сокращают полный перебор всех вариантов. Многие из них находят не самый эффективный маршрут, но часто дают максимально приближённый. Это особенно полезно в случаях, когда нужно быстро получить решение для большого количества городов, где полный перебор становится непрактичным.

**Жадный алгоритм** – алгоритм, который на каждом этапе выбирает самое локально оптимальное решение из всех возможных, без дальнейшего анализа, надеясь, что в конце получится глобально оптимальное решение. Это продолжается, пока мы не посетим все города. Алгоритм отличается своей высокой скоростью работы, но не всегда выдаёт точные результаты в задачах относящимся к NP полным. Уровень сложности при использовании жадного алгоритма равен *,* где *n* — число городов.

Жадный алгоритм не всегда гарантирует нахождение оптимального решения. В некоторых случаях использование этого алгоритма приводит к худшему результату, чем при использовании других методов. Но при этом, иногда он может дать достаточно хороший результат за короткое время.

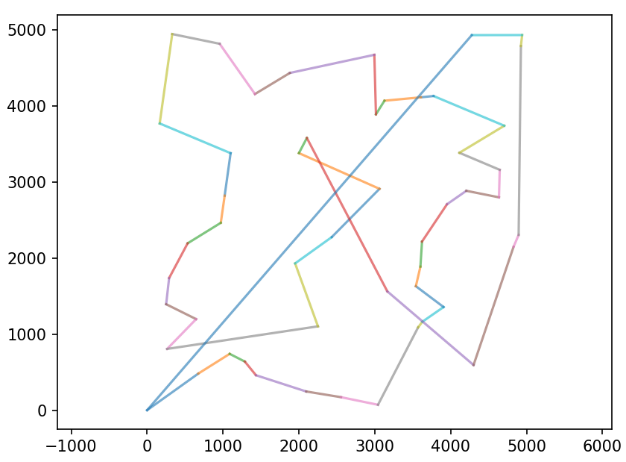
На рисунке ниже координаты городов подобраны так, что алгоритм не нашёл оптимального решения, поскольку последний путь оказался слишком длинным. Также, в классической задаче, в оптимальном решении не должно быть пересечений рёбер в графе.

Рисунок 7 — Неоптимальное решение задачи коммивояжёра жадным алгоритмом, при n (кол-во городов) = 51

Иногда встречаются такие координаты городов, для которых жадный алгоритм может найти оптимальное решение, например, такое решение представлено на рисунке ниже.

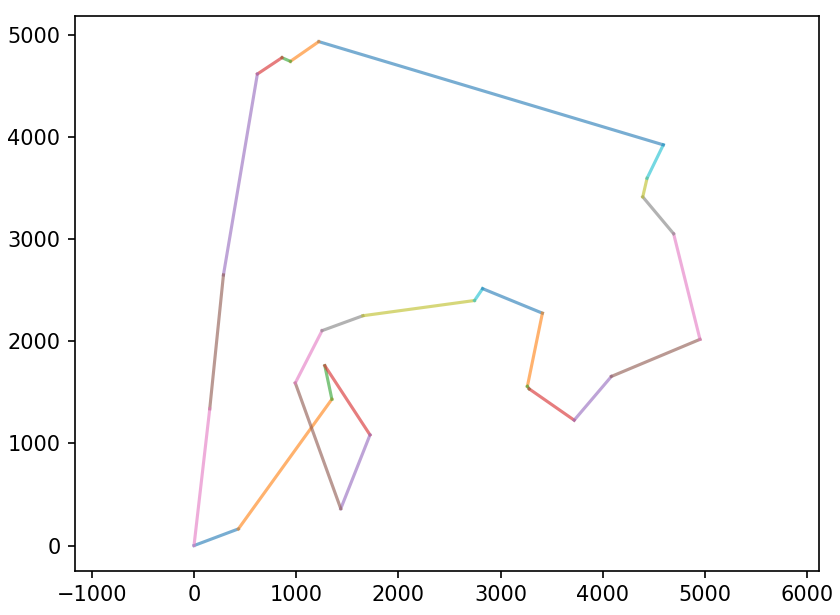
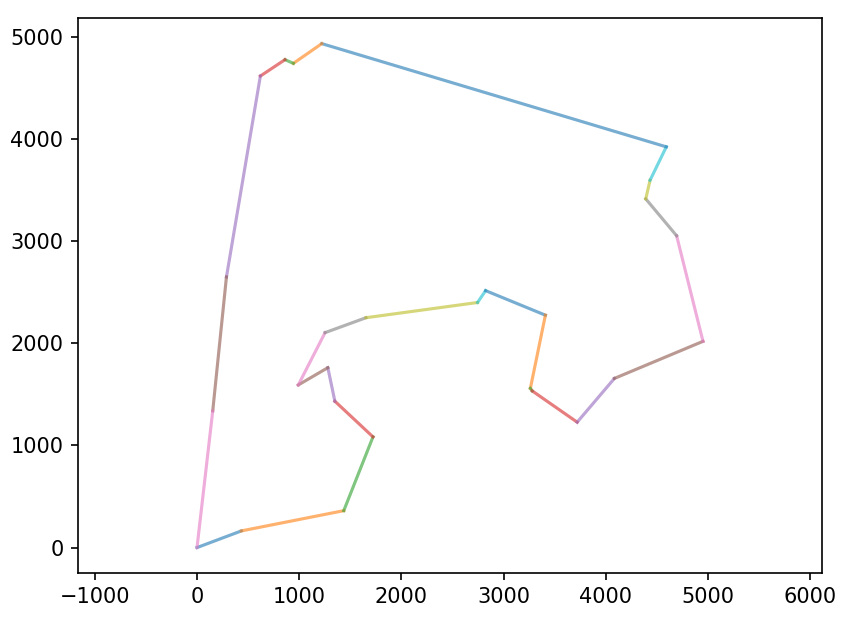
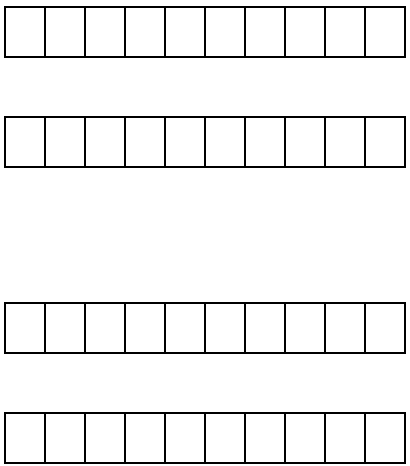


Рисунок 8 — Оптимальное решение задачи коммивояжёра жадным алгоритмом, при n (кол-во городов) = 27

Рисунок 9 — Оптимальное решение задачи коммивояжёра жадным алгоритмом с приоритетными городами, при n (кол-во городов) = 27

**Генетический алгоритм** – это эволюционный метод оптимизации, который моделирует процессы эволюции, такие как селекция, кроссинговер и мутация. В задаче коммивояжера генетический алгоритм определяет случайным образом начальный набор возможных решений (особей), каждое из которых является путём между всеми городами. На основе оценок целевой функции (длины маршрута) выполняется отбор "лучших" особей, которые затем используются для скрещивания и создания новой популяции. Для выполнения кроссинговера сначала генерируем точку разрыва, затем часть первого родителя до точки разрыва копируем в первого потомка, гены второго родителя копируем в первого потомка, если они ещё не были унаследованы. Аналогично поступаем со вторым потомком: копируем часть второго родителя до точки разрыва и часть первого родителя, которая ещё не была унаследована.

Рисунок 10 — Иллюстрация кроссинговера



0

6

2

7

3

8

1

4

9

5

0

3

9

4

1

7

5

6

2

8

0

6

2

7

3

9

4

1

5

8

0

3

9

4

6

2

7

8

1

5

1-ый родитель

2-ой родитель

1-ый ребёнок

2-ой ребёнок

Особи, созданные в результате скрещивания, с определённой вероятностью подвергаются мутации. Этот процесс повторяется несколько раз, пока не будет пройдено заданное количество итераций. В конечном итоге, лучшая особь, найденная в ходе этого процесса, представляет собой оптимальный маршрут для задачи коммивояжера.

Одним из преимуществ генетического алгоритма является его способность эффективно решать задачи оптимизации с неограниченным количеством решений. Однако, также существует ряд недостатков, таких как высокая временная сложность и трудность в подборе оптимальных параметров, таких как размер популяции, вероятность мутации и размер популяции. В целом, временная сложность генетического алгоритма может быть оценена как:

(2)

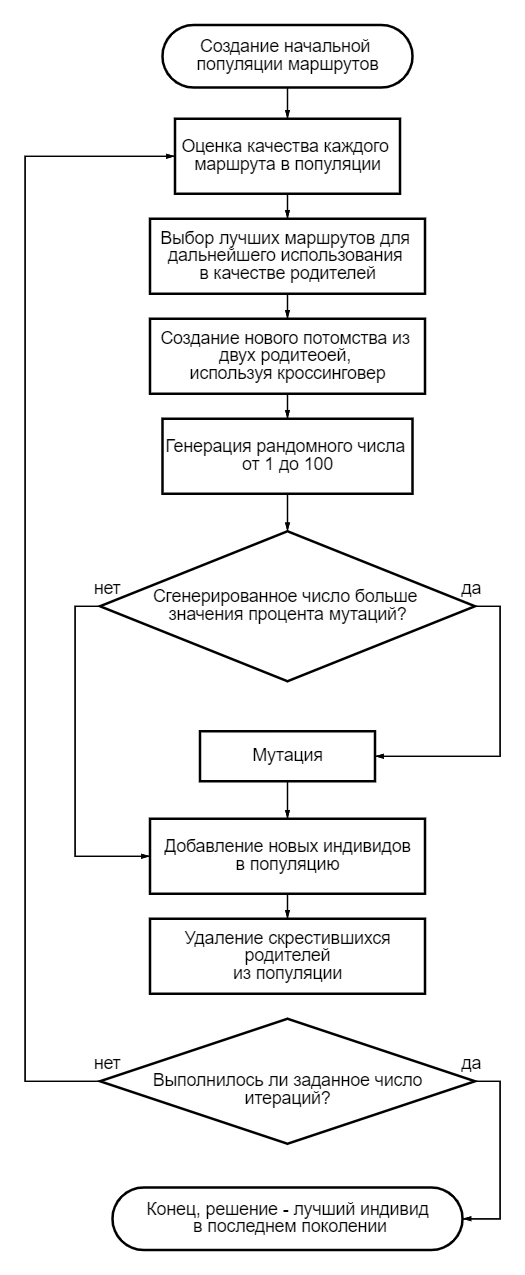
*,*

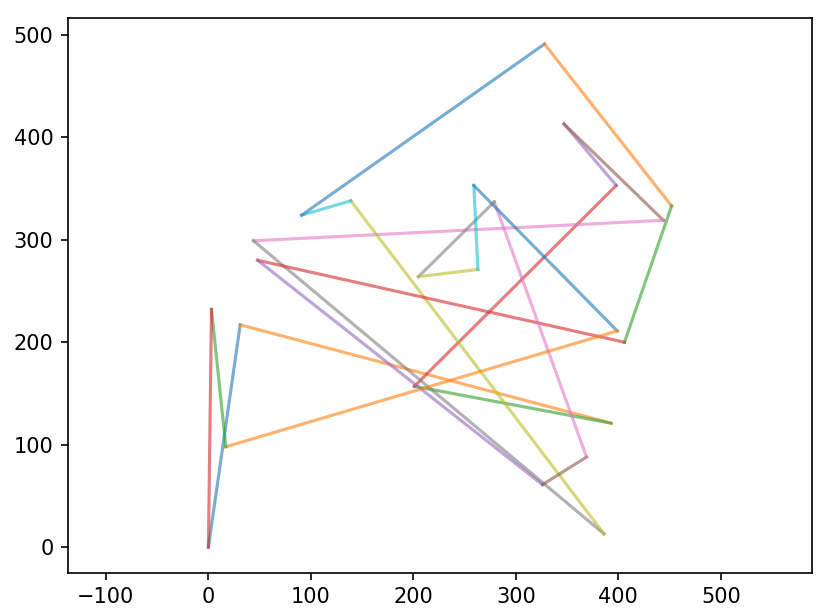
где *n* — число городов;

— число итераций;

*t* — время, затрачиваемое на оценку качества одного решения.

Рисунок 11 — Блок-схема логики работы генетического алгоритма

Генетические алгоритмы не всегда являются оптимальными в практическом применении, так как они требуют дополнительных ресурсов для хранения и обработки большого количества информации. Несмотря на эти недостатки, генетический алгоритм полезен для решения сложных задач оптимизации, особенно в случаях, когда нет эффективных алгоритмов для решения таких задач.

Рисунок 12 — Неоптимальное решение задачи коммивояжёра генетическим алгоритмом, при n (кол-во городов) = 23

**Метод Монте-Карло** – эвристический метод, который заданное число раз генерирует случайные перестановки городов и проверяет длину пути между ними. В представленном решении алгоритм столько раз генерирует перестановки городов, сколько и генетический алгоритм для заполнения популяции особями. В целом, метод не является наилучшим выбором, так как не гарантирует нахождение оптимального пути, а только приближенного решения. Хотя метод Монте-Карло не находит оптимального решения, он может быть полезен в тех случаях, когда другие алгоритмы слишком сложны или неэффективны. Кроме того, он может быть применен для анализа различных аспектов задачи коммивояжера и сравнения эффективности других алгоритмов решения.

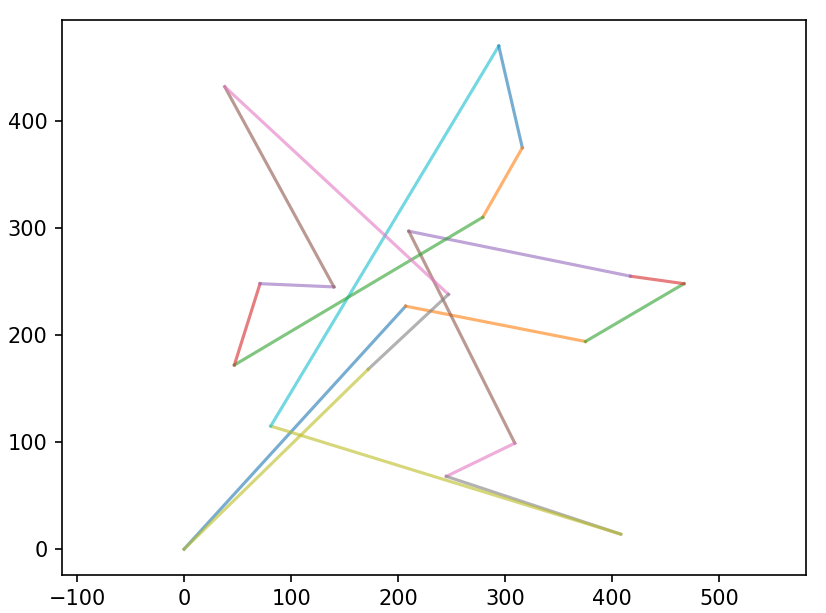


Рисунок 13 — Решение задачи коммивояжёра методом Монте-Карло, при n(кол-во городов) = 18

## Модификация генетического алгоритма

В стандартном генетическом алгоритме, основанном на строках-перестановках, мутация проходит максимум один раз в одной особи за одну итерацию, в модификации процент мутации считается для каждой хромосомы в особи, то есть для одной особи максимальное число количества мутаций будет равно длине самой особи. Так же в стандартном генетическом алгоритме из популяции удаляют скрестившихся родителей, а в модификации перед удалением популяцию сортируют и из неё удаляют наименее приспособленных индивидов в размере:

*,*

где — размер текущей популяции;

(3)

— размер старой популяции.

Размер популяции может изменяться, поэтому при каждой итерации это контролируется, и, если особей стало меньше, происходит генерация недостающих особей с проверкой, что новой особи нет среди уже удалённых.

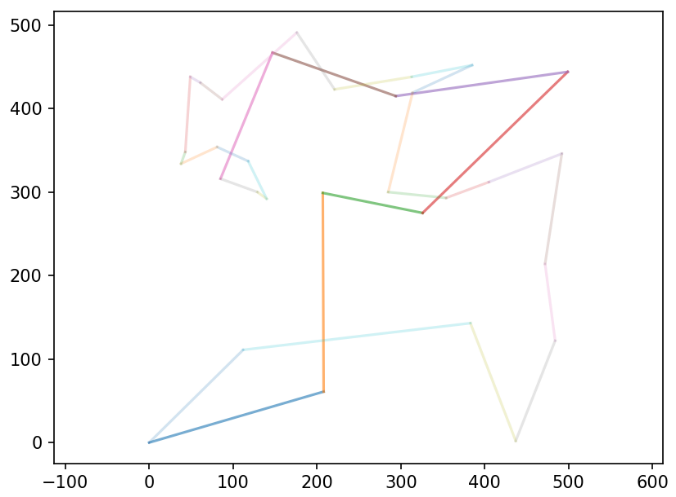
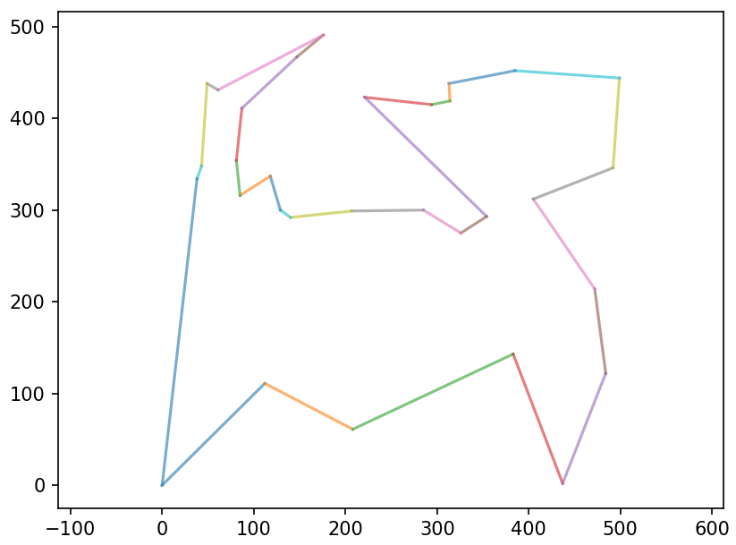


Рисунок 14 — Решение задачи коммивояжёра модифицированным генетическим алгоритмом, при n(кол-во городов) = 30

Рисунок 15 — Решение задачи коммивояжёра с приоритетными городами модифицированным генетическим алгоритмом, при n(кол-во городов) = 30

Ниже приведена таблица сравнения стандартного генетического алгоритма, основанного на строках-перестановках, и модификации генетического алгоритма для решения задачи коммивояжера. Для сравнения используется критерий Вилкоксона-Манна-Уитни [2, 3], который позволяет определить, существует ли статистически значимая разница в качестве решения, получаемого сравниваемыми алгоритмами. Для этого необходимо запустить алгоритмы на одних и тех же тестовых данных и сравнить результаты с теоретическим распределением значений статистики, основанном на предположении, что нулевая гипотеза верна (т.е. медианные значения выборок равны). Если значение W меньше или больше, чем критические значения, соответствующее выбранному уровню значимости (Q), то можно отклонить нулевую гипотезу и сделать вывод о статистически значимых различиях между двумя выборками. Когда значение W входит в промежуток нижних и верхних критических значений статистики, то делается вывод, что выборки были сформированы по одному и тому же закону распределения случайных чисел, а, значит, алгоритмы, которые генерировали данные выборки по соответствующе выбранному уровню значимости, не отличаются с точки зрения выбранного уровня значимости.

В каждой строчке таблицы ниже приведены средние арифметические значения длин путей десяти запусков каждого из алгоритмов при одинаковых координатах. Так, для получения данных, представленных ниже, каждый алгоритм был запущен по сто раз. Для данных приведённых «в таблице 1» значение W равно 58. При уровне значимости (Q) равном 0.002 распределение значений статистики принадлежит промежутку: [65; 145]. Поскольку значение W, полученное при сравнении двух алгоритмов критерием Вилкоксона-Манна-Уитни, не находится в указанном промежутке, то можно утверждать, что выборки неоднородны, следовательно модификация гораздо лучше стандартного генетического алгоритма.

Таблица 1 — Сравнение стандартного генетического алгоритма и модификации генетического алгоритма для решения задачи коммивояжёра (зелёное – самое оптимальное решение, красное – самое наихудшее)

|  |  |
| --- | --- |
| **Стандартный генетический алгоритм** | **Модификация генетического алгоритма** |
| 9298 | 4212 |
| 6090 | 2469 |
| 6340 | 2726 |
| 6566 | 2987 |
| 8235 | 4397 |
| 6581 | 2938 |
| 6136 | 2710 |
| 7922 | 3448 |
| 3414 | 1960 |
| 5160 | 2370 |

## Сравнение эффективности алгоритмов

Ниже приведена таблица 2, в которой сравниваются алгоритмы, используемые для решения задачи коммивояжёра без учёта приоритетных городов. В каждой ячейке таблицы представлены усреднённые данные десяти запусков при одном и том же количестве городов. В столбце «Полный перебор» стоят прочерки после строки, в которой количество городов равнялось шести, поскольку уже при восьми городах решение полным перебор требует колоссальных временных и ресурсных затрат. В последней строчке таблицы 2 приведены средние арифметические значения усреднённых длин каждого маршрута. По этим данным видно, что наихудшим методом стал метод Монте-Карло, который практически на всех запусках выдавал неоптимальные решения. Самым лучшим алгоритмом стал метод ветвей и границ, который ненамного обошёл модифицированный генетический алгоритм. Для более тщательного анализа автором было проведено дополнительное сравнение метода ветвей и границ и модифицированного генетического алгоритма критерием Вилкоксона-Манна-Уитни. Значение статистики W для этих двух алгоритмов равно 107.5, при уровне значимости Q равном 0,2 границы интервала критических значений статистики принадлежат следующему промежутку: [87; 123]. Значение W входит в промежуток нижних и верхних критических значений статистики, следовательно выборки были сформированы по одному и тому же закону распределения случайных чисел, а, значит, алгоритмы, которые генерировали данные выборки не отличаются с точки зрения выбранного уровня значимости, следовательно можно утверждать, что метод ветвей и границ и модификация генетического алгоритма являются одинаково оптимальными для решения задачи коммивояжёра без приоритетных городов.

Таблица 2 — Сравнение эффективности алгоритмов для решения задачи коммивояжёра (зелёное – самое оптимальное решение, красное – самое наихудшее)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Полный перебор** | **Жадный**  **алгоритм** | **Метод ветвей и границ** | **Метод Монте-Карло** | **Модификация генетического алгоритма** |
| Карта №1  (n\* = 6) | 1559.89 | 1666.34 | 1571.02 | 1603.5 | 1559.89 |
| Карта №2  (n = 8) | – | 2301.89 | 1783.79 | 2125.83 | 1783.79 |
| Карта №3  (n = 10) | – | 1829.93 | 1854.24 | 2204.77 | 1825.34 |
| Карта №4  (n = 12) | – | 1724.41 | 1848.05 | 2561.22 | 1750.6 |
| Карта №5  (n = 14) | – | 2377.31 | 2059.43 | 2902.45 | 1988.89 |
| Карта №6  (n = 16) | – | 2129.32 | 1959.11 | 3207.31 | 1974.16 |
| Карта №7  (n = 18) | – | 1892.47 | 1851.06 | 3413.71 | 1820.52 |
| Карта №8  (n = 20) | – | 1959.81 | 1914.43 | 4265.27 | 2096.53 |
| Карта №9  (n = 22) | – | 2967.1 | 2439.42 | 4743.18 | 2610.25 |
| Карта №10  (n = 24) | – | 2350.89 | 2257.23 | 4870.63 | 2314.47 |
| **Среднее арифме-тическое длин маршру-тов** | **–** | **2119.95** | **1953.79** | **3189.79** | **1972.44** |

\*n – количество городов

В таблице 3, приведённой ниже, сравниваются алгоритмы решения задачи коммивояжёра с приоритетными городами. Данные для таблицы 3 брались по такому же принципу, как и для таблицы 2. Последняя строка таблицы 3 также содержит средние арифметические значения усреднённых длин путей за сто запусков, из которых каждые десять запускались с одинаковыми координатами для каждого маршрута. По приведённым значениям видно, что наихудшим методом снова стал метод Монте-Карло, хотя далеко не всегда решения, найденные этим алгоритмом, становились самыми неоптимальными. В сравнении алгоритмов для решения задачи коммивояжёра с учётом приоритетных городов самым лучшим стал модифицированный генетический алгоритм.

Таблица 3 — Сравнение эффективности алгоритмов для решения задачи коммивояжёра с приоритетными городами (зелёное – самое оптимальное решение, красное – самое наихудшее)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Полный перебор** | **Жадный**  **алгоритм** | **Метод ветвей и границ** | **Метод Монте-Карло** | **Модификация генетического алгоритма** |
| Карта №1  (n\* = 6) | 1303.26 | 1346.47 | 1303.26 | 1303.26 | 1387.51 |
| Карта №2  (n = 8) | – | 1998.92 | 2106.42 | 1982.78 | 2011.12 |
| Карта №3  (n = 10) | – | 2263.48 | 2447.14 | 2301.3 | 2207.85 |
| Карта №4  (n = 12) | – | 2076.99 | 2037.92 | 2072.42 | 1989.39 |
| Карта №5  (n = 14) | – | 2239.57 | 2547.22 | 2551.34 | 2664.15 |
| Карта №6  (n = 16) | – | 3124.4 | 2972.47 | 3099.93 | 2779.89 |
| Карта №7  (n = 18) | – | 2695.38 | 3284.61 | 3464.22 | 2885.33 |
| Карта №8  (n = 20) | – | 3222.98 | 2996.09 | 3633.9 | 2961.87 |
| Карта №9  (n = 22) | – | 2726.6 | 2943.53 | 4377.45 | 2992.37 |
| Карта №10  (n = 24) | – | 3551.75 | 3412.36 | 4606.68 | 3280.6 |
| **Среднее арифме-тическое длин маршру-тов** | **–** | **2524.65** | **2605.1** | **2934.83** | **2516.01** |

\*n – количество городов

Согласно данным, представленным в таблице 3, ненамного хуже модификации генетического алгоритма оказался жадный алгоритм, чтобы точнее сравнить эти два алгоритма было проведено повторное сравнение этих двух алгоритмов, результаты которого приведены в таблице 4. Алгоритмы сравнивались критерием Вилкоксона-Манна-Уитни. Для сравниваемых алгоритмов значение статистики W равно 138, при уровне значимости Q равном 0,2 границы интервала критических значений статистики принадлежат следующему промежутку: [87; 123]. Значение статистики W не входит в промежуток нижних и верхних критических значений статистики, следовательно можно считать, что есть значимые различия между двумя выборками. Поскольку выборки неоднородны и жадный алгоритм практически на всех запусках выдавал неоптимальные решения, можно утверждать, что решения задачи коммивояжёра с приоритетными городами модификацией генетического алгоритма всегда будут оптимальнее, чем любым другим представленным алгоритмом.

Таблица 4 — Сравнение жадного алгоритма и модификации генетического алгоритма для решения задачи коммивояжёра с приоритетными городами (зелёное – самое оптимальное решение, красное – самое наихудшее)

|  |  |  |
| --- | --- | --- |
|  | **Жадный алгоритм** | **Модификация генетического алгоритма** |
| Карта №1  (n\* = 20) | 3081.71 | 3144.31 |
| Карта №2  (n = 22) | 3183.97 | 3056.35 |
| Карта №3  (n = 24) | 4019.6 | 3134.38 |
| Карта №4  (n = 26) | 3809.03 | 3217.5 |
| Карта №5  (n = 28) | 4237.59 | 3406.47 |
| Карта №6  (n = 30) | 3921.98 | 3339.06 |
| Карта №7  (n = 32) | 4148.53 | 3353.6 |
| Карта №8  (n = 34) | 3873.34 | 3744.9 |
| Карта №9  (n = 36) | 4174.08 | 3733.7 |
| Карта №10  (n = 38) | 3960.88 | 3860.5 |

\*n – количество городов

Таким образом, для решения стандартной задачи коммивояжёра одинаково оптимальными являются метод ветвей и границ и модифицированный генетический алгоритм. Для решения задачи коммивояжёра с приоритетными городами самым эффективным стал модифицированный генетический алгоритм. Однако, в зависимости от конкретных условий задачи, может быть более эффективно использовать другие методы решения. Например, если задача коммивояжера имеет очень большой размер, то генетический алгоритм может быть неэффективным из-за большого количества вычислений, которые необходимо выполнить. В этом случае использование метода ветвей и границ позволит значительно сократить время выполнения при незначительном отклонении от оптимального решения.

## Взаимодействие с пользователем

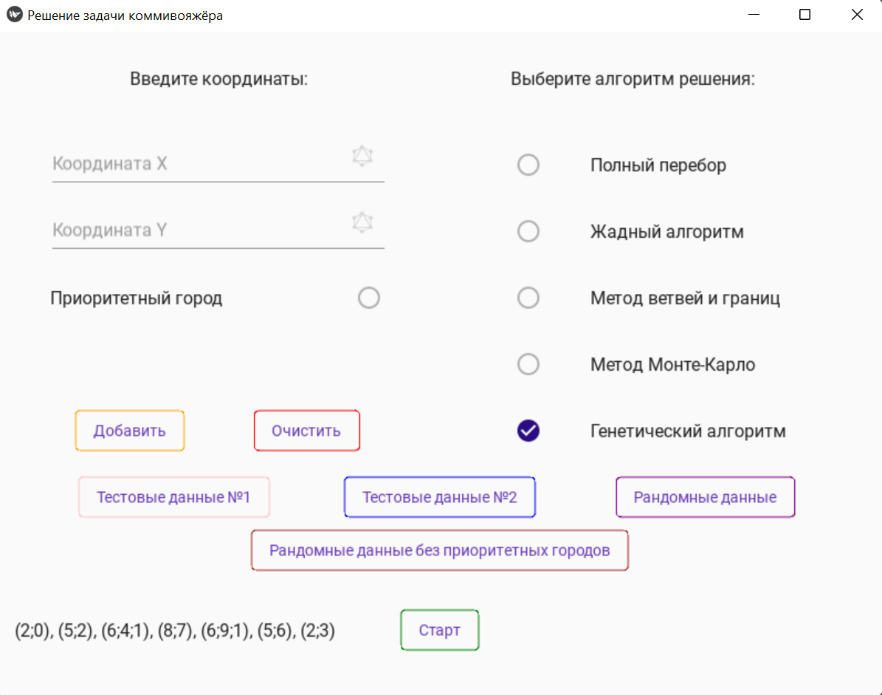
В качестве сервера выступает любой компьютер с запущенной программой для решения задачи коммивояжёра. Все алгоритмы решения и интерфейсное приложение написаны на высокоуровневом интерпретируемом языке программирования Python в PyCharm – кроссплатформенной интегрированной среде разработки для языка программирования Python, разработанной компанией JetBrains. Взаимодействие с пользователем осуществляется через KivyMD, фреймворк для создания кроссплатформенных мобильных приложений и интерфейсов. Для решения задачи коммивояжёра через KivyMD создан пользовательский интерфейс, который позволяет пользователю задать необходимые параметры задачи, такие как количество городов и координаты городов, указать приоритетные города решения или воспользоваться автозаполнением координат, а также выбрать необходимый алгоритм. После того, как пользователь задал все необходимые параметры, запускается выбранный алгоритм решения, а после выполнения отображается граф с решением задачи коммивояжера в виде карты с указанием маршрута и выводятся координаты городов, отсортированные в оптимальном для посещения порядке, с итоговой длиной маршрута. Для визуализации данных, полученных при решении задачи коммивояжёра, используется библиотека для создания графиков Matplotlib.

Рисунок 16 — Интерфейс программы для решения задачи коммивояжёра

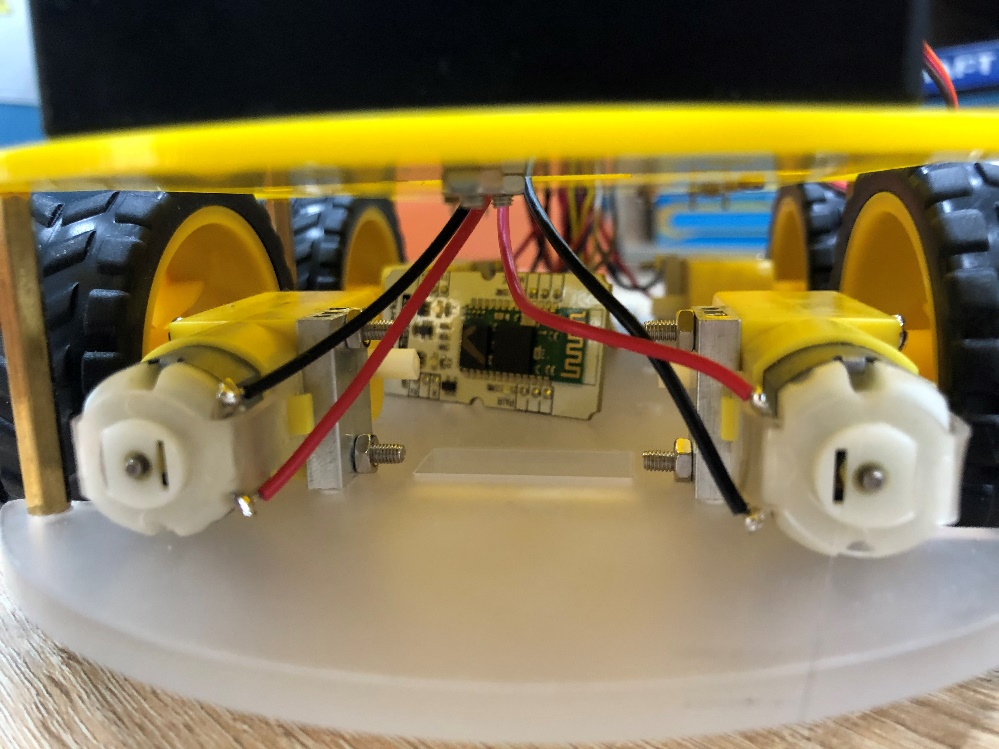
 После получения отсортированных координат городов они передаются Arduino, через Bluetooth-модуль HC-05, установленный на роботе, который выступает в качестве клиента, по технологии беспроводной передачи данных – Bluetooth с помощью библиотеки для работы с последовательными портами – Serial, которая позволяет осуществлять чтение и запись данных через серийный (COM) порт, который используется для связи между устройствами, такими как микроконтроллеры.

Рисунок 17 — Расположение Bluetooth-модуля в роботе

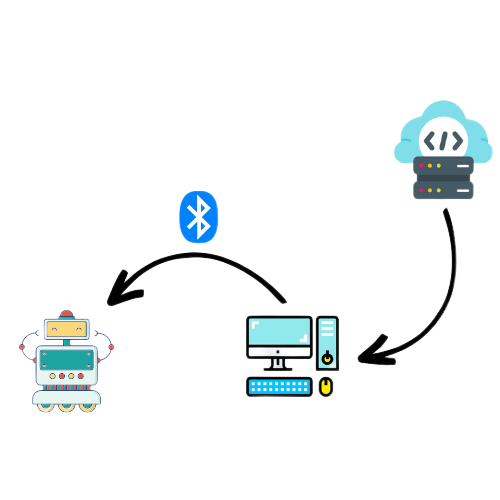
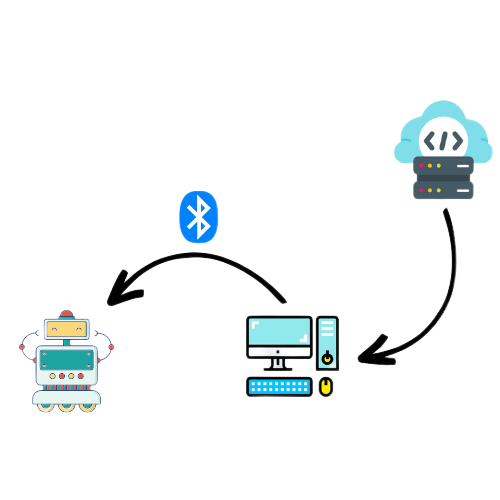
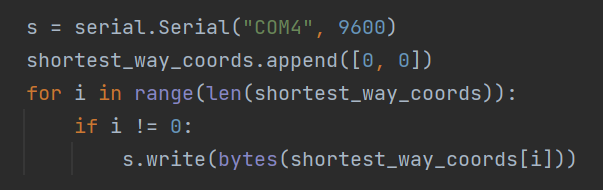
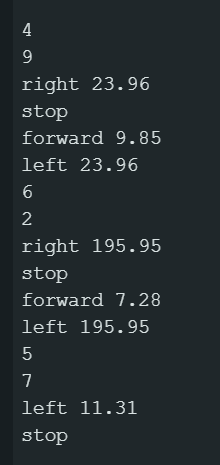


Рисунок 18 — Схема передачи данных от сервера к клиенту по Bluetooth

## Робототехническая часть

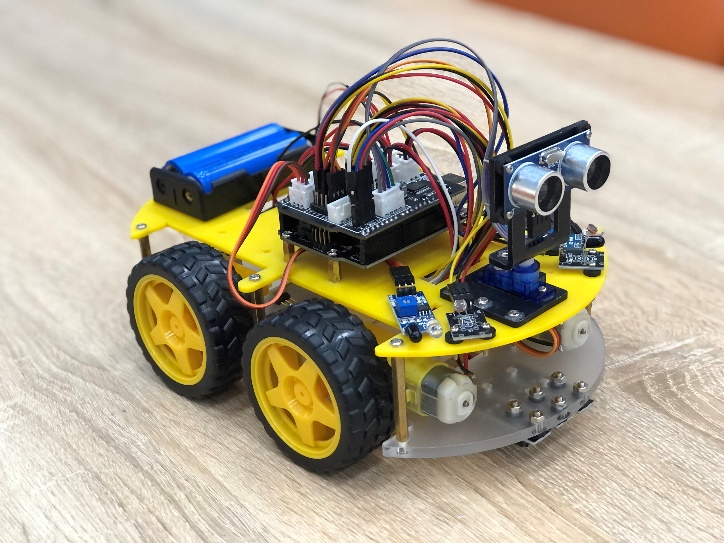
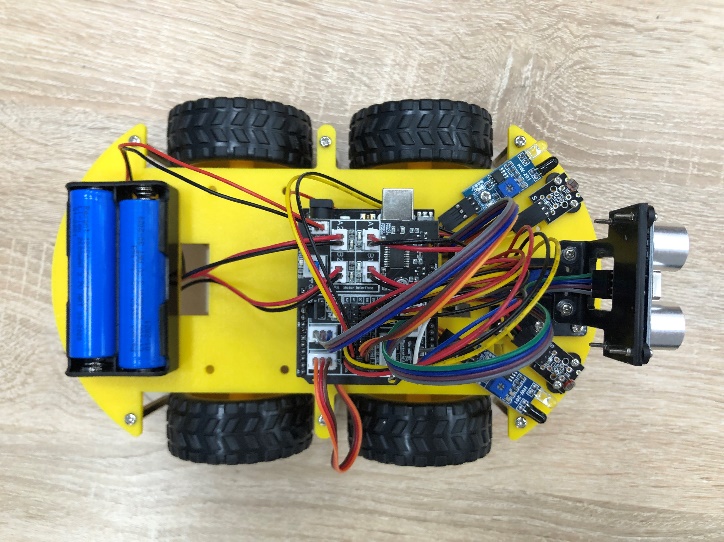
В проекте используется четырёхколёсный робот, управляемый с помощью платы Arduino Uno, оснащённой микроконтроллером ATmega328P. Микроконтроллер программировался на компилируемом языке программирования C++ с использованием интегрированной среды разработки Arduino IDE. Управление моторами робота осуществляется через двухканальный драйвер моторов TB6612. Одним из главных преимуществ использования этой платы является, наличие встроенной защиты от перегрева, короткого замыкания и перегрузки тока, что делает её абсолютно безопасной в использовании. Для приведения в действие колес используются четыре мотора-редуктора. Моторы-редукторы – это электромеханические устройства, состоящие из мотора и механического редуктора для увеличения крутящего момента и снижения скорости вращения. Используемые моторы-редукторы имеют коэффициент передачи 48:1, что означает, что скорость вращения на выходе моторов снижена в 48 раз, а крутящий момент увеличен в 48 раз по сравнению с вращением на выходе от двигателей без редуктора. Каркас робота выполнен из акрила, лёгкого и прочного материала, который позволяет создавать роботов, которые могут выдерживать долгое время эксплуатации и не ломаться при малейших ударах.

Рисунок 19 — Вид робота сбоку

Рисунок 20 — Вид робота сверху

Как только координаты городов передались через Bluetooth-соединение микроконтроллеру робота, Arduino принимает их через соответствующий серийный порт для чтения сигнала Bluetooth, а после обрабатывает, вычисляя угол поворота и длину пути для достижения нужной координаты.

Рисунок 21 — Фрагмент кода на C++, считающий путь и угол поворота до координаты в первой четверти от текущего местоположения робота

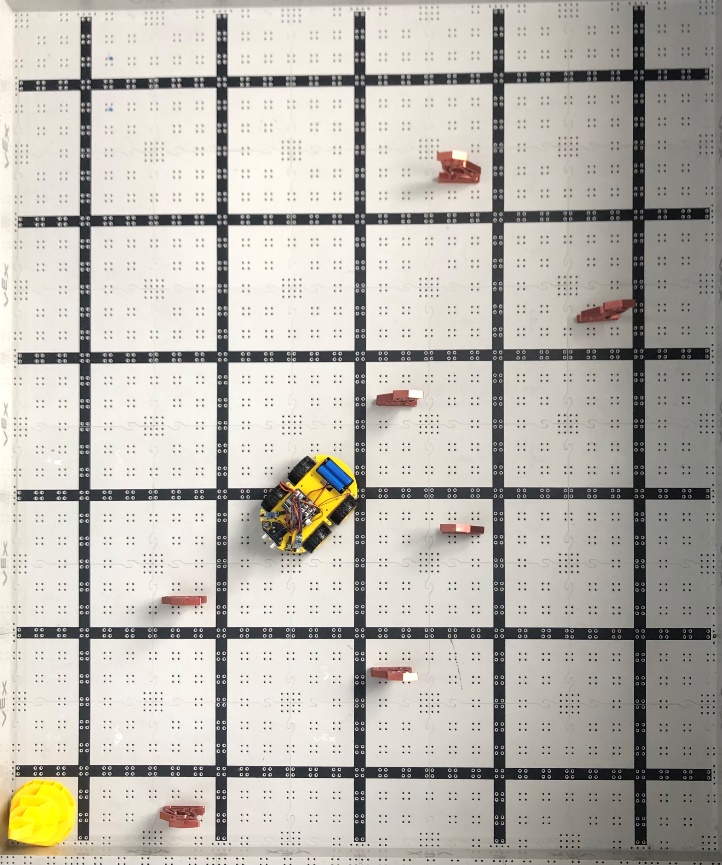


Рисунок 22 — Движение робота по маршруту, рассчитанному стандартной задачей коммивояжёра

# Заключение

В результате проведенной работы были исследованы различные методы решения стандартной задачи коммивояжера и задачи коммивояжёра с приоритетными городами, а также применение роботехнических систем для их решения. Было реализовано шесть алгоритмов решения задачи коммивояжера. Для оценки эффективности алгоритмов использовался критерий качества найденного решения и критерий Вилкоксона-Манна-Уитни. Также была разработана система управления роботом, позволяющая решать задачу коммивояжёра с использованием роботехнических систем. В процессе экспериментов было установлено, что для решения стандартной задачи коммивояжёра одинаково оптимальными являются метод ветвей и границ и модифицированный генетический алгоритм. Для решения задачи коммивояжёра с приоритетными городами самым эффективным стал модифицированный генетический алгоритм. Проведенные эксперименты показали, что выбор оптимального метода решения зависит от конкретных условий задачи. Таким образом, в данной работе были исследованы различные методы решения задачи коммивояжера, а также реализована система управления роботом для её решения.

# Список использованных источников

1. Бондарева, А. А. (2015). "Предельные теоремы теории вероятностей и их применения". М.: КноРус.
2. Mann, H. B., & Whitney, D. R. (1947). "On a test of whether one of two random variables is stochastically larger than the other". The annals of mathematical statistics, 18(1), 50-60.
3. Wilcoxon, F. (1945). "Individual comparisons by ranking methods". Biometrics bulletin, 1(6), 80-83.
4. Марков, А. А., Антонов, А. С., Михайлов, Е. А. (2017). "Робототехнические системы". СПб.: БХВ-Петербург.
5. Applegate, D. L., Bixby, R. E., Chvatal, V., Cook, W. J. (2006). "The Traveling Salesman Problem: A Computational Study". Princeton University Press.
6. Lawler, E. L. (1985). "The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization" (1st ed.). Wiley.
7. Laporte, G. (1992). "The Traveling Salesman Problem: An Overview of Exact and Approximate Algorithms". European Journal of Operational Research, 59(2), 231-247.