Министерство образования и науки Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н.Г. ЧЕРНЫШЕВСКОГО»

Кафедра математического и компьютерного моделирования

Решение задачи коммивояжера методами искусственного интеллекта

АВТОРЕФЕРАТ БАКАЛАВОРСКОЙ РАБОТЫ

студента 4 курса 413 группы

направления 01.03.02 - Прикладная математика и информатика

механико-математического факультета

Демахина Николая Николаевича

Научный руководитель

доцент, к.ф.-м.н., доцент ________ С. П. Шевырев

Зав. кафедры

зав. каф., д.ф. – м.н. ________ Ю. А. Блинков

Саратов 2017

ВВЕДЕНИЕ

Комбинаторика – это раздел математики, в котором изучаются вопросы о том, сколько существует комбинаций, подчиняющихся тем или иным условиям, можно сформировать из заданных объектов. Комбинаторика связана со многими областями математики, например, такие как алгебра, геометрия, теорией вероятности и имеет широкий спектр применения в различных областях знаний (генетика, информатика, статистическая физика).

Большой вклад в развитие комбинаторных методов был сделан Г. Лейбницем (диссертация «Комбинаторное искусство»), Я. Бернулли (работа «Искусство предположений»), Л. Эйлером. Можно сказать, что после написания работ Я. Бернулли и Г. Лейбница комбинаторные методы выделились в самостоятельный раздел математики.

Возвращение интереса к комбинаторному анализу относится к 50-м годам XX в. из-за бурного развития кибернетики и дискретной математики и с широким использованием ЭВМ.

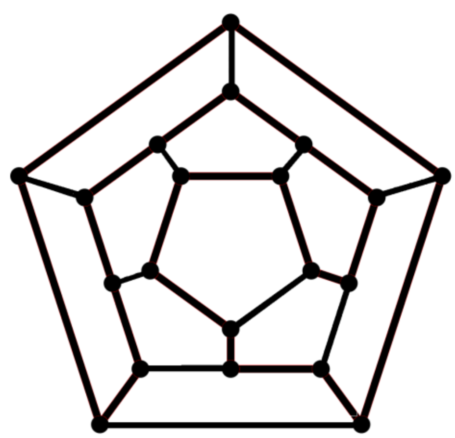

Рисунок 1 – игра «Кругосветное путешествие».

В 1859 г. У. Гамильтон придумал игру «Кругосветное путешествие», состоящую в нахождение пути, такого чтобы он проходил через все вершины (города) графа в соответствии с рисунком 1, чтобы посетить каждую вершину один раз и после этого возвратиться в начало. Пути, обладающие таким свойством, называются гамильтоновыми циклами.

Задача о гамильтоновых циклах в графе получила различные обобщения. Одно из них называется задача коммивояжера, которая имеет ряд применений в различных сферах современного мира.

Целью данной работы является изучение методов искусственного интеллекта как способа оптимизации, их эффективности и трудоемкости. В качестве решаемой задачи была выбрана задача коммивояжера. Она очень хорошо изучена, имеет эффективные способы решения, для того, чтобы сравнить с полученными результатами.

Основным инструментом для практического исследования была выбрана среда Qt Creator, поскольку она имеет множество встроенных функций для решения задачи коммивояжера методами искусственного интеллекта.

1 ОБЩАЯ СТРУКТУРА РАБОТЫ

1.1 Актуальность работы

Развития цивилизованного общества 21 века по праву называют этапом информатизации.

Характерной чертой данного периода является тот факт, что основным видом деятельности в сфере общественного производства становится сбор, обработка, хранение, передача и использование информации, осуществляемые на базе современных информационных технологий.

Задача коммивояжера – одна из наиболее популярных задач комбинаторной оптимизации, в простейшем варианте сводящаяся к поиску замкнутого маршрута минимальной длины, проходящего через все указанные вершины только по одному разу. Данная задача находит своё применение во многих областях современного мира, например, в сельском хозяйстве, в военном деле, в экономике и многих других.

Всё чаще даёт о себе знать проблема низкой производительности каких-либо расчётов, поэтому, перед нами встаёт цель проанализировать несколько методов для решения задачи коммивояжера и выбрать самый оптимальный.

Так как данная задача относится к числу трансвычислительных, то уже при относительно малом числе городов никакой компьютер не сможет решить методом полного перебора из-за того что время вычисления идеального маршрута может достигнуть нескольких миллиардов лет. Благодаря этому, существует множество методов нахождения выгодного маршрута на n количестве вершин. Эта задача является технически интересной, и как было сказано ранние используется во многих сферах современной жизни. Оптимальное решение задачи будет минимизировать затраты и путь.

1.2 Цели и задачи работы

Целью данной работы является проанализировать выбранные методы искусственного интеллекта и реализовать программу для решения задачи коммивояжера. Задачи в рамках поставленных целей:

- Описать формулировку задачи коммивояжера.
- Рассмотреть методы искусственного интеллекта для решения задачи коммивояжера: метод полного перебора, муравьиный алгоритм, метод отжига, генетический алгоритм.
- Реализовать программу для решения задачи коммивояжера методами искусственного интеллекта, протестировать ее на различных входных данных.
- Провести сравнительный анализ результатов взятых методов по определенным параметра, таким как время и путь.

1.3 Практическая значимость работы

Полученные в работе результаты позволяют понять, какой из 4 методов наиболее точно рассчитывает наименьшей путь для задачи с n количеством вершин и насколько быстро эти вычисления происходят.

2 СОДЕРЖАНИЕ ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ

2.1 Постановка задачи

Задача коммивояжера одна из знаменитых задач теории комбинаторики. Она заключается в том, чтобы найти самый выгодный маршрут, который будет проходить через указанные вершины (города) один раз с возвратом в начало пути.

История задачи коммивояжёра началась почти 200 лет: в 1832 г. в Германии была издана книга «Коммивояжёр: как он должен вести себя и что должен делать для того, чтобы доставлять товар и иметь успех в своих делах — советы старого курьера». В данной книге впервые была сформулирована проблема, но математическое решение не рассматривалось. Математическую постановку задачи сформулировал Карл Менгер в 1929 г., а позднее Хасслером Уитни назвал ее – Traveling Salesman Problem.

Сама постоновка задачи звучит так: пусть дано N городов. Коммивояжер (бродячий торговец) должен выйти из первого города, посетить по разу в неизвестном порядке города N-1 и вернуться в первый город. Расстояния между всеми городами известны. Требуется установить, в каком порядке коммивояжёру следует посетить города, чтобы суммарное пройденное расстояние было минимальным.

2.2 Концепция муравьиного алгоритма

Идея муравьиного алгоритма - моделирование поведения муравьёв, связанного с их способностью быстро находить кратчайший путь от муравейника к источнику пищи и адаптироваться к изменяющимся условиям, находя новый кратчайший путь. При своём движении муравей метит путь феромоном, и эта информация используется другими муравьями для выбора пути. Это элементарное правило поведения и определяет способность муравьёв находить новый путь, если старый оказывается недоступным.

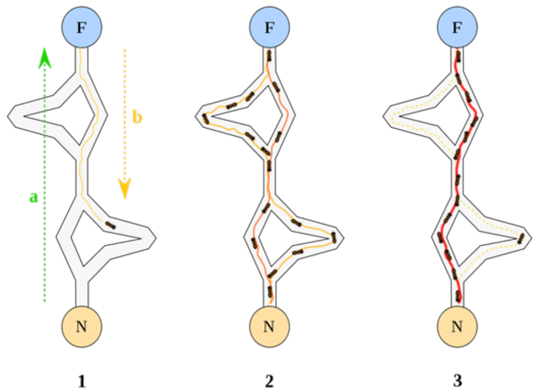

Рисунок 2 - Обход препятствий муравьями.

Рассмотрим случай, в соответствии с рисунком 2, когда у муравьев есть несколько путей чтобы добраться до еды. В этом случае необходимо определение нового оптимального пути. Дойдя до разветвления, муравьи с равной вероятностью могут пойти по тому или иному направлению. Однако, те муравьи, которые случайно выберут кратчайший путь, будут быстрее его проходить, и за несколько передвижений оно будет более обогащён феромоном. Поскольку движение муравьёв определяется концентрацией феромона, то следующие будут предпочитать именно этот путь, продолжая обогащать его феромоном до тех пор, пока этот путь по какой-либо причине не станет недоступен.

Очевидная положительная обратная связь быстро приведёт к тому, что кратчайший путь станет единственным маршрутом движения большинства муравьёв. Моделирование испарения феромона - отрицательной обратной связи - гарантирует нам, что найденное локально оптимальное решение не будет единственным - муравьи будут искать и другие пути. Если мы моделируем процесс такого поведения на некотором графе, рёбра которого представляют собой возможные пути перемещения муравьёв, в течение определённого времени, то наиболее обогащённый феромоном путь по рёбрам этого графа и будет являться решением задачи, полученным с помощью муравьиного алгоритма.

2.3 Описание метода имитации отжига

Метод отжига предназначен для поиска глобального минимума некоторой функции f(x), заданной для x из некоторого множества S. Элементы множества S представляют собой состояния воображаемой физической системы («энергетические уровни»), а значение функции f (x) в этих точках используется как энергия системы $E = f(x)$. В каждый момент времени предполагается заданной температура системы T , которая, как правило, уменьшается с течением времени. После попадания в состояние x при температуре T следующее состояние системы выбирается в соответствии с заданным порождающим семейством вероятностных распределений Q(x,T), которое при фиксированных x и T задает случайный элемент со значением G(x,T) в пространстве S. После генерации нового состояния $x' = G(x, T)$ система с вероятностью h(ΔE,T) переходит к следующему шагу в это состояние, в противном случае процесс генерации x' повторяется. Здесь ΔE обозначает приращение функции энергии $f(x') = f(x)$. Если ΔE меньше нуля, то новое состояние принимается всегда. Величина h(ΔE,T) называется вероятностью принятия нового состояния. Как правило, в качестве функции h(ΔE,T) выбирается либо точное значение соответствующей физической величины

$$h(\Delta E, T) = \frac{1}{1 + exp\left(\frac{\Delta E}{T}\right)}, \tag{3.1}$$

либо приближенное значение

$$h(\Delta E, T) = exp\left(\frac{\Delta E}{T}\right), \tag{3.2}$$

Вторая формула используется наиболее часто. При ее использовании h(E,T) оказывается больше единицы в случае E<0, и тогда соответствующая вероятность считается равной 1. Таком образом, если новое состояние дает лучшие значение оптимизируемой функции, то переход в этот состояние произойдет в любом случае.

Итак, конкретная схема метода отжига задается следующими параметрами:

1. выбором закона изменения температуры T(k), где k — номер шага;
2. выбором вероятностного распределения Q (x, T);
3. выбором функции вероятности принятия h(Δ E, T).

2.4 Описание генетических алгоритмов

Генетический алгоритм представляет собой метод, отражающей естественную эволюцию методов решения проблем, и в первую очередь задач оптимизации. Генетические алгоритмы – это процедуры поиска, основанные на механизмах естественного отбора и наследования. В них используется эволюционный принцип выживания наиболее приспособленных особей. Они отличаются от традиционных методов оптимизации несколькими базовыми элементами. В частности, генетические алгоритмы обладают рядом отличительных свойств:

1. кодирование параметров;
2. операции на популяции;
3. использование минимума информации о функции;

4. рандомизация операций.

Перечисленные свойства приводят в результате к устойчивости генетических алгоритмов.

Сфера применения генетических алгоритмов – это в основном оптимизация многопараметрических функций. Прикладное же применение генетических алгоритмов весьма обширно. Они применяются при разработке программного обеспечения в системах искусственного интеллекта, оптимизации, искусственных нейронных сетях и в других отраслях знаний. Следует отметить, что с их помощью решаются задачи, для которых ранее использовались нейронные сети. В этом случае генетические алгоритмы выступают просто в роли независимого от нейронных сетей метода, предназначенного для решения той же самой задачи. Примером может служить задача коммивояжера, изначально решавшаяся при помощи сети Хопфилда. Генетические алгоритмы часто используются совместно с нейронными сетями. Они могут поддерживать нейронные сети или наоборот, либо оба метода взаимодействуют в рамках одной гибридной системы, предназначенной для решения конкретной задачи. Генетические алгоритмы так же применяются совместно с нечеткими системами.

Целесообразность использования генетических алгоритмов изложена очень емко и кратко во фразе из статьи К. Де Йонга:

«Решающий аргумент при использовании генетических алгоритмов тесно связан с вопросом о том, какое пространство поиска будет исследовано. Если это пространство легко анализировать и его структура позволяет использовать специализированные методы поиска, то использование генетических алгоритмов менее эффективно с точки зрения затрат вычислительных ресурсов. Если же пространство поиска не поддается анализу и относительно слабо структурировано, и если возможен эффективный способ представления в генетических алгоритмах этого пространства, то они оказываются удивительно эффективным методом эвристического поиска в больших и сложных областях».

Но не стоит расценивать генетические алгоритмы как своеобразную панацею для задач оптимизации. С большой вероятностью генетические алгоритмы покажут как минимум не лучшие результаты по сравнению со специально разработанными методами для решения специализированных задач.

2.5 Алгоритм полного перебора

Полный перебор (или метод «грубой силы», англ. brute force) — метод решения математических задач. Относится к классу методов поиска решения исчерпыванием всевозможных вариантов. Сложность полного перебора зависит от количества всех возможных решений задачи. Если пространство решений очень велико, то полный перебор может не дать результатов в течение нескольких лет или даже столетий.

Любая задача из класса NP может быть решена полным перебором. При этом, даже если вычисление целевой функции от каждого конкретного возможного решения задачи может быть осуществлено за полиномиальное время, в зависимости от количества всех возможных решений полный перебор может потребовать экспоненциального времени работы.

2.6 Вычисления и анализ эксперимента

Был проведён сравнительный тест работы 4 алгоритмов на графах разных размерностей. В качестве тестовых задач использовались с разными количествами вершин (городов). Результаты тестирования приведены в соответствии с «Таблица 1» и «Таблица 2».

Таблица 1 - Сравнение точности решения задачи коммивояжера при помощи 4 алгоритмов.

Количество вершин (городов)	Муравьиный алгоритм	Метод отжига	Генетический алгоритм	Алгоритм полного перебора
4	356	356	356	356
5	590	590	590	590
6	678	678	678	678
7	725	725	725	725
8	897	897	897	897
9	1026	1026	1026	1026
10	1159	1159	1159	1159
11	1356	1356	1356	-
12	1429	1429	1429	-
15	1600	1600	1600	-
20	2033	2081	2056	-
25	2227	2402	2249	-
30	2636	2744	2654	-
35	3025	3095	3025	-
40	3336	3579	3336	-
45	3558	3797	3558	-
50	3844	4005	3891	-

Табоица 2 - Сравнение точности решения задачи коммивояжера при помощи 4 алгоритмов.

Количество вершин (городов)	Муравьиный алгоритм	Метод отжига	Генетический алгоритм	Алгоритм полного перебора
4	5.361	5.357	5.237	0.003
5	5.370	5.414	5.297	0.231
6	5,385	5,501	5,324	0,024
7	5,396	5,599	5,359	0,061
8	5,412	5,659	5,401	1,002
9	5,439	5,784	5,425	2,132
10	5,468	5,831	5,479	15,947
11	5,625	5,847	5,568	-
12	5,731	5,860	5,597	-
15	5,862	5,879	5,624	-
20	5.918	6.037	5.937	-
25	6.283	6.640	6.256	-
30	6.570	6.771	6.509	-
35	6.810	7.256	6.775	-
40	7.104	7.391	7.101	-
45	7.534	7.674	7.504	-
50	7.835	7.970	7.826	-

В ходе эксперимента установлено, что на относительно малых графах алгоритм полного перебора даёт точное решение и затрачивает малое количество времени в соответствии с рисунком 3. Однако по мере увеличения количества вершин временные затраты метода полного перебора изменяются согласно теоретической оценке (графы размерности 4-10 вершин). Использование полного перебора на графах большого размера нецелесообразно.

В соответствии с «Таблица 1» и «Таблица 2», очевидно, что муравьиный алгоритм находит более точное решение по сравнению с методом отжига и генетическим алгоритмом, так как эти два способа имеют относительно плохую сходимость.

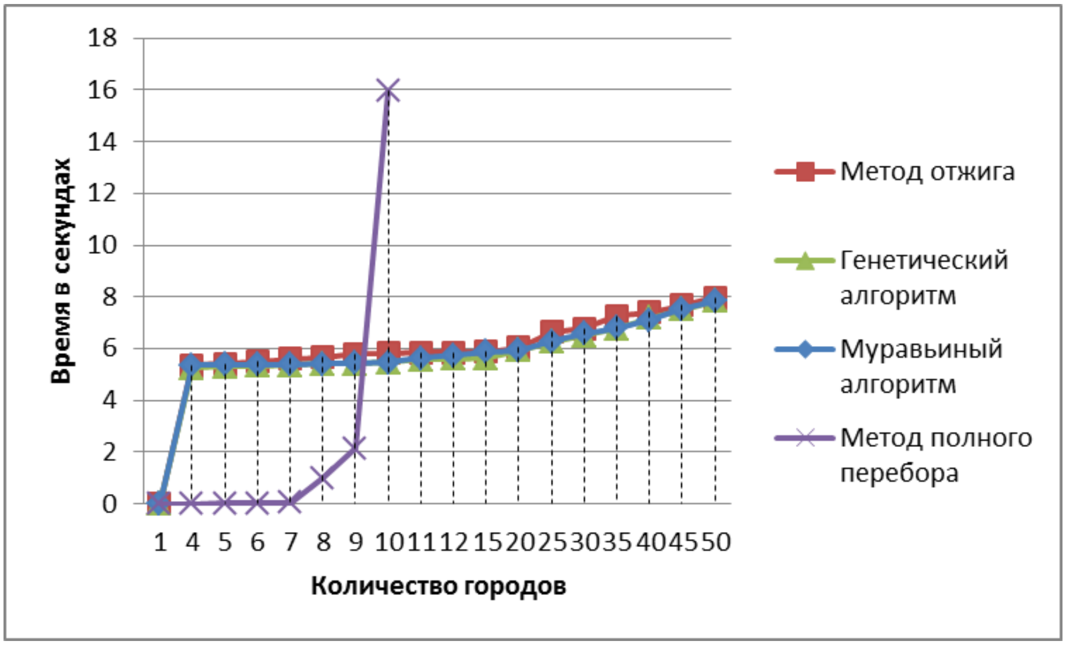

Рисунок 3 - Рост временных затрат.

ЗАКЛЮЧЕНИЕ

Как было сказано ранее, что задача коммивояжера имеет множество способов применения в современном мире. Так как нет единственного правильного решения этой задачи, то попытки его разработать привели к появлению множества методов и алгоритмов, которые имеют различия по времени вычисления и выбору пути. Выбор какого-либо метода или алгоритма осуществляется в зависимости от критериев, который важен в конкретном случае.

Из большого списка решений задачи коммивояжера были выбраны 4 метода, которые были рассмотрены в теоретической части и реализованы в практической части на языке C++. Генетический алгоритм, муравьиный алгоритм, метод отжига являются приближенными методами, а полный перебор дает точный результат.

Было проведено тестирование методов и алгоритмов на графах размерностью от 5 до 50, замерено время их выполнения. Сравнительный анализ который был проведен по показателям эффективности, такие как наименьший путь и время, дал понять, какую точность и скорость вычисления имеет каждый метод или алгоритм, какие у них плюсы и минусы. Лучшим по скорости вычисления был признан генетический алгоритм, а по сходимость лучше всего себя показал муравьиный алгоритм.