1. Как формулируется задача коммивояжера?

Необходимо посетить все города ровно один раз и вернуться в начальный город, затратив при этом, наименьшее расстояние (т.е. Гамильтонов цикл)

1. Какими методами может быть решена задача коммивояжера?
2. Переборный метод - это наиболее простой и прямолинейный подход, который состоит в переборе всех возможных маршрутов и выборе того, который обеспечивает наименьшую стоимость. Однако, при большом количестве городов, переборный метод становится непрактичным из-за высокой вычислительной сложности.
3. Метод ветвей и границ - это более эффективный метод, чем переборный метод, который заключается в разделении множества всех возможных маршрутов на подмножества и применении оценок для сужения области поиска. Этот метод может значительно сократить время поиска наиболее оптимального маршрута.

 **Разделяем** множество решений на **подмножества**.

 **Оцениваем** нижние границы стоимости решений в каждом подмножестве.

 **Отбрасываем** подмножества, которые не могут привести к оптимальному решению (**отсечение ветвей**).

 **Продолжаем** искать оптимальный путь, углубляясь в более перспективные ветви.

1. Методы приближенного решения - это подход, который не гарантирует точное решение задачи коммивояжера, но может найти достаточно хорошее приближенное решение за разумное время. Среди методов приближенного решения можно отметить метод ближайшего соседа, метод минимального остовного дерева и методы линейного программирования.

**Метод ближайшего соседа (Nearest Neighbor)**

Принцип:

1. Начинаем с **случайного города**.
2. Переходим в **ближайший** (по минимальному расстоянию) **непосещённый** город.
3. Повторяем, пока **не посетим все города**.
4. Возвращаемся в стартовый город.

### ****Муравьиный алгоритм (ACO - Ant Colony Optimization)****

1. Муравьи случайно ищут маршруты.
2. Запоминают **короткие маршруты**, откладывая **феромоны**.
3. Чем **короче путь**, тем больше **феромона**.
4. Следующие муравьи **предпочитают пути с большим количеством феромона**.
5. Постепенно **концентрируемся на лучшем пути**.
6. Методы эволюционной оптимизации - это группа методов, основанных на имитации эволюционных процессов в природе. Эти методы включают генетические алгоритмы, муравьиные алгоритмы, алгоритмы роя частиц и другие. Они могут использоваться для поиска наиболее оптимального маршрута в задаче коммивояжера.
7. Методы машинного обучения - это подход, основанный на использовании алгоритмов машинного обучения для обучения модели, которая может предсказывать наиболее оптимальный маршрут в задаче коммивояжера. Эти методы могут использоваться для нахождения оптимального маршрута на больших наборах данных.
8. Чем симметричная задача коммивояжера отличается от несимметричной?

Симметричная задача коммивояжера отличается от несимметричной задачи коммивояжера тем, что в симметричной задаче расстояния между парами городов одинаковы в обоих направлениях, т.е. расстояние между городами A и B равно расстоянию между городами B и A. В несимметричной задаче расстояние между городами может отличаться в зависимости от направления пути.

1. Чем замкнутая задача коммивояжера отличается от незамкнутой?

Замкнутая задача коммивояжера отличается от незамкнутой задачи коммивояжера тем, что в замкнутой задаче коммивояжера требуется найти наименьший циклический маршрут, проходящий через все города, в то время как в незамкнутой задаче коммивояжера маршрут не обязательно должен быть циклическим, т.е. может заканчиваться в любом городе. Другими словами, в замкнутой задаче коммивояжера город, с которого начинается маршрут, должен быть также и городом, в котором маршрут заканчивается.

1. В чем заключается принцип решения задачи коммивояжера методом ветвей и границ?

Метод ветвей и границ - это алгоритм решения задачи коммивояжера, который основывается на переборе всех возможных вариантов путей и отсечении ненужных ветвей. Алгоритм начинается с построения начального нижнего оценки длины пути и выбора первого города. Затем производится перебор всех возможных вариантов путей, путем построения дерева с отсечением неподходящих вариантов. При этом на каждом шаге вычисляется верхняя и нижняя границы длины пути, и отбрасываются варианты, которые не могут привести к более короткому пути, чем уже найденный. Если ветвь дерева не приводит к уменьшению длины пути, то она отбрасывается. Алгоритм заканчивает работу, когда перебраны все возможные пути и выбран оптимальный вариант.

### ****Общая схема работы****

1. **Инициализация**
   * Задаётся **матрица расстояний** между городами.
   * Начинаем с **нулевой оценки** и полного набора всех городов.
   * Устанавливаем **начальную нижнюю границу** для всех маршрутов.
2. **Редукция матрицы расстояний (снижение границы)**
   * Находим **минимальное значение** в каждой строке и **вычитаем его** из всех элементов строки.
   * Аналогично делаем **для столбцов**.
   * Сумма всех вычтенных значений — это **нижняя граница (LB - Lower Bound)**.
3. **Выбор дуги (ребра) для разбиения (ветвления)**
   * Выбираем **ноль в редуцированной матрице** с **наибольшей оценкой** (он даёт наибольшее влияние на путь).
   * Разбиваем задачу на два варианта:
     + **Включаем ребро** в маршрут.
     + **Исключаем ребро**, заменяя его на "бесконечность" (**INF**) в матрице.
4. **Рассчитываем нижнюю границу для новых подзадач**
   * Повторяем редукцию **с новой матрицей** для обеих подзадач.
   * Если нижняя граница **превышает текущий лучший маршрут**, эта ветвь **отбрасывается**.
5. **Продолжаем разбиение, пока не найдём оптимальный путь**
   * Когда **все города посещены**, путь замыкается, и мы получаем **оптимальный маршрут**.
6. Из каких процедур состоит метод ветвей и границ?

-Начальная инициализация данных и построение начальной нижней границы длины пути.

-Построение дерева с перебором всех возможных вариантов путей. Каждая вершина дерева соответствует выбору очередного города.

-Вычисление верхней и нижней границ длины пути для каждой вершины дерева.

-Нижняя граница - это сумма двух минимальных элементов в каждой строке матрицы, проходящих проверку на допустимость (элементы, равные бесконечности, не учитываются). Верхняя граница - это сумма длины найденного пути и оценки длины оставшейся части пути, рассчитанной на основе аналогичных минимальных элементов в каждой непосещенной строке.

-Отбрасывание недопустимых ветвей. Если верхняя граница длины пути больше уже найденной минимальной длины пути, то ветвь отбрасывается.

-Поиск оптимального пути. Алгоритм продолжает перебор ветвей до тех пор, пока не будут перебраны все возможные варианты путей, и выберет оптимальный вариант с наименьшим затраченным на это путём.

7. Какова область применения метода ветвей и границ?

Метод ветвей и границ - это эффективный алгоритм оптимизации, который может быть использован для решения широкого круга задач в различных областях. Он находит применение в следующих областях:

1. Комбинаторика - метод ветвей и границ может использоваться для решения различных задач комбинаторики, таких как задача о рюкзаке, задача о расписании, задача о назначениях и другие.
2. Производственный менеджмент - метод ветвей и границ может использоваться для решения задач оптимизации производственных процессов, например, для оптимизации распределения ресурсов, планирования производственных цепочек и других задач.
3. Логистика - метод ветвей и границ может использоваться для решения задач оптимизации логистики, например, для оптимизации маршрутов доставки, оптимизации загрузки транспортных средств и других задач.
4. Финансы - метод ветвей и границ может использоваться для решения задач оптимизации в финансовой сфере, например, для оптимизации портфеля инвестиций, оптимизации расписания облигаций и других задач.
5. Искусственный интеллект - метод ветвей и границ может использоваться для решения различных задач искусственного интеллекта, например, для оптимизации решения задачи обучения с подкреплением и других задач.

Таким образом, метод ветвей и границ может быть применен в широком спектре задач, которые требуют поиска оптимального решения из множества возможных вариантов.

8. Что такое жадный алгоритм?

Алгоритм, в котором на каждом шагу выбирается наиболее оптимальный вариант, но в конечном итоге, это будет не всегда приводить к рациональному и оптимальному решению (наиболее известные задачи – задача об рюкзаке и составлении расписания). В качестве примера, который способствует правильному решению задачи при помощи жадного алгоритма, можно вспомнить, как продавец выдаёт сдачу покупателю (сразу крупные купюры, а затем - мелкими).

9\*. В чем суть муравьиного алгоритма?

Каждый муравей откладывает феромоны (чем больше муравьев пройдет по пути, тем больше феромонов там останется, но муравьи так же учитывают и вероятность перехода (а она определяется благодаря расстоянию)). Подробнее – в лекциях Шимана за прошлый семестр.

10\*. В чем суть генетического алгоритма и какова его область применения?

Генетический алгоритм - это метод оптимизации, основанный на механизмах эволюции в природе. Он использует механизмы наследования, мутации и отбора для поиска наилучшего решения задачи оптимизации.

Суть генетического алгоритма заключается в том, что на начальном этапе создается популяция случайных решений (индивидов), которые кодируются генетически, например, в виде бинарных строк. Затем каждый индивид оценивается по определенному критерию (функции приспособленности), которая определяет, насколько хорошо данный индивид решает задачу.

Далее происходят операции, аналогичные механизмам эволюции. На первом этапе происходит отбор наиболее приспособленных индивидов, т.е. тех, которые решают задачу лучше остальных. Эти индивиды скрещиваются для создания новых потомков. На следующем этапе происходит мутация, т.е. случайное изменение генетического кода у некоторых индивидов, чтобы обеспечить более широкий поиск в пространстве решений. Таким образом, на каждом поколении популяции происходит постепенное улучшение качества решения.

Генетический алгоритм может применяться в различных областях, в которых требуется решение задач оптимизации, например:

1. Промышленность - генетический алгоритм может использоваться для оптимизации дизайна продуктов, оптимизации процессов производства, оптимизации логистических цепочек и т.д.
2. Финансы - генетический алгоритм может использоваться для оптимизации инвестиционного портфеля, прогнозирования финансовых рынков и т.д.
3. Медицина - генетический алгоритм может использоваться для оптимизации диагностики и лечения заболеваний, оптимизации процессов лекарственной разработки и т.д.
4. Искусственный интеллект - генетический алгоритм может использоваться для оптимизации процессов обучения нейронных сетей, оптимизации архитектуры нейронных сетей.