Задачи по дисциплине «Оценка сложности алгоритма»

Преподаватель Лапина О.Н.

«NP-полные задачи». 2024 г.

Задание:

1. ***Определить принадлежность задачи к классу NP задач (****для задач принятия решений)*.

   Для задач оптимизации описать двухэтапный недетерминированный процесс решения задач (см. Лекция NP-задачи). В ответ должен входить формат выхода недетерминированного шага, причем он должен включать в себя все элементы решения задачи. Например, в задаче о коммивояжере выходные данные должны представлять собой список всех городов в порядке их посещения. Кроме того, необходимо описать процесс проверки того, что предложенный вариант действительно является решением задачи.

2. ***Определить принадлежность задачи к классу NP –полных или NP-трудных задач.*** Каждая из NP-полных задач может быть сведена к любой другой за полиномиальное время. Опишите соответствующее преобразование какой-либо NP-полной задачи к ЗАДАННОЙ. Примеры возможного сведения:

   а) Упаковка рюкзака - раскладка по ящикам

   б) Раскладка по ящикам - планирование работ

   в) Планирование работ - суммы элементов подмножеств

   г) Суммы элементов подмножеств - коммивояжер

   д) Коммивояжер - планирование работ и т.п.

3. Определить, является ли задача NP-полной в сильном смысле (существует ли для ее решения псевдополиномиальный алгоритм).

4. Изучить приближенные методы решения задачи. Реализовать точный и 3-4 приближенных алгоритма на произвольном ЯП. Дать оценку сложность алгоритмов.

5. Оформить письменный отчет.

Задачи:

1. Задача о клике.

Дан граф G с m вершинами и целое положительное число n.

Граф называется кликой, если каждая вершина в нем связана ребром с каждой. Количество вершин в клике назовем ее мощностью.

- Задача принятия решения: Найдется ли в данном графе G клика мощности не менее, чем n?

- Задача оптимизации: найти максимальный размер клики в графе.

1.	Задача принятия решения	
2.	Задача оптимизации	

2. Вершинное покрытие.

Дан граф G с m вершинами и целое положительное число n.

Вершинным покрытием называется подмножество вершин графа $Z \subseteq V$, такое, что любое ребро графа G инцидентно хотя бы одной вершине множества Z.

- Задача принятия решения: Существует ли вершинное покрытие не более, чем из n вершин.

- Задача оптимизации: найти минимальное вершинное покрытие.

1.	Задача принятия решения	
2.	Задача оптимизации	

3. Независимое множество вершин.

Множество вершин $V_1 \subset V$ графа $G(V, E)$, независимо, если никакие две вершины из V_1 не связаны ребром.

- Задача оптимизации: Найти максимальное независимое множество.

- Задача принятия решения: Существует ли в графе независимое множество из k вершин?

1.	Задача принятия решения	

2.	Задача оптимизации	

4. Задача о самом длинном пути.

Дан граф $G(V, E)$ и номера 2-х вершин A и B, а также число k.

- Задача оптимизации: Найти путь из A в B максимальной длины.

- Задача принятия решения: Существует ли путь из A в B длиной не менее k?

1.	Задача принятия решения	
2.	Задача оптимизации	

5. Трехмерное сочетание.

Даны 3 множества A, B, C. Мощности их равны $|A| = |B| = |C| = m$, их множества не пересекаются. Задано множество троек $N \subseteq A \times B \times C$, $|N| = n$. Трехмерным сочетанием называется подмножество $M \subseteq N$, $|M| = m$ в котором никакие две разные тройки не имеют ни одной равной координаты, т.е. если $(x_1, y_1, z_1) \in M$ и $(x_2, y_2, z_2) \in M$, то $x_1 \neq x_2$, $y_1 \neq y_2$, $z_1 \neq z_2$.

- Задача оптимизации: Найти в множестве N трехмерное сочетание максимальной мощности $m' \leq m$.

- Задача принятия решения: содержит ли множество N трехмерное сочетание?

1.	Задача принятия решения	
2.	Задача оптимизации	

6. Задача раскраски графа.

- Задача оптимизации: найти минимальное число цветов, необходимых для раскраски вершин графа.

- Задача принятия решения: можно ли раскрасить вершины в C или менее цветов.

1.	Задача принятия решения	
2.	Задача оптимизации	

7. Раскладка по ящикам

Пусть у нас есть несколько ящиков единичной емкости и набор объектов различных размеров $s_1, s_2, \ldots s_N$.

В задаче оптимизации нас интересует наименьшее количество ящиков, необходимое для раскладки всех объектов,

В задаче принятия решения — можно ли упаковать все объекты в B или менее ящиков.

1.	Задача принятия решения	
2.	Задача оптимизации	

8. Упаковка рюкзака

Имеется набор объектов объемом $s_1, s_2, \ldots s_N$ стоимости $w_1, w_2, \ldots w_N$

- В задаче оптимизации нужно упаковать рюкзак объемом K так, чтобы его стоимость была максимальной.

- В задаче принятия решения нас интересует, можно ли добиться, чтобы суммарная стоимость упакованных объектов была по меньшей мере W .

1.	Задача принятия решения	
2.	Задача оптимизации	

9. Задача о суммах элементов подмножеств.

Пусть у нас есть множество объектов различных размеров $s_1, s_2, \ldots s_N$ и некоторая положительная верхняя граница L.

В задаче оптимизации необходимо найти набор объектов, сумма размеров которых наиболее близка к L и не превышает этой верхней границы.

В задаче принятия решения нужно установить, существует ли набор объектов с суммой размеров L.

Это упрощенная версия задачи об упаковке рюкзака.

1.	Задача принятия решения	
2.	Задача оптимизации	

10.Задача планирования работ

Пусть у нас есть набор работ, и мы знаем время, необходимое для завершения каждой из них, $t_1, t_2, \ldots t_N$, сроки $d_1, d_2, \ldots d_N$ к которым эти работы должны быть обязательно завершены, а также штрафы $p_1, p_2, \ldots p_N$, которые будут наложены при не завершении каждой работы в установленные сроки.

- Задача оптимизации требует установить порядок работ, минимизирующий накладываемые штрафы.
- Задача принятия решений: есть ли порядок работ, при котором величина штрафа будет не больше P.

1.	Задача принятия решения	

2.	Задача оптимизации	

11. **Задача о многопроцессорном расписании**.

Даны m одинаковых процессоров, n независимых задач, время решения каждой задачи t_i и время T.

- Задача оптимизации: нужно распределить задачи по процессорам, так, чтобы время решения было минимальным.
- Задача принятия решения: можно ли распределить задачи по процессорам так, чтобы время решения было не больше T?

1.	Задача принятия решения	
2.	Задача оптимизации	

12. Множество ребер, разрезающих циклы.

Для произвольного графа $G(V,E)$ и целого числа k выяснить, существует ли множество ребер $E' \subseteq E$, такое, что $|E'| = k$ и каждый цикл графа $G(V,E)$ содержит ребро из E'.

- Задача оптимизации: найти множество ребер минимальной мощности.
- Задача принятия решения: существует ли множество ребер, мощностью k?

1.	Задача принятия решения	
2.	Задача оптимизации	

13.Разбиение.

Дано множество положительных целых чисел $x_1, x_2, \ldots x_n$.

Можно ли разбить его на 2 подмножества так, чтобы суммы чисел в обоих подмножествах были равны?

1.	Задача принятия решения	

2	Задача оптимизации	

14.**Множество вершин, разрезающих циклы**.

Для произвольного графа $G(V, E)$ и целого числа k выяснить, существует ли множество вершин $V' \subseteq V$, такое, что $|V'| = k$ и каждый цикл графа $G(V, E)$ содержит вершину из V'.

- Задача оптимизации: найти множество вершин минимальной мощности.
- Задача принятия решения: существует ли множество вершин, мощностью k?

1.	Задача принятия решения	
2.	Задача оптимизации	

15. Задача изоморфизма подграфу.

Даны 2 графа $G_1 = (V_1, U_1)$ и $G_2 = (V_2, U_2)$. Пусть $|V_1| > |V_2|$ Требуется ответить на вопрос: Найдется ли в графе G_1 подграф H, изоморфный графу G_2 ?

- Задача оптимизации: Найти максимальное множество подграфов в графе G_1, изоморфных графу G_2.
- Задача принятия решения: Найдется ли в графе G_1 подграф H, изоморфный графу G_2 ?

1.	Задача принятия решения	
2.	Задача оптимизации	

16. Задача проверки простоты числа.

- ***Метод Монте-Карло:***

Выполните проверку числа на простоту алгоритмом Монте Карло для чисел 182 и 255, выписывая случайно выбранные числа и полученный результат. Сколько раз приходится выполнять проверку прежде, чем будет получен правильный результат? (В каждом случае начинайте с начала таблицы случайных чисел.)

- ***Реализовать классический и вероятностный алгоритм (Китайская теорема об остатках).***

- ***Реализовать любой другой алгоритм проверки числа на простоту***

1.	Вероятностный алгоритм	
2.	Другой алгоритм	

17. Задача факторизации длинных целых чисел.

Требуется разложит целое число на простые множители. Реализовать классический алгоритм и приближенный алгоритм.

1.		
2.		

18. Задача коммивояжера.

Постановка классической задачи Коммивояжера (КЗК).

Задан набор городов и «стоимость» (d_{ij}) путешествия между любыми двумя из них. Нужно определить такой порядок, в котором следует посетить все города (по одному разу) и вернуться в исходный город, чтобы общая стоимость путешествия оказалась минимальной.

При этом на «стоимость» накладываются условия неотрицательности ($d_{ij} >=0$), симметричности ($d_{ij} =d_{ji}$), запрета на петли.

- Задача оптимизации: найти путь минимальной стоимости.

-Задача принятия решения: существует ли путь коммивояжера со стоимостью, меньшей заданной константы С.

К настоящему времени считается, что наиболее эффективный из точных методов —метод ветвей и границ — применим при задаче, содержащей около 100 городов, поэтому будем полагать, что размерность задачи может считаться большой, если она близка или превышает эту величину.

Неклассические задачи Коммивояжера.

Задача коммивояжера возникает в обширном классе приложений. В КЗК вводятся ограничений на количество отображаемых объектов (только города и дороги), используется единственное отношение типа «находиться на расстоянии … от …» и только детерминированные переменные. Эти ограничения позволяют свести представление системы к реберно-взвешенному неориентированному графу, но не учитывают многих аспектов задач, возникающих на практике.

Классическая задача Коммивояжера не учитывает такие аспекты практических задач, как динамичность, нечеткость, неточность, недоопределенность и др., что не позволяет представить их в виде КЗК. Поиск оптимального решения КЗК без учета этих свойств реального мира ведет к решению, которое может стать неудовлетворительным даже при незначительных изменениях условий задачи.

Это приводит к появлению новых формулировок КЗК введением новых типов отношений и переменных либо дополнительных моделируемых объектов. Такие задачи будем называть *неклассическими*, *например,* *обобщенную ЗК, ЗК с временными окнами, ЗК с движущимися целями, динамическую ЗК, вероятностную ЗК.*

Важный признак классификации ЗК — топологические особенности задачи — совокупность ограничений на отношения между объектами. На практике эти ограничения, особенно условие симметричности в КЗК, часто нарушаются, например, если пункты i и j соединяются двумя дорогами с односторонним движением разной длины. В этой связи различают два варианта КЗК: *симметричную,* когда условие симметричности выполнено, и *асимметричную* — в противном случае.

Различают ***метрическую ЗК*** при выполнении условия: стоимость проезда из города в город подчиняется правилу треугольника $(d_{ik} >= d_{ij}+d_{jk})$ и неметрическую — в обратном случае (проезд из i в k напрямую стоит дороже, чем проезд транзитом через j).

Если не нужно возвращаться в исходный пункт, задача — *незамкнутая* (открытая).

ЗК можно также классифицировать по количеству критериев (однокритериальные, многокритериальные); по количеству коммивояжеров (с одним и с несколькими) и др. Рассматриваются также частные случаи ЗК, когда накладываются дополнительные ограничения на граф, описывающий задачу.

Задача коммивояжера с временными окнами.

Задача коммивояжера с временными окнами — задача поиска маршрута автомобиля, развозящего заказы потребителям. Все заказы загружены в машину на складе — исходной точке маршрута — и для каждого потребителя задан интервал времени, в течение которого он должен быть посещен машиной, заданы матрицы расстояний и времени проезда между потребителями. Если машина прибывает к потребителю до начала интервала его обслуживания, то она ожидает его начала, после чего начинается разгрузка, а если после завершения временного интервала, маршрут либо считается недопустимым *(задача с жесткими временными окнами),* либо подвергается штрафу за опоздание *(задача с мягкими временными окнами).* Требуется найти маршрут посещения потребителей, обладающий, по возможности, наивысшим качеством.

Если сравнить этот вариант задачи коммивояжера с КЗК, можно заметить, что добавились две детерминированные переменные: «время проезда между потребителями» и «интервал обслуживания потребителя», причем граф модели стал вершинно- и реберновзвешенным.

Нечеткая задача Коммивояжера.

В традиционной постановке этой задачи предполагается, что «расстояние» между узлами транспортной сети – детерминированная величина. Однако если в качестве расстояния используется, например, время перехода (переезда) от одного узла к другому, это предположение становится нереалистичным. На практике время перемещения по дуге сети между узлами зависит от множества случайных факторов: погоды и связанным с ней состоянием покрытия дороги, соотношения между пропускной способностью участка транспортной сети и интенсивностью потока транспортировок, типа транспортного средства, дня недели, времени суток, в которое осуществляется транспортировка и т.д. При этом важно отметить, что характер и уровень влияния перечисленных факторов не являются стабильными. Это обстоятельство ставит под сомнение целесообразность попыток описания численных значений продолжительностей перевозок между пунктами (узлами) транспортной сети как случайных величин с некоторым законом распределения, поскольку числовые характеристики этого закона не могут быть однозначно оценены. В связи с этим для адекватного описания указанных величин естественно использовать нечеткие числа.

Требуется решить одну из неклассических задач Коммивояжера.

1.	Задача принятия решения	
2.	Задача оптимизации	

Список источников.

1. Колесников А.В., Кириков И.А., и др. Решение сложных задач коммивояжера методами функциональных гибридных интеллектуальных систем / Под ред. А.В. Колесникова. — М.: ИПИ РАН, 2011. — 295 с., ил. — ISBN 978-5-902030-88-1
2. Серая О.В. Нечеткая задача коммивояжера. Мат. мод № 2 (17), 2007