

Постановка задачи

Неформальная постановка задачи

В классической формулировке задачи главным действующим лицом является почтальон, которому необходимо забрать корреспонденцию в почтовом отделении, разнести почту по всем улицам города, и затем вновь вернуться в почтовое отделение. Задача состоит в том, чтобы отыскать наиболее оптимальный путь для почтальона. Сформулированная выше задача имеет много потенциальных приложений. Например, оптимальный обход улиц снегоуборочной техникой, а также решение других проблемы, связанных с маршрутизацией.

Примечание: только в случае ориентированного и неориентированного графа задача китайского почтальона имеет полиномиальное решение. В случае смешанного графа задача является NP -полной.

Задача, сформулированная в терминах теории графов

Из неформальной постановки задачи следует, что существует связный неориентированный граф G . Рёбрам графа G приписаны положительные веса. Требуется найти цикл, проходящий через каждое ребро графа G по крайней мере один раз и такой, что для него общий вес (а именно сумма величин $n_j w(a_j)$, где число n_j показывает, сколько раз проходило ребро a_j , а $w(a_j)$ – вес ребра a_j) минимален. Очевидно, что если G содержит эйлеров цикл, то любой такой цикл будет оптимальным, так как каждое ребро проходится только один раз и вес этого цикла равен тогда $\sum_{j=1}^m w(a_j)$.

Основные определения:

Эйлеровый граф – граф, содержащий эйлеров цикл.

Эйлеров цикл – цикл проходящий по всем рёбрам и при том только по одному разу.

Цикл - замкнутый обход, состоящий из последовательности вершин, начинающейся и заканчивающейся в той же самой вершине, и каждые две последовательные вершины в последовательности смежны, то есть, соединены ребром.

Взвешенный граф - граф, каждому ребру которого поставлено в соответствие некое значение (вес ребра)

Полный граф - граф, в котором каждая пара различных вершин смежна.

Двудольный граф - граф, множество вершин которого можно разбить на две части таким образом, что каждое ребро графа соединяет какую-то вершину из одной части с какой-то вершиной другой части, то есть не существует ребра, соединяющего две вершины из одной и той же части.

Паросочетание в двудольном графе - произвольное множество рёбер двудольного графа, такое что никакие два ребра не имеют общей вершины.

Паросочетание графа G называется **совершенным** (или **полным**), если оно покрывает все вершины графа.

Идея решения задачи

Рассмотрим неориентированный граф $G = (V, E)$. Среди вершин из V некоторые вершины (скажем, из множества V^+) будут иметь чётные степени, а остальные (из множества $V \setminus V^+ = V^-$) – нечётные степени. Далее сумма степеней d_i всех вершин $v_i \in V$ равна удвоенному числу рёбер в E (так как каждое ребро

добавляет по единице к степеням двух его концевых вершин) и поэтому равна чётному числу $2m$, следовательно:

$$\sum_{i:v_i \in V} d_i = \sum_{i:v_i \in V^-} d_i + \sum_{i:v_i \in V^+} d_i = 2 * m \quad (1)$$

Сумма степеней всех вершин равна удвоенному количеству рёбер, а значит она чётная. Сумма степеней всех чётных вершин чётная, следовательно, сумма степеней всех нечётных вершин так же чётная. Отсюда следует, что количество вершин с нечётной степенью чётно.

Пусть M – множество таких цепей (скажем, μ_{ij}) в $G = (V, E)$ между концевыми вершинами v_i и v_j , $v_i, v_j \in V^-$, что никакие две цепи не имеют одинаковых конечных вершин, т.е. цепи соединяют различные пары вершин из V^- и покрывают все вершины множества V^- . Число цепей μ_{ij} в M равно $\frac{1}{2}|V^-|$, а как было показано выше, $|V^-|$ всегда чётно, следовательно, это число всегда целое.

Предположим теперь, что все рёбра, образующие цепь, добавлены к G в качестве искусственной параллельной цепи, причём эти ребра остаются в графе G . Это означает прежде всего, что все рёбра из G , образующие цепь μ_{ij} , теперь удвоены. Так поступаем с каждой цепью $\mu_{ij} \in M$, и полученный граф обозначим через $G^-(M)$. Поскольку некоторые рёбра из графа G могут входить более чем в одну цепь μ_{ij} , некоторые рёбра из $G^-(M)$ могут быть (после того как добавлены все «новые» цепи μ_{ij}) утроены, учетверены и т.д.

Теорема 1

Для любого цикла, проходящего по графу G , можно выбрать множество M , для которого граф $G^-(M)$ имеет эйлеров цикл, соответствующий первоначально взятому циклу в графе G . Это соответствие таково, что если цикл проходит по ребру (v_i, v_j) из G l раз, то в $G^-(M)$ существует l рёбер (одно реальное и $l - 1$

искусственных) между v_i и v_j , каждое из которых проходится ровно один раз эйлеровым циклом из $G^-(M)$. Справедливо и обратное утверждение.

Доказательство

Если цикл проходит по графу G , то по крайней мере одно ребро, инцидентное каждой вершине v_i нечётной степени, должно проходиться дважды. Ребро, проходящее дважды, можно рассматривать как два параллельных ребра – одно реальное и одно искусственное – и каждое из них проходится один раз. Пусть это ребро (v_i, v_k) . В случае нечётности степени d_k вершины v_k графа G добавление искусственного ребра прежде всего сделает d_k чётным, и значит только ребро (v_i, v_k) нужно будет проходить дважды, если ограничиться рассмотрением лишь вершин v_i и v_k . В случае, когда d_k чётно, добавление искусственного ребра сделает d_k нечётным, а второе ребро, выходящее из v_k , должно быть пройдено дважды (т.е. добавляется еще одно искусственное ребро). Такая ситуация сохраняется до тех пор, пока не встретится вершина нечётной степени, о чём говорилось выше. Следовательно, чтобы удовлетворить условию возвращения в вершину v_i , нужно дважды пройти всю цепь из v_i в некоторую другую вершину нечётной степени v_r , принадлежащую множеству V^- . Это автоматически приводит к выполнению условия прохождения вершины v_r . Аналогично обстоит дело для всех других вершин v_i из V^- . Это значит, что всё множество M цепей из G , определённое выше, должно проходиться дважды, и так как отсюда вытекает, что каждое ребро из $G^-(M)$ должно проходиться один раз, то теорема доказана.

Алгоритм решения задачи китайского почтальона непосредственно следует из доказанной теоремы, так как всё, что для этого необходимо, состоит в нахождении множества цепей M^* (цепного паросочетания для множества вершин нечётной степени), дающего наименьший дополнительный вес. Цикл наименьшего веса, проходящий по G , будет иметь вес, равный сумме весов всех рёбер из G плюс сумма весов рёбер в цепях из M^* . Это то же самое, что и сумма весов всех рёбер – реальных и искусственных – графа $G^-(M^*)$.

Алгоритм

Шаг 1. Проверить, является ли граф G эйлеровым: проверить являются ли степени всех вершин чётными. В случае положительного результата перейти сразу к шагу 5, иначе перейти к шагу 2.

Шаг 2. Пусть $C = (c_{ij})$ матрица весов ребер связного графа G . Используя алгоритм кратчайшей цепи Дейкстры, образуем $|V^-| \times |V^-|$ матрицу $D = (d_{ij})$, где d_{ij} – вес цепи наименьшего веса, идущей из некоторой вершины $v_i \in V^-$ в вершину $v_j \in V^-$, где V^- – множество вершин, степени которых нечётные.

Шаг 3. В полном графе на множестве вершин V^- найдем полное паросочетание M^* с минимальным весом (в соответствии с матрицей весов D). Это можно сделать эффективно с помощью венгерского алгоритма решения задачи о назначениях.

Шаг 4. Если вершина v_a сочетается с другой вершиной v_b , то определим цепь μ_{ab} наименьшего веса (из v_a в v_b), соответствующую весу d_{ab} , используя алгоритм кратчайшей цепи Дейкстры (шаг 2). Добавим искусственные ребра в G , соответствующие ребрам из $\mu_{\alpha\beta}$, и проделаем это для всех других цепей из множества M^* , в результате чего получится Эйлеров граф $G^-(M^*)$ (вес искусственного ребра берётся равным весу параллельного ему реального ребра).

Примечание: Следует заметить, что поскольку на шаге 2 мы используем минимальное паросочетание, никакие две кратчайшие цепи μ_{ij} и μ_{pq} при таком паросочетании (скажем, идущие из v_i в v_j и из v_p в v_q) не могут иметь общего ребра. Если бы они имели общее ребро (v_a, v_b) , как показано на рис. 1, то сочетание вершин v_p и v_j (использующее подцепи от v_p к v_a и от v_a к v_j) и сочетание пары вершин v_i и v_q (использующее подцепи v_i к v_b и от v_b к v_q) давало бы общее паросочетание меньшего веса чем первоначальное паросочетание, что противоречит предположению о минимальности исходного паросочетания.

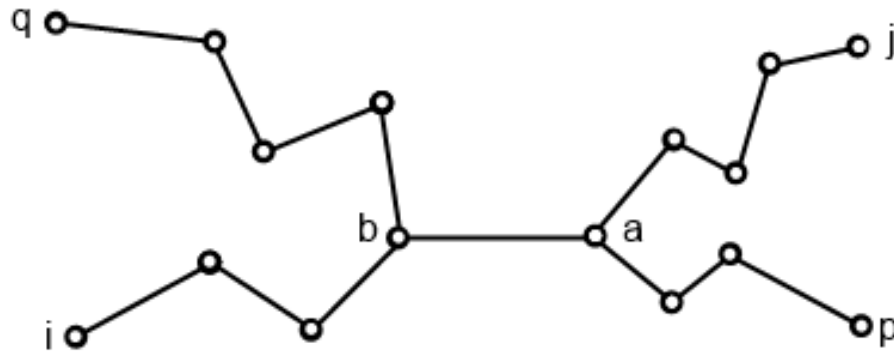


Рис. 1. Цепи μ_{ij} и μ_{pq} имеющие общее ребро (a, b) .

Следовательно, граф $G^-(M^*)$ не должен содержать более двух параллельных ребер между любыми двумя вершинами, т.е. оптимальный цикл не проходит ни по какому ребру графа G более чем два раза. Это значит, что степени всех нечётных вершин увеличились на один (так как это начала и концы новых искусственных цепей), а степени чётных вершин, через которые проходят искусственные цепи, увеличились на два (так как цепь в вершину входит и исходит), следовательно, в полученном графе $G^-(M^*)$ все вершины имеют чётную степень, а это значит, что граф $G^-(M^*)$ Эйлеров.

Шаг 5. Построим Эйлеров цикл используя алгоритм Флёрис.

Примечание: Сумма весов всех ребер графа $G^-(M^*)$ равна минимальному весу цикла, проходящего по G . При этом число прохождений цикла по ребру (v_i, v_j) равно общему числу параллельных ребер между $v_i \in V^-$ и $v_j \in V^-$ в $G^-(M^*)$.

Описания элементов алгоритма

Алгоритм кратчайшей цепи Дейкстры

Алгоритм Дейкстры применяется для решения задачи о кратчайшей цепи при условии, что $w(e) > 0$ для всех $e \in E$ (т.е. веса для всех рёбер графа положительные). Алгоритм основан на присвоении вершинам меток. Первая часть метки $l(u)$ вершины u указывает текущее кратчайшее расстояние от вершины s до вершины u , вторая – предшествующую вершину в текущей цепи от вершины s до вершины u . Метки могут быть временными или постоянными. Постоянная метка не может изменяться в процессе выполнения алгоритма. Для вершины $p \in V$ пусть $G(p) = \{x \in V : (p, x) \in E\}$.

Шаг 1. Положить метку вершины s равной $(0, s)$ и считать эту метку постоянной. Для всех вершин $u \neq s$ положить метки равными (∞, s) и считать эти метки временными. За текущую рассматриваемую вершину с постоянной меткой взять вершину $p = s$.

Шаг 2. Для каждой вершины $u \in G(p)$ с временной меткой изменить метку в соответствии со следующим выражением: $l(u) = \min\{l(u), l(p) + w(p, u)\}$. При этом если первая часть метки изменилась, то изменить вторую часть метки, положив ее равной p .

Шаг 3. Среди всех вершин с временными метками найти вершину u с минимальной первой частью $l(u)$ метки.

Шаг 4. Считать метку вершины u постоянной и положить $p = u$.

Шаг 5. (а) (При нахождении цепи из вершины s в вершину t) Если $p = t$, то $l(t)$ является длиной кратчайшей цепи из s в t . Алгоритм завершает работу. Если $p \neq t$, то перейти к шагу 2.

(б) (При нахождении цепей от s ко всем остальным вершинам.) Если все вершины помечены постоянными метками, то эти метки дают длины кратчайших

цепей. Алгоритм завершает работу. Если некоторые метки являются временными, то перейти к шагу 2.

По завершении алгоритма первые части меток дают искомые кратчайшие расстояния. Сами кратчайшие цепи можно получить с помощью рекурсивной процедуры, начиная с вершины t , для которой ищется кратчайшая цепь от s . Вторая часть метки вершины t указывает вершину u , непосредственно предшествующую ей в кратчайшей цепи от s к t . Берем вторую часть метки вершины u и повторяем действия, пока не достигнем вершины s .

Венгерский алгоритм решения задачи о назначениях

Венгерский алгоритм (англ. *Hungarian algorithm*) — алгоритм, решающий задачу о назначениях за полиномиальное время. Задача формулируется следующим образом: пусть дан взвешенный полный граф с целыми весами ребер, нужно найти в нем полное паросочетание минимального веса. Вес паросочетания определяется как сумма весов его ребер.

Алгоритм, решающий задачу, работает с графом, как с матрицей весов: для удобства, рассмотрим алгоритм на примере конкретной задачи (полный граф из 4 вершин: рис. 2), представленной в матричном виде:

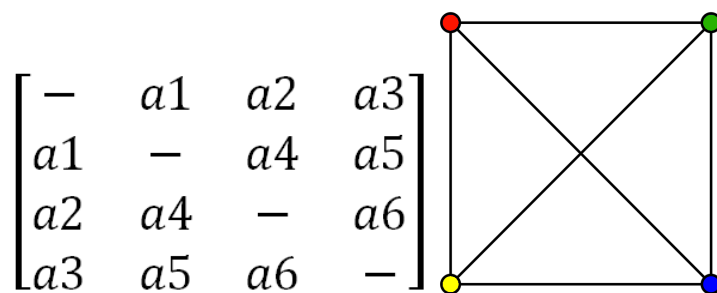


Рис. 2.

Данную матрицу можно интерпретировать как двудольный граф $K_{n,n}$, где $n = 4$ (рис. 3). В матрице значения элементов на диагонали равны ∞ т.е. бесконечно возможный вес, чтобы исключить возможность паросочетания в двудольном графе двух вершин, которые в изначальном полном графе являются одной и той же.

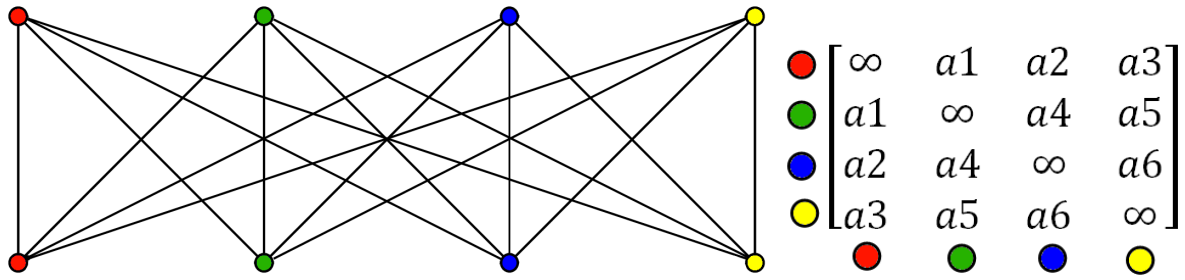


Рис. 3.

Далее будем обозначать левую и правую доли графа за X и Y соответственно, вес ребра xy — как $w(xy)$.

Вспомогательные леммы:

1. Если веса всех ребер графа, инцидентных какой-либо вершине, изменить (увеличить или уменьшить) на одно и то же число, то в новом графе оптимальное паросочетание будет состоять из тех же ребер, что и в старом.
2. Выделим в множествах X и Y подмножества X' , Y' . Пусть $d = \min\{w(xy) | x \in X \setminus X', y \in Y'\}$. Прибавим d ко всем весам ребер, инцидентных вершинам из X' . Затем отнимем d от всех весов ребер, инцидентных вершинам из Y' (далее для краткости эта операция обозначается как $X' \uparrow \downarrow Y'$)

Тогда:

- Веса всех ребер графа останутся неотрицательными.
 - Веса ребер вида xy , где $x \in X'$, $y \in Y'$ или $x \in X \setminus X'$, $y \in Y \setminus Y'$, не изменятся.
3. Если веса всех ребер графа неотрицательны и некоторое полное паросочетание состоит из ребер нулевого веса, то оно является оптимальным.

Шаг 1. Вычитаем из каждой строки значение ее минимального элемента. Теперь в каждой строке есть хотя бы один нулевой элемент (рис. 4).

0	a2'	a3'	a4'
b1'	b2'	b3'	0
0	c2'	c3'	c4'
d1'	0	d3'	d4'

Рис. 4.

Ищем в текущем графе полное паросочетание из ребер нулевого веса: если оно найдено, то желаемый результат достигнут, алгоритм закончен, если нет, то переходим к шагу 2.

Шаг 2. Вычитаем из каждого столбца значение его минимального элемента. Теперь в каждом столбце есть хотя бы один нулевой элемент (рис. 5).

0	a2'	a3'	a4'
b1'	b2'	b3'	0
0	c2'	c3'	c4'
d1'	0	0	d4'

Рис. 5.

Ищем в текущем графе полное паросочетание из ребер нулевого веса: если оно найдено, то желаемый результат достигнут, алгоритм закончен, если нет, то переходим к шагу 3.

Шаг 3. Покроем нули матрицы весов минимальным количеством строк и столбцов - это не что иное, как нахождение минимального вершинного покрытия в двудольном графе (рис 6). Пусть X_c и Y_c — множества вершин минимального вершинного покрытия из левой и правой долей (то есть, строк и столбцов).

0	a2'	a3'	a4'
b1'	b2'	b3'	0
0	c2'	c3'	c4'
d1'	0	0	d4'

Рис. 6

Шаг 4. Применим преобразование $X_c \uparrow \downarrow (Y \setminus Y_c)$. Для этого преобразования d будет минимумом по всем ребрам между $X \setminus X_c$ и $Y \setminus Y_c$ то есть, ребер нулевого веса здесь нет, поэтому, после его выполнения в матрице весов появится новый нуль. После этого перейдем к шагу 1.

Иными словами, после шагов 1 и 2 закрываем все нули в полученной матрице, используя минимальное количество «вертикальных и горизонтальных линий» (рис. 6). Если для этого требуется n «линий», то между нулями существует оптимальное значение. Если требуется менее n «линий», то среди элементов, непокрытых «линиями», находится наименьший, вычитается из всех непокрытых элементов и прибавляется ко всем элементам, покрытым дважды (рис. 7), после чего возвращаемся к шагу 1.

0	0	a3'-a2'	a4'-a2'
b1'+a2'	b2'	b3'	0
0	c2'-a2'	c3'-a2'	c4'-a2'
d1'+a2'	0	0	d4'

Рис. 7

Алгоритм Флёрри (построение Эйлера цикла)

Пусть G – эйлеров граф. Тогда следующая процедура всегда возможна и приводит к эйлеровой цепи графа G : выйдя из произвольной вершины, идём по рёбрам графа произвольным образом, соблюдая лишь следующие правила:

а) стираем рёбра по мере их прохождения, а также изолированные вершины (т.е. вершины, не инцидентные никакому ребру), которые при этом образуются;

б) на каждом шаге идём по мосту (т.е. ребру, удаление которого увеличивает число компонент связности) только тогда, когда нет других возможностей.

Оценка сложности алгоритма

Шаг 1. Проверить, является ли граф G эйлеровым: сложность алгоритма равна $O(V)$, то есть линейная относительно количества вершин V в данном графе.

Шаг 2. Используем алгоритм кратчайшей цепи Дейкстры: сложность алгоритма равна $O(|V|^2)$, то есть квадратичная относительно количества вершин V в данном графе.

Шаг 3. Используем венгерский алгоритм решения задачи о назначениях: сложность алгоритма равна $O(|V|^3)$, то есть кубическая относительно количества вершин V в начальном графе, степени которых нечётные.

Шаг 4. Используем алгоритм кратчайшей цепи Дейкстры: сложность алгоритма равна $O(|V|^2)$, то есть квадратичная относительно количества вершин V в данном графе.

Шаг 5. Используем алгоритм Флэри: сложность алгоритма равна $O(E)$, то есть линейная относительно количества рёбер E в данном графе.

Пример работы программы

Для примера возьмём граф, изображённый на рис. 8:

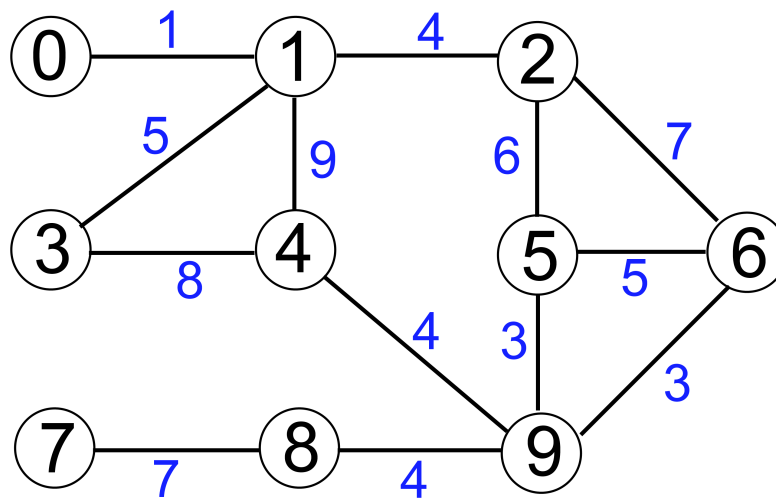


Рис. 8

Его матрица смежности:

0	1	0	0	0	0	0	0	0	0
1	0	2	5	9	0	0	0	0	0
0	2	0	0	0	6	7	0	0	0
0	5	0	0	8	0	0	0	0	0
0	9	0	8	0	0	0	0	0	4
0	0	6	0	0	0	5	0	0	3
0	0	7	0	0	5	0	0	0	3
0	0	0	0	0	0	0	0	7	0
0	0	0	0	0	0	0	7	0	4
0	0	0	0	4	3	3	0	4	0

(2)

Результат работы программы:

Эйлеров Цикл

[0, 1, 4, 9, 8, 7, 8, 9, 6, 5, 9, 4, 3, 1, 2, 6, 5, 2, 1, 0]

Суммарный вес

87