Алгоритм Форда-Беллмана

Введение

граф алгоритм маршрут

При конструировании радиоаппаратуры и других электронных устройств возникает задача об оптимальном размещении компонентов на печатной плате, или отдельных элементов в корпусе устройства. Помощь в ее решении могут оказать различные алгоритмы теории графов, один из них будет изложен в этой работе.

1. Теоретическая часть

.1 Постановка задачи

1) Определить минимальный путь из υ1 в υ7 в нагруженном орграфе D, причем он должен иметь минимальное число дуг среди всех возможных минимальных путей. Орграф D: около каждой дуги указана ее длина.

Составлена (7 ˟ 7) - матрица C(D) длин дуг нагруженного орграфа D.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | υ1 | υ2 | υ3 | υ4 | υ5 | υ6 | υ7 |
| υ1 | ∞ | ∞ | 9 | ∞ | ∞ | 2 | 12 |
| υ2 | 1 | ∞ | ∞ | ∞ | 1 | 2 | 4 |
| υ3 | 2 | 1 | ∞ | ∞ | 1 | ∞ | 2 |
| υ4 | ∞ | 1 | 1 | ∞ | ∞ | 1 | ∞ |
| υ5 | 1 | 2 | ∞ | 2 | ∞ | ∞ | ∞ |
| υ6 | ∞ | ∞ | ∞ | ∞ | 1 | ∞ | 8 |
| υ7 | ∞ | 2 | 1 | ∞ | 1 | 2 | ∞ |

2) Составить матрицу смежности A = (aij) для данного графа.

) Составить матрицу инцидентности B= (bij) для данного графа.

1.2 Основные понятия теории графов

Пусть υ - не пустое множество и Х - некоторый набор пар элементов из V вида (υ, ω), υ, ω € V. Элементы множества V - вершины, множества X - ребра графа.

X = {( υ1, υ2), (υ2, υ3), (υ1, υ4), (υ4, υ4)}

Ребро вида (υ, υ) называется петлей, одинаковые пары во множестве X называются кратными ребрами.

Количество одинаковых пар (υ, ω) называется кратностью ребра υ, ω.

Множество V и набор Х определяют псевдограф G(V, X), т. е. граф с кратными ребрами и петлями.

Псевдограф без петель называется мультиграфом.

Если во множестве X нет кратных ребер, то G(V, X) называется графом.

Если во множестве Х пары являются упорядоченными, то граф называется ориентированным графом или орграфом.

Ребра орграфа называются дугами.

Если граф не ориентированный и х = (υ, ω) - ребро, то вершины υ и ω называются концами ребра х.

Если же граф ориентированный, то υ называется началом дуги х, а ω - концом дуги х.

Вершина υ и ребро х называются инцидентными.

Вершины υ и ω называются смежными, если они имеют общую вершину.

Степенью вершины υ называется число δ(υ) ребер графа инцидентных данной вершине.

Вершина, имеющая степень 0 называется изолированной, 1 - висячей.

Последовательность υi1xi1, υi2xi2, …, υikxik, υik+1, k ≥ 1, υij € V, xij € X, в которой чередуются вершины и ребра (дуги) и ребро xij (υij, υij+1) называется маршрутом, соединяющим вершины υi1 и υik+1.

υi1 называется начальной вершиной. υik+1 называется конечной вершиной. Остальные - внутренние.

Одна и та же вершина может быть одновременно и начальной, и конечной, и внутренней.

Если нет кратных ребер, то маршрут однозначно восстанавливается по последовательности вершин.

Для орграфа маршрут называется путь.

Число ребер или дуг в маршруте (пути) называется его длинной.

Маршрут называется замкнутый, если начальная и конечная вершины совпадают.

Незамкнутый маршрут, в котором все ребра попарно различны, называется цепью.

Цепь, в которой все вершины попарно различны, называется простой.

Замкнутый маршрут, в котором все ребра попарно различны, называется циклом.

Цикл, в котором внутренние вершины попарно различны, называется простым.

Матричные способы задания графов.

Пусть ϧ(υ, x) - граф

V = {υ1, υ2, …, υn}

X = {x1, x2, …, xn}

Матрицей смежости называется квадратная матрица А n - ого порядка, элементы которой определяются формулой:

A = (aij) aij = 1, (υi, υj) € X

0, (υi, υj) не € X

Если дан псевдограф, то aij = 0, (υi, υj) € X k, k - кратность ребра

Матрицей инцидентности называется матрица В размера M x N, элементы которой определяются по формуле:

B = (bij) bij = 0, υi не инцидентна xj

1, υi есть конец дуги xj

-1, υi есть начало дуги xj

Пусть D - ориентированный мультиграф с непустым множеством дуг, тогда:

) сумма строк матрицы B(D) есть нулевая строка

) любая строка B(D) есть линейная комбинация остальных

) ранг матрицы B(D) не превосходит числа вершин - 1 (минус 1)

) для любого контура в D сумма столбцов матрицы B(D) соответствующих дугов входящих в этот контур равна нулевому столбцу.

A(G) - матрица смежности

k(G) = A \* A \* … \* A

Элемент qkij матрицы Ak(G) ориентированного псвдографа равен числу всех путей длины k из вершины υi в υj.

Теорема: для того, чтобы n - вершинный орграф D с матрицей смежности A(D) имел хотя бы один контур необходимо и достаточно, чтобы матрица k = A2 + + A3 + … + An имела не нулевые диагональные элементы.

Объединением графов G1(V1, X1) и G2(V2, X2) называется граф

G = G1 U G2 (V1UV2, X1UX2)

Пересечением графов G1 и G2, где V1 - пересечение V2 ≠ Ø называется граф (V1∩V2 = Ø) G1∩G2 (V1∩V2, X1∩X2)

Подграфом графа G называется граф, все вершины и ребра которого содержатся среди вершин и ребер графа G.

Подграф называется собственным, если он отличен от самого графа.

Вершина ω орграфа D (графа G) называется достижимой из вершины V, если существует путь (маршрут) из V в ω или V = ω.

Граф называется связным, если для любых двух его вершин существует маршрут, соединяющий их.

Компонентой связности графа G называется его связный подграф не являющийся собственным подграфом никакого другого связанного подграфа.

Пусть орграф D имеет множество вершин V = {V1, V2, …, Vn}.

Матрицей достижимости орграфа D называется квадратная матрица T(D) порядка n, у которой tij = 1, υj достижимо из υi

0, υj не достижимо из υi

Матрицей связности S(D) орграфа D называется квадратная матрица порядка n, у которой Sij = 1, υj достижимо из υi, υi достижимо из υj

, в противном случае

Матрицей связности неориентированного графа G называется квадратная матрица S(G) порядка n у которой:

Sij = 1, если i = j или если существует маршрут, соединяющий Vi и Vj

, в противном случае

.3 Выбор алгоритма

Для решения поставленной задачи будем использовать алгоритм Форда - Бэллмана. Назовем орграф D = (V, X) нагруженным, если на множестве дуг X определена некоторая функция l: X → R, которую часто называют весовой функцией. Тем самым в нагруженном орграфе D в каждой дуге x € X поставлено в соответствие некоторое действительное число l(x). Значение l(x) будем называть длинной дуги x. Для этого пути π нагруженного орграфа D обозначим через l(π) сумму длин входящих в π дуг, при этом каждая дуга учитывается столько раз, сколько она входит в путь. Величину l(π) будем называть длинной пути π в нагруженном орграфе D. Ранее так называлось количество дуг в пути π. В связи с этим заметим, что если длины дуг выбраны равными l, то l(π) выражает введенную ранее длину пути π в ненагруженном орграфе. Следовательно, любой ненагруженный орграф можно считать нагруженным с длинами дуг, равными l. Аналогично определяется и нагруженный граф, а также длина маршрута в нем.

Пусть в нагруженном орграфе D из вершины υ в вершину ω, где υ ≠ ω, называется минимальным, если он имеет минимальную длину среди всех путей орграфа D из υ в ω. Аналогично определяется и минимальный маршрут в нагруженном графе G. Если в нагруженном орграфе D имеются замкнутые пути отрицательной длины, то для заданных вершин υ1, ω орграфа D, где υ ≠ ω, минимального пути из υ в ω может и не быть. Действительно, если в D имеется замкнутый путь σ отрицательной длины и существует путь из υ в ω, проходящий хотя бы через одну вершину, содержащуюся в σ, то очевидно, что в D найдется путь π из υ в ω вида π = π1Ο σ Ο π2, где π1, π2 - пути в D (возможно также, что либо π1, либо π2 является пустой последовательностью; при этом предполагаем, что Ø Ο σ = σ Ο Ø = σ). Но тогда π1 Ο σ Ο σ Ο π2, π1 Ο σ Ο σ Ο σ Ο π2 … также пути в D из υ в ω, и длина каждого следующего пути в этой последовательности отличается от длины предыдущего на l(σ) < 0, а значит, длины путей из υ в ω могут принимать сколь угодно малые отрицательные значения. Аналогичная ситуация имеет место в случае, когда в нагруженном графе G вершины υ, ω находятся в одной компоненте связности, содержащей хотя бы одно ребро отрицательной длины. В таких случаях имеют смысл лишь задачи поиска минимальных путей (маршрутов) среди путей (маршрутов), число дуг (ребер) в которых ограничено сверху.

Приведем некоторые свойства минимальных путей (маршрутов) в нагруженном орграфе D = (V, X) (графе G = (V, X)):

) если существует x € X l(x) > 0, то любой минимальный путь (маршрут) является простой цепью;

) если υ1 υ2 … υk - минимальный путь (маршрут), то для любых номеров i, j таких, что 1 ≤ i < j ≤ k, путь (маршрут) υi υi+1 также является минимальным;

) если υ … и ω - минимальный путь (маршрут) среди путей из υ в ω (среди маршрутов, соединяющих υ, ω), содержащих не более k+1 дуг (ребер), то υ … u - минимальный путь (маршрут) среди путей из υ в u (среди маршрутов, соединяющих υ, u), содержащих не более k дуг (ребер).

Рассмотрим теперь задачу поиска минимальных путей (маршрутов) в нагруженном орграфе (графе). При этом для определенности рассуждения будем проводить для орграфа (для графа они аналогичны).

Замечание 1. При решении некоторых практических задач возникает необходимость поиска максимальных путей в нагруженном орграфе. Такая задача легко сводится к исследуемой ниже задаче поиска минимальных путей заменой знаков при длинах дуг на противоположные.

Пусть D = (V, X) - нагруженный орграф, V = { υ1, …, υn}, n > 2. Введем величины λi(k) , где i = 1, …, n, k = 1, 2, … . Для каждых фиксированных i и k величина λi(k) равна длине минимального пути среди путей из υ1 в υi, содержащих не более k дуг; если же таких путей нет, то λi(k) = ∞. Кроме того, если произвольную вершину υ € V считать путем из υ в υ нулевой длины, то величины λi(k) можно ввести также и для k = 0, при этом

λ1(0) = 0, λi(0) = ∞, i = 2, …, n. (1)

Введем также в рассмотрение квадратную матрицу C(D) = [сij] порядка n с элементами

cij = l(υi, υj), если (υi, υj) € X;

∞, если (υi, υj) € Х,

которую будем называть матрицей длин дуг нагруженного орграфа D.

Следующее утверждение дает простые формулы для вычисления величин λi(k).

Утверждение 1. При i - 2, …, n, k ≥ 0 выполняется равенство

λi(k+1) = min {λj(k)+cij} (2)

1 ≤ j ≤ n

а при i - 1, k ≥ 0 справедливо равенство

λi(k+1) = min{0; min {λj(k)=cj1}} (3)

1 ≤ j ≤ n

Используя утверждение 1, нетрудно описать алгоритм нахождения значений величин λi(k) (будем записывать в виде матрицы, где i - номер строки, k+1 - номер столбца). Действительно, используя рекуррентные соотношения (2), (3) и исходя из (1), последовательно определяем набор величин λ1(k), …, λn(k) ((k+1) столбец матрицы), начиная с k = 0, а затем шаг за шагом увеличивая значение k до любой необходимой величины.

Будем теперь предполагать, что в D отсутствуют простые контуры отрицательной длины (ниже в утверждении 5 приводится простой метод проверки этого условия). Для дальнейших рассуждений нам понадобятся дополнительные утверждения.

Утверждение 2. Из всякого замкнутого пути отрицательной длины можно выделить простой контур отрицательной длины.

Утверждение 3. Если в нагруженном орграфе отсутствуют простые контуры отрицательной длины, то в нем нет замкнутых путей отрицательной длины.

Утверждение 4. В нагруженном орграфе, в котором отсутствуют простые контуры отрицательной длины, можно из всякого незамкнутого пути выделить простую цепь с теми же начальной и конечной вершинами, длина которой не превышает длины исходного пути.

Замечание 2. При определении нагруженного графа мы предполагали, что величины l(x), x € X, конечны. Между тем величины λi(k) (i = 1, 2, …, n, k = 0, 1, …, n - 1) можно аналогичным образом определить и для случая, когда для некоторых x € X выполняется l(x) = ∞. При этом сохраняет силу утверждение 1, а следовательно, и алгоритм построения величин λi(k). Отметим, однако, что теперь могут существовать вершины, достижимые из υ1 с длинами минимальных путей из υ1 в эти вершины, равными ∞, т. е. мы уже не можем судить по λi(n-1) о достижимости вершин υi из υ1.

Замечание 3. Если при некотором k0, где 0 ≤ k0 ≤ n - 2, выполняются равенства λi(k0) = λi(k0 + 1), i = 1, 2, …, n, и, в частности , λi(n - 1) = λi(k0), i = 1, 2, …, n, т. е. в данном случае значения λi(k) при k > k0 не несут никакой дополнительной информации, и тогда имеет смысл оборвать процесс последовательного определения наборов величин λ1(k), …, λn(k) на значении k = k0.

Приведем теперь утверждение, дающее нам легко проверяемое необходимое и достаточное условие того, что в D отсутствуют простые контуры отрицательной длины. При этом будем рассматривать случай, когда из вершины υ1 достижимы все другие вершины орграфа D. Этого всегда можно добиться удалением вершин, не достижимых из υ1.

Утверждение 5. Пусть в орграфе D = (V, X), где V = { υ1, …, υn}, любая вершина υi, i € {2, …, n}, достижима из υ1. Тогда, для того чтобы в D отсутствовали простые контуры отрицательной длины, необходимо и достаточно, чтобы выполнялось условие λi(n -1) = λi(n), i = 1, …, n.

Алгоритм (алгоритм Форда - Беллмана) нахождения минимального пути в нагруженном орграфе D из υ1 в υi1, (i ≠ 1):

Шаг 1. Пусть мы уже составили величин λi(k), i = 1, 2, … , n, k = 0, 1, 2, …, n - 1. Если λi(n-1) = ∞, то вершина υi1 не достижима изυ1 (предполагаем, что все величины l(x), x € X конечны). В этом случае работа алгоритма заканчивается.

Шаг 2. Пусть λi1(n-1) < ∞, тогда число λi1(n-1) выражает длину любого минимального пути из υ1 в υi1 в нагруженном орграфе *D.* Определим минимальное число k ≥ 1,при котором выполняется равенство λi1(k1) =λi1(n-1). По определению чисел λi(k) получаем, чтоk1 - минимальное число дуг в пути среди всех минимальных путей υ1 в υi1 в нагруженном орграфе *D.*

Шаг 3. Последовательно определяем номера i2, …, ik1+1 такие, что

λi2(k1-1) + сi2,i1 = λi1(k1)

λi3(k1-2) + сi3,i2 = λi1(k1 - 1)

. . . . . . . . . . . . . . . . (4)

λi(0) + ci j = λi(1)+1 k1+1 k1 k1

(эти номера найдутся в силу (2)).

Из (4) с учетом того, что λi1(k1)=λi1(n - 1) < ∞, имеем ci2, i1<∞,…,ci, i < ∞, λi(0) < ∞,

k1+1 k1 k1+1

откуда, используя (1), получаем

(υi2, υi1), …, (υi, υi) € X, l(υi2, υi1) = ci2, i1, …, l(υi2, υi1) = ci,I;

k1+1 k1 k1+1 k1 k1+1 k1

λi(0) = 0, i = 1, υi = υ1  (5)

k1+1 k1+1 k1+1

Складывая равенства (4) и учитывая (5), имеем

l(υ1k1 υi … υi2 υi1) = λi1(k1), (6)

т. е. υ1 υi … υi2 υi1 - исходный минимальный путь из υ1 в υi1 в нагруженном орграфе D. Заметим, что в этом пути ровно ki дуг. Следовательно мы определили путь с минимальным числом дуг среди всех минимальных путей из υ1 в υi1 в нагруженном орграфе D.

Замечание 4. Номера i2, i3, …, ik1, удовлетворяющие (6), вообще говоря, могут быть выделены неоднозначно. Эта неоднозначность соответствует случаям, когда существует несколько различных путей изυ1 в υi1 c минимальным числом дуг среди минимальных путей изυ1 в υi1 в нагруженном орграфе *D.*

Замечание 5. Алгоритм можно модифицировать, с тем υ1 в υi1, содержащих не более k0 дуг, где k0 - заданное число, k0 > 1. Xтобы определить минимальный путь из υ1 в заданную вершину υi1 среди путей из υ1 в υi1. Для этого в алгоритме вместо λi(n - 1) следует воспользоваться λi1(k0). Отметим, что при использовании указанной модификации алгоритма предположение об отсутствии простых контуров отрицательной длины излишне. При том, если в орграфе *D* имеются простые контуры отрицательной длины, может выполняться неравенство λ1(k0) < 0.

Замечание 6. Алгоритм и его модификация, описанная выше, после соответствующего изменения в терминологии и обозначениях применимы и к неориентированному графу *G.* При этом условие отсутствия простых контуров отрицательной длины заменяется условием отсутствия ребер отрицательной длины.

Замечание 7. Нетрудно доказать, что утверждение 1, а следовательно, и алгоритм вместе с его модификацией останутся справедливыми и для нагруженного ориентированного псевдографа. При этом в случае кратных дуг следует учитывать лишь дуги минимальной длины, а при наличии петель - лишь, петли отрицательной длины. Это замечание переносится и на неориентированные псевдографы.

2. Практическая часть

граф алгоритм маршрут

2.1 Поиск минимального пути

Будем последовательно определять ее элементы, используя рекуррентное соотношение (2) и исходя из (1).

λ1(0) = 0, λi(0) = ∞, i = 2, …, n.

Используя (2) определим элементы λi(k), i = 2, 3, 4, 5, 6, 7; k = 1, 2, 3, 4, 5, 6.

λ2(1) = min{λj(0) + cj2}, 1 ≤ j ≤ 7

при j = 1 λ1(0) + с12 = 0 + ∞ = ∞

при j = 2 λ2(0) + с22 = ∞ + ∞ = ∞

при j = 3 λ3(0) + с32 = ∞ + 1 = ∞

при j = 4 λ4(0) + с42 = ∞ + 1 = ∞

при j = 5 λ5(0) + с52 = ∞ + 2 = ∞

при j = 6 λ6(0) + с62 = ∞ + ∞ = ∞

при j = 7 λ7(0) + с72 = ∞ + 2 = ∞

Следовательно λ2(1) = ∞

λ3(1) = min{λj(0) + cj3}, 1 ≤ j ≤ 7

при j = 1 λ1(0) + с13 = 0 + 9 = 9

при j = 2 λ2(0) + с23 = ∞ + ∞ = ∞

при j = 3 λ3(0) + с33 = ∞ + ∞ = ∞

при j = 4 λ4(0) + с43 = ∞ + 1 = ∞

при j = 5 λ5(0) + с53 = ∞ + ∞ = ∞

при j = 6 λ6(0) + с63 = ∞ + ∞ = ∞

при j = 7 λ7(0) + с73 = ∞ + 1 = ∞

Следовательно λ2(1) = 9

λ4(1) = min{λj(0) + cj4}, 1 ≤ j ≤ 7

при j = 1 λ1(0) + с14 = 0 + ∞ = ∞

при j = 2 λ2(0) + с24 = ∞ + ∞ = ∞

при j = 3 λ3(0) + с34 = ∞ + ∞ = ∞

при j = 4 λ4(0) + с44 = ∞ + ∞ = ∞

при j = 5 λ5(0) + с54 = ∞ + 2 = ∞

при j = 6 λ6(0) + с64 = ∞ + ∞ = ∞

при j = 7 λ7(0) + с74 = ∞ + ∞ = ∞

Следовательно λ4(1) = ∞

λ5(1) = min{λj(0) + cj5}, 1 ≤ j ≤ 7

при j = 1 λ1(0) + с15 = 0 + ∞ = ∞

при j = 2 λ2(0) + с25 = ∞ + 1 = ∞

при j = 3 λ3(0) + с35 = ∞ + 1 = ∞

при j = 4 λ4(0) + с45 = ∞ + ∞ = ∞

при j = 5 λ5(0) + с55 = ∞ + ∞ = ∞

при j = 6 λ6(0) + с65 = ∞ + ∞ = ∞

при j = 7 λ7(0) + с75 = ∞ + 1 = ∞

Следовательно λ5(1) = ∞

λ6(1) = min{λj(0) + cj6}, 1 ≤ j ≤ 7

при j = 1 λ1(0) + с16 = 0 + 2 = 2

при j = 2 λ2(0) + с26 = ∞ + 2 = ∞

при j = 3 λ3(0) + с36 = ∞ + ∞ = ∞

при j = 4 λ4(0) + с46 = ∞ + 1 = ∞

при j = 5 λ5(0) + с56 = ∞ + ∞ = ∞

при j = 6 λ6(0) + с66 = ∞ + ∞ = ∞

при j = 7 λ7(0) + с76 = ∞ + 2 = ∞

Следовательно λ6(1) = 2

λ7(1) = min{λj(0) + cj7}, 1 ≤ j ≤ 7

при j = 1 λ1(0) + с17 = 0 + 12 = 12

при j = 2 λ2(0) + с27 = ∞ + 4 = ∞

при j = 3 λ3(0) + с37 = ∞ + 2 = ∞

при j = 4 λ4(0) + с47 = ∞ + ∞ = ∞

при j = 5 λ5(0) + с57 = ∞ + ∞ = ∞

при j = 6 λ6(0) + с67 = ∞ + 8 = ∞

при j = 7 λ7(0) + с77 = ∞ + ∞ = ∞

Следовательно λ7(1) = 12

λ2(2) = min{λj(1) + cj2}, 1 ≤ j ≤ 7

при j = 1 λ1(1) + с12 = 0 + ∞ = ∞

при j = 2 λ2(1) + с22 = ∞ + ∞ = ∞

при j = 3 λ3(1) + с32 = 9 + 1 = 10

при j = 4 λ4(1) + с42 = ∞ + 1 = ∞

при j = 5 λ5(1) + с52 = ∞ + 2 = ∞

при j = 6 λ6(1) + с62 = 2 + ∞ = ∞

при j = 7 λ7(1) + с72 = 12 + 2 = 14

Следовательно λ2(2) = 10

λ2(2) = min{λj(1) + cj2}, 1 ≤ j ≤ 7

при j = 1 λ1(1) + с12 = 0 + ∞ = ∞

при j = 2 λ2(1) + с22 = ∞ + ∞ = ∞

при j = 3 λ3(1) + с32 = 9 + 1 = 10

при j = 4 λ4(1) + с42 = ∞ + 1 = ∞

при j = 5 λ5(1) + с52 = ∞ + 2 = ∞

при j = 6 λ6(1) + с62 = 2 + ∞ = ∞

при j = 7 λ7(1) + с72 = 12 + 2 = 14

Следовательно λ2(2) = 10

λ3(2) = min{λj(1) + cj3}, 1 ≤ j ≤ 7

при j = 1 λ1(1) + с13 = 0 + 9 = 9

при j = 2 λ2(1) + с23 = ∞ + ∞ = ∞

при j = 3 λ3(1) + с33 = 9 + ∞ = ∞

при j = 4 λ4(1) + с43 = ∞ + 1 = ∞

при j = 5 λ5(1) + с53 = ∞ + ∞ = ∞

при j = 6 λ6(1) + с63 = 2 + ∞ = ∞

при j = 7 λ7(1) + с73 = 12 + 1 = 13

Следовательно λ3(2) = 9

λ4(2) = min{λj(1) + cj4}, 1 ≤ j ≤ 7

при j = 1 λ1(1) + с14 = 0 + ∞ = ∞

при j = 2 λ2(1) + с24 = ∞ + ∞ = ∞

при j = 3 λ3(1) + с34 = 9 + ∞ = ∞

при j = 4 λ4(1) + с44 = ∞ + ∞ = ∞

при j = 5 λ5(1) + с54 = ∞ + 2 = ∞

при j = 6 λ6(1) + с64 = 2 + ∞ = ∞

при j = 7 λ7(1) + с74 = 12 + ∞ = ∞

Следовательно λ4(2) = ∞

λ5(2) = min{λj(1) + cj5}, 1 ≤ j ≤ 7

при j = 1 λ1(1) + с15 = 0 + ∞ = ∞

при j = 2 λ2(1) + с25 = ∞ + 1 = ∞

при j = 3 λ3(1) + с35 = 9 + 1 = 10

при j = 4 λ4(1) + с45 = ∞ + ∞ = ∞

при j = 5 λ5(1) + с55 = ∞ + ∞ = ∞

при j = 6 λ6(1) + с65 = 2 + 1 = 3

при j = 7 λ7(1) + с75 = 12 + 1 = 13

Следовательно λ5(2) = 3

λ6(2) = min{λj(1) + cj6}, 1 ≤ j ≤ 7

при j = 1 λ1(1) + с16 = 0 + 2 = 2

при j = 2 λ2(1) + с26 = ∞ + 2 = ∞

при j = 3 λ3(1) + с36 = 9 + ∞ = ∞

при j = 4 λ4(1) + с46 = ∞ + 1 = ∞

при j = 5 λ5(1) + с56 = ∞ + ∞ = ∞

при j = 6 λ6(1) + с66 = 2 + ∞ = ∞

при j = 7 λ7(1) + с76 = 12 + 2 = 14

Следовательно λ6(2) = 2

λ7(2) = min{λj(1) + cj7}, 1 ≤ j ≤ 7

при j = 1 λ1(1) + с17 = 0 + 12 = 12

при j = 2 λ2(1) + с27 = ∞ + 4 = ∞

при j = 3 λ3(1) + с37 = 9 + 2 = 11

при j = 4 λ4(1) + с47 = ∞ + ∞ = ∞

при j = 5 λ5(1) + с57 = ∞ + ∞ = ∞

при j = 6 λ6(1) + с67 = 2 + 8 = 10

при j = 7 λ7(1) + с77 = 12 + ∞ = ∞

Следовательно λ7(2) = 10

λ2(3) = min{λj(2) + cj2}, 1 ≤ j ≤ 7

при j = 1 λ1(2) + с12 = 0 + ∞ = ∞

при j = 2 λ2(2) + с22 = 10 + ∞ = ∞

при j = 3 λ3(2) + с32 = 9 + 1 = 10

при j = 4 λ4(2) + с42 = ∞ + 1 = ∞

при j = 5 λ5(2) + с52 = 3 + 2 = 5

при j = 6 λ6(2) + с62 = 2 + ∞ = ∞

при j = 7 λ7(2) + с72 = 10 + 2 = 12

Следовательно λ2(3) = 5

λ3(3) = min{λj(2) + cj3}, 1 ≤ j ≤ 7

при j = 1 λ1(2) + с13 = 0 + 9 = 9

при j = 2 λ2(2) + с23 = 10 + ∞ = ∞

при j = 3 λ3(2) + с33 = 9 + ∞ = ∞

при j = 4 λ4(2) + с43 = ∞ + 1 = ∞

при j = 5 λ5(2) + с53 = 3 + ∞ = ∞

при j = 6 λ6(2) + с63 = 2 + ∞ = ∞

при j = 7 λ7(2) + с73 = 10 + 1 = 11

Следовательно λ3(3) = 9

λ4(3) = min{λj(2) + cj4}, 1 ≤ j ≤ 7

при j = 1 λ1(2) + с14 = 0 + ∞ = ∞

при j = 2 λ2(2) + с24 = 10 + ∞ = ∞

при j = 3 λ3(2) + с34 = 9 + ∞ = ∞

при j = 4 λ4(2) + с44 = ∞ + ∞ = ∞

при j = 5 λ5(2) + с54 = 3 + 2 = 5

при j = 6 λ6(2) + с64 = 2 + ∞ = ∞

при j = 7 λ7(2) + с74 = 10 + ∞ = ∞

Следовательно λ4(3) = 5

λ5(3) = min{λj(2) + cj5}, 1 ≤ j ≤ 7

при j = 1 λ1(2) + с15 = 0 + ∞ = ∞

при j = 2 λ2(2) + с25 = 10 + 1 = 11

при j = 3 λ3(2) + с35 = 9 + 1 = 10

при j = 4 λ4(2) + с45 = ∞ + ∞ = ∞

при j = 5 λ5(2) + с55 = 3 + ∞ = ∞

при j = 6 λ6(2) + с65 = 2 + 1 = 3

при j = 7 λ7(2) + с75 = 10 + 1 = 11

Следовательно λ5(3) = 3

λ6(3) = min{λj(2) + cj6}, 1 ≤ j ≤ 7

при j = 1 λ1(2) + с16 = 0 + 2 = 2

при j = 2 λ2(2) + с26 = 10 + 2 = 12

при j = 3 λ3(2) + с36 = 9 + ∞ = ∞

при j = 4 λ4(2) + с46 = ∞ + 1 = ∞

при j = 5 λ5(2) + с56 = 3 + ∞ = ∞

при j = 6 λ6(2) + с66 = 2 + ∞ = ∞

при j = 7 λ7(2) + с76 = 10 + 2 = 12

Следовательно λ6(3) = 2

λ7(3) = min{λj(2) + cj7}, 1 ≤ j ≤ 7

при j = 1 λ1(2) + с17 = 0 + 12 = 12

при j = 2 λ2(2) + с27 = 10 + 4 = 14

при j = 3 λ3(2) + с37 = 9 + 2 = 11

при j = 4 λ4(2) + с47 = ∞ + ∞ = ∞

при j = 5 λ5(2) + с57 = 3 + ∞ = ∞

при j = 6 λ6(2) + с67 = 2 + 8 = 10

при j = 7 λ7(2) + с77 = 10 + ∞ = ∞

Следовательно λ7(3) = 10

λ2(4) = min{λj(3) + cj2}, 1 ≤ j ≤ 7

при j = 1 λ1(3) + с12 = 0 + ∞ = ∞

при j = 2 λ2(3) + с22 = 5 + ∞ = ∞

при j = 3 λ3(3) + с32 = 9 + 1 = 10

при j = 4 λ4(3) + с42 = 5 + 1 = 6

при j = 5 λ5(3) + с52 = 3 + 2 = 5

при j = 6 λ6(3) + с62 = 2 + ∞ = ∞

при j = 7 λ7(3) + с72 = 10 + 2 = 12

Следовательно λ2(4) = 5

λ3(4) = min{λj(3) + cj3}, 1 ≤ j ≤ 7

при j = 1 λ1(3) + с13 = 0 + 9 = 9

при j = 2 λ2(3) + с23 = 5 + ∞ = ∞

при j = 3 λ3(3) + с33 = 9 + ∞ = ∞

при j = 4 λ4(3) + с43 = 5 + 1 = 6

при j = 5 λ5(3) + с53 = 3 + ∞ = ∞

при j = 6 λ6(3) + с63 = 2 + ∞ = ∞

при j = 7 λ7(3) + с73 = 10 + 1 = 11

Следовательно λ3(4) = 6

λ4(4) = min{λj(3) + cj4}, 1 ≤ j ≤ 7

при j = 1 λ1(3) + с14 = 0 + ∞ = ∞

при j = 2 λ2(3) + с24 = 5 + ∞ = ∞

при j = 3 λ3(3) + с34 = 9 + ∞ = ∞

при j = 4 λ4(3) + с44 = 5 + ∞ = ∞

при j = 5 λ5(3) + с54 = 3 + 2 = 5

при j = 6 λ6(3) + с64 = 2 + ∞ = ∞

при j = 7 λ7(3) + с74 = 10 + ∞ = ∞

Следовательно λ4(4) = 5

λ5(4) = min{λj(3) + cj5}, 1 ≤ j ≤ 7

при j = 1 λ1(3) + с15 = 0 + ∞ = ∞

при j = 2 λ2(3) + с25 = 5 + 1 = 6

при j = 3 λ3(3) + с35 = 9 + 1 = 10

при j = 4 λ4(3) + с45 = 5 + ∞ = ∞

при j = 5 λ5(3) + с55 = 3 + ∞ = ∞

при j = 6 λ6(3) + с65 = 2 + 1 = 3

при j = 7 λ7(3) + с75 = 10 + 1 = 11

Следовательно λ5(4) = 3

λ6(4) = min{λj(3) + cj6}, 1 ≤ j ≤ 7

при j = 1 λ1(3) + с16 = 0 + 2 = 2

при j = 2 λ2(3) + с26 = 5 + 2 = 7

при j = 3 λ3(3) + с36 = 9 + ∞ = ∞

при j = 4 λ4(3) + с46 = 5 + 1 = 6

при j = 5 λ5(3) + с56 = 3 + ∞ = ∞

при j = 6 λ6(3) + с66 = 2 + ∞ = ∞

при j = 7 λ7(3) + с76 = 10 + 2 = 12

Следовательно λ6(4) = 2

λ7(4) = min{λj(3) + cj7}, 1 ≤ j ≤ 7

при j = 1 λ1(3) + с17 = 0 + 12 = 12

при j = 2 λ2(3) + с27 = 5 + 4 = 9

при j = 3 λ3(3) + с37 = 9 + 2 = 11

при j = 4 λ4(3) + с47 = 5 + ∞ = ∞

при j = 5 λ5(3) + с57 = 3 + ∞ = ∞

при j = 6 λ6(3) + с67 = 2 + 8 = 10

при j = 7 λ7(3) + с77 = 10 + ∞ = ∞

Следовательно λ7(4) = 9

λ2(5) = min{λj(4) + cj2}, 1 ≤ j ≤ 7

при j = 1 λ1(4) + с12 = 0 + ∞ = ∞

при j = 2 λ2(4) + с22 = 5 + ∞ = ∞

при j = 3 λ3(4) + с32 = 6 + 1 = 7

при j = 4 λ4(4) + с42 = 5 + 1 = 6

при j = 5 λ5(4) + с52 = 3 + 2 = 5

при j = 6 λ6(4) + с62 = 2 + ∞ = ∞

при j = 7 λ7(4) + с72 = 9 + 2 = 11

Следовательно λ2(5) = 5

λ3(5) = min{λj(4) + cj3}, 1 ≤ j ≤ 7

при j = 1 λ1(4) + с13 = 0 + 9 = 9

при j = 2 λ2(4) + с23 = 5 + ∞ = ∞

при j = 3 λ3(4) + с33 = 6 + ∞ = ∞

при j = 4 λ4(4) + с43 = 5 + 1 = 6

при j = 5 λ5(4) + с53 = 3 + ∞ = ∞

при j = 6 λ6(4) + с63 = 2 + ∞ = ∞

при j = 7 λ7(4) + с73 = 9 + 1 = 10

Следовательно λ3(5) = 6

λ4(5) = min{λj(4) + cj4}, 1 ≤ j ≤ 7

при j = 1 λ1(4) + с14 = 0 + ∞ = ∞

при j = 2 λ2(4) + с24 = 5 + ∞ = ∞

при j = 3 λ3(4) + с34 = 6 + ∞ = ∞

при j = 4 λ4(4) + с44 = 5 + ∞ = ∞

при j = 5 λ5(4) + с54 = 3 + 2 = 5

при j = 6 λ6(4) + с64 = 2 + ∞ = ∞

при j = 7 λ7(4) + с74 = 9 + ∞ = ∞

Следовательно λ4(5) = 5

λ5(5) = min{λj(4) + cj5}, 1 ≤ j ≤ 7

при j = 1 λ1(4) + с15 = 0 + ∞ = ∞

при j = 2 λ2(4) + с25 = 5 + 1 = 6

при j = 3 λ3(4) + с35 = 6 + 1 = 7

при j = 4 λ4(4) + с45 = 5 + ∞ = ∞

при j = 5 λ5(4) + с55 = 3 + ∞ = ∞

при j = 6 λ6(4) + с65 = 2 + 1 = 3

при j = 7 λ7(4) + с75 = 9 + 1 = 10

Следовательно λ5(5) = 3

λ6(5) = min{λj(4) + cj6}, 1 ≤ j ≤ 7

при j = 1 λ1(4) + с16 = 0 + 2 = 2

при j = 2 λ2(4) + с26 = 5 + 2 = 7

при j = 3 λ3(4) + с36 = 6 + ∞ = ∞

при j = 4 λ4(4) + с46 = 5 + 1 = 6

при j = 5 λ5(4) + с56 = 3 + ∞ = ∞

при j = 6 λ6(4) + с66 = 2 + ∞ = ∞

при j = 7 λ7(4) + с76 = 9 + 2 = 11

Следовательно λ6(5) = 2

λ7(5) = min{λj(4) + cj7}, 1 ≤ j ≤ 7

при j = 1 λ1(4) + с17 = 0 + 12 = 12

при j = 2 λ2(4) + с27 = 5 + 4 = 9

при j = 3 λ3(4) + с37 = 6 + 2 = 8

при j = 4 λ4(4) + с47 = 5 + ∞ = ∞

при j = 5 λ5(4) + с57 = 3 + ∞ = ∞

при j = 6 λ6(4) + с67 = 2 + 8 = 10

при j = 7 λ7(4) + с77 = 9 + ∞ = ∞

Следовательно λ7(5) = 8

λ2(6) = min{λj(5) + cj2}, 1 ≤ j ≤ 7

при j = 1 λ1(5) + с12 = 0 + ∞ = ∞

при j = 2 λ2(5) + с22 = 5 + ∞ = ∞

при j = 3 λ3(5) + с32 = 6 + 1 = 7

при j = 4 λ4(5) + с42 = 5 + 1 = 6

при j = 5 λ5(5) + с52 = 3 + 2 = 5

при j = 6 λ6(5) + с62 = 2 + ∞ = ∞

при j = 7 λ7(5) + с72 = 9 + 2 = 11

Следовательно λ2(6) = 5

λ3(6) = min{λj(5) + cj3}, 1 ≤ j ≤ 7

при j = 1 λ1(5) + с13 = 0 + 9 = 9

при j = 2 λ2(5) + с23 = 5 + ∞ = ∞

при j = 3 λ3(5) + с33 = 6 + ∞ = ∞

при j = 4 λ4(5) + с43 = 5 + 1 = 6

при j = 5 λ5(5) + с53 = 3 + ∞ = ∞

при j = 6 λ6(5) + с63 = 2 + ∞ = ∞

при j = 7 λ7(5) + с73 = 9 + 1 = 10

Следовательно λ3(6) = 6

λ4(6) = min{λj(5) + cj4}, 1 ≤ j ≤ 7

при j = 1 λ1(5) + с14 = 0 + ∞ = ∞

при j = 2 λ2(5) + с24 = 5 + ∞ = ∞

при j = 3 λ3(5) + с34 = 6 + ∞ = ∞

при j = 4 λ4(5) + с44 = 5 + ∞ = ∞

при j = 5 λ5(5) + с54 = 3 + 2 = 5

при j = 6 λ6(5) + с64 = 2 + ∞ = ∞

при j = 7 λ7(5) + с74 = 9 + ∞ = ∞

Следовательно λ4(6) = 5

λ5(6) = min{λj(5) + cj5}, 1 ≤ j ≤ 7

при j = 1 λ1(5) + с15 = 0 + ∞ = ∞

при j = 2 λ2(5) + с25 = 5 + 1 = 6

при j = 3 λ3(5) + с35 = 6 + 1 = 7

при j = 4 λ4(5) + с45 = 5 + ∞ = ∞

при j = 5 λ5(5) + с55 = 3 + ∞ = ∞

при j = 6 λ6(5) + с65 = 2 + 1 = 3

при j = 7 λ7(5) + с75 = 9 + 1 = 10

Следовательно λ2(6) = 3

λ6(6) = min{λj(5) + cj6}, 1 ≤ j ≤ 7

при j = 1 λ1(5) + с16 = 0 + 2 = 2

при j = 2 λ2(5) + с26 = 5 + 2 = 7

при j = 3 λ3(5) + с36 = 6 + ∞ = ∞

при j = 4 λ4(5) + с46 = 5 + 1 = 6

при j = 5 λ5(5) + с56 = 3 + ∞ = ∞

при j = 6 λ6(5) + с66 = 2 + ∞ = ∞

при j = 7 λ7(5) + с76 = 9 + 2 = 11

Следовательно λ6(6) = 2

λ7(6) = min{λj(5) + cj7}, 1 ≤ j ≤ 7

при j = 1 λ1(5) + с17 = 0 + 12 = 12

при j = 2 λ2(5) + с27 = 5 + 4 = 9

при j = 3 λ3(5) + с37 = 6 + 2 = 8

при j = 4 λ4(5) + с47 = 5 + ∞ = ∞

при j = 5 λ5(5) + с57 = 3 + ∞ = ∞

при j = 6 λ6(5) + с67 = 2 + 8 = 10

при j = 7 λ7(5) + с77 = 9 + ∞ = ∞

Следовательно λ7(6) = 8

Заполним получившимися значениями

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | λi(0) | λi(1) | λi(2) | λi(3) | λi(4) | λi(5) | λi(6) |
| i1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| i2 | ∞ | ∞ | 10 | 5 | 5 | 5 | 5 |
| i3 | ∞ | 9 | 9 | 9 | 6 | 6 | 6 |
| i4 | ∞ | ∞ | ∞ | 5 | 5 | 5 | 5 |
| i5 | ∞ | ∞ | 3 | 3 | 3 | 3 | 3 |
| i6 | ∞ | 2 | 2 | 2 | 2 | 2 | 2 |
| i7 | ∞ | 12 | 10 | 10 | 9 | 8 | 8 |

Величина λ7(6) = 8 выражает длину минимального пути из υ1 в υ7 в нагруженном орграфе D. Найдем минимальное число k ≥ 1, при котором выполняется равенство λ7(k+1) = λ7(6). Получаем, что k = 5. Таким образом, минимальное число дуг в пути среди всех минимальных путей из υ1 в υ7 в нагруженном орграфе D равняется 5. Определим теперь последовательность номеров i1, i2, i3, i4, i5, i6, где i1 = 7, удовлетворяющих (4), для этого используем формулу (2). Получаем, что в качестве такой последовательности гадо взять номера 7, 3, 4, 5, 6, 1, так как

λ3(4) +c37 = 6 + 2 = λ7(5)

λ4(3) +c43 = 5 + 1 = λ3(4)

λ5(2) +c54 = 3 + 2 = λ4(3)

λ6(1) +c65 = 2 + 1 = λ5(2)

λ1(0) +c16 = 0 + 2 = λ6(1)

Тогда υ1 υ6 υ5 υ4 υ3 υ7 - искомый минимальный путь из υ1 в υ7 в нагруженном орграфе D, причем он содержит минимальное число дуг среди всех возможных минимальных путей из υ1 в υ7.

.2 Построение матрицы смежности

Построим матрицу смежности для данного орграфа.

.3 Построение матрицы инцидентности

Произвольно обозначим ребра орграфа Х1, Х2, …, Х23.

Построим матрицу инцидентности для данного орграфа.

Заключение

Приведенный в данной работе алгоритм позволяет через λi(k), i = 1, 2, …, n, k = 0, 1, …, n - 1, определить минимальный путь в нагруженном орграфе D из υ1 в любую достижимую вершину, причем из всех возможных путей он выделяет путь с наименьшим числом дуг.

Список литературы

. Нефедов В.Н. Курс дискретной математики / В.Н. Нефедов, В.А. Осипова. Изд - во МАИ, 2012. 262 с.

. Емеличев В.А. Лекции по теории графов. / В.А. Емеличев, О.И. Мельников. М.: Наука, Гл. ред. физ. - мат. лит., 1990. 384 с.

. Свами М. Графы, сети, алгоритмы / М. Свами, К. Тхуласираман. М.: Мир, 2014. 455 с.

Приложение

Показан минимальный путь из V1 в V7 в нагруженном орграфе D, причем он содержит минимальное число дуг среди всех возможных минимальных путей из V1 в V7.