

Численные методы

Лекция 3

Е.А. Яревский

18 ноября 2020

Метод Холецкого

Рассматривается положительно определенная матрица.

Такая матрица, очевидно, невырождена.

Если A – положительно определенная матрица, то все ее главные подматрицы положительно определенные.

В частности, все диагональные элементы положительны.

Для симметричной положительно определенной системы, разложение $A = LDL^T$ существует и устойчиво к вычислениям.

Однако, в этом случае используется другое разложение – разложение Холецкого (Холесского).

Теорема:

Для симметричной положительно определенной матрицы A существует единственная нижняя треугольная матрица $G(n \times n)$ с положительными диагональными элементами, такая, что $A = GG^T$.

Метод Холецкого

Разложение Холецкого можно получить из LDL^T разложения.

L – нижняя унитреугольная, $D = \text{diag}(d_1, \dots, d_n)$ – диагональная с положительными элементами.

Матрица $G = L \text{diag}(\sqrt{d_1}, \dots, \sqrt{d_n})$ является вещественной нижней треугольной с положительными диагональными элементами.

$$A = GG^T = L(\text{diag}(\sqrt{d_1}, \dots, \sqrt{d_n}))^2 L^T = LDL^T.$$

Единственность следует из единственности LDL^T разложения.

Метод Холецкого

Разложение Холецкого можно получить из LDL^T разложения.

L – нижняя унитреугольная, $D = \text{diag}(d_1, \dots, d_n)$ – диагональная с положительными элементами.

Матрица $G = L \text{diag}(\sqrt{d_1}, \dots, \sqrt{d_n})$ является вещественной нижней треугольной с положительными диагональными элементами.

$$A = GG^T = L(\text{diag}(\sqrt{d_1}, \dots, \sqrt{d_n}))^2 L^T = LDL^T.$$

Единственность следует из единственности LDL^T разложения.

СЛАУ решается двумя треугольными системами.

Такой метод конструктивен (позволяет вычислить разложение), однако более эффективные методы могут быть получены непосредственно из представления $A = GG^T$.

Метод Холецкого

Сравним j -е столбцы в разложении $A = GG^T$:

$$A(:, j) = \sum_{k=1}^j G(j, k)G(:, k).$$

Получаем

$$G(j, j)G(:, j) = A(:, j) - \sum_{k=1}^{j-1} G(j, k)G(:, k) \equiv v. \quad (1)$$

Если знаем $(j - 1)$ первых столбцов G , то можем вычислить вектор v .
Сравнивая компоненты в (1), находим $G(j : n, j) = v(j : n) / \sqrt{v(j)}$.

Метод Холецкого

```
for  $j = 1:n$   
     $v(j:n) = A(j:n, j)$   
    for  $k = 1:j-1$   
         $v(j:n) = v(j:n) - G(j, k) G(j:n, k)$   
    end  
     $G(j:n, j) = v(j:n)/\sqrt{v(j)}$   
end
```

Замещение старой матрицы новой:

```
for  $j = 1:n$   
    if  $j > 1$   
         $A(j:n, j) = A(j:n, j) - A(j:n, 1:j-1) A(j, 1:j-1)^T$   
    end  
     $A(j:n, j) = A(j:n, j)/\sqrt{A(j, j)}$   
end
```

Прямые методы для разреженных матриц. Хранение

Для некоторых специальных типов матриц существуют простые способы экономного хранения.

(Ленточные матрицы – хранение по диагоналям.)

Для произвольной разреженной матрицы нужен универсальный подход.

Наиболее очевидный способ: **координатный формат**. Хранятся только ненулевые элементы матрицы, и их координаты – номера строк и столбцов.

Нужны три одномерных массива:

- массив ненулевых элементов матрицы A (values);

- массив номеров строк матрицы A для элементов values (обозначим rows);

- массив номеров столбцов матрицы A для элементов values (обозначим cols).

Данное представление называют **полным** (представлена вся матрица A) и **неупорядоченным** (элементы матрицы могут храниться в произвольном порядке).

Координатный формат

$$A = \begin{bmatrix} 1 & -1 & 0 & -3 & 0 \\ -2 & 5 & 0 & 0 & 0 \\ 0 & 0 & 4 & 6 & 4 \\ -4 & 0 & 2 & 7 & 0 \\ 0 & 8 & 0 & 0 & -5 \end{bmatrix}$$

```
values=(1, -1, -3, -2, 5, 4, 6, 4, -4, 2, 7, 8, -5);  
rows=(1, 1, 1, 2, 2, 3, 3, 3, 4, 4, 4, 5, 5);  
cols=(1, 2, 4, 1, 2, 3, 4, 5, 1, 3, 4, 2, 5).
```


Разреженный строчный формат

Разреженный строчный формат, РСФ.

(the compressed sparse row (CSR) data format).

Размерность матрицы A равна N , количество ненулевых элементов – NNZ .

РСФ представление матрицы S состоит из трех массивов:

- Массив A длины NNZ . Вещественный массив, содержащий все ненулевые элементы матрицы по порядку слева направо, начиная с первой строки до последней.
- Массив IA длины $N+1$. Целый массив, такой что $IA(1) = 1$, $IA(k+1) = IA(k) + nnz_k$, где nnz_k – число ненулевых элементов в k -ой строке.
- Массив JA длины NNZ . Это целый массив, содержащий номера столбцов для каждого элемента массива A .

Разреженный строчный формат

$$A = \begin{bmatrix} 1 & -1 & 0 & -3 & 0 \\ -2 & 5 & 0 & 0 & 0 \\ 0 & 0 & 4 & 6 & 4 \\ -4 & 0 & 2 & 7 & 0 \\ 0 & 8 & 0 & 0 & -5 \end{bmatrix}$$

$A = (1, -1, -3, -2, 5, 4, 6, 4, -4, 2, 7, 8, -5);$
 $JA = (1, 2, 4, 1, 2, 3, 4, 5, 1, 3, 4, 2, 5); \text{ (cols)}$
 $IA = (1, 4, 6, 9, 12, 14). \text{ (pointer)}$

Разреженный строчный формат

В зависимости от того, как записываются номера столбцов в массиве JA (по порядку или нет) различают **упорядоченное** и **неупорядоченное** представления.

Неупорядоченные представления нужны для алгоритмического удобства. Результат матричных операций – неупорядоченное представление, для его упорядочения требуется дополнительное время, а большинство алгоритмов не требует упорядоченности представления.

Иногда для представления разреженных матриц используется **разреженный столбцовый формат (РСтФ)**, который конструируется аналогично РСФ.

Умножение матрицы в формате РСФ на вектор

$$y = A \cdot x$$

Умножение матрицы в стандартном формате:

$$y(i) = \sum_{j=1}^N A(i,j) x(j)$$

Умножение матрицы в формате РСФ на вектор

$$y = A \cdot x$$

Умножение матрицы в стандартном формате:

$$y(i) = \sum_{j=1}^N A(i,j) x(j)$$

Умножение матрицы в РСФ

$$y(i) = \sum_{j=IA[i]}^{IA[i+1]-1} A[j] x(JA[j])$$

Умножение разреженных матриц

Каждый элемент матрицы $C = A \cdot B$ – произведение двух **разреженных векторов**.

Возможно сконструировать алгоритмы, **линейные** по числу ненулевых элементов.

Например, аналог сортировки слиянием, и более изощренные.

Прямые методы для разреженных матриц

При решении СЛАУ (Гауссом, разложениями...) происходит **заполнение** разреженных матриц.

ПРИМЕР: Заполнение стреловидной и обратной стреловидной матриц.

Проблемы, вызываемые заполнением:

- Необходимо отводить память для хранения возникших ненулевых элементов.
- Время, затрачиваемое при выполнении факторизации, быстро увеличивается с ростом заполнения, так как приходится выполнять гораздо большее количество арифметических операций.
- Границы ошибок увеличиваются вместе с ростом заполнения.

Прямые методы для разреженных матриц

Способ уменьшить заполнение – выполнить предварительную перестановку строк/столбцов.

Матрица P называется **матрицей перестановки**, если в каждой строке и столбце матрицы находится лишь один единичный элемент.

Умножение слева – перестановка строк, справа – перестановка столбцов.

$$PAQ^T Qx = Py.$$

Для симметричной матрицы A : $P = Q$, и преобразованная матрица тоже симметрична.

Задача нахождения перестановки, минимизирующей заполнение, является NP-трудной (минимум должен вычисляться по всем $N!$ перестановкам).

На практике используют эвристические алгоритмы, которые не гарантируют оптимальную перестановку, но, как правило, дают приемлемый результат.

Прямые методы для разреженных матриц

При решении симметричной положительно определенной матрицы, не требует перестановок для поддержания численной устойчивости.

Поскольку PAP^T также симметрична и положительно определена при любой перестановке P , можно симметрично переупорядочить, не заботясь о численной устойчивости и до начала реального численного разложения.

Этапы решения разреженной системы:

1. Переупорядочивание: вычисление матрицы перестановки P
2. Символическое разложение: построение портрета матрицы и выделение памяти под хранение ненулевых элементов
3. Численное разложение: вычисление значений матрицы и размещение их в выделенной памяти
4. Обратный ход: решение двух треугольных систем уравнений.

Для разреженных матриц общего вида, как правило, нельзя предсказать, где произойдет заполнение, пока не начались собственно вычисления.

Метод минимальной степени

Основные понятия теории графов.

Пусть исходной матрице A соответствует граф $G(A)$.

Алгоритм минимальной степени строит последовательность графов исключения G_i , каждый из которых получен из предыдущего удалением вершины с минимальной степенью и созданием клики между всеми вершинами, которые были смежными с удаленной.

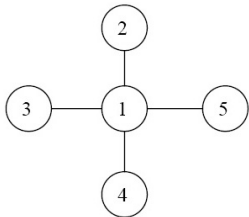
Когда вершин с минимальной степенью несколько, выбирается любая.

Алгоритм продолжается до тех пор, пока в очередном графе есть вершины.

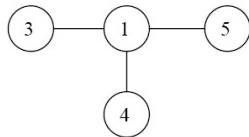
По мере удаления вершин, их номер записывается в перестановку π , по которой впоследствии строится матрица перестановки P .

Метод минимальной степени

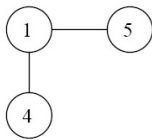
$$\begin{bmatrix} 4 & 1 & 2 & 0.5 & 2 \\ 1 & 0.5 & 0 & 0 & 0 \\ 2 & 0 & 3 & 0 & 0 \\ 0.5 & 0 & 0 & 0.625 & 0 \\ 2 & 0 & 0 & 0 & 16 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 17 \\ 3 \\ 7 \\ 6 \\ 12 \end{bmatrix}$$



$G_0, \pi = \{\emptyset\}$



$G_1, \pi = \{2\}$



$G_2, \pi = \{2, 3\}$



$G_3, \pi = \{2, 3, 4\}$



$G_4, \pi = \{2, 3, 4, 1\}$

$G_5 = \emptyset, \pi = \{2, 3, 4, 1, 5\}$

Полученная перестановка $\pi = \{2, 3, 4, 1, 5\}$ соответствует матрице

$$P = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Переупорядоченная матрица A :

$$\overline{A} = PAP^T = \begin{bmatrix} 0.5 & 0 & 0 & 1 & 0 \\ 0 & 3 & 0 & 2 & 0 \\ 0 & 0 & 0.625 & 0.5 & 0 \\ 1 & 2 & 0.5 & 4 & 2 \\ 0 & 0 & 0 & 2 & 16 \end{bmatrix}$$

При практической реализации метод минимальной степени в его исходном виде не используется в силу его значительной трудоемкости.

Широко известны две модификации метода:

- приближенный метод минимальной степени (Approximate Minimum Degree, AMD).

Идея: вычисление приближенной степени вершины с помощью следующего эвристического правила: степень вершины не превосходит сумму степеней ее соседей.

- множественный метод минимальной степени (Multiple Minimum Degree, MMD).

Используется следующая модификация: если на некотором шаге алгоритма нашлось несколько вершин с минимальной степенью, то можно одновременно удалить все из них, не являющиеся соседями.

Профильные методы.

Проблема прямых методов: **заполнение** матриц.

Определим

$$f_i = \min\{j : a_{ij} \neq 0\}, \quad \beta_i(A) = i - f_i(A).$$

Оболочка A :

$$Env(A) = \{\{i, j\} : 0 < i - j \leq \beta_i(A)\}.$$

Профиль A :

$$|Env(A)| = \sum_{i=1}^N \beta_i(A).$$

Число операций для разложения матрицы: $\sim \sum_{i=1}^N \beta_i^2(A)$.

Профиль множителя разложения Холецкого совпадает с профилем треугольника исходной матрицы.

Нужно найти перестановку, уменьшающую профиль!

Алгоритм Катхилла-Макки

Для каждой вершины i графа Ω , соответствующего матрице, вычислить степень ρ_i вершины. Затем выбрать какую-либо вершину i , для которой $\rho_{i1} = \min_i \rho_i$ и пометить эту вершину первой. Схему Катхилла-Макки можно рассматривать как метод уменьшения ширины ленты матрицы посредством локальной минимизации чисел β_i .

В основе метода следующее замечание.

Пусть Y – помеченный узел, а Z – непомеченный сосед Y .

Для того, чтобы уменьшить ширину ленты в строке, соответствующей Z , нужно присвоить Z номер, как можно менее отличающийся от номера Y .

Алгоритм Катхилла-Макки

- Выбираем вершину 1^* и обозначаем её как 1.
- Присвоить вершинам, смежным с вершиной 1, новые номера, начиная с 2 в порядке возрастания их степеней. Если степени некоторых смежных вершин совпадают, то выбирают любую из них. Эти вершины относят к первому уровню.
- Повторить эту процедуру последовательно для каждой из вершин первого уровня: (для вершины 2, затем – 3 и т.д.).
- Повторить процедуру для вершин каждого следующего уровня, пока все вершины графа не будут перенумерованы. Если граф состоит из нескольких несвязных подграфов, то процедура повторяется для каждого несвязного подграфа.
- Переставить строки и столбцы матрицы в соответствии с новыми номерами вершин.

Алгоритм Катхилла-Макки

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|----|---|---|---|---|---|---|---|---|---|----|
| 1 | X | X | | X | | | | | X | |
| 2 | X | X | X | | | | | | X | |
| 3 | | X | X | | X | | | | X | |
| 4 | X | | | X | X | X | | | X | X |
| 5 | | | X | X | X | | | X | X | X |
| 6 | | | | X | | X | X | | | X |
| 7 | | | | | | X | X | X | | X |
| 8 | | | | | X | | X | X | | X |
| 9 | X | X | X | X | X | | | | X | X |
| 10 | | | | X | X | X | X | X | X | X |

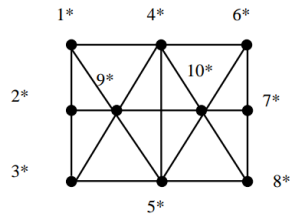


Рис. а

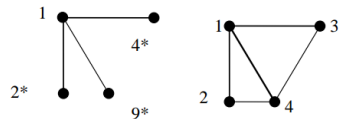


Рис.б.

Алгоритм Катхилла-Макки

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|------|---|---|---|---|---|---|---|----|---|----|
| | | | 4 | 9 | 3 | | 5 | 10 | 7 | 8 |
| 1 | X | X | X | X | | | | | | |
| 2 | X | X | | X | X | | | | | |
| 3 4 | X | | X | X | | X | X | X | | |
| 4 9 | X | X | X | X | X | | X | X | | |
| 5 3 | | X | | X | X | | X | | | |
| 6 | | | X | | | X | | X | X | |
| 7 5 | | | X | X | X | | X | X | | X |
| 8 10 | | | X | X | | X | X | X | X | X |
| 9 7 | | | | | | X | | X | X | X |
| 10 8 | | | | | | | X | X | X | X |

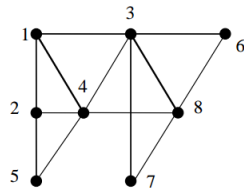


Рис.с.

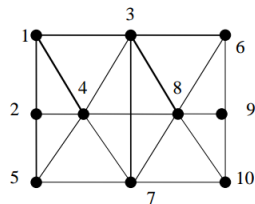


Рис. d

Обратный алгоритм Катхилла-Макки

Алгоритм Катхилла-Макки уменьшает ширину ленты посредством локальной минимизации чисел β_i .

Есть надежда, что он позволит уменьшить и их сумму.

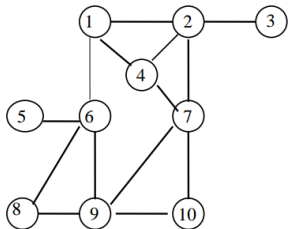
Упорядочение, получаемое обращением упорядочения Катхилла-Макки, часто гораздо сильнее уменьшает профиль, чем первоначальное упорядочение, хотя ширина ленты остается неизменной!

Это упорядочение – обратное упорядочение Катхилла-Макки (RCM) (Reverse Cuthill-McKee). Доказано, что обратная схема всегда **не хуже прямой** в отношении хранения и обработки оболочки.

Обратный алгоритм Катхилла-Макки

- Определить начальный узел r и выполнить присвоение $r \rightarrow X_1$. (Выбор начального узла важен!).
- (Основной цикл). Для $i = 1, \dots, N$ найти всех нумерованных соседей узла X_i и занумеровать их в порядке возрастания степеней.
- (Обратное упорядочение). Обратное упорядочение Катхилла-Макки есть y_1, y_2, \dots, y_N , где $y_i = X_{N-i+1}$ для $i = 1, \dots, N$.

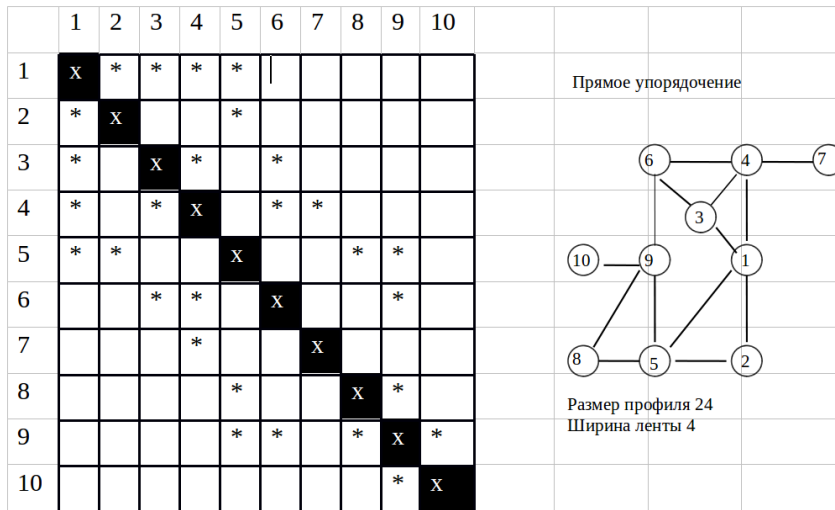
Обратный алгоритм Катхилла-Макки



Размер профиля 23
Ширина ленты 5

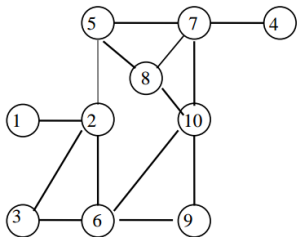
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|----|---|---|---|---|---|---|---|---|---|----|
| 1 | X | * | | * | | * | | | | |
| 2 | * | X | * | * | | | * | | | |
| 3 | | * | X | | | | | | | |
| 4 | * | * | | X | | | * | | | |
| 5 | | | | | X | * | | | | |
| 6 | * | | | | * | X | | * | * | |
| 7 | | * | | * | | | X | | * | * |
| 8 | | | | | | * | | X | * | |
| 9 | | | | | | * | * | * | X | * |
| 10 | | | | | | | * | | * | X |

Обратный алгоритм Катхилла-Макки



Обратный алгоритм Катхилла-Макки

Обратное упорядочение
в соответствии с шагом 3.



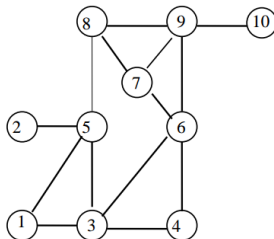
Размер оболочки = 22
Ширина ленты 4

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|----|---|---|---|---|---|---|---|---|---|----|
| 1 | x | * | | | | | | | | |
| 2 | * | x | * | | * | * | | | | |
| 3 | | * | x | | | * | | | | |
| 4 | | | | x | | | * | | | |
| 5 | | * | | | x | | * | * | | |
| 6 | | * | * | | | x | | | * | * |
| 7 | | | | * | * | | x | * | | * |
| 8 | | | | | * | | * | x | | * |
| 9 | | | | | | * | | | x | * |
| 10 | | | | | | * | * | * | * | x |

Обратный алгоритм Катхилла-Макки

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|----|---|---|---|---|---|---|---|---|---|----|
| 1 | x | | * | | * | | | | | |
| 2 | | x | | | * | | | | | |
| 3 | * | | x | * | * | * | | | | |
| 4 | | | * | x | | * | | | | |
| 5 | * | * | * | | x | | | * | | |
| 6 | | | * | * | | x | * | | * | |
| 7 | | | | | | * | x | * | * | |
| 8 | | | | | * | | * | x | * | |
| 9 | | | | | | * | * | * | x | * |
| 10 | | | | | | | | | * | x |

Размер профиля 18



Начальный узел 3.

Определение начального узла

Цель состоит в том, чтобы найти пару узлов, удаленных друг от друга на максимальное или почти максимальное расстояние. Практический опыт свидетельствует, что такие узлы хороши в качестве начальных для нескольких алгоритмов упорядочения, в том числе для алгоритма RCM (Reverse Cuthill-McKee).

Расстояние $d(x, y)$ между двумя узлами x и y связного графа $G = (X, E)$ – длина кратчайшего пути, соединяющего эти узлы.

Эксцентриситет узла x : $e(x) = \max\{d(x, y), \text{ где } y \in X\}$.

Узел $x \in X$ называется периферийным, если эксцентриситет равен диаметру графа.

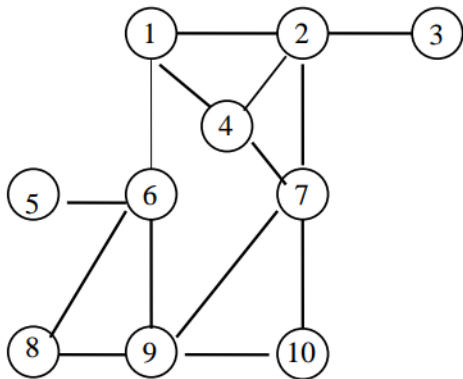
Цель: описание эвристического эффективного алгоритма для определения узлов с большим эксцентриситетом.

Алгоритм **не дает гарантии**, что будет найден периферийный узел или хотя бы узел, близкий к периферийному.

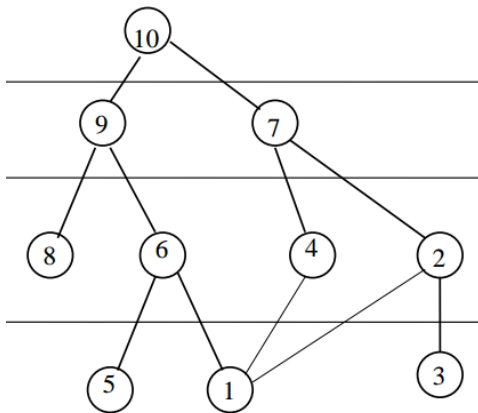
Определение начального узла

Алгоритм Гиббса et al (1976)

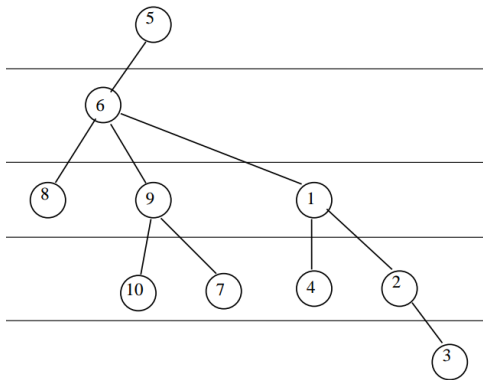
1. инициализация: выбрать в X произвольный узел r .
2. построение структуры уровней:
 - а) построить структуру уровней с корнем в r .
 - б) определить $e(r)$.
3. стягивание последнего уровня: выбрать в последнем уровне узел x с минимальной степенью.
4. построение структуры уровней:
 - а) построить структуру уровней с корнем в x .
 - б) если $e(x) > e(r)$, положить $x \rightarrow r$ и перейти к шагу 3.
5. Узел x является псевдопериферийным.



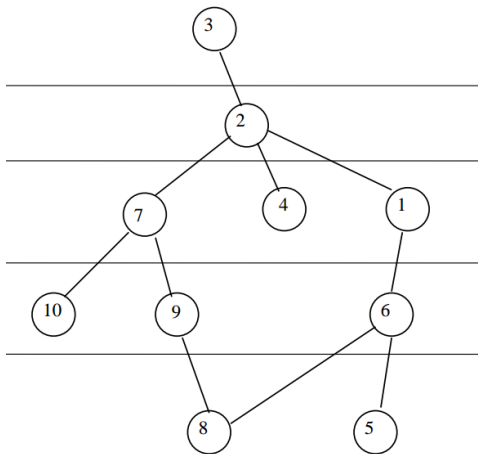
Исходный граф.



Построение с узла 10, $e(10) = 3$.



Выбираем узел 5.
Построение с узла 5, $e(5) = 4$.



Выбираем узел 3.
Построение с узла 3, $e(3) = 4$.

1. Бахвалов Н.С., Жидков Н.П., Кобельков Г.М., *Численные методы*, М.: Бином. Лаборатория знаний, 2012. - 636 с. Параграфы 6.1, 6.3, 6.6, 6.7, 6.11.
2. Faires J.D., Burden R.L. *Numerical methods* (3ed., Thomson Brooks Cole, 2003). Глава 7.