УДК 004

АННОТАЦИЯ
К РАСЧЕТУ ПАРАМЕТРОВ СЕТЕВЫХ СТРУКТУР

Гладких Александр Петрович

gap84@rambler.ru

Кафедра Распределенных вычислительных систем, ТГТУ

Ключевые слова и фразы: сеть, сетевая структура, алгоритм, максимальный поток между точками, путь с наименьшим временем, наибольшая пропускная способность, пример расчета сети.

Аннотация: в данной статье рассмотрим методы расчета некоторых ключевых параметров сетевых структур, таких как путь между двумя заданными узлами сети с минимальным временем, наибольшая (максимальная) общая пропускная способность между точками сети.

Введение

Известно, что задача определения пути с наименьшим временем, как и задача нахождения общей максимальной пропускной способности между узлами в сети имеет важное практическое значение. Существуют различные способы и подходы к решению, однако, в последнее время в связи с большим и стремительным развитием компьютерной техники, с нашей точки зрения, удобен метод динамического программирования, предложенный Р.Беллманом [1,2]. Этот метод хорошо реализуем практически, легко программируется с использованием компьютеров. Далее мы рассмотрим то, что уже предложено, для решения поставленных задач, и попытаемся определить некоторые аспекты, которые могли бы улучшить моделируемые ситуации для обозначенных выше задач. Эти предложенные алгоритмы решения также достаточно легко программируются.

1 Расчет пути с наименьшим временем

Перейдем к первому из перечисленных выше понятий. Мы будем определять данный параметр, опираясь на методы, разработанные в теории динамического программирования. Для примера рассмотрим представленную ниже сетевую структуру.

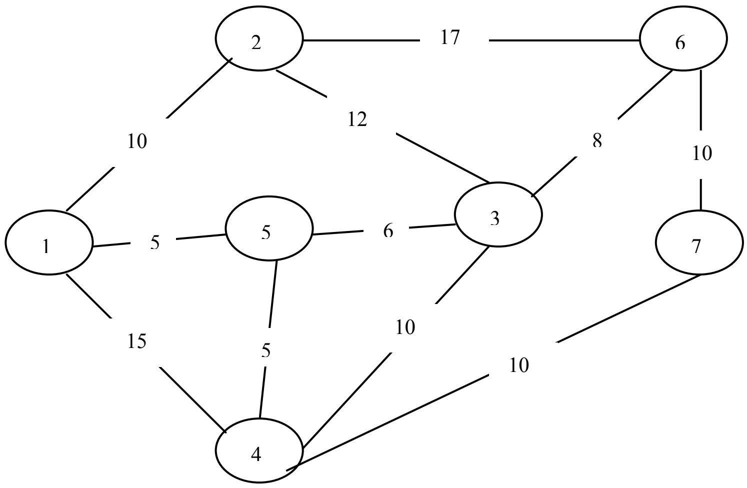

Рис.1 Рассматриваемая сетевая структура

В основе метода определения пути с наименьшим временем лежит формула последовательных приближений [1,2], которая была предложена в теории динамического программирования, и которая в дополнении с соответствующими ограничениями выглядит следующей совокупностью уравнений:

$$\left.\begin{aligned} &\mu^{k}(i) = \min_{i \neq j}\{\tau\ (i,j) + \mu^{k-1}(j)\}, i = 1,2..N-1 \\ &\mu^{k}(M) = 0 \end{aligned}\right\} \qquad (1)$$

где $k = 0,1..N-2$

$$\mu^{0}(i) = \begin{cases} \tau\ (i,M), i \neq M, i = 1,2..N, прямой\_путь\_существует \\ \infty, i \neq M, i = 1,2..N, прямого\_пути\_нет \end{cases}$$

где $\mu(i)$ - время для перехода из i-го в M-ый узел через k промежуточных узлов, N – общее число узлов, $\tau(i, j)$ - время перехода из i - го в j – ый узел.

В данном случае количество промежуточных узлов k изменяется от 0 до N-2, т.к. минимальный путь не может иметь петель, вычитанием 2-х узлов исключаются из общего числа исходный и конечный узлы.

Как показано в [1], если решение исходной задачи существует, то найденная последовательность будет сходиться к искомому решению не более, чем за $N-2$ итераций. Это решение единственно, но достигаться может на различных путях.

Здесь мы не будем более детально рассматривать процесс непосредственного решения по формулам (1), т.к. он схож с рассмотренным далее для задачи определения общей максимальной пропускной способности по принципу решения.

2 Расчет общей максимальной пропускной способности между узлами

Теперь мы рассмотрим другую несколько более сложную задачу о нахождении наибольшей, максимально возможной пропускной способности между некоторыми двумя точками.

В данном случае мы разобьем задачу на более локальные. Прежде всего, мы просто определим максимальную пропускную способность между двумя точками. В этом случае логический алгоритм решения выглядит приблизительно так (для примера применительно к схеме на рисунке 2, при выборе направления движения из узла 1):

1. Надо выбрать максимальное из значений по направлениям 1->2, 1->3, 1->4; т.е. это значение 15, направление 1->3

2. Но на направлении 1->3 далее пропускная способность падает до значения равного 3, поэтому, поэтому более трех далее не пройдет.

Следовательно, необходимо выбрать наименьшее значение из значений оставшейся части ветви от 3 и далее и новой выбранной части от 1 к 3.

3. Т.к. при выборе направления от первого узла уже необходимо руководствоваться и значениями пропускных способностей оставшихся частей сетевой структуры, то пункты 1 и 2 нужно выполнять наоборот, т.е сначала 2, потом 1.

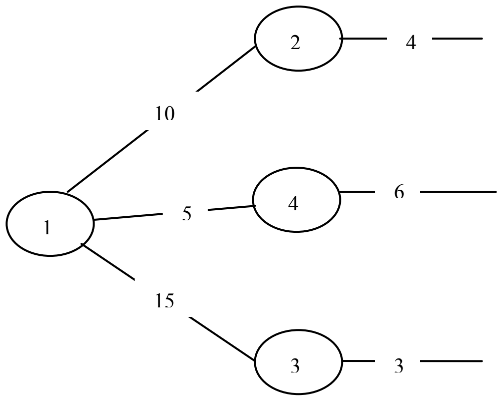

Рис. 2 Схема для иллюстрации логического алгоритма

Более строгая математическая формулировка данного алгоритма может быть выражена законом (max-min) композиции [3]. Это представлено ниже:

$$\left.\begin{aligned} &\mu^{k}(i) = \max_{i \neq j} \min\{\mu(i,j);\mu^{k-1}(j)\}, i = 1,2..N-1, k = 0,1..N-2 \\ &\mu^{k}(M) = \infty, k = 0,1..N-2 \end{aligned}\right\}$$

$$u \qquad (2)$$

$$\mu^{0}(i) = \begin{cases} \mu\ (i,M), i \neq M, i = 1,2..N, прямой\_путь\_существует \\ 0, i \neq M, i = 1,2..N, прямого\_пути\_нет \end{cases}$$

где $\mu^{k}(i)$ - пропускная способность из i-го в М-ый узел через k промежуточных узлов , N – общее число узлов, $\mu\ (i,j)$ - пропускная способность прямого пути из i - го в j – ый узел.

Теперь возвращаясь к первоначальной задаче определения наибольшей общей пропускной способности между точками в сетевой структуре, используя полученный выше метод нахождения максимальной пропускной способности между двумя точками, предлагается следующий алгоритм решения:

1. Первоначально по формулам (2) находится максимальная пропускная способность между точками, найденный путь запоминается в случае если найденное значение пропускной способности положительно. Если оно равно нулю, то при заданных исходных условиях положительного решения не существует, т.е. требуется поменять исходные данные для задачи.

2. Найденное значение максимальной пропускной способности добавляется к значению общей максимальной пропускной способности.

3. Из значений всех отрезков, на которые данный путь разбит узлами, вершинами, вычитается найденное значение максимальной пропускной способности.

4. Образуется сетевая структура с новыми пропускными значениями по всем веткам.

5. Для новой структуры выполняется пункт 1-4 при условии, что при выполнении пункта 1 существует, т.е. находится путь из исходной точки в конечную через заданное количество промежуточных с ненулевым значением пропускной способности, иначе выполняется переход к пункту 6. Если ненулевое значение пропускной способности найдено, оно прибавляется к общему максимальному значению пропускной способности между исходной и конечной точками.

6. Путей из исходной точки в конечную через заданное количество промежуточных с ненулевым значением пропускной способности больше не существует, поэтому общая максимальная пропускная способность из исходной точки в конечную найдена в данной сетевой структуре.

3 Пример расчета общей максимальной пропускной способности между двумя узлами в сети

Далее приводятся рассуждения для конкретного примера сети, схема которого изображена на рис. 3

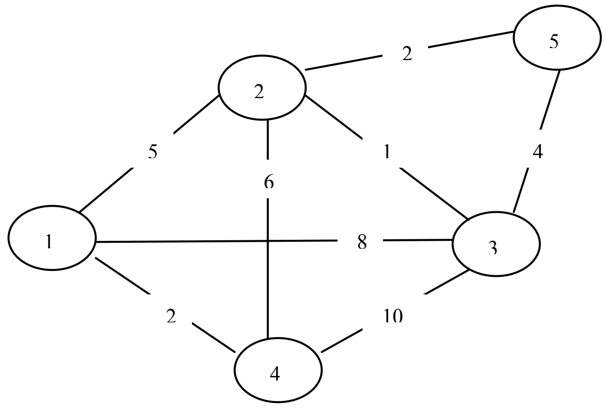

Рис.3 Схема сетевой структуры, взятая для примера пояснения алгоритма решения задачи

Примем, что исходный, стартовый узел – это 1, а конечный – 3 и количество промежуточных узлов, требующихся нам при определении общей максимальной пропускной способности из исходного узла в конечный равно 1, т.е.

$$i = 1$$
$$M = 3$$
$$k = 1$$

Исходные данные по формулам (2) для $\mu^0(i)$ и $\mu(i, j)$:

$$\mu^0(1) = 8; \mu^0(2) = 1; \mu^0(3) = \infty; \mu^0(4) = 10; \mu^0(5) = 4;$$

$$\mu(1,2) = 5; \mu(1,3) = 8; \mu(1,4) = 2; \mu(1,5) = 0;$$
$$\mu(2,1) = 5; \mu(2,3) = 1; \mu(2,4) = 6; \mu(2,5) = 2;$$
$$\mu(3,1) = 8; \mu(3,2) = 1; \mu(3,4) = 10; \mu(3,5) = 4;$$
$$\mu(4,1) = 2; \mu(4,2) = 8; \mu(4,3) = 10; \mu(4,5) = 0;$$
$$\mu(5,1) = 0; \mu(5,2) = 2; \mu(5,3) = 4; \mu(5,4) = 0;$$

Рассчитаем значения $\mu^1(i)$ согласно первому пункту алгоритма:

$$\mu^1(1) = \max \min_{i \neq j} \begin{bmatrix} \mu(1,2); \mu^0(2) \\ \mu(1,3); \mu^0(3) \\ \mu(1,4); \mu^0(4) \\ \mu(1,5); \mu^0(5) \end{bmatrix} = \max \min_{i \neq j} \begin{bmatrix} 5;1 \\ 8;\infty \\ 2;10 \\ 0;4 \end{bmatrix} = \max \begin{bmatrix} 1 \\ 8 \\ 2 \\ 0 \end{bmatrix} = 8$$

следующий узел

$\text{next} = 3$

путь : 1 - 3 - 3

аналогично :

$\mu^1(2) = 6$

$\text{next} = 4$

путь : 2 - 4 - 3

$\mu^1(3) = \infty$

$\text{next} = 3$

путь : 3 - 3 - 3

$\mu^1(4) = 10$

$\text{next} = 3$

путь : 4 - 3 - 3

$\mu^1(5) = 4$

$\text{next} = 3$

путь : 5 - 3 - 3

Нас интересует значение $\mu^1(1)$ по постановке задачи в рассматриваемом примере и согласно 2 пункту алгоритма значение общей максимальной пропускной способности сейчас $MaxFlow = \mu^1(1) = 8$.

Теперь, выполняя пункты 3 и 4 алгоритма получаем $\mu(1,3) = \mu(1,3) - \mu^1(1) = 0$ и новую сетевую структуру, изображенную на рис.4

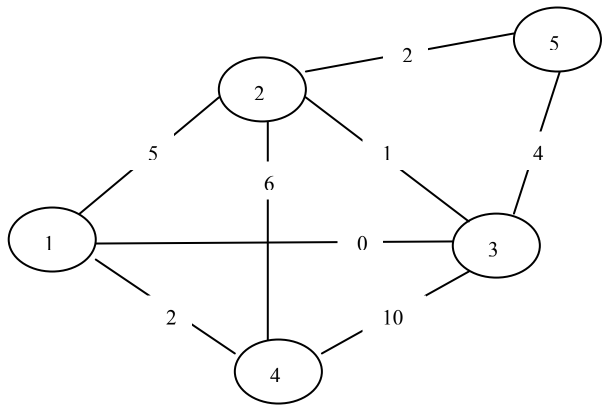

Рис.4 Новая полученная сетевая структура

Переходим к пункту 5 и выполняем пункт 1 с целью нахождения ненулевого решения для новой схемы, представленной на рис.4.

Теперь исходные данные:

$$\mu^0(1) = 0; \mu^0(2) = 1; \mu^0(3) = \infty; \mu^0(4) = 10; \mu^0(5) = 4;$$

$$\mu(1,2) = 5; \mu(1,3) = 0; \mu(1,4) = 2; \mu(1,5) = 0;$$
$$\mu(2,1) = 5; \mu(2,3) = 1; \mu(2,4) = 6; \mu(2,5) = 2;$$
$$\mu(3,1) = 0; \mu(3,2) = 1; \mu(3,4) = 10; \mu(3,5) = 4;$$
$$\mu(4,1) = 2; \mu(4,2) = 8; \mu(4,3) = 10; \mu(4,5) = 0;$$
$$\mu(5,1) = 0; \mu(5,2) = 2; \mu(5,3) = 4; \mu(5,4) = 0;$$

Далее алгоритм расчета повторяется, и через несколько итераций может быть получено окончательное значение для общей максимальной пропускной способности между узлами, которое равно $MaxFlow = 11$

4 Программная реализация

Программа для реализации описываемого выше алгоритма была написана на языке С++.

Функциональная блок схема программы показана ниже на рис.7

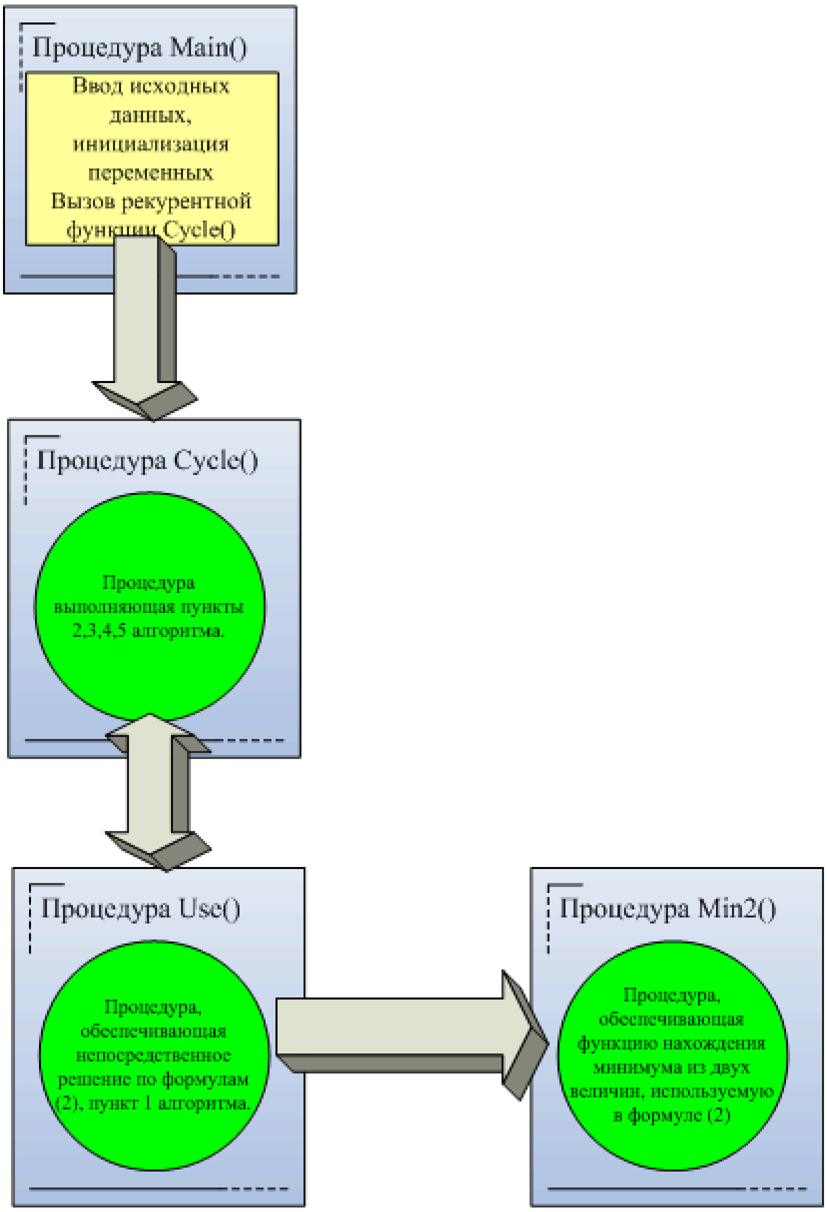

Рис.7 Основные блоки программы

На рисунке стрелками изображена взаимосвясь между основными блоками программы:

←→ - показана рекуррентная зависимость между процедурами Cycle() и Use(), взаимовызывающими друг друга;

→ - показана обычный вызов из одной процедуры другой.

Заключение

Нами были рассмотрены вопросы определения пути с наименьшим временем и нахождения общей максимальной пропускной способности между узлами. Были рассмотрены методы и предложены алгоритмы решения, основанные на теории динамического программирования.

В данной статье предполагалось, что значения ребер сетевых структур заранее известны и определены, в дальнейшем мы постараемся рассмотреть вариант с изменяющимися весовыми значениями ребер сетевой структуры для задачи определения пути с наименьшим временем и нахождения общей максимальной пропускной способности между узлами. Это более близко к реальности и поэтому представляет дополнительный интерес.

Список литературы

1. Беллман Р., Дрейфус С. Прикладные задачи динамического программирования: пер. с англ. – М.: Наука, 1965, 460 с.
2. Беллман Р., Калаба Р. Динамическое программирование и современная теория управления, М.: 1969 – 120 с.
3. Кофман А. Введение в теорию нечетких множеств: пер. с франц. – М.: Радио и связь, 1982 – 432 с.

TO CALCULATION OF PARAMETERS OF NETWORK STRUCTURES

Gladkih A.P.

gap84@rambler.ru

Faculty of the Distributed computing systems, TSTU

Keywords: a network, network structure, algorithm, the maximal stream between points, a way with the least time, the greatest throughput, an example of calculation of a network.

Abstract: in this article we shall consider methods of calculation of some key parameters of network structures, such as a way between two set points of a network with minimal time, the greatest (maximal) general throughput between points of a network.