МИНИСТЕРСТВО ОБЩЕГО И ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ РФ
Санкт-Петербургский
государственный университет аэрокосмического приборостроения

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

к выполнению лабораторной работы

"ПРОЕКТИРОВАНИЕ ГЛОБАЛЬНЫХ ВЫЧИСЛИТЕЛЬНЫХ СЕТЕЙ"

Чуркин В.И., Горбачев С.В.

Рекомендуется для студентов специальностей 2201 и 5528 по курсу "Вычислительные комплексы, системы и сети", «Телекоммуникации в сетях ЭВМ»

С.-Петербург
2000 г.

Составители: С.В.Горбачев, В.И.Чуркин
Рецензенты: кафедра автоматизированных систем управления Санкт– Петербургского государственного университета аэрокосмического приборостроения;
канд. техн. наук *Г.С.Бритов*

Содержатся методические указания для изучения проблем проектирования глобальных вычислительных сетей с коммутацией пакетов и сообщений. Рассматриваются вопросы по проектированию структуры распределенной вычислительной сети и по определению максимальных пропускных способностей. Описывается постановка задачи формализованного проектирования вычислительных сетей общего пользования с распределенной структурой, где в качестве критерия при синтезе структуры сети используется стоимость.

С использованием формального аппарата графов в качестве модели соединения аппаратных средств распределенных вычислительных сетей описываются приближенные эвристические алгоритмы оптимизации топологии глобальных вычислительных сетей. Приводится один из популярных алгоритмов определения кратчайших расстояний – алгоритм Дейкстры, исследуется алгоритм Форда-Фалкерсона, предназначенный для нахождения максимального потока и минимального разреза в сети.

Предназначены для студентов специальности 2201 "Вычислительные машины, комплексы системы и сети", студентов магистерской подготовки по специальности 5528 «Информатика и вычислительная техника» и для самостоятельного изучения вопросов проектирования вычислительных сетей.

Подготовлены к публикации кафедрой вычислительных систем и сетей Санкт-Петербургского государственного университета аэрокосмического приборостроения.

ВВЕДЕНИЕ

Настоящие методические указания посвящены двум направлениям в проектировании глобальных вычислительных сетей с коммутацией сообщений и пакетов, а именно: проектированию структуры распределенной вычислительной сети и выбору пропускных способностей.

Обоснованному решению этих задач должен предшествовать выбор абонентских подсистем и распределение по ним прикладных программ.

Решение этих задач рассмотрено в учебном пособии /1/.

1. МЕТОДЫ ТЕОРИИ ГРАФОВ И ПРОЕКТИРОВАНИЕ СТРУКТУР РАСПРЕДЕЛЕННЫХ ВЫЧИСЛИТЕЛЬНЫХ СЕТЕЙ

1.1. Показатели эффективности проектирования вычислительных сетей

Наличие множества критериев эффективности функционирования вычислительных сетей обусловливает многокритериальный характер задачи их проектирования. Для упрощения задачи проектирования определяют главный показатель эффективности, который подлежит оптимизации, а остальные показатели переводят в разряд ограничений. В зависимости от основного показателя эффективности различают следующие варианты постановки задач формализованного проектирования вычислительных сетей:

- Синтез сети по критерию времени – определить такую структуру вычислительной сети, которая обеспечивает минимальную задержку сообщений при заданных стоимости и надежности.
- Синтез сети по критерию стоимости – при заданных средней задержке в передаче сообщений, показателях надежности сети и заданных объемах информации абонентов, подлежащих обработке в сети. Спроектировать сеть такой структуры, которая будет иметь минимальную стоимость.
- Синтез сети по критерию надежности – при заданных общей стоимости сети и средней задержке в передаче сообщений. Спроектировать сеть, имеющую максимальную надежность.

Для коммерческих вычислительных сетей общего пользования наиболее приемлемым является синтез по критерию стоимости, а для вычислительных сетей специального назначения – синтез по критерию времени и надежности.

Сети передачи данных, которые лежат в основе вычислительных сетей, делятся на межцентровые (с распределенной структурой) для обмена информацией между узлами сети, и абонентские (со структурой дерева) для обеспечения доступа абонентов к сети.

Методы проектирования зависят от вида сети передачи данных. Так для более простых абонентских сетей используются точные методы проектирования оптимальных деревьев, в то время как для проектирования межцентровых сетей с распределенной топологией используются приближенные эвристические алгоритмы. Качество получаемых результатов при этом определяется тем, насколько удачные эвристики используются в алгоритме.

В рамках данных методических указаний мы будем рассматривать лишь проектирование распределенных (межцентровых) вычислительных сетей общего пользования, где в качестве критерия при синтезе структуры сети используется стоимость. В качестве модели соединения аппаратных средств распределенных вычислительных сетей используются графы. Узлам коммутации вычислительной сети при этом сопоставляются вершина, а каналам передачи данных – дуги графа. Если все каналы вычислительной сети являются дуплексными или полудуплексными, то соответствующий граф будет неориентированным. Если сеть имеет симплексные каналы, то соответствующий граф будет ориентированным графом или орграфом. Орграф является более общим объектом, чем неориентированный граф, так как последний преобразуется в орграф посредством замены каждого его ребра в пару разнонаправленных дуг. Исходя из этого в качестве модели вычислительной сети мы будем использовать орграфы и называть их просто "графы".

1.2. Алгоритм определения кратчайших расстояний

При проектировании распределенной вычислительной сети непременным требованием является связность графа этой сети. Причем, в случае орграфа речь может идти как о сильной связанности (когда из каждой вершины можно попасть в любую другую вершину), так и о слабой связности (когда для некоторых пар вершин из одной в другую можно попасть, а обратно – нет), для определения связности графа может быть использован алгоритм определения кратчайшего расстояния. Действительно, если в результате выполнения алгоритма расстояния всех вершин графа от некоторой выделенной вершины конечны, то граф будет связным, а в противном случае – нет.

Рассмотрим один из популярных алгоритмов определения кратчайших расстояний – алгоритм **Дейкстры**. Этот алгоритм определяет кратчайшие расстояния от одной вершины взвешенного графа до всех остальных вершин.

Примечание 1. Граф называется взвешенным, если каждой его дуге приписан вес (расстояние).

Алгоритм является наилучшим по порядку числа операций (сложения и сравнения) и порядок (роста) его трудоемкости составляет $\mathbf{O}(3\mathbf{n}^2)$, где $\mathbf{n}$ – число вершин графа.

Примечание 2. Символ "**O**" в данном случае происходит от английского слова order – порядок.

Обозначим через $\mathbf{G = (V,R)}$ ориентированный связный граф,

где $\mathbf{V = \{v_1,v_2,..., v_n\}}$ – множество вершин графа,

$\mathbf{R \subset V \times V}$ – множество дуг графа.

Пусть $\mathbf{L^0} = \|\mathbf{L^0_{ij}}\|$ – $(\mathbf{n}\times\mathbf{n})$-матрица положительных весов дуг графа $\mathbf{G}$. Причем, $\mathbf{L^0_{ij}}$ – вес дуги $(\mathbf{v_i}, \mathbf{v_j})$.

Если $(\mathbf{v_i}, \mathbf{v_j}) \not\subset \mathbf{R}$, то $\mathbf{L^0_{ij}} = +\infty$, где $+\infty$ – несобственное число с его известными свойствами из расширенной системы вещественных чисел.

Введем на элементах $\mathbf{(n \times n)}$-матриц $\mathbf{L} = \|\mathbf{Lij}\|$ трехоперандную операцию:

$$\mathbf{<i, k, j> = \min \{ L_{ij}, L_{ik} + L_{kj} \}}.$$

Для удобства изложения алгоритма Дейкстры будем считать, что требуется найти кратчайшие пути от вершины $\mathbf{v_1}$ графа $\mathbf{G}$.

Инициализация: пусть $\mathbf{S = \{ 2, 3,...,n \}}$, $\mathbf{L_{ij} := L^0_{ij}}$ для всех $\mathbf{i, j}$.

Шаг 1. Положить $\kappa = 2$ и перейти к шагу 2.

Шаг 2. Найти $\mathbf{i^*}$ такое, что $L_{1i^*} = \min\limits_{j \in S} L_{1j}$.

Шаг 3. $\mathbf{S:= S \backslash i^*}$.

Шаг 4. Для $\mathbf{j \subset S}$ выполнить $\mathbf{L_{1i}} := <\mathbf{1, i^*, j}>$.

Шаг 5. Присвоить $\mathbf{\kappa=\kappa+1}$ и, если $\mathbf{n \geq \kappa}$, перейти к шагу 2. В противном случае веса кратчайших путей найдены.

1.3. Выбор пропускных способностей сетевых каналов

Введем понятие диаметра графа, которое играет важную роль при проектировании структур вычислительных сетей с минимальным или ограниченным временем передачи информации. ***Диаметром*** графа называется максимальная длина кратчайших маршрутов в графе (максимум берется по всем парам вершин).

Самым "бедным" по числу дуг связным графом является дерево. Этот граф обладает тем свойством, что между каждой парой вершин существует единственная цепь. Однако, учитывая ненадежность реальных каналов вычислительных сетей, часто этого недостаточно. Одним из показателей структурной живучести вычислительных сетей является число непересекающихся независимых путей между парами узлов сети, и этот показатель важно учитывать при проектировании графа вычислительной сети.

Примечание 3. С формальной стороны понятия цепи и пути (маршрута) в теории графов отличаются. ***Маршрут*** называется цепью, если все его дуги отличны. Если, кроме того, отличны все вершины, то маршрут называется ***простой цепью***.

Существуют два понятия непересекающихся цепей:

– *вершинно*-непересекающиеся цепи, когда ни одна из цепей не содержит общих вершин;

– непересекающиеся *по дугам* цепи, когда ни одна из цепей не содержит общих дуг.

Назовем ***VW****-разделяющим* множеством множество **A** дуг графа, обладающее тем свойством, что любая дуга, принадлежащая всякой простой цепи из **V** в **W**, содержится в **A**.

Аналогично ***VW****-отделяющим* множеством назовем множество **B** вершин графа (не содержащее **V** и **W**), обладающее тем свойством, что любая простая цепь из **V** в **W** проходит через вершину из **B** и все вершины, через которые проходит всякая простая цепь из **V** в **W**, включены в **B**.

Существуют две важные теоремы, которые связаны с именем ***Менгера***.

ТЕОРЕМА 1. Максимальное число *вершинно*-непересекающихся простых цепей, соединяющих две различные несмежные вершины **V** и **W** графа **G**, равно минимальному числу вершин в ***VW****-отделяющем* множестве.

ТЕОРЕМА 2. Максимальное число непересекающихся *по дугам* простых цепей, соединяющих две различные вершины **V** и **W** графа **G**, равно минимальному числу дуг ***VW****-разделяющего* множества.

Назовем ***разрезом*** такое разделяющее множество, никакое собственное подмножество которого не является разделяющим. В силу минимальности этого **VW**-разделяющего множества оно может содержать только одну дугу, принадлежащая всякой простой цепи из **V** в **W**.

Теорема 2 по существу утверждает, что задача поиска числа непересекающихся по дугам цепей эквивалентна задаче определения мощности минимального разреза.

Практический способ определения минимального разреза дает теорема Форда-Фалкерсона. Для ее формулировки введем понятие сети, которое возникло в связи с рассмотрением транспортных сетей.

Сеть можно определить как граф или орграф **D=(N,R)**, рассматриваемый вместе с функцией **b**, приписывающей каждой дуге $(\mathbf{r_i}, \mathbf{r_j}) \subset \mathbf{R}$ некоторое положительное число $\mathbf{b_{ij}}$, называемое пропускной способностью.

Вершины сети (их множество **N**) принято называть узлами.

По каждой дуге $(\mathbf{r_i}, \mathbf{rj})$ сети **D** может протекать поток $\mathbf{x_{ij}}$, причем $\mathbf{0<x_{ij}<b_{ij}}$, для всех i,j.

$$\sum_i x_{ij} - \sum_i x_{ji} = 0, \forall j \neq N_S \text{ и } N_T, \qquad (1)$$

где узлы $\mathbf{N_S}$ и $\mathbf{N_T}$ называют полюсами или *источником* и *стоком* соответственно. Поток втекает в сеть через источник и вытекает из сети через сток. Величина потока через двухполюсную сеть с полюсами $\mathbf{N_S}$ и $\mathbf{N_T}$ равна:

$$\sum_j x_{Sj} = \sum_j x_{jT}. \qquad \textbf{(2)}$$

Пропускной способностью разреза назовем сумму пропускных способностей принадлежащих ему дуг. Имеет место замечательный результат, называемый теоремой о максимальном потоке и минимальном разрезе.

ТЕОРЕМА 3 (Форда-Фалкерсона). Если в сети существует цепь, идущая из $\mathbf{N_S}$ в $\mathbf{N_T}$, то максимальный поток из $\mathbf{N_S}$ в $\mathbf{N_T}$ равен минимальной пропускной способности разрезов, разделяющих узлы $\mathbf{N_S}$ и $\mathbf{N_T}$.

Если теперь положить пропускную способность каждой дуги равной единице, то пропускная способность минимального разреза даст число независимых по дугам цепей.

Для решения задачи определения числа вершинно-независимых цепей достаточно каждый узел (кроме источника и стока) заменить парой узлов и соединяющей их дугой таким образом, как показано на рис.1.

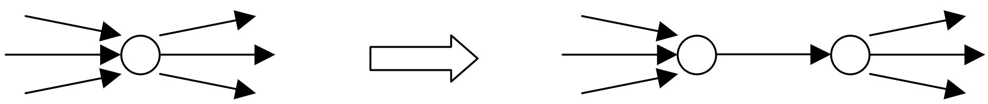

Рис. 1

Пропускная способность полученной сети (с единичными пропускными способностями дуг) даст число независимых цепей.

1.4. Нахождение максимального потока в двухполюсной сети

Рассмотрим теперь алгоритм **Форда-Фалкерсона**, предназначенный для нахождения максимального потока и минимального разреза в сети.

Алгоритм начинается с произвольного потока из источника $\mathbf{N_S}$ в сток $\mathbf{N_T}$. Алгоритм состоит в систематическом поиске всех возможных путей их $\mathbf{N_S}$ в $\mathbf{N_T}$, увеличивающих поток (процесс расстановки пометок), и в соответствующем увеличении потока (изменение потока).

ШАГ 1. Процесс расстановки пометок. На шаге 1 каждый узел находится в одном из трех состояний:

– "не помечен",

– "помечен и не просмотрен",

– "помечен и просмотрен".

Вначале все узлы не помечены. Пометка произвольного узла всегда состоит из двух частей. Первая часть – индекс узла $\mathbf{N_i}$, который указывает, что можно послать поток из $\mathbf{N_i}$ в $\mathbf{N_j}$. Вторая часть пометки – число, указывающее максимальную величину потока, который можно послать из источника $\mathbf{N_S}$ в $\mathbf{N_j}$, не нарушая ограничений на пропускные способности дуг. Прежде всего источник $\mathbf{N_S}$ получает пометку (**S+, E(s)=** ∞). Первая часть этой пометки означает, что можно послать поток из узла $\mathbf{N_S}$ в этот узел; символ ∞ означает, что величина потока не ограничена сверху. Теперь узел $\mathbf{N_S}$ "помечен и не просмотрен". А все остальные узлы "не помечены".

Выберем любой помеченный и не просмотренный узел $\mathbf{N_j}$. Пусть он имеет пометку (**i+, E(j)**) или (**i–, E(j)**).

Два узла будем называть соседними, если они соединены дугой. Из всех узлов, соседних с $\mathbf{N_j}$, выделим те узлы $\mathbf{N_k}$, которые не помечены и для которых $\mathbf{x_{jk}} < \mathbf{b_{jk}}$.

Припишем каждому узлу $\mathbf{N_k}$ пометку (**j+, E(k)**), где $\mathbf{E(k)=min\ \{\ E(j),\ b_{jk} - x_{jk}\ \}}$.

Такие узлы $\mathbf{N_k}$ теперь "помечены и не просмотрены". После этого всем соседним с $\mathbf{N_j}$ узлам $\mathbf{N_k}$, которые не помечены и для которых $\mathbf{x_{kj}} > 0$, приписываем пометку (**j–, E(k)**), где $\mathbf{E(k)=min\ \{\ E(j),\ x_{kj}\ \}}$.

Такие узлы $\mathbf{N_k}$ теперь также "помечены и не просмотрены". Сейчас все узлы, соседние с $\mathbf{N_j}$, имеют пометки. Тогда узел $\mathbf{N_j}$ считается помеченным и просмотренным, и его можно больше не рассматривать на этом шаге.

Может оказаться, что некоторые соседние с $\mathbf{N_j}$ узлы помечены, а остальные не могут быть помечены (либо все соседние с $\mathbf{N_j}$ узлы не могут быть помечены). В таких случаях узел $\mathbf{N_j}$ также считается помеченным и просмотренным. Знаки "+" и "–" в первой части пометок указывают, как должен меняться поток на шаге 2.

Продолжим приписывать пометки узлам, которые являются соседними для помеченных и не просмотренных узлов до тех пор, пока либо узел $\mathbf{N_T}$ окажется помеченным, либо нельзя будет больше пометить ни один узел и сток $\mathbf{N_T}$ окажется непомеченным. Если $\mathbf{N_T}$ не может быть помечен, то не существует пути из $\mathbf{N_S}$ в $\mathbf{N_T}$, увеличивающего поток, и, следовательно, полученный поток максимален. Если же $\mathbf{N_T}$ помечен, то на шаге 2 можно найти путь, увеличивающий поток.

ШАГ 2. Изменение потока. Предположим, что сток $\mathbf{N_T}$ имеет пометку (**k+, E(t)**). Тогда заменим $\mathbf{x_{kt}}$ на ($\mathbf{x_{kt} + E(t)}$). Если же он имеет пометку (**k–, E(t)**), то $\mathbf{x_{kt}}$ заменим на ($\mathbf{x_{kt}} - \mathrm{E}\mathbf{(t)}$). Затем в любом из этих случаев переходим к узлу $\mathbf{N_k}$. Вообше, если узел $\mathbf{N_k}$ имеет пометку (**j+, E(k)**), то $\mathbf{x_{jk}}$ заменим на ($\mathbf{x_{jk} + E(k)}$), и перейдем к узлу $\mathbf{N_j}$. Если узел $\mathbf{N_k}$ имеет пометку (**j–, E(k)**), то $\mathbf{x_{kj}}$ заменим на ($\mathbf{x_{kj} - E(k)}$) и перейдем к узлу $\mathbf{N_j}$. После этого сотрем все старые пометки узлов и вновь перейдем к шагу 1.

1.5. Оптимизация структуры сети с использованием X-трансформаций Стейглица

Рассмотрим теперь непосредственно один из алгоритмов оптимизации структуры распределенной вычислительной сети. Этот алгоритм использует в качестве критерия стоимость, а ограничения представлены указанием числа непересекающихся по дугам цепей между парами узлов сети. Алгоритм применим к сетям с дуплексными каналами передачи данных.

Алгоритм начинает работу с генерации полного графа и затем последовательно пытается выбросить из него ребро за ребром, выбирая их с помощью датчика случайных чисел. Если при этом требования по связности и числу реберно независимых путей удовлетворяются, то ребро удаляется. Если подряд определенное число таких попыток безуспешны, то получена базовая структура и алгоритм переходит к следующему этапу оптимизации структуры сети.

Этот этап состоит в последовательном применении так называемых X-трансформаций Стейглица.

Суть X-трансформаций состоит в следующем. Выбираем два ребра ($\mathbf{v_i, v_j}$) и ($\mathbf{v_k, v_m}$) такие, что ребра ($\mathbf{v_i, v_m}$) и ($\mathbf{v_k, v_j}$) отсутствуют в исходном графе. Если стоимость последней пары ребер удовлетворяет неравенству $\mathbf{w_{im}+w_{kj} < w_{ij}+w_{km}}$, то пара ребер ($\mathbf{v_i, v_j}$) и ($\mathbf{v_k, v_m}$) может быть заменена в графе на ($\mathbf{v_i, v_m}$) и ($\mathbf{v_k, v_j}$) соответственно. Такая замена (X-трансформация графа) выполняется, если полученный граф связан и удовлетворяет требованиям по числу реберно независимых цепей. Алгоритм заканчивается, когда к графу не может быть применена ни одна X-трансформация. Таким образом, полученная структура является локально оптимальной по отношению к X-трансформациям.

После нахождения локально оптимальной структуры алгоритм может быть повторен. Это целесообразно в том случае, если используется другая реализация (или другой отрезок реализации) последовательности случайных чисел. Тогда получающаяся базовая структура в общем случае будет отличаться от прежней и в результате будет найдена новая локально оптимальная структура.

Окончательный выбор среди локально оптимальных структур выполняется по критерию стоимости.

2. УЧЕБНО-ИССЛЕДОВАТЕЛЬСКАЯ РАБОТА N1

"Оптимизация структуры распределенных вычислительных сетей"

ЦЕЛЬ работы: изучение алгоритмов оптимизации на графах и их применение.

2.1. Методические указания к оптимизации структуры распределенных вычислительных сетей

Программы, реализующие алгоритм оптимизации на IBM PC, называются ARTNET.

Стоимость аренды каналов передачи данных в зависимости от их протяженности определяются в программах в соответствии с тарифами США по состоянию на 1977г. Стоимость арендуемых дуплексных каналов тональной частоты составляет ($ /мес./ миля):

– первые 25 миль – 3.3;

– следующие 150 миль – 1.65.

Для удобства работы программы имеют систему меню и подсказок.

2.2. Пример использования алгоритма Форда-Фалкерсона

Вторым пунктом выполнения лабораторной работы является определение числа реберно-независимых цепей между парой вершин графа с помощью алгоритма Форда-Фалкерсона.

Рассмотрим применение этого алгоритма к сети, изображенной на рис.2.

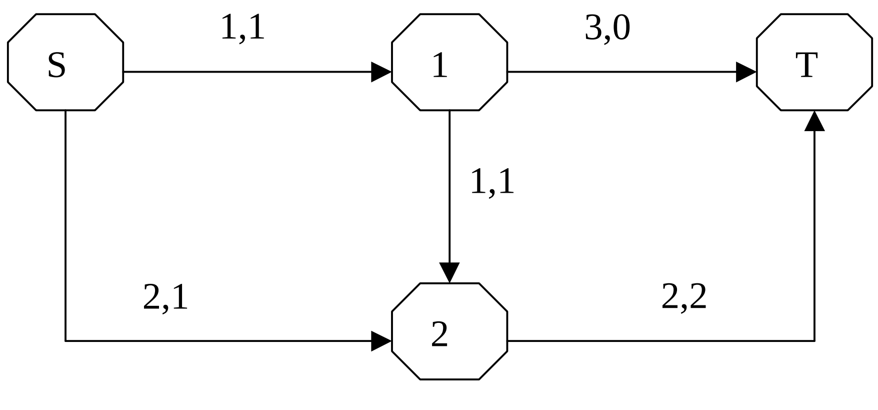

Рис. 2

Здесь каждой ориентированной дуге поставлено в соответствии два числа. Первое из них – пропускная способность дуги, а второе – исходный поток по дуге (в качестве исходного потока можно использовать любые числа, удовлетворяющие условиям (1) и (2). В частности, их можно взять нулевыми).

ШАГ 1. Припишем узлу $\mathbf{N_S}$ пометку ($\mathbf{S+, \infty}$), где ∞ – бесконечность. Узел $\mathbf{N_S}$ имеет два соседних узла. Пометить $\mathbf{N_1}$ мы не можем, так как $\mathbf{b_{S1}–x_{S1}=0}$ и нет дугового потока $\mathbf{x_{1S}>0}$. Узлу $\mathbf{N_2}$ припишем пометку ($\mathbf{S+,1}$), поскольку $\mathbf{b_{S2}–x_{S2}=1}$ и $\mathbf{E(2)} = \min \mathbf{(E(S), b_{S2}–x_{S2})} = \min(\infty, \mathbf{1}) = \mathbf{1}$.

Теперь узел $\mathbf{N_S}$ помечен и просмотрен. А узел $\mathbf{N_2}$ помечен и не просмотрен. Узел $\mathbf{N_2}$ имеет два непомеченных соседних узла, а именно $\mathbf{N_1}$ и $\mathbf{N_T}$. Узел $\mathbf{N_T}$ в данный момент не может быть помечен, а узел $\mathbf{N_1}$получает пометку (2–,1), поскольку $\mathbf{x_{12}=1 > 0}$ и $E(1)=\min(\mathbf{E(2), x_{12}})=1$.

Теперь узел $\mathbf{N_2}$ помечен и просмотрен, а узел $\mathbf{N_1}$ помечен и не просмотрен. Узел $\mathbf{N_1}$ имеет только один соседний непомеченный узел, а именно $\mathbf{N_T}$. Узел $\mathbf{N_T}$ должен получить пометку (1+,1), поскольку $\mathbf{E(T) = \min (E(1), b_{1T}–x_{1T})} = 1$.

Результат этой расстановки пометок изображен на рис.3.

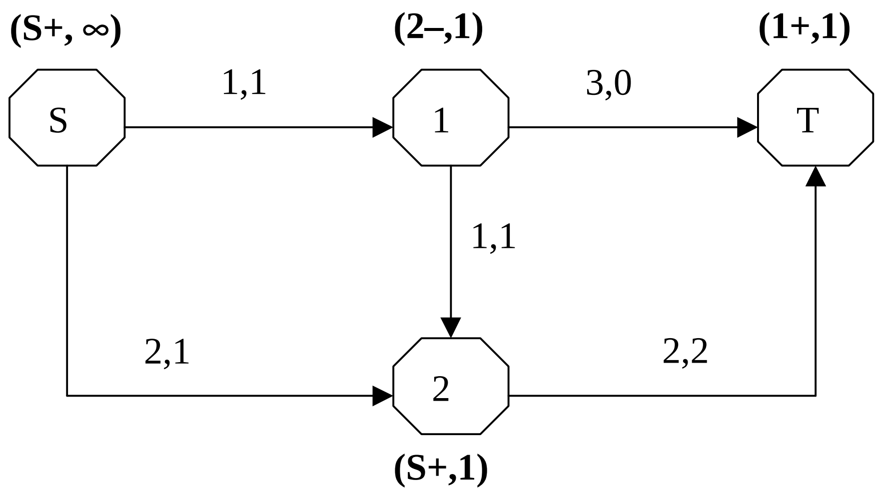

Рис. 3

Так как узел $\mathbf{N_T}$ оказывается помеченным, то переходим к шагу 2.

ШАГ 2. Поскольку пометка узела $\mathbf{N_T}$ есть (1+,1), то увеличиваем $\mathbf{x_{1T}}$ на 1. В результате получаем $\mathbf{x'_{1T}=x_{1T}+1=1}$.

Переходим к узлу $\mathbf{N_1}$, имеющему пометку (2–,1), и уменьшаем $\mathbf{x_{12}}$ на 1. Получаем $\mathbf{x'_{12}=x_{12}–1=0}$.

Переходим к узлу $\mathbf{N_2}$, имеющему пометку (S+,1), и прибавляем 1 к $\mathbf{x_{S2}}$. Получаем $\mathbf{x'_{2S}=x_{2S}+1=2}$.

Окончательный результат операции изменения потока изображен на рис.4.

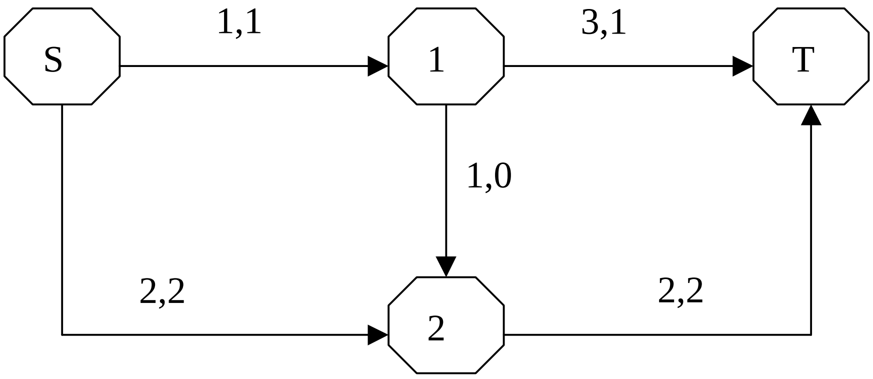

Рис. 4

ШАГ 3. Приписываем пометку (**S**+, ∞).

Теперь узлы $\mathbf{N_1}$ и $\mathbf{N_2}$ не могут быть помечены, и узел $\mathbf{N_T}$ оказывается непомеченным.

Конец.

ВЫВОДЫ: С помощью алгоритма Форда-Фалкерсона между парой вершин графа S и T найдено две реберно-независимых цепи:

1. (S-1), (1-T);
2. (S-2), (2-T).

ST-разделяющим множеством является множество дуг: A={(S-1), (1-T), (1-2), (S-2), (2-T)}.
ST-отделяющим множеством является множество вершин: A={1, 2}.

Любой из возможных вариантов разреза в данном случае является минимальным. Пример разреза: (S-1), (2-T) с пропускной способностью, равной трем.

2.3. Описание лабораторной установки и требования безопасности

Описание лабораторной установки и требования безопасности приведено в соответствующих разделах /2/.

2.4. Порядок выполнения работы

1. Получить у преподавателя вариант задания с номером **i**, содержащий:

– число узлов коммутации проектируемой сети **N**=7+**i**(mod 4);

– матрицу ||**A**||, содержащую требование на нижнюю границу числа $\mathbf{A_{km}}$ реберно-независимых цепей между заданными парами узлов коммутации

$$\mathbf{A_{km} = max\ \{|(k+i)(mod\ 4) - (m+i)(mod\ 4)|,\ i2\ (mod\ 4),\ 1\}}\ ,\ \forall k, m = \overline{1, N}\ .$$

– матрицу ||**D**||, содержащую расстояния (в километрах) между узлами коммутации

$$\mathbf{D_{km} = 5[(k+m)(mod\ 5)] + 20(i)(mod\ 6) + i2(mod\ 10);}$$

– граф **G=(V,R)** с **N** вершинами и множеством ребер

$$R = \bigcup_{j=0}^{N-1}(v_j, v_{(j+1)}) \cup \bigcup_{j=0}^{N+1}(v_j, v_{(j+i)(\mathrm{mod}\ N)})\text{ , где } v_j \in V, \forall j = \overline{0, N-1}.$$

2. Используя алгоритм Форда-Фалкерсона, определить число реберно-независимых цепей между вершинами графа G с номерами **i** (mod **N**) и $\mathbf{(i+1)^2}$ (mod **N**).
3. Для заданного варианта в соответствии с определениями построить:

- **VW**-отделяющее множество,
- **VW**-разделяющее множество,
- разрез,
- диаметр.

4. Сдать коллоквиум преподавателю.
5. Для заданных исходных данных получить на ПЭВМ IBM PC три локально оптимальные структуры распределенной вычислительной сети.
6. Сделать выводы по результатам выполненной работы (сравнить по стоимости локально оптимальные структуры, а также базовые и локально оптимальные структуры, проверить правильность соблюдения условий при выполнении X-трансформаций для одной из локально оптимальных структур).
7. Сдать отчет преподавателю.

2.5. Содержание отчета

1. Формулировка задания.
2. Описание методов оптимизации структур распределенных вычислительных сетей.
3. Краткое описание используемого алгоритма оптимизации.
4. Описание с детализацией по шагам применения алгоритма Форда-Фалкерсона для нахождения числа реберно-независимых цепей.
5. Анализ результатов исследования заданного графа сети (**VW**-отделяющее множество, **VW**-разделяющее множество, разрез, диаметр).
6. Результаты выполнения X-трансформаций (стоимость полносвязной структуры, базовые и локально оптимальные структуры и их стоимость).
7. Выводы.
8. Литература.

2.6. Контрольные вопросы

1. Каковы критерий и ограничения при оптимизации структуры коммерческой вычислительной сети общего пользования?
2. Каковы критерий и ограничения при оптимизации структуры специализированной вычислительной?
3. В чем отличие задач оптимизации структуры абонентской и межцентровой сети?
4. Чем отличаются маршрут (путь), цепь и простая цепь?
5. Чем отличаются сильно связный граф от слабо связного?
6. Что называется диаметром графа?
7. Что называется разделяющим множеством?
8. Какое множество называется отделяющим?
9. В чем суть вершинной формы теоремы Менгера?
10. В чем суть реберной (дуговой) формы теоремы Менгера?
11. Как определяется двухполюсная сеть?
12. Как определяется поток и величина потока в двухполюсной сети?
13. В чем суть теоремы Форда-Фалкерсона?
14. Каковы основные этапы используемого алгоритма оптимизации структуры распределенной вычислительной сети?
15. Каковы основные шаги алгоритма Форда-Фалкерсона?
16. В чем суть первого шага алгоритма Форда-Фалкерсона?
17. В чем суть второго шага алгоритма Форда-Фалкерсона?
18. Каковы основные условия для выполнения X-трансформации?

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Чуркин В.И. Проектирование вычислительных сетей: учебное пособие / С.-П., СПИАП. 1992. 86 с.
2. Дудник В.М. и др. Введение в программирование: методические указания к выполнению лабораторных работ / Л., ЛИАП. 1983. 60 с.

СОДЕРЖАНИЕ