ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ

Сборник трудов
1962 г. Института математики СО АН СССР Выпуск 3

О ЗАДАЧЕ СОЕДИНЕНИЯ ЭЛЕМЕНТОВ ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ

Ю.Г. Решетняк

Как показали Э.В.Евреинов и Ю.Г.Косарев [I], единственно реальным путем создания электронных вычислительных машин производительностью более миллиарда операций в секунду в настоящее время является путь построения вычислительных систем. Под вычислительной системой (ВС) понимается устройство,состоящее из большого числа однотипных частей - элементарных машин (ЭМ), каждая из которых представляет собой самостоятельную электронную вычислительную машину.

В процессе работы ВС из одних ее элементов в другие должны передаваться различные сообщения. Для этой цели элементарные машины, входящие в ВС, необходимо соединить между собой. Возникает задача нахождения оптимального способа соединения. При этом должны учитываться требования двоякого рода - математические (ВС должна допускать реализацию достаточно большого класса алгоритмов) и технические.

Требования эти противоречивы, и само по себе каждое из них имеет в значительной мере неопределенный характер. Поэтому, для того чтобы искать оптимальное соединение элементов ВС, необходимо, прежде всего, дать точный критерий оптимальности.

Цель настоящей статьи состоит в том, чтобы указать некоторые критерии такого рода. Этому посвящены § I и 2.

В заключение приводятся некоторые примеры и гипотезы относительно способов соединения, оптимальных в смысле, определяемом ниже.

§ I. Понятие m -кратной коммутации

Пусть M – произвольная вычислительная система, содержащая N отдельных вычислительных машин. Будем считать, что все машины, составляющие систему M, определенным образом занумерованы. Машину с номером i обозначим через m_i, $(i = 1,2,\dots, N)$.

Предположим, что выделены некоторые машины $m_{i_1}, m_{i_2},\dots, m_{i_k}$, где $k \leq N$, машине m_{i_s} сопоставлено некоторое множество машин A_s, $s = 1,2,\dots, k$.

Допустим, что для каждого s в машине m_{i_s} вырабатывается некоторый сигнал σ_s, который передается во все машины множества A_s. В случае, если такая ситуация имеет место, мы будем говорить, что в вычислительной системе задана некоторая коммутация. Машину m_{i_s} мы будем называть машиной, управляющей множеством A_s.

Дадим общее определение. Пусть S – произвольное конечное множество, состоящее из $N>0$ элементов. Для простоты будем считать, что S совпадает с отрезком $\{1,2,\dots, N\}$ натурального ряда. Коммутацией в множестве S будем называть некоторое, вообще говоря, многозначное отображение $f(x)$ его подмножества S' в множество S. $f(x)$ здесь означает совокупность всех элементов, отвечающих элементу x. Мы будем говорить, что элемент x управляет множеством $f(x)$.

Задание коммутации в вычислительной системе, очевидно, равносильно заданию коммутации на множестве, образованном номерами машин, составляющих ВС. При этом S' есть, очевидно, множество номеров $i_1, i_2,\dots, i_k$, $f(i_s)$ есть множество номеров машин, входящих в A_s.

Коммутация $f(x)$ называется простой или однократной, если при $x \neq y$ множества $f(x)$ и $f(y)$ не имеют общих элементов, $f(x) \cap f(y) = \Lambda$. Коммутация $f(x)$ называется m-кратной, если никакой элемент множества S не может принадлежать более чем m различным множествам $f(x)$, причем существует элемент, принадлежащий в точности m множествам.

В случае, если мы имеем дело с вычислительной системой, то сформулированные понятия имеют простой смысл. Коммутация простая, если в каждую машину системы может поступить сигнал самое большее из одной машины. Коммутация m -кратная, если в каждую машину поступает не более m сигналов из остальных машин системы, причем есть машина, в которую поступает в точности m сигналов.

Каждый алгоритм, допускающий реализацию на электронной вычислительной машине, как хорошо известно, сводится к последовательному вычислению значений некоторых простейших логических функций, каждая из которых зависит от ограниченного числа аргументов. В качестве таких функций можно брать, например, функции – отрицание, $f(x) = \bar{x}$, дизъюнкцию $f(x_1, x_2) = x_1 \vee x_2$ и конъюнкцию $f(x_1, x_2) = x_1 \,\&\, x_2$.

Сведение всех алгоритмов непосредственно к указанным трем функциям нерационально, с точки зрения программирования, и обычно для каждой машины устанавливается некоторый более богатый набор простейших функций – операций машины. При этом важно заметить, что эти простейшие функции представляют собой в современных машинах, как правило, функции одного или двух (редко трех) аргументов.

Предположим, что рассматриваемая вычислительная система однородна в том смысле, что набор простейших стандартных операций один и тот же для всех машин, составляющих вычислительную систему. Пусть при реализации некоторого алгоритма в данный момент времени требуется вычислить величины:

$$f_1(x_1),\ f_2(x_2),\dots, f_k(x_k), F_1(y_1, z_1) F_2(y_2, z_2),\dots, F_p(y_p, z_p)$$

где $k+p \leq N$ (N – общее число машин в ВС). Для этого необходимо выделить отдельные машины, составляющие ВС, и заслать в них соответствующие значения аргументов. Так как в каждую машину должно поступить не более двух аргументов, то, очевидно, она должна быть связана самое большее с двумя машинами вычислительной системы. Поэтому связи, которые должны быть осуществлены для реализации данного этапа вычислительного алгоритма, имеют характер двухкратной коммутации.

Если процесс счета организован так, что для каждого k один из двух аргументов y_k и z_k уже хранится в той машине, которой поручено вычисление $F_k(x_k, y_k)$, то можно при реализации данного вычислительного алгоритма ограничиться однократными коммутациями.

Если в число операций машины включены операции, представляющие собой функции трех и более переменных, то для осуществления каждого этапа вычислительного алгоритма необходимо уметь быстро осуществлять трех-, соответственно, четырех- и более кратные коммутации.

Из сказанного ясно, таким образом, что для удовлетворительного функционирования вычислительной системы достаточно иметь такую систему связей, которая позволила бы легко осуществлять различные коммутации какой-либо определенной кратности, обычно не слишком большой. В этом и состоят математические требования к вычислительной системе.

В заключение настоящего параграфа приведем некоторые подсчеты общего числа коммутаций различной кратности. Эти подсчеты представляют интерес в том отношении, что они позволяют делать некоторые суждения относительно предполагаемой системы связей, удовлетворяющей требованиям реализуемости различных коммутаций.

Пусть $f(x)$ - некоторая коммутация кратности m. Обозначим через S' область, где эта коммутация определена. Для произвольного $z \in S$ обозначим через $f^{-1}(z)$ совокупность всех $x \in S'$, таких, что $z \in f(x)$. Множество $f^{-1}(z)$ состоит не более чем из m элементов и, в частности, может быть пустым. Отображение $f^{-1}(z)$ назовем обратным к $f(x)$. Таким образом, отображение, обратное m -кратной коммутации, есть многозначное отображение, при котором образ каждого элемента состоит, самое большее, из m элементов. Ясно, что при этом существует z, для которого $f^{-1}(z)$ состоит в точности из m элементов.

Обратно, пусть задано многозначное отображение $g(z)$, при котором $g(z)$ состоит не более чем из m элементов (в частности, $g(z)$ может быть пусто); предполагается, однако, что $g(z) \neq \Lambda$, хотя бы для одного z. Для произвольного $x \in S$ пусть $f(x)$ означает совокупность всех z, для которых $x \in g(z)$. Тогда $f(x)$ представляет собой коммутацию кратности не большей m. Действительно, $z \in f(x)$ в том и только в том случае, если $x \in g(z)$. Но $g(z)$ содержит не более m элементов и, значит, z принадлежит не более чем m из множеств $f(x)$. Отображение, обратное коммутации $f(x)$, есть как раз отображение $g(z)$. Действительно, пусть $x \in f^{-1}(z)$. Тогда $z \in f(x)$ и, значит, по определению, $x \in g(z)$. Значит, $f^{-1}(z) \in g(z)$. Обратно, пусть $x \in g(z)$. Тогда $z \in f(x)$ и, значит, $x \in f^{-1}(z)$, откуда $f^{-1}(z) \supset g(z)$. Следовательно,

$$f^{-1}(z) = g(z).$$

Нетрудно видеть, что если для некоторого z $g(z)$ состоит из m элементов, то $f(z)$ есть коммутация кратности m.

Таким образом, между множеством коммутаций кратности не большей m и m -значными отображениями, то есть отображениями, при которых образ каждого элемента состоит не более чем из m элементов, существует взаимно однозначное соответствие. Число m -значных отображений легко подсчитать.

Число подмножеств множества S, состоящих не более чем из m элементов, равно:

$$R_{m,N} = 1 + C_N^1 + C_N^2 + \ldots + C_N^m ,$$

где

$$C_N^K = \frac{N(N-1)\ldots(N-K+1)}{K!} .$$

Сопоставляя каждому $x \in S$ произвольное из этих подмножеств, мы получим некоторое m -значное отображение. Общее число m -значных отображений поэтому равно:

$$\Gamma_{m,N} = (R_{m,N})^N = \Big(\sum_{K=0} C_N^K\Big)^N .$$

На основании сказанного $\Gamma_{m,N}$ равно также общему числу всех коммутаций кратности не выше m. Число коммутаций кратности в точности, равной m, есть $\Gamma_{m,N} - \Gamma_{m-1,N}$. Из формальных соображений целесообразно ввести коммутацию кратности 0 - это коммутация, для которой обратное отображение $f^{-1}(z)$ таково, что $f^{-1}(z)$ пусто при всех z. Число простых коммутаций, в частности, оказывается равным

$$\Gamma_{1,N} - \Gamma_{0,N} = (N+1)^N - 1 .$$

Укажем простую асимптотическую формулу для числа $\Gamma_{m,N}$, справедливую при фиксированном m и достаточно большом N.

Имеем:

$$\Gamma_{m,N} = (R_{m,N})^N = \Big(1 + N + \frac{N(N-1)}{2!} + \ldots + \frac{N(N-1)\ldots(N-m+1)}{m!}\Big)^n =$$

$$= \frac{N^{mN}}{(m!)^N}\Big[\Big(1-\frac{1}{N}\Big)\Big(1-\frac{2}{N}\Big)\ldots\Big(1-\frac{m-1}{N}\Big) + \frac{m}{N}\Big(1-\frac{1}{N}\Big)\ldots\Big(1-\frac{m-2}{N}\Big) + \ldots$$

$$+ \frac{m!}{N^{m-1}} + \frac{m!}{N^m}\Big]^N = \frac{N^{mN}}{(m!)^N}\Big[1 - \frac{1+2\ldots+(m-1)-m}{N} + \frac{a}{N^2} + \ldots\Big]^N ,$$

$a = const.$

Отсюда

$$\Gamma_{m,N} = \frac{N^{mN}}{(m!)^N} = \Big[1 - \frac{m^2-3m}{2N} + \frac{a}{N^2} + \ldots\Big]^N =$$

$$= \frac{N^{mN}}{(m!)^N} e^{-\frac{m^2-3m}{2}} \Big[1 + 0\Big(\frac{1}{N}\Big)\Big] ,$$

и окончательно

$$\Gamma_{m,N} = \frac{N^{mN}}{(m!)^N} e^{-\frac{m^2-3m}{2}} \left[1+O\left(\frac{1}{N}\right)\right]. \qquad (2)$$

§ 2. Информационный граф вычислительной системы

Элементарные машины, из которых составляется вычислительная система, соединяются посредством каналов связи. Мы будем предполагать каналы связи ориентированными, то есть допускающими передачу информации лишь в одном определенном направлении. Каждый канал связи имеет ровно два конца, из которых один соединен с одной элементарной машиной, второй - с другой. Один из концов канала связи называется входом, другой - выходом. Мы предполагаем, что информация передается всегда в направлении от входа к выходу.

Совокупность всех каналов связи образует некоторый ориентированный граф, вершинами которого служат элементарные машины, составляющие вычислительную систему, а ребрами являются каналы связи. Этот граф мы будем называть информационным графом вычислительной системы.

Для произвольной машины m_i обозначим через K_i^+ совокупность всех каналов связи, подключенных к машине m_i своим выходом. По этим каналам информация поступает в машину m_i. Через K_i^- обозначим совокупность всех каналов связи, подключенных к m_i своим входом. По этим каналам связи происходит передача информации из машины m_i. Число каналов связи, принадлежащих множествам K_i^+ и K_i^-, обозначим через k_i^+ и k_i^-, соответственно. Нетрудно видеть, что имеет место равенство:

$$\sum_{i=1}^{N} k_i^+ = \sum_{i=1}^{N} k_i^- = R ,$$

где R - общее число каналов связи.

Передача информации из машины m_i вычислительной системы в машину m_j может осуществляться следующим образом. Найдем цепочку каналов связи $\ell_1, \ell_2, \ldots, \ell_k$, такую, что каналы ℓ_j и ℓ_{j+1} при $j = 1,2,\ldots,k-1$ исходят из одной ЭМ, причем один из них присоединен к этой ЭМ своим входом, а другой - своим выходом. При этом вход канала ℓ подключен к ЭМ m_i, а выход канала ℓ_k - к ЭМ m_j. Существование такой цепочки каналов связи есть необходимое условие для того,

чтобы какая-либо информация могла быть передана из ЭМ m_i в ЭМ m_j. Процесс передачи можно мыслить происходящим двумя возможными способами.

Во-первых, по каналу ℓ_1 сигнал из ЭМ m_i передается в элементарную машину m_{i_1}, к которой подключен выход канала ℓ_1. В ЭМ m_{i_1} этот сигнал задерживается на некоторое время, после чего подается на вход канала ℓ_2. По каналу ℓ_2 сигнал поступает в следующую ЭМ m_{i_2} и после некоторой задержки передается на вход канала ℓ_3 и т.д. В конце концов сигнал поступит в ЭМ m_j.

Задержка на каждом этапе описываемого процесса связана с необходимостью некоторой обработки передаваемого сигнала, которая может состоять, например, в прочтении адресной части сигнала с целью выяснить, на вход какого канала связи следует подать этот сигнал, в передаче сигнала из одного регистра ЭМ в другой и т.п. Требуемая цепочка каналов связи здесь находится непосредственно в процессе передачи информации.

Во-вторых, можно представить себе способ передачи, при котором выход канала ℓ_s непосредственно соединен со входом канала ℓ_{s+1}, и передаваемый сигнал проходит по цепочке каналов связи, не задерживаясь в ЭМ m_{i_s}. В этом случае вся цепочка каналов связи работает как единый канал связи. Задержка сигналов на промежуточных передаточных пунктах не происходит. Путь, по которому передается сигнал, в данном случае определяется заранее. Соединение отдельных каналов связи в цепочку производится по управляющим сигналам, образуемым либо в каждой отдельной ЭМ, через которую проходит линия связи, либо в некотором управляющем устройстве, общем для всей вычислительной системы. В том случае, когда время, необходимое для выработки этих управляющих сигналов и приведения линии связи в готовность, мало по сравнению с суммарным временем задержки при первом способе, более выгодным является данный второй способ соединения.

Для краткости мы будем говорить, что в первом случае происходит передача сигнала с задержками, а во втором - прямая передача сигнала.

Представим теперь, что сигнал σ из ЭМ m_i должен быть передан нескольким элементарным машинам $m_{j_1}, m_{j_2}, \ldots, m_{j_K}$. В этом случае процесс передачи можно мыслить происходящим следующими способами.

Предположим, что существует цепочка каналов связи, такая, что элементарные машины $m_{j_1}, m_{j_2}, \ldots, m_{j_k}$ являются концами некоторых из каналов связи цепочки. В случае, если сигнал подается по этой цепочке с задержками, сигнал σ автоматически будет получен ЭМ $m_{j_1}, m_{j_2}, \ldots, m_{j_k}$. При прямой передаче сигнала по цепочке он также может быть получен каждой из элементарных машин $m_{j_1}, m_{j_2}, \ldots, m_{j_k}$, если в них предусмотрены специальные устройства для считывания сигналов, проходящих по цепочке каналов связи.

При прямой передаче сигнала в некоторых случаях может оказаться целесообразно вместо цепочки использовать систему каналов связи, соединенных друг с другом таким образом, что они образуют дерево, то есть линию связи с разветвлениями (см., например, рис.I). При этом в точках ветвления линии связи,вообще говоря, необходимо усиление сигнала.Это должно быть предусмотрено в конструкции ВС.

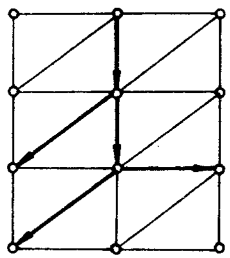

Рис. I

Предположим теперь, что в вычислительной системе задана некоторая коммутация $f(x)$. Пусть $f(x)$ определено для $x = i_1, i_2, \ldots, i_k$. Положим, $f(i_s) = A_s$. Допустим, что из m_{i_s} передается сигнал σ_s. Для того, чтобы этот сигнал был принят машинами множества A_s, мы должны построить линию связи L_s, исходящую из ЭМ m_{i_s} и соединяющую ее со всеми ЭМ множества A_s. Построим соответствующие линии связи L_s. Передача сигналов из управляющих ЭМ m_{i_s} будет возможна по этим линиям L_s, очевидно, в том и только в том случае, если никакой канал связи не входит одновременно в две линии L_{s_1} и L_{s_2}. В противном случае будет, очевидно, происходить смешение различных сигналов, и передача информации из управляющих машин станет невозможной.

Для реализации передачи сигналов, соответствующих произвольной коммутации, посредством информационного графа мы должны прежде всего иметь в каждой ЭМ m_i устройство, которое обеспечивает возможность соединения любого выхода каналов группы K_i^+ с любым входом группы K_i^-. Сложность этого устройства (назовем его коммутационным устройством) зависит от значений чисел K_i^+ и K_i^-. Если эти числа велики,то указанное устройство будет весьма сложным. Например, если k_i^+ и k_i^- одного порядка с общим числом N элементарных машин, то коммутационное устройство ЭМ m_i по сложности сравнимо с информационным графом всей ВС. Ясно, что такая ситуация является неприемлемой.

Заметим здесь, что при построении коммутационного устройства каждой отдельной ЭМ могут быть использованы те же принципы, что и при построении информационного графа всей ВС.

Таким образом, мы естественным путем приходим к следующему требованию, которому должен удовлетворять оптимальный информационный граф вычислительной системы: для каждого i числа k_i^+ и k_i^- должны быть не слишком велики.

Дадим точную формулировку. Положим $k_i = k_i^+ + k_i^-$ и пусть $k(M) = \max k_i$. Допустим, что информационный граф вычислительной системы $i = 1, 2, \ldots, N$ устроен так, что в нем может быть реализована передача сигналов,отвечающая любой коммутации кратности не выше m. Мы будем говорить, что граф оптимален, если величина $k(m)$ в нем достигает наименьшего из возможных значений, причем ни одно ребро информационного графа не может быть устранено без того, чтобы граф не потерял того свойства, что в нем может быть реализована любая коммутация заданной кратности.

Задача построения вычислительных систем с оптимальной структурой информационного графа является, по-видимому, достаточно сложной.

§ 3. Понятие пути в информационном графе

В этом параграфе мы дадим уточненные формулировки некоторых понятий, введенных в предыдущем параграфе.

Пусть Q - произвольный ориентированный граф. Простым путем в графе Q называется всякая последовательность ребер графа $\bar{a}_1, \bar{a}_2, \ldots, \bar{a}_m$, такая, что никакое ребро не встречается в этой последовательности дважды и начало ребра a_i

совпадает с концом ребра $\bar{a}_{i-1}$ для всех $i=1,\ldots,m$.

Пусть R – произвольное частично упорядоченное конечное множество. Отношение порядка в R будем обозначать символом $>$. Говорим, что элемент $x \in R$ непосредственно предшествует y, если $x > y$ и в R не существует элемента z, такого, что $x > z > y$. Будем также говорить, что y непосредственно следует за x.

Множество R называется упорядоченным деревом, если для всякого $x \in R$ существует не более одного элемента множества R, непосредственно предшествующего x. Для всякого частично упорядоченного конечного множества можно построить ориентированный граф, полностью определяющий строение R. Для этого будем рассматривать элементы множества R как вершины графа, и каждый элемент x считать соединенным ребром со всеми теми и только теми вершинами R, которые непосредственно следуют за x. При этом ребро считаем направленным от вершины x к вершине, следующей за ней. Ясно, что задание описанного графа в свою очередь полностью определяет множество R.

Пусть R – упорядоченное дерево. Вершина $x \in R$ называется корнем дерева R, если всякий элемент y следует за x. Точка x упорядоченного дерева R называется точкой ветвления дерева, если в R существует, по крайней мере, две различные вершины y и z, следующие за x, для которых ни одно из отношений $y < z$ и $z < y$ не выполняется. Точка x называется концом упорядоченного дерева R_1, если в R нет элементов, следующих за x.

Пусть Q – информационный граф вычислительной системы, R – произвольное упорядоченное дерево. Пусть φ – отображение множества вершин R в множестве вершин графа Q. Это отображение назовем гомоморфным отображением, если выполняются следующие условия:

а) Если x, $y \in R$ – концы произвольного ребра дерева R, и $x > y$, то $\varphi(x)$ и $\varphi(y)$ различны и являются концами ребра графа Q, направленного от $f(x)$ к $\varphi(y)$. Это ребро назовем образом ребра xy при отображении φ и обозначим через $\varphi(xy)$.

б) Образы любых двух различных ребер графа R суть различные ребра графа Q.

Путем в графе Q мы будем называть совокупность L ребер графа Q, являющегося образом некоторого упорядоченного дерева R с единственным корнем при гомоморфном отображении R в Q. Вершину m_i графа Q, являющегося образом корня упорядоченного дерева R, мы будем называть началом пути L. Мы будем также говорить, что путь L исходит из вершины m_i.

В том случае, когда частично упорядоченное множество R есть множество натуральных чисел $R=\{1,2,\ldots,n\}$, понятие пути в смысле данного определения, очевидно, совпадает с понятием простого пути.

Пусть в вычислительной системе M задана произвольная коммутация $f(x)$ и пусть $L_1, L_2, \ldots, L_m$ – некоторое множество путей в информационном графе Q системы S, попарно не имеющих общих ребер. Будем говорить, что пути $L_1, L_2, \ldots, L_m$ реализуют коммутацию $f(x)$, если каждая вершина x графа Q, для которой $f(x)$ определено, является началом некоторого пути L_i, причем все вершины соответствующего множества $f(x)$ принадлежат пути L_i.

§ 4. Некоторые примеры и гипотезы

Предположим, что информационный граф ВС устроен так, что в нем может быть реализована любая простая коммутация. Тогда из него легко построить информационный граф, в котором реализуется любая m-кратная коммутация. Для этого достаточно каждое ребро графа Q заменить на m ребер той же ориентации и с теми же концами.

Необходимо при этом, однако, отметить, что если исходный граф Q был оптимальным, то построенный новый граф, по-видимому, может не быть таковым. Сравнивая энтропию множества всех коммутаций кратности не выше m с энтропией множества всех простых коммутаций, мы видим, что их отношение

$$\frac{\log_2 \Gamma_{m,N}}{\log_2 \Gamma_{1,N}} \to m$$

при $N \to \infty$. Это обстоятельство позволяет предположить, что асимптотически данный способ соединения все же близок к оптимальному.

Тривиальным примером графа, в котором может быть реализована любая простая коммутация, является так называемый ориентированный полный граф. Это граф Q, в котором любые две вершины соединены двумя (и только двумя) ребрами взаимно противоположного направления. Если общее число вершин ориентированного графа Q равно N, то число всех связей будет равно $N(N-1)$. Если N весьма велико, то число связей также будет очень большим.

Заметим, что в ориентированном полном графе может быть реализована не только любая простая, но и любая ($N-1$)-кратная коммутация, где N – число вершин графа. Действительно, пусть $f(x)$ – такая коммутация. Каждая вершина x, для которой $f(x)$ определено, соединяется с любой из вершин множества $f(x)$ двумя ребрами, из которых одно входит, а другое выходит из x. Вершине x_i сопоставим все ребра, исходящие из нее и оканчивающиеся в точках множества $f(x)$. Эти ребра образуют путь L_x, соединяющий x с вершинами множества $f(x)$.

Нетрудно видеть, что ни одно ребро не может принадлежать одновременно двум из этих путей, и, значит, они реализуют данную коммутацию.

Из сказанного видно, что соединение машин по способу полного ориентированного графа, с точки зрения функционирования вычислительной системы, является наиболее выгодным. В тех случаях, когда N невелико, по-видимому, соединение машин ВС целесообразно осуществлять именно таким образом.

В заключение настоящей статьи мы сформулируем одну гипотезу. Рассмотрим n-мерный куб, и пусть P_n означает ориентированный граф, который получается, если в графе одномерных ребер куба каждое ребро заменить двумя, имеющими взаимно-противоположные направления. На прилагаемых чертежах 2 и 3 изображены графы, получаемые таким образом в случае $n = 2$ и $n = 3$. Вершины графа здесь изображаются кружками, а направление каждого отдельного ребра указано стрелкой.

Общее число вершин графа P_n равно 2^n. Количество ребер k^-, исходящих из каждой вершины, равно числу ребер, входящих в нее, и равно n, $k = k^+ = k^- = n$. Общее число ребер графа P_n равно $2^n n$. Число k в данном случае равно $\log_2 N$, где $N = 2^n$ – число вершин графа P_n.

Наша гипотеза состоит в том, что в графе P_n любая коммутация кратности I может быть реализована и притом с использованием только простых путей.

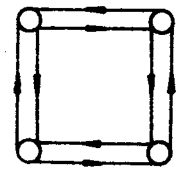

Рис. 2

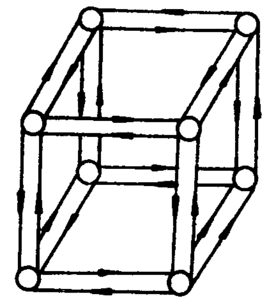

Рис. 3

Автор не располагает доказательством этой гипотезы в общем виде. Для случая $n = 2$ общее число коммутаций кратности I согласно формуле, приведенной выше, равно $5^4 = 625$, и для данного случая вопрос удается решить перебором всех возможных коммутаций. Перебор в данном случае значительно облегчается симметрией графа P_n, благодаря чему общее число случаев, которые в действительности должны быть рассмотрены, оказывается значительно меньше 625.

Для $n = 3$ и $n = 4$ нами было рассмотрено большое число отдельных частных примеров. Для этого строились случайным путем различные простые коммутации. Случаев, когда такую коммутацию было бы невозможно реализовать в графах P_3 и P_4, не встречалось.

Намечается проведение экспериментов с рассмотрением случайным образом полученных коммутаций для больших значений посредством электронной вычислительной машины Института математики СО АН СССР.

Рассмотрение отдельных примеров показывает, что наибольшие трудности при реализации коммутаций встречаются, как правило, в случае простых коммутаций вида $f(x)$, где $f(x)$ взаимно однозначное отображение. Коммутации такого рода мы будем называть перестановками. В этом случае, в каждой вершине должен заканчиваться по крайней мере один путь, из числа реализующих коммутацию, и, по крайней мере, один путь должен исходить из нее. Это обстоятельство ведет к повышенному "расходу" ребер при реализации перестановок.

Максимальный "расход" ребер достигается при перестановке, для которой вершины x и $f(x)$ максимально удалены друг от друга. Длина пути, соединяющего вершины x и $f(x)$, должна быть в этом случае не меньше n. Так как число всех путей равно 2^n, то общее количество ребер, используемых при реализации данной перестановки, не меньше $2^n n$, и поскольку $2^n n$ есть общее число ребер графа P_n, оно должно быть равно $2^n n$. Таким образом, при реализации данной коммутации используются все ребра графа.

Рассмотренная перестановка может быть реализована следующим образом. В кубе выберем n линейно независимых ребер $\bar{\ell}_1, \bar{\ell}_2, \dots, \bar{\ell}_n$, исходящих из одной вершины. Возьмем произвольную вершину x графа P_n. Из вершины x проведем сначала ребро $\ell_{x,1}$, параллельное вектору $\bar{\ell}_1$. Из конца ребра $\bar{\ell}_{x,1}$ проведем ребро $\ell_{x,1}$, параллельное $\bar{\ell}_2$, и т.д. Если построено ребро $\ell_{x,k}$, $k \neq n$, то из конца ребра $\ell_{x,k}$ мы проводим ребро $\ell_{x,k+1}$, параллельное вектору ℓ_{k+1}. При $k = n$ построение заканчивается.

Последовательность ребер $\ell_{x,1}, \ell_{x,2}, \dots, \ell_{x,n}$ образует путь L_x, соединяющий вершину x с вершиной, максимально удаленной от нее. Нетрудно показать, что пути L_x, построенные таким образом для различных x, попарно не имеют общих ребер и, следовательно, реализуют рассматриваемую коммутацию.

Литература

I. Евреинов Э.В., Косарев Ю.Г. О возможности построения вычислительных систем высокой производительности. Новосибирск, Издательство СО АН СССР, 1962.