РАЗДЕЛ II

ТЕЛЕКОММУНИКАЦИИ И ПОТОКИ В СЕТЯХ

«Равномерные» алгоритмы последовательного заполнения потоковой сети потоками продуктов

А. П. Афанасьев, Я. Р. Гринберг, И. И. Курочкин

Сформулирована задача определения оптимальных путей в потоковой сети, если элементарные требования на организацию потоков продуктов между полюсами возникают последовательно. Проанализировано принципиальное отличие этой задачи от классической многопродуктовой проблемы. Предложены два алгоритма решения задачи и получены вычислительные процедуры нахождения оптимальных путей. Оказалось, что для обоих алгоритмов определение очередного оптимального пути сводится к применению алгоритмов для поиска пути минимальной стоимости. Разработана математическая модель процесса последовательного заполнения потоковой сети потоками продуктов. Принята вероятностная модель потока элементарных требований. Математическая модель содержит несколько типов сетей — стохастические сети, «колесо», «двойное колесо», «связанные кластеры». В целях сокращения объема вычислений предложены модификации оптимальных вычислительных процедур. Приведены некоторые результаты численных экспериментов.

1. Введение

В теории потоков в сетях хорошо известна так называемая «проблема многопродуктового потока», которая заключается в следующем. В заданной потоковой сети (с ограничениями на пропускные способности дуг) определена матрица интенсивностей потоков разных продуктов между полюсами (называемая матрицей требований или требованием); требуется определить, возможен ли в этой сети многопродуктовый поток, определяемый данным требованием.

Элементарным требованием назовем элемент матрицы требований. Таким образом, элементарное требование — это интенсивность потока специфического (отдельного) продукта между некоторой парой полюсов в сети. На практике нередко возникают задачи, в которых матрица требований не задана сразу, одномоментно, а формируется постепенно, путем последовательного добавления элементарных требований. Однако выбор пути, по которому осуществляется удовлетворение данного элементарного требования, необходимо производить последовательно, на каждом шаге.

В качестве примера можно привести телекоммуникационные сети связи с разделяемой средой передачи данных. Определение канала связи между абонентами, а в особенности тот факт, что необходимость установления связей между различными абонентами возникает последовательно во времени и вполне укладывается в описанную схему удовлетворения элементарных требований.

Реальные задачи IP-маршрутизации в глобальных сетях, насколько можно судить из литературы [1], полностью этой схеме не соответствуют. Но фрагменты глобальных сетей — автономные системы, а также их части этим свойством (т. е. укладываются в вышеописанную схему) обладают. В данной математической модели можно учесть не только накопление заявок, но и их аннулирование. Именно такая постановка задачи более близка для телекоммуникационных сетей. Однако в данной работе рассматривается только накопление заявок, так как в целом мы считаем, что указанная постановка задачи и ее модификации могут оказаться полезными для тех, кто заинтересован в рациональном экономичном и оптимальном функционировании физических сетей связи.

Классическая проблема многопродуктового потока решается методом линейного программирования (ЛП) [2]. Почти одновременно с выходом в свет этой книги в [3] было получено необходимое и достаточное условие разрешимости (допустимости) многопродуктовой проблемы, однако нам неизвестны ни вычислительные процедуры, реализующие это условие, ни примеры его практического применения. Любые алгоритмы решения многопродуктовой задачи будем называть **синхронными**, подчеркивая тем самым, что требование задается сразу, одномоментно.

Значительно более хорошо в теории потоков в сетях исследована однопродуктовая задача, т. е. задача удовлетворения элементарного требования. Кроме классической задачи определения максимального потока между некоторой парой полюсов, имеются также алгоритмы по определению кратчайшего пути или пути минимальной стоимости — Дейкстры [4], Беллмана—Форда [5], Диница [6, 7]. Эти алгоритмы востребованы и широко используются в практических целях. Приведем только один пример. Некоторые протоколы маршрутизации в глобальных сетях — например, OSPF — используют алгоритм Беллмана—Форда для определения канала связи между двумя абонентами в телекоммуникационной сети с разделяемой средой передачи. При этом оптимизация может производиться по каждому из нескольких критериев — минимальная временная задержка, максимальная надежность, максимальная пропускная способность и др. Такая постановка задачи требует не только знания топологии сети или ее некоторого участка (в терминах телекоммуникационных сетей — автономной системы), но и задания стоимостей (метрик) дуг сети. Эти характеристики определяются с помощью мониторинга сети или иным образом. Более ранний, но все еще широко применяемый протокол маршрутизации в глобальных сетях связи — RIP-протокол — использует совершенно другую методику вектора расстояния. Путем обмена информацией с другими соседними маршрутизаторами, RIP-маршрутизатор составляет таблицу маршрутизации, в которой указана «длина» пути для каждого адресата. Длина понимается как количество дуг пути. Если до адресата есть несколько путей, то в таблице сохраняются только пути с наименьшей длиной. Если подходить строго, то поскольку RIP-протокол не «знает» и не использует прямо топологию сети, нельзя говорить о какой-либо сетевой оптимизации. Кроме того, следует заметить, что протокол RIP использует «запоздалую» информацию, так как маршрутизатор рассылает служебные пакеты только соседям. И информация о пути к моменту ее поступления может быть уже устаревшей. Все же с некоторой натяжкой можно говорить о том, что RIP-протокол осуществляет оптимизацию по критерию кратчайшего пути.

Возвращаясь к описанной выше постановке задачи по выбору оптимальных путей для удовлетворения **последовательно** возникающих элементарных требований, надо констатировать, что рассмотрение такой проблемы в литературе отсутствует. Цель настоящей статьи — предложить алгоритмы решения этой проблемы, вывести оптимальные вычислительные процедуры и разработать математическую модель процесса последовательного заполнения потоковой сети потоками различных продуктов.

2. Постановка задачи. Различные варианты алгоритмов решения

В самой общей постановке задача может быть сформулирована следующим образом. В потоковой сети последовательно (во времени) возникают элементарные требования на организацию потоков разных продуктов, удовлетворение которых следует произвести до возникновения очередного элементарного требования. Как следует производить выбор пути (путей) в сети для удовлетворения очередного элементарного требования, чтобы суммарный поток удовлетворенных элементарных требований был максимальным. Алгоритмы, реализующие удовлетворение элементарных требований на каждом шаге, будем называть **последовательными**, подчеркивая тем самым их отличие от синхронных алгоритмов.

Введем понятия **относительно оптимального** и **абсолютного** последовательных алгоритмов.

Относительно оптимальным последовательным алгоритмом будем называть такой последовательный алгоритм, который обеспечивает не меньший суммарный поток по сравнению с любым другим последовательным алгоритмом и любым потоком элементарных требований в произвольной сети.

В каждый данный момент совокупность элементарных требований образует матрицу требований многопродуктовой задачи в данной сети. Очевидно, что пока очередные элементарные требования выполняются последовательным алгоритмом, суммарное требование является допустимым. Однако оно может оказаться допустимым и при таком количестве элементарных требований, которое невозможно удовлетворить, применяя последовательный алгоритм. С другой стороны, если суммарная матрица требований, рассматриваемая как требование в многопродуктовой задаче, оказалась недопустимой в данной сети, то никакой последовательный алгоритм не сможет удовлетворить эту совокупность элементарных требований.

Абсолютным последовательным алгоритмом назовем такой, который обеспечивает такой же суммарный поток удовлетворенных требований, как и синхронный алгоритм. Другими словами, если очередное элементарное требование оказалось невыполнимо для абсолютного последовательного алгоритма, то многопродуктовая задача с суммарным требованием, включающим первое невыполненное элементарное требование — также не имеет решения. И это справедливо для всех типов сетей и всех потоков элементарных требований.

Конечно, можно распространить понятие относительно оптимального последовательного алгоритма также и на определенные типы сетей и (или) на определенные типы потоков элементарных требований; соответственно, искать именно такие специфические оптимальные алгоритмы. Такая программа весьма обширна, и ее выполнение не входит в задачу настоящей работы. Кроме того, существование не только абсолютного, но и относительно оптимального последовательных алгоритмов нам представляется весьма проблематичным.

Идеология настоящей работы заключается в том, что мы выбираем несколько (три, из них два оригинальных) алгоритмов, которые кажутся нам «разумными», некоторую модель потока элементарных требований, и методом математического моделирования сравниваем их между собой и с синхронным алгоритмом на разных типах сетей.

Кратко опишем идеи, заложенные в эти алгоритмы, а также резоны, по которым мы выбрали вероятностную модель потока элементарных требований.

Первый из них, идея которого принадлежит проф. А. П. Афанасьеву, заключается в следующем: очередной путь следует проводить по дугам сети с наибольшей пропускной способностью.

Другой последовательный алгоритм, предложенный Я. Р. Гринбергом, основан на измерении пропускных способностей минимальных разрезов между всеми парами полюсов в сети. Из этих величин формируется критерий неравномерности сети, а оптимальный очередной путь определяется как такой, который максимально не увеличивает этот критерий.

В обоих этих алгоритмах присутствует идея использовать преимущественно «избыточный ресурс» сети, т. е. при удовлетворении очередного элементарного требования приблизить сеть к некому «идеальному» или «равномерному» состоянию. Правда, и «равномерность» и «избыточность» в этих алгоритмах понимаются по-разному.

Третий алгоритм, который мы применяем в данной работе, — это «классический» алгоритм кратчайшего по количеству дуг пути. Этот алгоритм используется как некий «эталонный» алгоритм для сравнения. Во-первых, он является последовательным алгоритмом. Во-вторых, он никак не был «приспособлен» к решению задачи о максимизации суммарного потока. Именно с ним мы сравниваем два новых алгоритма, которые, естественно, должны давать лучшие результаты.

При выборе математической модели для потока элементарных требований мы рассматривали два варианта — минимаксный (игровой) и вероятностный подходы.

Первый заключается в следующем. Каков должен быть наихудший поток элементарных требований, и как в этом случае следует себя вести оптимальным образом, т. е. так, чтобы удовлетворить максимальное число требований. Вполне правдоподобная гипотеза, решающая такую задачу, описана ниже. Выбрать пару полюсов в сети, которая характеризуется тем, что пропускная способность минимального разреза имеет минимальное значение среди всех других пар полюсов в сети. Поток требований установить как поток требований только между этой парой полюсов, а очередные требования удовлетворять так, чтобы между этой парой полюсов можно было «пропустить» максимальный поток. При подробном исследовании этой гипотезы необходимо также включить в рассмотрение и такие потоковые сети, в которых минимальное значение минимального разреза имеют две или несколько пар полюсов. С практической точки зрения минимаксный подход нам представляется менее перспективным, поэтому в настоящей работе мы его далее нигде не рассматриваем.

Второй подход заключается в задании вероятностных характеристик для потока элементарных требований. Например, можно считать, что элементарные требования статистически равномерно распределены между всеми парами полюсов. Или — с заданными вероятностями, что является наиболее общим случаем для вероятностной модели.

Отметим, что алгоритм Дейкстры, также как и равномерный по дугам алгоритм, «нечувствительны» к модели потока элементарных требований. Напротив, равномерный по минимальным разрезам алгоритм прямо зависит от вероятностных характеристик потока элементарных требований, однако он нуждается в дополнительных уточнениях, если модель потока не является вероятностной.

Будем далее считать, что элементарное требование во всех случаях является требованием обеспечить поток только единичной интенсивности, поэтому математически представляет собой просто целое число, указывающее на порядковый номер той пары полюсов, между которыми надо организовать поток продукта.

3. Вероятностная модель потока элементарных требований

Пусть имеется сеть G, состоящая из N узлов $A_1, A_2, \ldots, A_N$ и K дуг $B_1, B_2, \ldots, B_K$, имеющих пропускные способности b_k, $k = 1, 2, \ldots, K$. Узлы A_i, $i = 1, 2, \ldots N_1$, $N_1 \leqslant N$ будем называть также полюсами. Упорядочим каким-либо образом все пары полюсов. Общее количество M различных пар полюсов в сети G равно $M = N_1(N_1 - 1)/2$.

Последовательность целых неотрицательных чисел $\prod_t t = 1, 2, \ldots, t, \ldots$ будем называть вероятностной моделью потока элементарных требований, если выполнены следующие условия:

— все числа последовательности меньше или равны M;

— существуют частоты p_m, с которыми каждое число $1 \leqslant m \leqslant M$ встречается в этой последовательности.

Частота p_m определяется как следующий предел:

$$\lim_{t \to \propto} \frac{q_{t,m}}{Q_t} = p_m, \tag{1.1}$$

где $q_{t,m}$ — количество чисел m, содержащихся в отрезке последовательности $\prod_t$ длиной Q_t.

Очевидно, что

$$\sum_{m=1}^{M} p_m = 1.$$

Вероятностный поток элементарных требований с частотами p_m для всех пар полюсов будем обозначать $\prod(p_m)$.

Если все частоты одинаковы и равны $1/M$, то такой поток будем называть равномерным потоком и обозначать $\prod(1)$.

4. Равномерный по минимальным разрезам алгоритм

В обозначениях п. 3 введем в рассмотрение следующие величины:

— R_m — пропускная способность минимального разреза между m-ой парой полюсов, равная, по теореме о максимальном потоке и минимальном разрезе максимальному потоку между этой парой полюсов;

— $\overline{R} = (1/M) \sum\limits_{m=1}^{M} R_m$ — среднее значение пропускных способностей минимальных разрезов между всеми парами полюсов.

Если поток элементарных требований равномерен, то естественно предположить, что чем «ближе» все R_m к среднему значению, тем больше элементарных требований «вместится» в сеть. Поэтому образуем из введенных величин критерий χ

$$\chi = \sum_{m=1}^{M} \left(\frac{1}{M} \frac{R_m}{\overline{R}} - p_m \right)^2, \tag{4.1}$$

который, будучи пропорционален среднеквадратичному отклонению величин R_m от среднего значения, и является той «мерой неравномерности», которую следует максимально не увеличивать при удовлетворении очередного элементарного требования. В случае равномерного потока $p_m = 1/M$, и определенный таким образом критерий имеет следующий вид:

$$\chi = \sum_{m=1}^{M} \left(\frac{R_m}{\overline{R}} - 1 \right)^2. \tag{4.2}$$

(Постоянный множитель $1/M^2$ несущественен и не принимается во внимание.)

Удовлетворение очередного элементарного требования есть определение пути между некоторой парой полюсов и «пропускание» по нему потока единичной интенсивности. При этом сеть G меняется, так как изменяются пропускные способности дуг b_k. Это изменение должно максимально не увеличивать критерий χ. Выразим критерий и его приращение через величины b_k. Для этого нам потребуется знание всех минимальных разрезов между всеми парами полюсов сети G.

Пусть b_s^{lm}, $s = 1, 2, \ldots, S_{lm}$, где s-я дуга l-ого минимального разреза между m-й парой полюсов, S_{lm} — количество дуг в этом разрезе. Тогда

$$R_m = \frac{1}{L_m} \sum_{l=1}^{L_m} \sum_{s=1}^{S_{lm}} b_s^{lm}, \tag{4.3}$$

где L_m — полное количество минимальных разрезов между m-ой парой полюсов. Преобразуем это выражение, определив числа u_{mk}, $m = \overline{1, M}$, $k = \overline{1, K}$, как такие, которые обозначают количество различных минимальных разрезов между m-й парой полюсов, в которые входит дуга b_k. Тогда очевидно, что

$$R_m = \frac{1}{L_m} \sum_{k=1}^{K} u_{mk} b_k. \tag{4.4}$$

С учетом (4.4) неравномерность сети (4.1), (4.2) оказывается выраженной через величины b_k. После удовлетворения очередного элементарного требования потока П сеть G изменяется в части изменения величин b_k, следовательно, изменится и величина χ. В случае, если относительное изменение величин b_k невелико, изменение неравномерности сети определяется следующей формулой:

$$\Delta\chi = \sum_{k=1}^{K} \frac{\partial\chi}{\partial b_k}\Delta b_k = \sum_{k=1}^{K} \left(\frac{\partial\chi}{\partial\overline{R}}\frac{\partial\overline{R}}{\partial b_k} + \sum_{m=1}^{M} \frac{\partial\chi}{\partial R_m}\frac{\partial R_m}{\partial b_k}\right)\Delta b_k. \tag{4.5}$$

Далее имеем:

$$\frac{\partial\chi}{\partial\overline{R}} = -\frac{2}{\overline{R}}\left[\frac{\overline{R^2}}{\overline{R}^2} - 1\right], \tag{4.6}$$

$$\frac{\partial\overline{R}}{\partial b_k} = r_k = \frac{1}{M}\sum_{m=1}^{M}\frac{u_{mk}}{L_m}, \tag{4.7}$$

$$\frac{\partial\chi}{\partial R_m} = \frac{2}{M\overline{R}}\left(\frac{1}{M}\frac{R_m}{\overline{R}} - p_m\right), \tag{4.8}$$

$$\Delta\chi = \frac{2}{M\overline{R}}\sum_{k=1}^{K}\Delta b_k\left[\sum_{m=1}^{M}\left(\frac{R_m}{\overline{R}}r_k - r_{mk}\right)p_m - \frac{\overline{R^2}}{\overline{R}^2}r_k + {} \right.$$
$$\left. {} + \frac{1}{M\overline{R}}\sum_{m=1}^{M} R_m r_{mk}\right], \tag{4.9}$$

где

$$r_{mk} = \frac{u_{mk}}{L_m}, \qquad \sum_{m=1}^{M} r_{mk} = M r_k.$$

Если поток элементарных требований равномерен, то (4.9) упрощается и принимает следующий вид:

$$\Delta\chi = \frac{2}{M\overline{R}}\sum_{k=1}^{K}\Delta b_k\left[-\frac{\overline{R^2}}{\overline{R}^2}r_k + \frac{1}{M\overline{R}}\sum_{m=1}^{M}R_m r_{mk}\right]. \tag{4.10}$$

Из (4.9) и (4.10) видно, что приращение меры неравномерности сети имеет вид линейной формы по приращениям пропускных способностей дуг. Задача заключается в том, чтобы при удовлетворении очередного элементарного требования сделать это приращение минимальным (в алгебраическом смысле, т. е. отрицательные величины приращения возможны). Таким образом, мы приходим к задаче поиска пути минимальной стоимости. С учетом того, что для любого пути все приращения Δb_k отрицательны (или равны нулю), выпишем явным образом выражения для стоимостей дуг сети:

$$W_k = \sum_{m=1}^{M}\left(-\frac{R_m}{\overline{R}}r_k + r_{mk}\right)p_m + \frac{\overline{R^2}}{\overline{R}^2}r_k - \frac{1}{M\overline{R}}\sum_{m=1}^{M}R_m r_{mk}, \tag{4.11}$$

$$W_k = \frac{\overline{R^2}}{\overline{R}^2} r_k - \frac{1}{M\overline{R}} \sum_{m=1}^{M} R_m r_{mk}. \tag{4.12}$$

В выражения (4.11), (4.12) входят величины, полностью определяемые пропускными способностями дуг сети, которые изменяются после удовлетворения каждого очередного элементарного требования. Таким образом, вообще говоря, на каждом шаге для определения оптимального по данному критерию пути требуется полный перерасчет стоимостей дуг. Экономные алгоритмы для этого неизвестны в литературе, однако их поиск и описание не является целью настоящей работы. Некоторые упрощения вычислительной процедуры, цена за которые — нахождение лишь субоптимальных относительно данного критерия путей, описаны в п. 7 настоящей работы.

5. Равномерный по дугам алгоритм*

Напомним, что идея этого алгоритма заключается в том, чтобы каждый очередной путь проводить по дугам сети с максимальной пропускной способностью. Для формализации этой идеи введем некоторые определения. Распределим все дуги сети по классам, в каждый из которых поместим дуги с одинаковой пропускной способностью, и упорядочим все классы в порядке возрастания пропускной способности входящих в них дуг. Пусть $B_1, \ldots, B_q, \ldots, B_Q$ — обозначения введенных классов, $K_1, \ldots, K_Q$ — количество дуг в этих классах, $b_1, \ldots, b_Q$ — величины пропускных способностей дуг, входящих в эти классы.

Пусть теперь C — путь из одного полюса в другой и пусть этот путь содержит c_1 дуг из класса B_1, ..., c_Q дуг из класса B_Q. По определению $c_q \geqslant 0$: c_q — целые числа; кроме того, $\sum\limits_{q=1}^{Q} c_q \leqslant N - 1$, так как любой путь не может содержать больше чем $(N-1)$ дуг. Можно сказать, что каждый путь представляет собой целочисленный неотрицательный вектор в Q-мерном пространстве $\mathbf{C_Q}$. Теперь переходим к главному — упорядочиванию этих векторов. Пусть даны два пути c_1 и c_2, которые представляются двумя векторами c_q^1 и c_q^2. Образуем из них вектор $d_q = c_q^1 - c_q^2$ (этот вектор не имеет смысла пути в сети G). Пусть $d_{1,2}$ — первая ненулевая координата вектора d_q. Если $d_{1,2} > 0$, то будем говорить, что путь c_2 следует за путем c_1. Наоборот, если $d_{1,2} < 0$, то будем говорить, что путь c_1 следует за путем c_2. Если $d_{1,2} = 0$, т. е. векторы c_q^1 и c_q^2 путей c_1 и c_2 полностью совпадают, то пути c_1 и c_2 будем называть эквивалентными. Для обозначения отношения следования также будем использовать знаки «$>$» или «$<$».

* Метод реализации алгоритма А. П. Афанасьева путем назначения стоимостей дугам сети G, предложен и разработан Я. Р. Гринбергом.

По определению, если $c_1 > c_2$, то $c_2 < c_1$. Очевидно также, что введенное отношение обладает свойством транзитивности, т. е. если $c_1 < c_2$ и $c_2 < c_3$, то $c_1 < c_3$. Введенное отношение порядка позволяет разбить все пути из одного полюса сети G в другой полюс, на следующие друг за другом классы эквивалентных путей. Теперь мы можем сформулировать критерий оптимальности по Афанасьеву.

При удовлетворении очередного элементарного требования оптимальным путем является любой представитель последнего класса эквивалентных путей.

Может показаться, что введенное отношение порядка похоже на лексикографический порядок, однако это не так. Лексикографический порядок устанавливается по первой «букве» «слова», при совпадении — по второй, и т. д., причем порядок следования «букв» в «слове» — произвольный. Порядок, введенный в настоящем разделе, строго ранжирует «буквы», делая их больше похожими на разряд числа, и в каждом разряде оставляет только количество «букв», вне зависимости от порядка следования «букв». Так что, если, например, «слово» начинается на букву «н», то в нем уже не может быть никаких букв от «а» до «м».

Опишем алгоритм определения оптимального пути. Идея алгоритма заключается в назначении стоимостей дугам сети, причем дуги одного класса имеют одинаковые стоимости. Стоимости убывают по мере увеличения пропускной способности дуг, так что самые большие стоимости у дуг первого класса, самые маленькие — у дуг последнего класса. Стоимости назначаются таким образом, что если в сети существует путь, не содержащий представителя какого-либо класса, то все пути, содержащие представителя этого или меньших классов, будут иметь большие стоимости.

Введем дополнительно ряд определений. Пусть количество элементов в каждом классе B_i равно n_i. Для каждого номера класса i, $1 \leqslant i < Q$, выполняется одно и только одно из следующих трех взаимоисключающих условий:

1)
$$K_{i+1} + K_{i+2} + \ldots + K_Q \leqslant N - 1; \tag{5.1}$$

2) существует такое k, $i + k < Q$, что
$$K_{i+1} + K_{i+2} + \ldots + K_{i+k} = N - 1; \tag{5.2}$$

3) существует такое k, $i + k < Q$, что
$$\begin{gathered} K_{i+1} + K_{i+2} + \ldots + K_{i+k} < N - 1; \\ K_{i+1} + K_{i+2} + \ldots + K_{i+k+1} > N - 1. \end{gathered} \tag{5.3}$$

В случае (3) существует такое число K^*_{i+k+1}, что
$$K_{i+1} + K_{i+2} + \ldots + K^*_{i+k+1} = N - 1. \tag{5.4}$$

Теперь определим стоимости дуг W_i, принадлежащих классу B_i следующим рекуррентным образом, начиная с последнего класса:
$$W_Q = \varepsilon, \tag{5.5}$$

$$W_i = W_{i+1}K_{i+1} + W_{i+2}K_{i+2} + \ldots + W_Q K_Q + \varepsilon_1, \tag{5.6}$$

если выполнено 1-е условие;

$$W_i = W_{i+1}K_{i+1} + W_{i+2}K_{i+2} + \ldots + W_{i+k}K_{i+k} + \varepsilon_1, \tag{5.7}$$

если выполнено 2-е условие;

$$W_i = W_{i+1}K_{i+1} + W_{i+2}K_{i+2} + \ldots + W_{i+k+1}K^*_{I+k+1} + \varepsilon_1, \tag{5.8}$$

если выполнено 3-е условие.

Величины ε, ε_1 — положительны.

Обозначим через $W(c)$ — стоимость пути c, т. е.

$$W(c) = W_1 c_1 + W_2 c_2 + \ldots + \varepsilon c_Q.$$

Поскольку стоимости классов монотонно убывают, то из (5.1)–(5.8) легко усмотреть, что стоимость любого пути в G, в котором имеются только дуги классов, следующих за B_i, строго меньше, чем стоимость только одной дуги класса B_i и всех других классов, ему предшествующих. Именно в этом и заключается смысл введенных стоимостей.

Теорема. *Путь минимальной стоимости из одного полюса в другой в сети G со стоимостями дуг, определенными соотношениями* (5.5)–(5.8), *является оптимальным путем.*

Доказательство проведем методом от противного. Пусть c_1, c_q^1 — путь минимальной стоимости в сети G, и c_2, c_q^2 — оптимальный по Афанасьеву путь в сети G. По предположению теоремы путь c_1 предшествует пути c_2. Это означает, что в векторе $d_q = c_q^1 - c_q^2$ величина $d_{1,2} > 0$. Докажем, что путь c_1 имеет большую стоимость, чем путь c_2. Пусть число $d_{1,2} > 0$ находится в классе B_i. Тогда $W(c_1) - W(c_2) = W_i d_{1,2} + \Delta$, где

$$\Delta = W_{i+1}\left(c_{i+1}^1 - c_{i+1}^2\right) + W_{i+2}\left(c_{i+2}^1 - c_{i+2}^2\right) + \ldots + \varepsilon\left(c_Q^1 - c_Q^2\right).$$

Величина Δ может быть как положительной, так и отрицательной. Докажем, что $W(c_1) - W(c_2) > 0$. Во-первых,

$$W_i \leqslant W_i d_{1,2}. \tag{5.9}$$

Во-вторых,

$$-\Delta \leqslant W_{i+1}c_{i+1}^2 + W_{i+2}c_{i+2}^2 + \ldots + \varepsilon c_Q^2.$$

Это выражение представляет собой стоимость некоторого отрезка пути в сети G. Следовательно, $-\Delta < W_i$. Совмещая это с (5.9), получаем $W(c_1) - W(c_2) = W_i d_{1,2} + \Delta > 0$. Таким образом, оказалось, что оптимальный путь имеет меньшую стоимость, чем путь минимальной стоимости, что невозможно. Это противоречие доказывает теорему. □

Пусть $S(c)$ — стоимость пути минимальной стоимости c в сети G с метрикой дуг, определенной выражениями (5.5)–(5.8). По этому числу легко восстанавливается вектор c_q:

$$S_1 = S, \quad c_q = \left[\frac{S_q}{W_q}\right], \quad S_q = S_{q-1} - c_{q-1}W_{q-1}, \tag{5.10}$$

где квадратные скобки обозначают целую часть числа, заключенного в скобки. Это непосредственно следует из метода назначения стоимостей дугам сети, при котором «разряды», соответствующие классам эквивалентности дуг сети, не «смешиваются».

6. Описание блоков математической модели по заполнению потоковой сети последовательно потоками продуктов

Математическая модель по заполнению потоковой сети последовательными потоками продуктов состоит из следующих блоков:

1. Генератор сетей.
2. Генератор случайного потока элементарных требований.
3. Блок вычисления стоимостей дуг.
4. Блок, реализующий алгоритмы поиска пути минимальной стоимости.
5. Блок, реализующий визуализацию результатов.
6. Блок, реализующий архивацию результатов.

Генератор сетей

В настоящей работе для имитации потоковых, в том числе телекоммуникационных, сетей было выбрано несколько характерных топологий, а именно: стохастическая топология, «колесо» и «двойное колесо», «связанные кластеры». Каждая из этих топологий определяется своим набором параметров, которые перечислены ниже. Общим для процесса генерации сети является задание количества узлов и полюсов, определение того, существует или нет связь (дуга) между данными двумя узлами, определение пропускной способности данной связи (дуги), проверка связности сети. В результате генерации мы получаем фактически неориентированный граф с определенными весами (пропускными способностями) дуг этого графа, являющийся связным по всем парам полюсов. Выбранные типы сетей имеют, на наш взгляд, еще и то преимущество, что позволяют «конструировать» гибридные сети, полезные при имитации реальных потоковых сетей. Алгоритм генерации сетей приведен на рис. 1.

Для различных типов сетей нужно задавать разные параметры, но в целом можно выделить несколько групп:

1. Параметры графа в целом (симметричность, ориентированность и т. п.).
2. Количественный состав узлов и полюсов в сети.

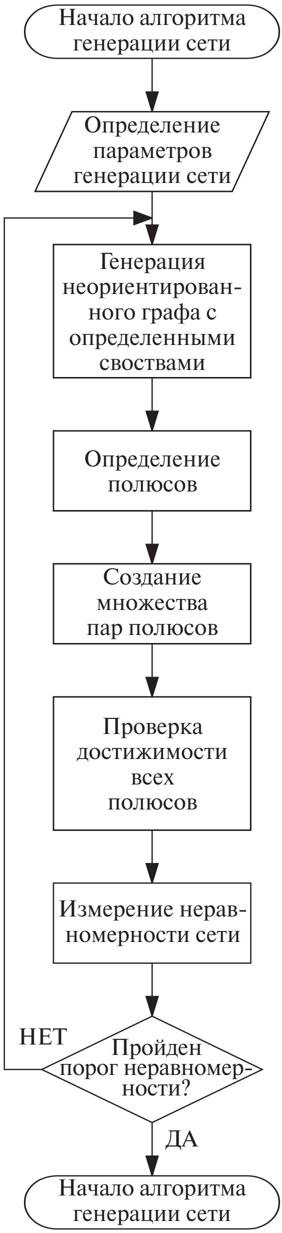

Рис. 1. Алгоритм генерации сетей

3. Параметры, описывающие количество и величины ненулевых дуг в сети, в том числе характеристики функции определения веса дуги (к примеру: равномерный закон случайного распределения величин в определенном интервале).
4. Параметры, характерные для определенной топологии сети.

1. Стохастическая топология сети

При стохастической топологии — пример на рис. 2 — нет выраженной структуры сети, полюса равнозначны по связности. Дуги имеют случайную пропускную способность в заданном интервале.

При генерации сетей со случайной топологией необходимо задать следующие параметры:

- Количество узлов сети (количество узлов в графе).
- Количество полюсов в сети (параметр необходим для определения связности между полюсами).
- Процент нулевых дуг (отсутствующих дуг).
- Интервал пропускных способностей дуг сети.
- Тип распределения значений пропускных способностей дуг сети при их стохастической генерации и параметры этого типа распределения.
- Флаг симметричности графа сети.
- Минимальный порог меры неравномерности для начальной сети.

2. Радиальная топология («колесо»)

Данная топология — рис. 3 — характеризуется выделением центрального узла из общего множества. Дуги, связывающие периферические узлы с центральным имеют более низкую пропускную способность, чем дуги связывающие периферические узлы между собой. Периферические узлы связаны между собой дугами в «кольцо». Таким образом, название «колесо» достаточно четко описывает особенности топологии: центральный узел — ось,

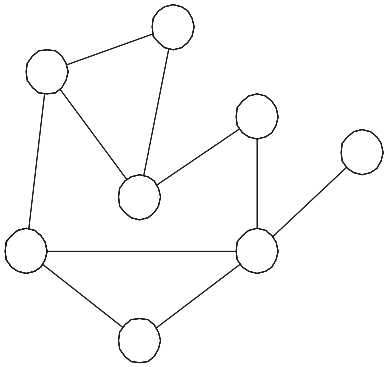

Рис. 2

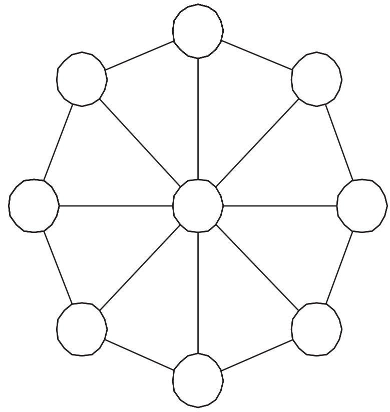

Рис. 3

обод — периферические узлы, связанные дугами большой пропускной способности в «кольцо», спицы — дуги с малой пропускной способностью, которые связывают центральный и периферические узлы.

Стоит отметить, что в данной работе в любой топологии, даже если центральный узел представляет собой сильносвязанный граф (т. е. сеть с дугами большой пропускной способности), то логически в топологии этот граф будет представлен единым узлом.

- Общее количество вершин.
- Количество периферических узлов.
- Количество полюсов.
- Процент нулевых дуг к центральному узлу.
- Максимальное значение пропускной способности дуги на периферии.
- Максимальное значение пропускной способности дуги к центральному узлу.
- Тип распределения случайных величин дуг при генерации сети и параметры этого типа распределения.
- Связность кольца с центром.

3. Топология типа «двойное колесо»

Данная топология, графически представленная на рис. 4 и рис. 5, отличается от «колеса» наличием второго «обода» (второго множества узлов, объединенных в «кольцо» и связанных с центральным узлом).

Данная топология может иметь несколько вариаций:

- Два множества периферических дуг (два обода) не имеют связи между собой.
- Два множества периферических дуг связаны небольшим количеством дугам с малой пропускной способностью.

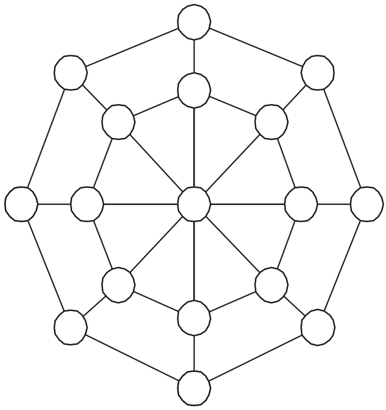

Рис. 4

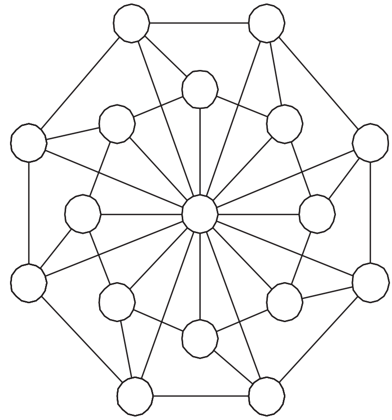

Рис. 5

- Два множества периферических дуг сильносвязаны и/или имеют большое количество дуг с большой пропускной способностью.
- Общее количество вершин.
- Общее количество вершин в каждом кольце.
- Количество полюсов.
- Процент нулевых дуг к центральному узлу.
- Максимальное значение пропускной способности дуги на периферии.
- Максимальное значение пропускной способности дуги к центральным узлам.
- Тип распределения случайных величин дуг при генерации сети и параметры этого типа.
- Связность колец меду собой.
- Связность колец с центральным узлом.

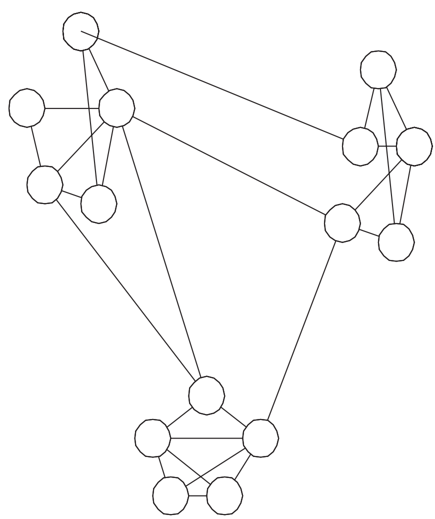

Рис. 6

4. Множество связанных кластеров

Топология связанных кластеров предполагает наличие нескольких сильносвязанных графов с дугами большой пропускной способности, которые соединены дугами малой пропускной способности в общий граф (единую сеть) (рис. 6).

- Общее количество вершин.

- Общее количество вершин в каждом кластере.
- Количество кластеров.
- Количество полюсов.
- Процент нулевых дуг внутри кластера.
- Максимальное количество ненулевых дуг, связывающих пару кластеров.
- Максимальное значение пропускной способности дуги внутри кластеров.
- Максимальное значение пропускной способности дуги между кластерами.
- Распределение случайных величин при генерации сети.

Отметим, что в качестве кластеров можно использовать все вышеперечисленные топологии, тем самым моделируя, например, сложную иерархию телекоммуникационных сетей.

Генератор случайного потока элементарных требований

Генератор случайного потока элементарных требований создает последовательность элементарных требований с заданными вероятностными характеристиками. С помощью генератора случайных чисел задается пара полюсов, между которыми организуется элементарный поток (удовлетворяется элементарное требование).

Блок вычисления стоимостей дуг

Блок вычисления стоимостей дуг для каждого алгоритма свой. Методы вычислений стоимостей дуг приведены в описаниях соответствующих алгоритмов заполнения.

Общим для всех алгоритмов остается вид матрицы стоимостей. Несуществующие дуги нулевые (признак отсутствия дуги) или бесконечно большая величина. Существующие дуги помечены ненулевыми весами.

Блок, реализующий алгоритмы поиска пути минимальной стоимости

В качестве алгоритмов нахождения пути минимальной стоимости используются:

- Алгоритм Дейкстры.
- Алгоритм Беллмана—Форда.

Блок, реализующий визуализацию результатов

Визуализация результатов осуществляется с помощью двухмерных графиков.

Блок, реализующий архивацию результатов

Результаты каждого завершившегося процесса заполнения сети «закладывается» в ячейку базы данных. В этой ячейке хранятся следующие параметры: тип и количественные характеристики исходной сети; алгоритм заполнения; тип потока элементарных требований и фактическая последовательность элементарных требований; выбранные пути; количество удовлетворенных элементарных требований по принятым критериям завершения процесса; данные для визуализации результатов.

Сохранение результатов осуществляется с помощью СУБД MS SQL Server 2000.

Алгоритм заполнения сети элементарным потоком

1. Нахождение минимального пути по матрице весов для заданной пары полюсов.
2. Нахождение максимальной пропускной способности минимального пути.
3. Если максимальная пропускная способность дуг минимального пути больше, чем элементарный поток, то веса (пропускные способности) дуг уменьшаются на значение элементарного потока и происходит возвращение в начало алгоритма.
4. Если заполнение сети элементарным потоком завершилось неудачей, то выставляется флаг на пересчет весов (алгоритм определения весов) или завершается в соответствии с критерием завершения заполнения сети.

Используются следующие критерии завершения процесса:

- 1-й отказ в удовлетворении очередного элементарного требования.
- Все пары полюсов несвязанны (граф сети несвязный для всех пар полюсов).

Общая схема заполнения сети приведена на рис. 7.

7. Математическое моделирование. Примеры численных результатов

В этом разделе приведены некоторые результаты численных экспериментов по заполнению сетей элементарными потоками продуктов в соответствии с последовательно возникающими элементарными требованиями. Во всех этих экспериментах в качестве потока элементарных требований был принят равномерный поток. На каждом типе сети были опробованы оба новых алгоритма — равномерный по минимальным разрезам и равномерный по дугам алгоритмы, каждому из которых соответствует свой график. На каждом графике представлены две кривые, представляющие собой меру неравномерности той сети, которая образовалась из исходной после удовлетворения всех предыдущих элементарных требований. Одна из кривых соответствует новому алгоритму, другая — алгоритму кратчайшего пути.

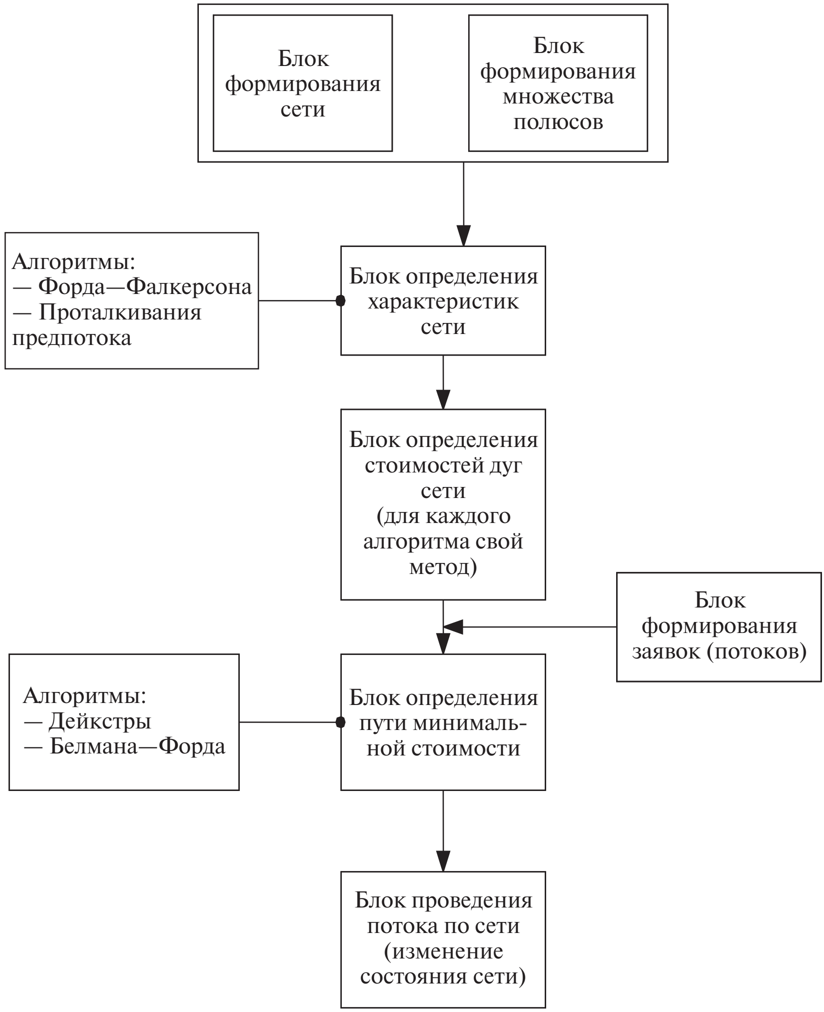

Рис. 7. Общая схема заполнения сети (одна итерация)

Стохастическая сеть

- 10 вершин;
- 4 полюса;
- нулевых дуг 45%;
- максимальное значение 10;
- порог начальной неравномерности 0,004.

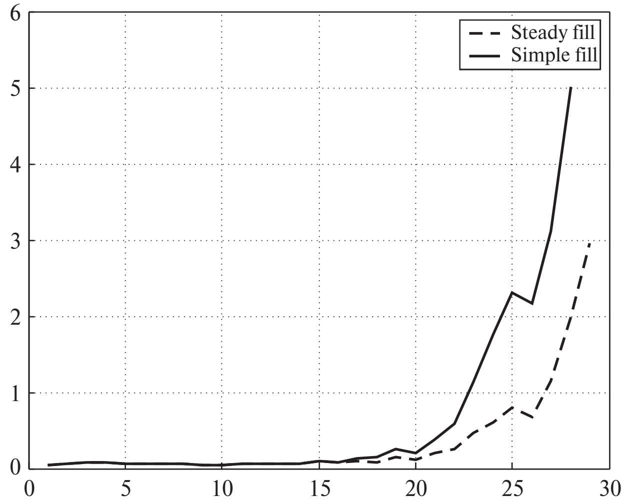

Рис. 8. Равномерный по минимальным разрезам алгоритм

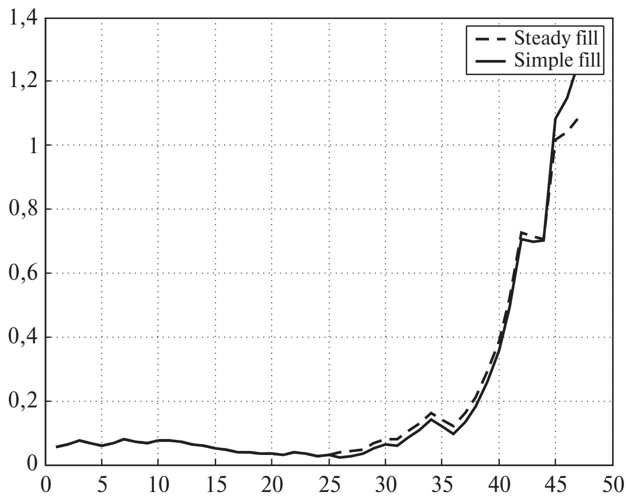

Рис. 9. Равномерный по дугам алгоритм

Измерение неравномерности после проведения каждого элементарного потока.

Топология сети «колесо»

- 11 вершин;
- 4 полюса;

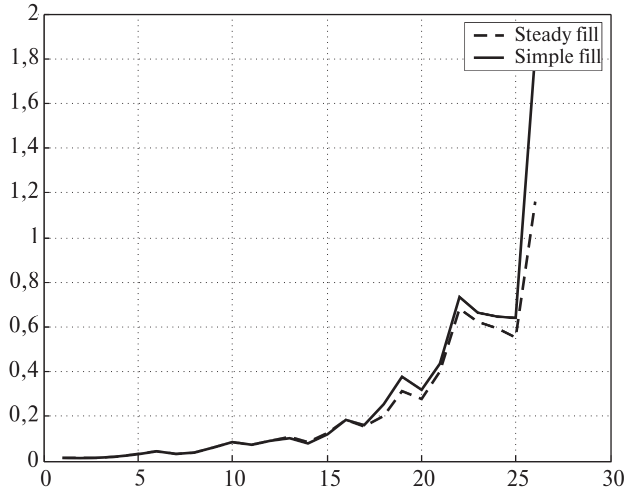

Рис. 10. Равномерный по минимальным разрезам алгоритм

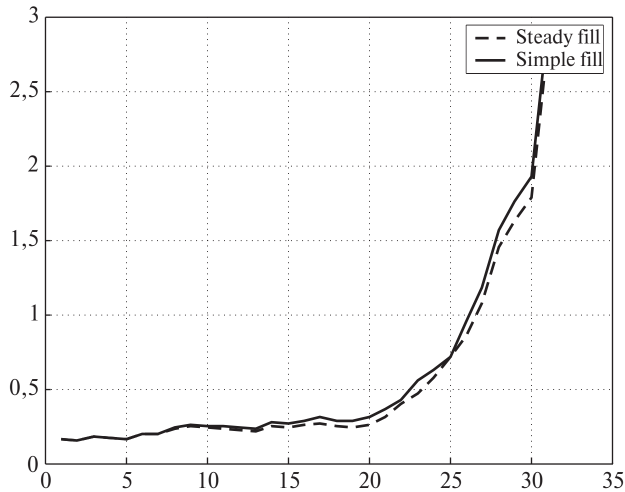

Рис. 11. Равномерный по дугам алгоритм

- нулевых дуг 0%;
- максимальное значение 10;
- порог начальной неравномерности 0,004.

Измерение неравномерности после проведения каждого элементарного потока.

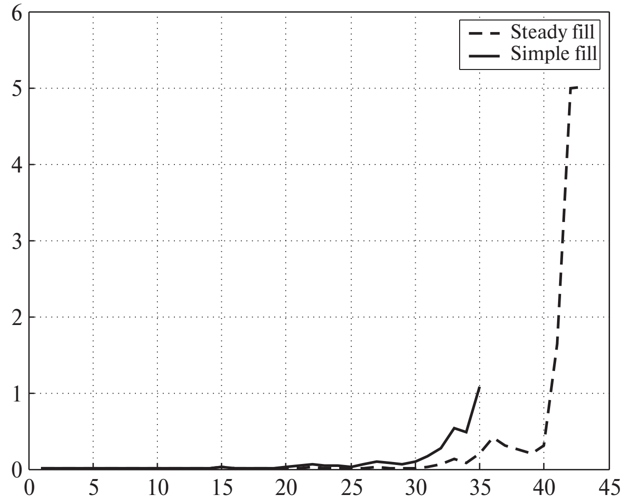

Рис. 12. Равномерный по минимальным разрезам алгоритм

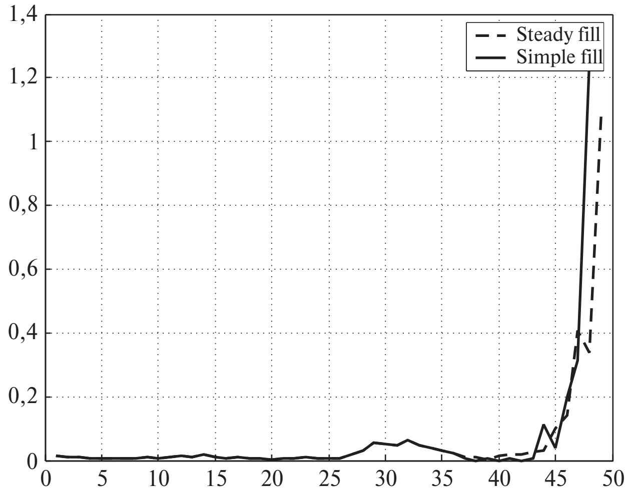

Рис. 13. Равномерный по дугам алгоритм

Топология сети «двойное колесо»

- 13 вершин;
- 4 полюса;
- нулевых дуг 0%;
- максимальное значение 10;
- порог начальной неравномерности 0,004.

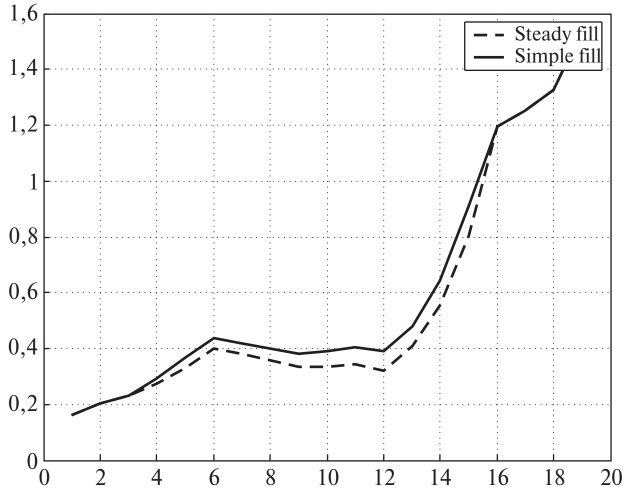

Рис. 14. Равномерный по минимальным разрезам алгоритм

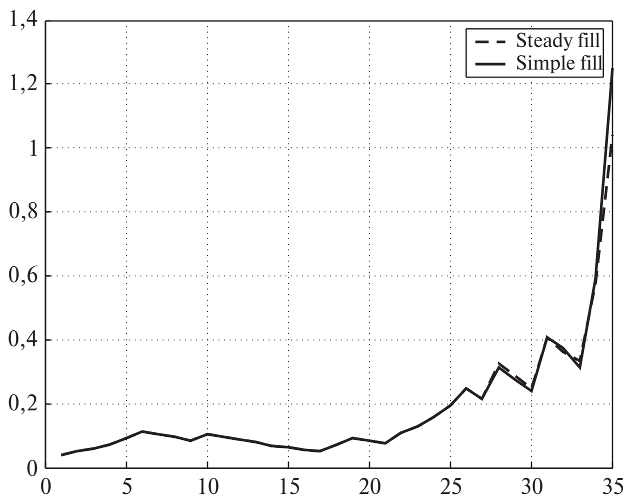

Рис. 15. Равномерный по дугам алгоритм

Измерение неравномерности после проведения каждого элементарного потока.

Топология сети «связанные кластеры»

- 13 вершин;
- 4 полюса;
- нулевых дуг 0%;

- максимальное значение 10;
- порог начальной неравномерности 0,004.

Измерение неравномерности после проведения каждого элементарного потока.

Из представленных результатов невозможно сделать какие-либо окончательные выводы, можно лишь отметить, что применение различных алгоритмов к одной сети приводит к существенно разным результатам, также сильно разнятся и результаты для разных типов сетей.

Литература

1. *Афанасьев А. П.*, *Гринберг Я. Р.*, *Курочкин И. И.* Принципы, методы, алгоритмы и протоколы маршрутизации в сетях связи. Обзор // Сетевые и алгоритмические задачи распределенных вычислений. Труды ИСА РАН. М.: УРСС, 2004. С. 5–104.
2. *Ху Т.* Целочисленное программирование и потоки в сетях. М.: Мир, 1974.
3. *Onaga K.*, *Kakusho O.* On Feasibility Conditions of Multicommodity Flows in Networks // IEEE Trans. on Circuit Theory. 1971. V. CT-18. № 4. P. 425–429.
4. *Dijkstra E. W.* A Note on Two Problems in Connection with Graphs // Num. Math. 1959. 1. 269–271.
5. *Bellman R.* On a Routing Problem. Quarterly of Applied Mathematics. 1958. 16. Pp. 87–90.
6. *Диниц Е. А.* Экономные алгоритмы нахождения кратчайших путей в сети // Транспортные сети: Сборник трудов. М.: ВНИИСИ, 1978.
7. *Диниц Е. А.*, *Карзанов А. В.* Программа расчета кратчайших путей и расстояний в сети // Транспортные сети.