

**Ю.Ю. ГРОМОВ, В.О. ДРАЧЕВ,
К.А. НАБАТОВ, О.Г. ИВАНОВА**

СИНТЕЗ И АНАЛИЗ ЖИВУЧЕСТИ СЕТЕВЫХ СИСТЕМ

МОСКВА

«ИЗДАТЕЛЬСТВО МАШИНОСТРОЕНИЕ-1»

2007

**Ю.Ю. ГРОМОВ, В.О. ДРАЧЕВ,
К.А. НАБАТОВ, О.Г. ИВАНОВА**

СИНТЕЗ И АНАЛИЗ ЖИВУЧЕСТИ СЕТЕВЫХ СИСТЕМ

Монография



МОСКВА
«ИЗДАТЕЛЬСТВО МАШИНОСТРОЕНИЕ-1»
2007

УДК 519.7

ББК 281

C387

Р е ц е н з е н т ы:

Доктор физико-математических наук, профессор
Московского энергетического института

Е. Ф. Кустов

Доктор физико-математических наук, профессор
Института радиоэлектроники РАН

В. Ф. Крапивин

C38
7

Синтез и анализ живучести сетевых систем : монография / Ю.Ю. Громов, В.О. Драчев, К.А. Набатов, О.Г. Иванова. – М. : «Издательство Машиностроение-1», 2007. – 152 с. – 400 экз. – ISBN 978-5-94275-386-3.

Рассмотрены вопросы моделирования и анализа живучести систем. Предложены модели и алгоритмы для решения задач анализа и оценки живучести сетей, анализа информационных и энергетических сетей.

Предназначена для научных, инженерно-технических работников и студентов вузов.

УДК 519.7

ББК 281

ISBN 978-5-94275-386-3

© «Издательство Машиностроение-1», 2007
© ГОУ ВПО «Тамбовский государственный
технический университет» (ТГТУ), 2007

Научное издание

**ГРОМОВ Юрий Юрьевич,
ДРАЧЕВ Виталий Олегович,
НАБАТОВ Константин Александрович,
ИВАНОВА Ольга Геннадьевна**

СИНТЕЗ И АНАЛИЗ ЖИВУЧЕСТИ СЕТЕВЫХ СИСТЕМ

МОНОГРАФИЯ

Редактор Т.М. Глинкина
Инженер по компьютерному макетированию Т.А. Сынкова
Корректор О.М. Ярцева

Подписано в печать 7.12.2007.
Формат 60×84/16. 8,84 усл. печ. л.
Тираж 400 экз. Заказ № 796

«Издательство Машиностроение-1»,
107076, Москва, Стромьинский пер., 4

Подготовлено к печати и отпечатано в Издательско-полиграфическом центре
Тамбовского государственного технического университета
392000, Тамбов, Советская, 106, к. 14

По вопросам приобретения книги обращаться по телефону 8(4752)638108

ВВЕДЕНИЕ

Сетевая информационная система (СИС) представляет собой многоуровневую иерархическую структуру, включающую в себя множество узлов, связанных между собой определенным образом. Такой конструкции присуще свойство уязвимости, которая определяется тем, что за счет многочисленных узлов и связей между ними (учитывая, что нормальное функционирование нескольких узлов иерархической сети возможно только при нормальном функционировании одного основного узла, называемого управляющим) нередко проявляется «каскадный эффект», когда сбой в одном каком-либо месте провоцирует перегрузки и выход из строя многих других элементов СИС.

Проектирование новых ИС и развитие уже существующих связаны с проблематикой принятия решений по использованию имеющихся сетевых структур: управлению потоками, распределению ресурсов между узлами. Перечисленные проблемы тесно связаны с задачей определения связности и живучести существующей или проектируемой ИС. Для рассматриваемых систем характерно наличие не только объективной, но и субъективной неопределенности, когда некоторые параметры системы известны отдельным пользователям, но не известны ЛПР (лицу, принимающему решения) или другим пользователям. Ответственность за принятые решения обязывает аккуратно разграничить неопределенные и случайные неконтролируемые факторы: случайность должна быть теоретически обоснована (и подтверждена результатами применения статистических методов), имеющаяся информация о функциях распределения, используемых случайных величинах должна быть указана явно. Взаимная зависимость элементов СИС приводит к немарковости случайных процессов, протекающих в них.

Проблеме оценки живучести СИС посвящен ряд работ (А.Г. Додонов, М.Г. Кузнецова, В.М. Вишневский, Д.Л. Белоцерковский, Ю.Е. Мельников, Ж.С. Сарыпбеков, Ю.Е. Малашенко, С.Л. Colbourn, К. Sekine, Н. Imai, S. Tani, А.Е. Smith и др.), в которых разработаны аналитические модели, адекватно описывающие процесс расчета живучести СИС, тем не менее, в настоящее время актуальной является задача разработки аналитического описания, обобщающего полученные ранее результаты и позволяющего не только осуществить разработку новых методов проектирования и анализа СИС, но и ставить и решать задачи расчета живучести СИС большой размерности и сложной структуры.

1. АНАЛИТИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ ОЦЕНКИ ЖИВУЧЕСТИ СЕТЕВЫХ СТРУКТУР

1.1. ОБЩЕЕ ОПИСАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ ОЦЕНКИ ЖИВУЧЕСТИ СЕТЕВЫХ СТРУКТУР

В инфраструктуре современного информационно-индустриального общества сетевые информационные системы (СИС) занимают одно из ключевых мест. Это вызвано возрастающей ролью информации в наукоемком промышленном производстве. Информация в современных условиях выступает, как ресурс, позволяющий минимизировать расходы других ресурсов (сырьевых, материальных, энергетических, трудовых, финансовых и т.д.).

Требования к качеству информации в современных условиях настолько возросли, что трудно представить нормальное функционирование общества без соответствующего информационного обеспечения. Например, в условиях рыночной экономики особое значение приобретают достоверность и полнота информации, без которых невозможна маркетинговая, финансовая и инвестиционная деятельность [1].

С другой стороны, в сферах телекоммуникаций, энергетики, в вопросах водоснабжения и водоотведения крупных промышленных предприятий или мегаполисов на первое место выходят вопросы анализа существующих (или вновь проектируемых) сетевых структур с точки зрения живучести при функционировании в режиме экстремальных нагрузок или быстрого роста абонентов – потребителей того или иного вида продукта, передаваемого по данным сетям.

Типичными задачами, возникающими перед аналитиками и инженерами-проектировщиками, являются расчет распределения потоков внутри информационной сети (или транспортной сети связи, так называемой сети передачи данных – СПД) при воздействии на сетевую структуру неблагоприятных внешних факторов (например, стихийных бедствий), выявление «узких мест» СИС (каналов, подверженных перегрузке, узлов, отказывающих в обслуживании при увеличении нагрузки и т.д.).

Учитывая, что задачи анализа и синтеза сетевых структур средней и большой размерности являются NP-сложными [2 – 6], а для их решения часто приходится строить отдельную модель, объемы затрачиваемого на расчеты времени, различных физических ресурсов могут быть велики. Исследования в данной области ведутся с середины XX в., и выработано множество подходов для решения указанных выше задач, основными из которых являются:

- 1) вероятностные полиномиальные процедурные модели расчета, предложенные авторами [7 – 17];
- 2) процедурные модели, построенные с использованием элементов искусственного интеллекта (так называемая искусственная нейронная сеть ИНС, Artificial Neural Network, ANN), рассмотренные в работе [3];
- 3) потоковые модели, основанные на критерии допустимости СИС, предложенные авторами [4].

Принципиальная схема функционирования сетевой структуры формализуется известной математической моделью, которая называется многопродуктовой потоковой сетью (МП-сетью) и задается с помощью графа [4 – 6, 18 – 21].

Так как количество параметров модели велико и, к тому же, может варьироваться в зависимости от использования модели из п. 1 – 3, возникает потребность хранения большого количества данных, т.е. потребность в базе данных (БД) параметров модели. Кроме того, часто приходится анализировать не только модель самой сетевой структуры, но и модели неблаго-

приятных факторов (НФ), при воздействии которых система должна функционировать. НФ разделяют на внешние (ВнНФ) и внутренние (ВНФ). ВНФ моделирует отказы программных средств, а ВнНФ – действия всех НФ, лежащих вне системы [18]. После параметризации полученные данные по НФ также нужно хранить в специализированной БД.

Ситуация развития сетевой структуры с течением времени приводит к необходимости создания БД готовых решений (или моделей), к которым можно будет вернуться в дальнейшем, не производя параметризацию СИС снова. Совокупность перечисленных выше баз данных и связей (через параметры хранящихся в них моделей, так называемые знания о СИС) между ними можно представить как систему знаний (СЗ) об исследуемой сетевой структуре. Наличие же СЗ требует наличия системы поиска (поисковой системы) необходимой информации в БД № 1 – БД № N. Тип баз данных может отличаться, соответственно могут различаться и процедуры поиска в этих базах, следовательно, поисковая система должна быть унифицирована и состоять из набора систем поиска (система поиска 1... – система поиска N, в зависимости от количества БД) [18]. Для работы с полученным набором структур необходимо наличие пользовательского интерфейса. Для функционирования баз данных (БД № 1 – БД № N), СЗ, поисковой системы необходим комплекс программно-аппаратных средств (компьютерных систем, вычислительных сетей, сетей хранения данных (Storage Area Networks, SAN) или присоединенных сетевых хранилищ данных (Network – Attached Storage, NAS), систем обработки информации, СУБД и т.д.) [22].

Совокупность перечисленных систем, комплекса программно-аппаратных средств и интерфейса представляет собой информационную систему (ИС) – ИС оценки живучести сетевых структур (ИСЖС), представленную на рис. 1.1.

Функционирование ИСЖС можно представить следующим образом.

1. Пользователь вводит параметры модели СИС через пользовательский интерфейс и максимальное значение живучести (так называемый граничный критерий живучести) или делает соответствующие выборки из БД через пользовательский интерфейс и/или поисковую систему. Параметры модели могут также выбираться автоматически или автоматизированно по запросу модуля анализа текущего состояния (АТС), который может также быть подключен к интерфейсам существующей СИС. В таком случае возможно отслеживание состояния СИС в режиме реального времени и на основе анализа полученных от структуры данных и параметров изменение и принятие решений ИСЖС «на лету» [4]. Процедурную модель модуля АТС см. в параграфе 2.2, рис. 2.2.

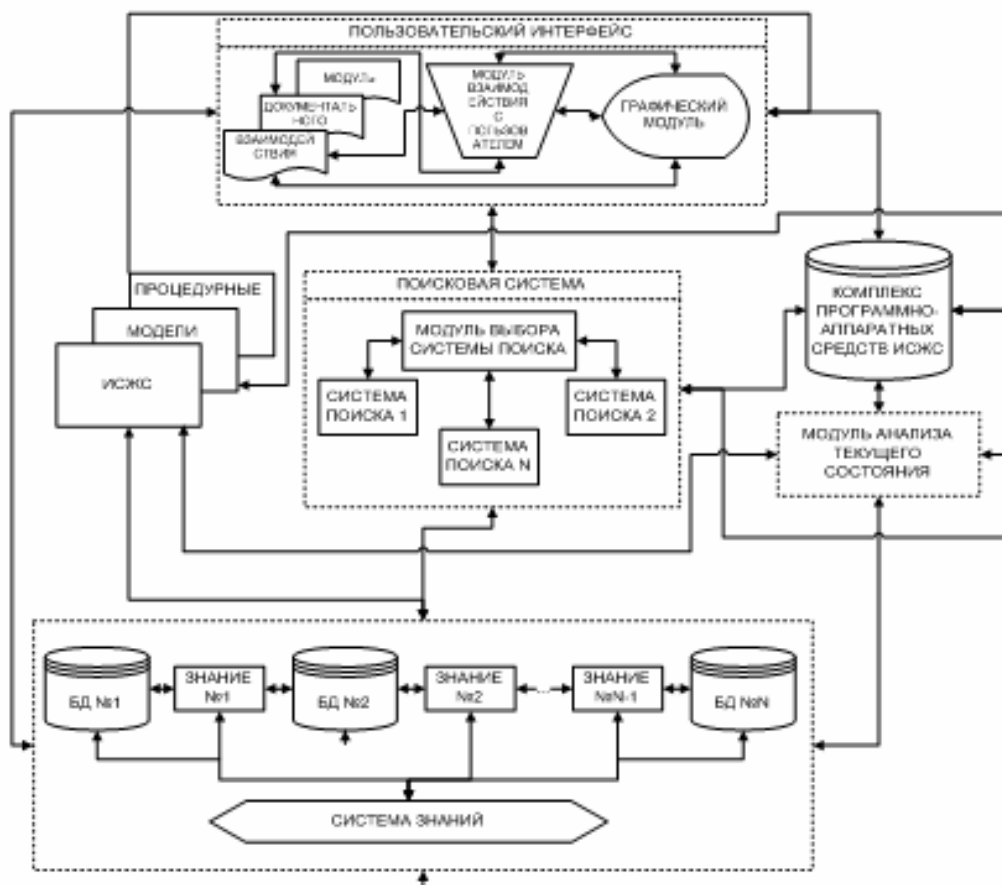


Рис. 1.1. Схема информационной системы для оценки живучести сетевых структур

Далее набор параметров передается в блок анализа модели. Выбор процедурной модели, по которой будет вестись расчет, зависит от топологии сетевой структуры, ее параметров и таких вычисляемых критериев, как, например, диаметр графа СИС [19] (см. параграф 2.2).

2. На этом шаге происходит расчет живучести СИС по выбранной модели, а также набор необходимых вычисляемых параметров. Процесс расчета контролируется модулем АТС.

3. Результаты вычисления живучести СИС вместе с исходным набором параметров передаются в блок синтеза СИС с «повышенной» живучестью, при этом графовая модель СИС будет отличаться от заданной пользователем (так называемая улучшенная модель).

4. На этом шаге производится проверка соответствия синтезированной модели по критерию, заданному пользователем на шаге 1. Если сравнение удовлетворительное, пользователю выдается готовое решение, иначе переходим на шаг 1, и предлагается изменить исходную модель СИС.

Схема функционирования ИСЖС представлена на рис. 1.2.

Таким образом, ИСЖС должна содержать в себе:

1. Систему знаний:
 - а) Набор БД (БД № 1 – БД № N), связанных между собой определенными параметрами хранящихся в них моделей СИС и НФ (знаниями о СИС, НФ и типах воздействия НФ на СИС, промежуточными значениями расчетов);
 - б) БД готовых решений.
2. Систему поиска информации по указанным в п. 1 БД.
3. Процедурную модель анализа СИС, в функции которой входят отнесение сетевой структуры к тому или иному типу и передача необходимых данных из системы знаний в выбранную процедурную модель для расчета.
4. Процедурные модели расчета живучести сетевых структур в зависимости от типа оцениваемой СИС.
5. Модуль анализа текущего состояния.
6. Процедурную модель синтеза сетевой структуры с учетом рассчитанных значений живучести.
7. Комплекс программно-аппаратных средств.
8. Пользовательский интерфейс (для ввода новых данных пользователем, задания критериев оценки и вывода результатов расчета пользователю).

Данная ИС позволяет проектировать, анализировать или отслеживать состояние существующей СИС без значительных затрат различных видов ресурсов, в том числе благодаря тому, что в ее структуре не используются процедурные модели прямого перебора (brute force models) [23 – 30], требующие повышенных затрат всех типов ресурсов.

Рис. 1.2. Схема функционирования информационной системы оценки живучести сетевых структур

Сетевая информационная система представляет собой распределенную структуру, размещенную на большой территории (рис. 1.3). Схема функционирования ее задается с помощью графа, который определяет физическую структуру СИС, его ребра r_i соответствуют физическим компонентам информационной СИС (таким, как каналы связи), проложенным от одной вершины графа (узла) V_i к другому.

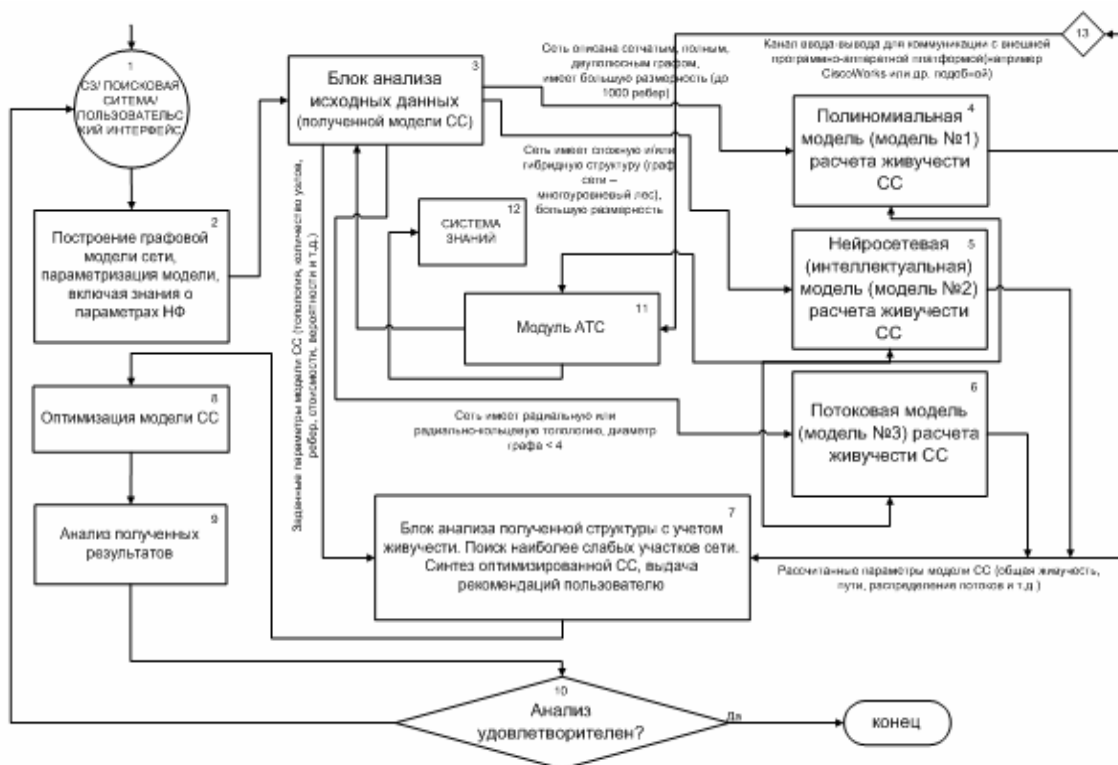


Рис. 1.2. Схема функционирования информационной системы оценки живучести сетевых структур

Узлы СИС соответствуют источникам/приемникам потоков либо осуществляют транзитные функции для существующих потоков. Такие вершины графа информационной СИС носят название транзитных. Совокупность ребер, которые надо прой-

ти потока из вершины V_i до вершины V_j , называется путем (V_i, V_j) . Если для любых двух вершин графа существует путь (V_i, V_j) , то граф называется связным (рис. 1.4, а), в противном случае граф не связан (рис. 1.4, б).

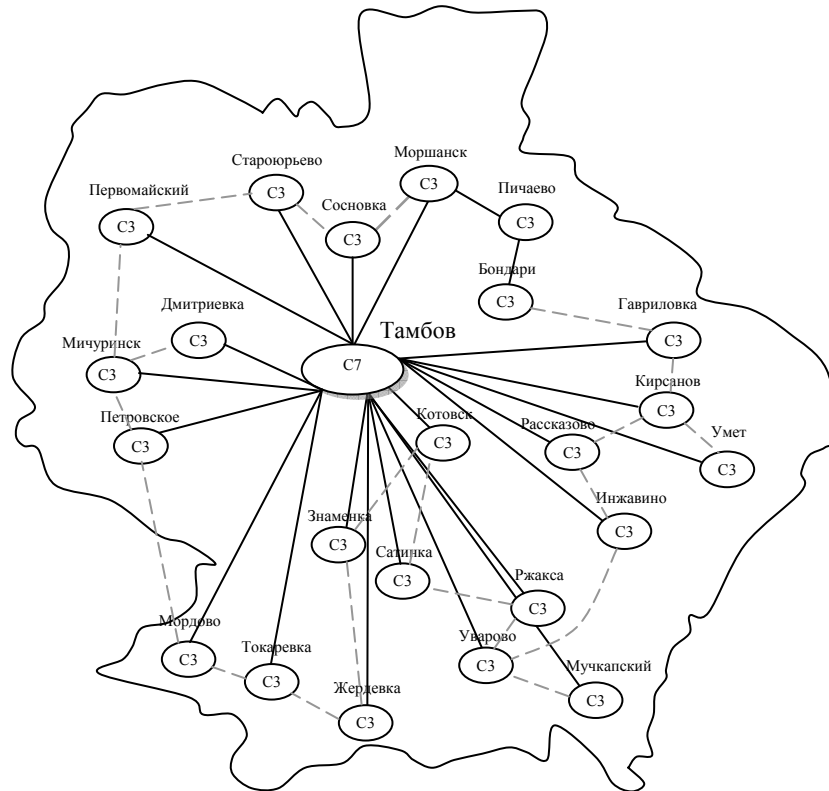


Рис. 1.3. Общая схема СИС:
С3 – Cisco 3xxx; С7 – Cisco 7xxx; черные линии – основные каналы связи между узлами, серый пунктир – резервные кольцевые структуры

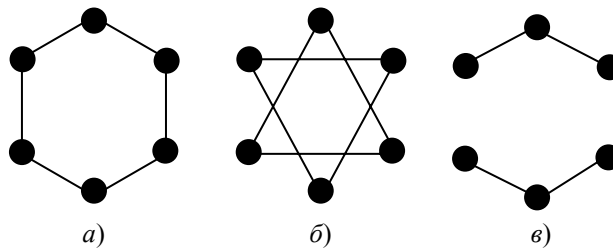


Рис. 1.4. Примеры связного (а), несвязного (б) графа и сечения графа по ребрам (в)

Каждое ребро, входящее в вершину или исходящее из нее, называется инцидентным этой вершине. Общее количество ребер $d(i)$, инцидентных вершине, называется степенью вершины. Если граф ориентирован, то различают полустепень исхода $d^+(i)$ и захода $d^-(i)$.

Сечением по ребрам (рис. 1.4, в) называют наименьшее количество ребер, удаление которых из графа разбивает последний на два несвязанных между собой компонента.

Разбиение множества V всех вершин графа на два подмножества $V_1 \cup V_2 = V$ называется разрезом по вершинам. Это разбиение определяет разрез по ребрам: $E_1 \cup E_2 = E$, где E_1 – множество всех ребер, выходящих из вершин V_1 и входящих в вершины V_2 [31].

Если элементу графа (ребру, вершине) приписана какая-либо физическая величина (например, длина ребра, пропускная способность, задержка обработки информации), эта величина отмечается числом, называемым весом элемента (ребра, вершины).

Длина пути между вершинами определяется матрицей расстояний $\|V_i, V_j\|$: выбирается j столбец и суммируются по i все длины ребер, расположенные в столбце.

Вершина (или точка на ребре), расположенная на наименьшем расстоянии от всех остальных вершин, называется медианой графа, медианное расстояние R – радиусом графа. Удаление даже одного ребра увеличивает радиус графа, так как в таком случае необходимо отыскать обходной и потому более длинный путь. Таким образом, удаление одного или даже нескольких ребер не всегда уничтожает связность графа. Такое свойство графа носит название живучести.

1.2. КРИТЕРИЙ ЖИВУЧЕСТИ ГРАФА. МАКСИМИЗАЦИЯ ЖИВУЧЕСТИ. УСЛОВИЕ СУЩЕСТВОВАНИЯ ФИЗИЧЕСКОГО ГРАФА СЕТЕВОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ

Удаление всех ребер, инцидентных некоторой вершине, изолирует ее, прерывая все пути к другим вершинам – граф становится не связным, живучесть графа – равной нулю.

Для обеспечения наибольшей живучести надо строить граф с наибольшей степенью $d(i)$ всех его вершин. Таким является полный граф, в котором каждая вершина связана ребром непосредственно с каждой другой вершиной. Число ребер $m = \frac{n(n-1)}{2}$, где n – число вершин. Каждая вершина имеет максимальную степень $d = n - 1$, но на практике не все вершины нуждаются в подобной «защите», такой усиленный граф нерентабелен [6, 20, 31, 32]. Необходимо создать такой граф из n вершин, чтобы каждая из них имела заданную ей степень K_{α_i} , $i = \overline{1, n}$, где $1 \leq K_{\alpha_i} \leq n - 1$.

Решение задачи возможно при использовании так называемого «сжимающегося множества» чисел [20, 31, 32].

Некоторое множество чисел $\{K_1, \dots, K_n\}$ при $n > 1$ реализуется в качестве множества степеней вершин ненаправленного графа G с n вершинами, и $d = k$ тогда и только тогда, когда $\{K_1, \dots, K_n\}$ последовательно сжимаема.

$$K_i \geq K, \quad i = \overline{1, n}. \quad (1.1)$$

Если $K = 1$, то $\sum_{i=1}^n K_i \geq 2(n-1)$. задается n вершин, строится последовательность чисел $n - 1, n - 2, \dots, 1$, из которой и выбираются необходимые значения степени d . Построение графа осуществляется последовательным соединением смежных вершин, не превышая при этом их степени (см. пример в прил. 1).

Введем следующие обозначения и допущения:

- n – число вершин V_i , $i = \overline{1, n}$, графа;
- m – число ребер (V_i, V_{i+1}) графа;
- $d(i)$ – степень вершины V_i графа;
- граф не имеет кратных ребер;
- все ребра графа имеют одинаковый вес, равный 1.

Условие связности графа: $m \geq n - 1$ [31].

$\sum_{i=1}^n d(i) = 2m$, так как каждое ребро инцидентно двум вершинам. Но распределение степеней по вершинам неоднородно, поэтому $\sum_i d(i) \geq n \min d(i)$, где $\min d(i)$ – наименьшее значение степени одной из вершин графа

$$2m \geq n \min d(i) \Rightarrow \min d(i) \leq \frac{2m}{n}. \quad (1.2)$$

Таким образом, в графе существует, по крайней мере, одна вершина с наименьшей степенью, не превышающей значение $v = \left\lfloor \frac{2m}{n} \right\rfloor$. Изъятие ребер, инцидентных этой вершине, разбивает граф на две компоненты $\{V_i\}$ и $\{V_1, \dots, V_{i-1}, V_{i+1}, \dots, V_n\}$. Следовательно, величину v можно принять за критерий живучести графа [20, 31].

В детерминированных сетевых задачах считается, что исследователю доступна полная информация о СИС. Однако, как правило, исследователю не известны ни число источников потоков, ни величины потоков, ни величины конечных задержек при передаче сообщений. Приходится довольствоваться вероятностной оценкой того или иного параметра. Имеется и другая составляющая неопределенности. Случается, что большой объем информации невозможно передать по одному каналу, тогда его делят на «пакеты», каждый из которых передается далее по произвольно выбранному свободному каналу с присущими последнему задержками, имеющими случайную природу [33 – 37]. СИС всегда функционирует в условиях неопределенности, но эта неопределенность индуцируется процессами, происходящими в самой СИС и потому поддающимися мониторингу, что позволяет осуществить статистическую обработку процессов.

Задержки каналов, узлов, отказы технических элементов могут привести к блокаде значительного участка СИС, но не разрушить ее. СИС останется живучей, но временно бездействующей.

Имеется иной тип воздействий на СИС – внешний, и потому совершенно непредсказуемый (например, события, носящие стихийный характер). Значительная часть СИС при таком типе воздействия может быть разрушена, но оставшиеся связанные между собою сегменты функционируют, пусть и в «усеченном» режиме. Исследованию процессов, протекающих при этом, посвящена теория связности и живучести СИС [21], использующая при этом только стохастические методы.

При теоретико-графовом рассмотрении сетевых задач не выделяется специальное множество тяготеющих пар, полученный при таком подходе критерий живучести может оказаться иллюзорным: связность графа не будет разрушена, но про-

дуктовый поток не осуществится – СИС будет недопустима. Более действенным представляется потоковый подход к анализу живучести СИС [19, 38, 39].

Любая из комбинаторно-графовых задач может быть рассмотрена путем прямого перебора, требующего больших временных затрат. Процедурные модели, не требующие перебора в той или иной форме, эффективны не для всех задач, а для многих из них процедурных моделей вообще не существует. Наиболее полный обзор процедурных моделей вычисления основных графовых характеристик, связанных с проблематикой живучести информационных СИС, осуществлен в работе [39]. Приведем некоторые из них:

1. К-связность как мера живучести СИС. Неориентированный граф $G = (V, R)$ называется k -связным относительно пары вершин $v', v'' \in V$, если после удаления любых $k - 1$ ребер обязательно останется путь, соединяющий вершины v', v'' . Граф G называется k -связным, если он является k -связным относительно каждой пары своих вершин. В k -связном графе для любой пары вершин существует не менее K реберно-непересекающихся путей их соединения. Основываясь на этих определениях, можно поставить задачу синтеза графа гарантированной высокой живучести [40]: задан граф $G = (V, R)$, для каждой пары вершин задано целое неотрицательное число $K(v', v'')$. Требуется в графе G найти подграф, в котором для любой пары узлов $v', v'' \in V$ существует не менее $K(v', v'')$ реберно-непересекающихся путей соединения. В общем виде сформулированная задача не поддается решению [40], но предложены многие эвристические методы [41, 42]. Для небольших значений K задача дополнения заданного графа до k -связного рассмотрена в работе [43].

2. Живучесть и диаметр графа СИС. В реально функционирующих информационных сетях используется ограничение на число переприемов одного сообщения. Соответственно, в модели такой СИС будут считаться связными только те пары узлов (вершин графа), между которыми существует путь, имеющий длину не более заданной. При анализе подобных СИС используется понятие диаметра графа.

Определение. Пусть в графе G найдены $L(V', V'')$ – длины кратчайших путей между всеми парами вершин $V', V'' \in V$. Тогда величину $L = \max_{V', V'' \in V} L(V', V'')$ называют диаметром графа. В работе [5] исследуется проблема сложности общей задачи отыскания максимального числа вершинно-(реберно-)непересекающихся путей ограниченной длины L . В частности, в работе отражено, что эффективное решение задачи существует только для графов с $L \leq 3$. При анализе живучести информационных СИС используют также верхние и нижние оценки диаметра графа. В работе [6] вводится обобщенное понятие диаметра $L' = \max_{x, y} \rho(x, y)$, где x, y – произвольные точки на ребрах графа СИС, а $\rho(x, y)$ – длина пути в графе между этими точками. В работе [19] приводится обширная библиография по оценкам живучести сетевых систем. Подобные оценки исследованы также в работах [44 – 47].

3. Условная связность. Введением понятия диаметра графа на понятие связности было наложено определенное ограничение.

Определение. Вершины графа СИС считаются связными, если длина соединяющего их пути не превосходит заданной величины. Имеется также обширный класс утверждений, рассмотренных в работе [48], в которых связность понимается в обычном смысле, но на компоненты графа, образующиеся в результате нарушения связности, налагаются дополнительные условия. Например, граф G считается условно связным, если удаление некоторого минимального числа ребер (вершин) оставляет в образовавшихся компонентах присущие исходному графу свойства: планарность, двудольность, заданную степень вершин и т.д., если критерием связности положить минимальное число ребер, удаление которых в каждой компоненте оставляет некоторое N_0 число вершин. Задача разбиения графа СИС на «одинаковые» компоненты рассматривается в работе [49].

4. Стойкость. При синтезе СИС максимальной живучести возникает вопрос о минимальной величине затрат, обеспечивающих эту живучесть, т.е. проблема стойкости. Стойкость численно равна наименьшей средней стоимости создания новой компоненты связности. Авторами работ [50, 51] было получено соотношение между стойкостью графа и его живучестью. Если стойкость графа $\sigma(G) \geq \sigma_0$, то граф содержит не менее σ_0 реберно-непересекающихся остовных деревьев. При этом было найдено важное приложение к определению живучести – вычисление плотности графа. Пусть граф $G' = (V', R')$ – подграф графа G . Плотностью $\rho(G')$ подграфа G' называется отношение мощности множества его ребер к мощности множества его вершин [52]:

$$\rho(G') = |R' / V'|.$$

Плотные графы являются менее уязвимыми.

5. Минимальный разрез как характеристика уязвимости СИС. Понятия сечения и разреза совпадают, но не всегда. Сечение – более общее понятие. При построении процедурной модели живучести СИС под воздействием НФ для моделирования полного разрушения структуры пытаются разделить тяготеющие пары (v_{s_i}, v_{t_i}) , $i \in M$, т.е. удалить множество таких ребер, что их удаление из СИС разрушает все пути соединения для всех тяготеющих пар (создают разрез). Пропускная способность такого разреза равна сумме пропускных способностей всех входящих в него ребер. Минимальный разрез – разрез с минимальной пропускной способностью (т.е. включающий в себя наименьшее число ребер). При моделировании считается, что именно этот разрез будет подвергнут наибольшему воздействию НФ и именно этот разрез укрепляют. Метод отыскания минимального разреза в общем виде неэффективен, лишь в некоторых специальных случаях его поиск сводится к простым комбинаторным задачам [40]. При разрушении части СИС происходит перераспределение (перемаршрутизация) потоков. Вопросы, связанные с перемаршрутизацией потоков после выхода из строя отдельных ребер, рассматриваются в работах [53,

54], а задача синтеза потоковой СИС с учетом живучести – в работах [55, 56]. Отмеченные выше показатели структурной живучести мало представительны: в основу их построения полагается лишь один из многих аргументов целевой функции живучести графа, как правило – связность. Отыскивается наиболее слабое звено графа, определяется минимальное сечение, которое, в той или иной степени, используется в выражении показателя живучести. Так называемое гарантированное значение живучести графа СИС задается наихудшим состоянием деформирования графа.

В работах [7 – 17] предлагается универсальное определение коэффициента живучести, при котором никакое разрушение ребер не приводит к потере связности.

При удалении ребер одна из вершин графа оказывается в изоляции, но, если ее заранее объединить с какой-либо устойчивой вершиной (stable node), связность графа удается сохранить. Продолжая далее этот часто употребляемый прием удаления и контракции ребер, сводят исходный граф G к тривиальной петле.

Если всем удалениям ребер приписать одинаковую вероятность удаления p , то контракция их будет иметь вероятность $q = 1 - p$.

В результате совокупности всех таких действий создается многочлен из произведений p и q различной степени. Численное значение такого многочлена при заданном значении p принимают за критерий живучести $R(G)$ графа G .

Приняв p и q за x и y , получают форму полинома Тутте:

$$\sum_{S,t} K_x x^{f_x(t,s)} K_y y^{f_y(t,s)}, \quad (1.3)$$

где K_y, K_x – константы; $f(s, t)$ – функции от параметров графа E, V .

$$T(G; x, y) = \sum_{E,A} (x-1)^{\rho(E)-\rho(A)} (y-1)^{|A|-\rho(A)}. \quad (1.4)$$

$$R(G) = \begin{cases} (1-p(e))R(G/e); \\ R(G \setminus e); \\ p(e)R(G \setminus e) + (1-p(e))R(G/e), \end{cases} \quad (1.5)$$

где «\» – знак удаления; «/» – контракции.

Применив $R(G)$ к графам с различным содержанием ребер, получают график зависимости живучести от вероятности удаления ребра (рис. 1.5).

Преимущества такой постановки очевидны: живучесть графа определяется только числом ребер и вероятностью их удаления при воздействии на физический граф СИС внешних НФ. Предлагаемая процедурная модель универсальна и применима к любым графам.

Численные значения критерия живучести, предлагаемые авторами [7 – 17], не совпадают с тем, что предлагают другие исследователи, использующие вероятностные и теоретико-графовые подходы, но едва ли это отличие велико, к тому же все оценки живучести весьма условны [4].

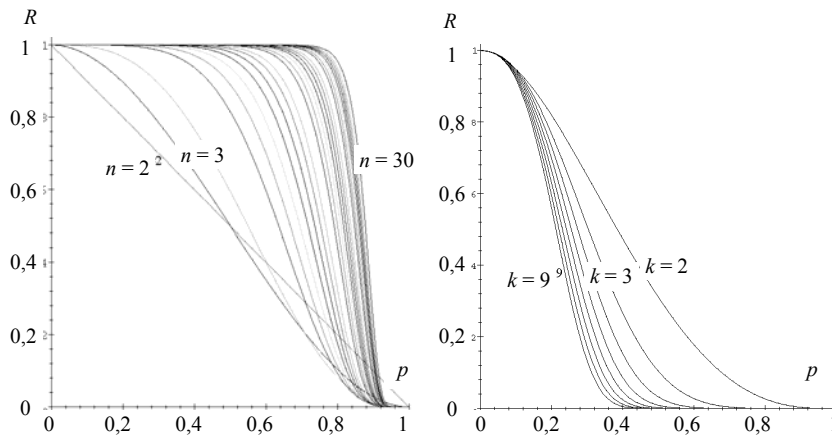


Рис. 1.5. График зависимости живучести от вероятности удаления ребра

1.3. ВЫЧИСЛЕНИЕ ОБЩЕЙ ЖИВУЧЕСТИ СЕТЕВОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ В ПОЛИНОМИАЛЬНОЙ ФОРМЕ

Киоко Секине (Kyoko Sekine) и Хироши Имаи (Hiroshi Imai) Департамента информационных наук Токийского университета предложили процедурную модель расчета живучести многополюсной СИС через полином Тутте (Tutte polynomial) для любого графа с максимум 14 вершинами и $\binom{14}{2} = 91$ ребром (полный граф), а также для планарного графа, например, решетчатого графа 12×12 , содержащего 144 вершины и 264 ребра.

Сама по себе живучесть СИС исследовалась на протяжении долгого времени (например, см. [10, 46]), и рассматривалось много видов живучести, а также процедурные модели для их расчета. Учитывая, что расчет живучести СИС – NP-сложная задача [12], интенсивно исследовалась приближительная процедурная модель расчета верхнего и нижнего предела живучести СИС [11].

Пусть $G = (V, E)$ – простой, связный, неориентированный граф с количеством вершин V и количеством ребер E . Рассмотрим СИС (граф) $G = (V, E)$. Каноническая живучесть многополюсной СИС $R(G, p)$ определяется как вероятность того, что G останется связным после того, как каждое ребро будет удалено с одинаковой вероятностью p .

$R(G, p)$ можно рассчитать с помощью перечисления остовов G . На практике это тесно связано с полиномом Тутте, который является инвариантом в теории графов. Полином Тутте графа G – это полином с двумя переменными $T(G; x, y)$, рассчитываемый по формуле (1.4):

$$T(G, x, y) = \sum_{A \subseteq E} (x-1)^{p(E)-p(A)} (y-1)^{|A|-p(A)}, \quad (1.6)$$

где $p: 2^E \rightarrow \mathbf{Z}$ – это ранжирующая (рэнкинговая) функция графа G . Это значит, что $p(A)$ – это ранг подграфа $G' = (V(A), A)$: количество вершин, $|V(A)|$, минус количество связных компонентов G' .

Исходя из определения полинома Тутте, его значения в определенных точках будут следующими:

- $T(G; 1, 1)$ рассчитывает количество остовных деревьев G , которое вычисляется полиномиально;
- $T(G; 2, 1)$ рассчитывает количество лесов G , которое NP-сложно для вычисления;
- $T(G; 1, 2)$ рассчитывает количество остовов G , которое также NP-сложно для вычисления.

Чтобы получить информацию о других смыслах и множестве вариантов применения полинома Тутте, см. [9, 10].

Связь между канонической живучестью СИС и полиномом Тутте рассчитывается по формуле:

$$\begin{aligned} R(G, p) &= \sum_{A \subseteq E, p(A)=p(E)} (1-p)^{|A|} p^{|E-A|} = \\ &= p^{|E|-p(E)} (1-p)^{p(E)} \sum_{A \subseteq E, p(A)=p(E)} \left\{ (1-p) \frac{1}{p} \right\}^{|A|-p(A)} = \\ &= p^{|E|-p(E)} (1-p)^{p(E)} T\left(G, 1, \frac{1}{p}\right). \end{aligned} \quad (1.7)$$

Заметим, что когда $p = 1/2$, оно точно соответствует подсчету количества остовов. Так как полином Тутте имеет следующую рекурсивную формулу, каноническая живучесть многополюсной СИС можно вывести из следующего соотношения:

$$T(G, x, y) = \begin{cases} xT(G/e, x, y); \\ yT(G \setminus e, x, y); \\ T(G \setminus e, x, y) + T(G/e, x, y). \end{cases} \quad (1.8)$$

Здесь для ребра $e \in E$ мы обозначаем с помощью $G \setminus e$ граф, получаемый удалением e из G , а с помощью G/e – граф, получаемый контракцией e из G . Петля – это ребро, соединяющее одну и ту же вершину, а копетля – это ребро, удаление которого уменьшает ранг графа на 1. По определению, полином Тутте для петли – это y , а для копетли – x . Полином Тутте для пустого графа равен 1.

Общая живучесть СИС в полиномиальной форме. Пусть $p(e)$ – заданная вероятность удаления ребра $e \in E$. Тогда общая живучесть СИС определяется как вероятность того, что граф останется связным после того, как каждое ребро e будет удалено с вероятностью $p(e)$. Эта живучесть будет просто определяться как $R(G)$.

В этом общем случае мы не находим никакой связи с полиномом Тутте. Тем не менее, его все-таки можно рассчитать перечислением остовов. Тогда в виде полинома Тутте будет использоваться следующая формула удаления/контракции ребра.

Утверждение 1. Для ребра e

$$R(G) = \begin{cases} (1-p(e))R(G/e); \\ R(G \setminus e); \\ p(e)R(G \setminus e) + (1-p(e))R(G/e). \end{cases} \quad (1.9)$$

Обоснование утверждения 1 приведено в прил. 3.

Процесс расчета с использованием этой формулы может быть представлен в виде двоичного дерева, в котором корень соответствует начальному графу G и у каждого предка есть максимум два потомка. Узлы в этом двоичном дереве соответствуют минорам G . Здесь граф, получаемый из G путем удаления и контракции ребер, называется минором G . Для каждого маршрута от корня к конечной вершине ряд контрактированных ребер однозначно соответствует остовному дереву G . Тогда количество листьев равно количеству остовных деревьев.

Более того, для каждого маршрута ребра, состоящие из удаленных петель, соответствуют внешне активным ребрам соответствующего остовного дерева. Здесь мы говорим, что e_j с концевыми вершинами a и b является внешне активным для остовного дерева T , если для каждого e_k , которое является ребром в простом маршруте в T , соединяющим a и b , выполняется условие, что $k < j$ [12].

Расчет с помощью формул контракции и удаления ребра. Новый подход к расчету полинома Тутте графа, используя тот факт, что многие изоморфные миноры появляются в этом расчете согласно рекурсивной формуле, предложен авторами Sekine, Imai, Tani [31].

Легко заметить, что эту процедурную модель можно использовать для расчета живучести СИС в общем случае. Путем разделения изоморфных миноров расширительное дерево видоизменяется в корневой ациклический граф с одним источником (корнем) и одним стоком (ребра ориентированы от корня к стоку). Например, рис. 1.6 отражает расчет живучести многополюсной СИС для заданного графа. На этой схеме p_i обозначает вероятность удаления ребра e_i , а $q_i = 1 - p_i$. Этот ациклический граф тесно связан с БСПР (бинарной схемой принятия решений: средство работы с булевой функцией, предложенной авторами в работе [15]), репрезентирующей все остовные деревья заданного графа. Это потому, что, как отмечалось выше, каждый маршрут однозначно соответствует остовному дереву графа.

Исходя из этого, живучесть многополюсной СИС в общем случае может быть рассчитана по следующей процедурной модели, описанной в параграфе 2.2.

Для двух заданных графов, так как нет эффективной процедурной модели для определения того, изоморфны они или нет, мы ограничимся следующим нахождением изоморфных миноров при таком же расположении ребер.

Сначала для каждого графа зададим ребра e_1, e_2, \dots, e_m ($m = |E|$). Затем мы применим формулу для контракции/удаления ребер в том же порядке. Предположим, что $E_i = \{e_1, e_2, \dots, e_i\}$ и $\bar{E}_i = \{e_{i+1}, e_{i+2}, \dots, e_m\}$. Тогда миноры G на уровне i имеют совокупность ребер \bar{E}_i . Для $i = 1, \dots, m$ определим фронт исключения на i -уровне V_i как подмножество вершин, состоящее из вершин v , таких, что v инцидентны некоторым ребрам в E_i и некоторым ребрам в \bar{E}_i . Ребрами, ограниченными в данном процессе, задается отношение эквивалентности на V_i , такое, что две вершины находятся в одном и том же классе эквивалентности, только если в процессе получения минора они унифицируются в одну вершину путем ограничений.

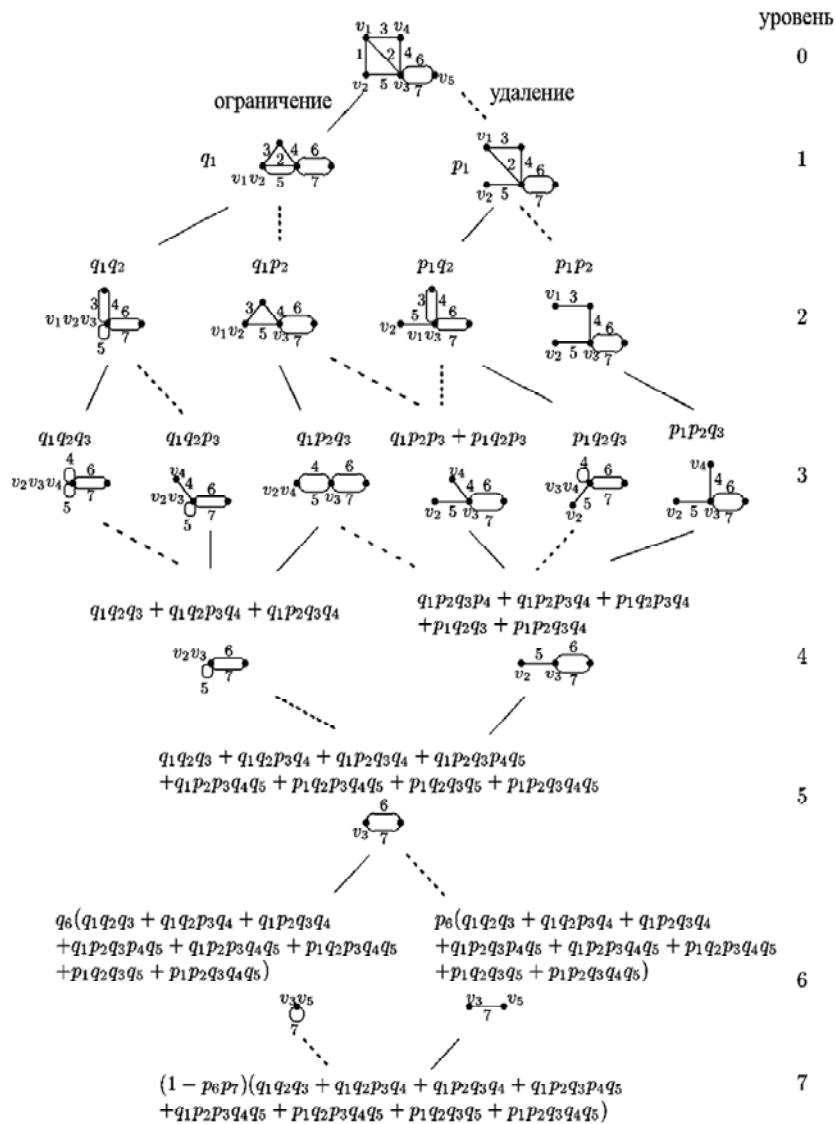


Рис. 1.6. Графическая схема аналитической модели процесса вычисления полинома общей живучести СИС

Далее рассмотрим разделение v_i на классы эквивалентности с помощью этого отношения. Мы называем это разделение разделением исключения минора на уровне i . Например, на рис. 1.6 фронтом исключения третьего уровня является $\{v_2, v_3, v_4\}$, так как все инцидентные ребра v_1 подверглись контракции или удалению. Когда e_1 и e_2 подвергаются контракции, а e_3 удаляется, v_2 и v_3 унифицируются в одну вершину, а разделение исключения этого минора будет иметь вид $\{\{v_2, v_3\}, \{v_4\}\}$.

Используя эти определения, получаем следующее.

Утверждение 2. Пусть H_1 и H_2 – два минора G с одинаковой совокупностью ребер \bar{E}_i . H_1 и H_2 изоморфны и имеют одинаковое расположение ребер, только если их разделения исключения на уровне i являются идентичными.

Обоснование утверждения 2 приведено в прил. 4.

Размер ациклического графа определяется количеством миноров в нем. Ширина ациклического графа считается максимальной среди количества миноров на каждом уровне. Следует отметить, что по отношению к рабочей области памяти эта процедурная модель может быть применена пропорционально размеру ациклического графа. Глубина ациклического графа – это количество ребер. Тогда важной является ширина ациклического графа. Мы продемонстрировали некоторые числовые способы расчета размера и ширины ациклического графа для полных графов и решетчатых графов [11].

Утверждение 3. Пусть l – максимальный размер фронта исключения. Тогда ширина процесса расчета ациклического графа ограничивается B_l , где B_l – называемое числом Белла – это число разделений l элементов.

Общая живучесть двухполюсной СИС в полиномиальной форме. При заданной паре двух вершин s и t живучесть двухполюсной СИС $R(G|s, t)$ определяется как вероятность, что будет существовать маршрут, соединяющий s и t в графе, остающемся после того, как каждое ребро e удалено с вероятностью $p(e)$.

Вначале мы модифицируем данный граф G в G' путем представления двух новых вершин s' и t' и соединения их ребрами s и t (причина объясняется ниже). Вероятность удаления этих двух ребер принимается равной нулю. Более того, предположим, что расположение этих двух ребер является последним для следующей рекурсивной формулы. Тогда двухполюсная живучесть между s и t эквивалентна живучести между s' и t' .

Мы называем ребро e активным, если существует простой маршрут, соединяющий s' и t' и содержащий e . Иначе мы называем его неактивным. Согласно этому определению, получается следующая рекурсивная формула.

Утверждение 4. Для ребра e , кроме вновь добавленных двух ребер:

$$R(G'|s', t') = \begin{cases} R(G' \setminus e|s', t'); \\ (1 - p(e))R(G' / e|s', t'); \\ p(e)R(G' \setminus e|s', t') + (1 - p(e))R(G' / e|s', t'). \end{cases} \quad (1.10)$$

Обоснование утверждения 4 приведено в прил. 5.

Рассматривая декомпозиционное дерево для двусвязных компонентов, мы можем решить, является ли каждое ребро активным или неактивным в G' . Здесь каждый узел декомпозиционного дерева соответствует двусвязному компоненту или точке сочленения. Если точка сочленения расположена в каком-либо двусвязном компоненте, существует ребро между двумя соответствующими узлами. Оба компонента, которые включают s' или t' , называются активными. Другие компоненты считаются активными, если их соответствующие узлы в декомпозиционном дереве находятся на маршруте между двумя узлами, соответствующими двум компонентам, содержащим s' и t' ; другие компоненты неактивны. Тогда ребро e в G' активно, если это – активный компонент; другие ребра неактивны. Это приводит к следующему утверждению.

Утверждение 5. Дана декомпозиция двусвязных компонентов G' с маркированными компонентами на маршруте между s' и t' . Является ли ребро активным или неактивным можно проверить в режиме реального времени.

Декомпозиционное дерево должно поддерживаться для удаления активного ребра, так как возможно, что некоторые активные ребра в одном и том же компоненте станут неактивными ребрами. Неактивные компоненты абсолютно нерелевантны для живучести двухполюсной СИС между s' и t' .

Причина, почему нужно добавлять два новых ребра, состоит в исключении следующей возможности: когда все инцидентные ребра s подверглись контракции или удалению (самое большее одно ребро подверглось контракции по определению), вершина s выходит из границ фронта исключения и объединяется с другой вершиной. Тогда, даже если два минора имеют одинаковое разделение исключения на уровне i , возможно, что вершины, которые унифицируются к вершине s , различны. В этом случае одно из активных ребер в одном миноре может быть неактивным в другом миноре. Например, на третьем уровне (см. рис. 1.6), если e_8 и e_9 не добавляются, минор, который определяется ограничением ребра e_1 и удалением e_2, e_3 , и минор, который определяется ограничением e_2 и удалением e_1, e_3 , изоморфны с одинаковым расположением ребер. Однако e_5 активно в предшествующем миноре, но неактивно в другом миноре.

Процедурная модель расчета полинома Тутте может быть также использована для расчета живучести двухполюсной СИС (рис. 1.7).

В отличие от случая многополюсной СИС, когда e – неактивная копелля, e удаляется по рекурсивной формуле. Однако один из двух компонентов, которые содержат точки сочленения, соединенные e , неактивны в G' .

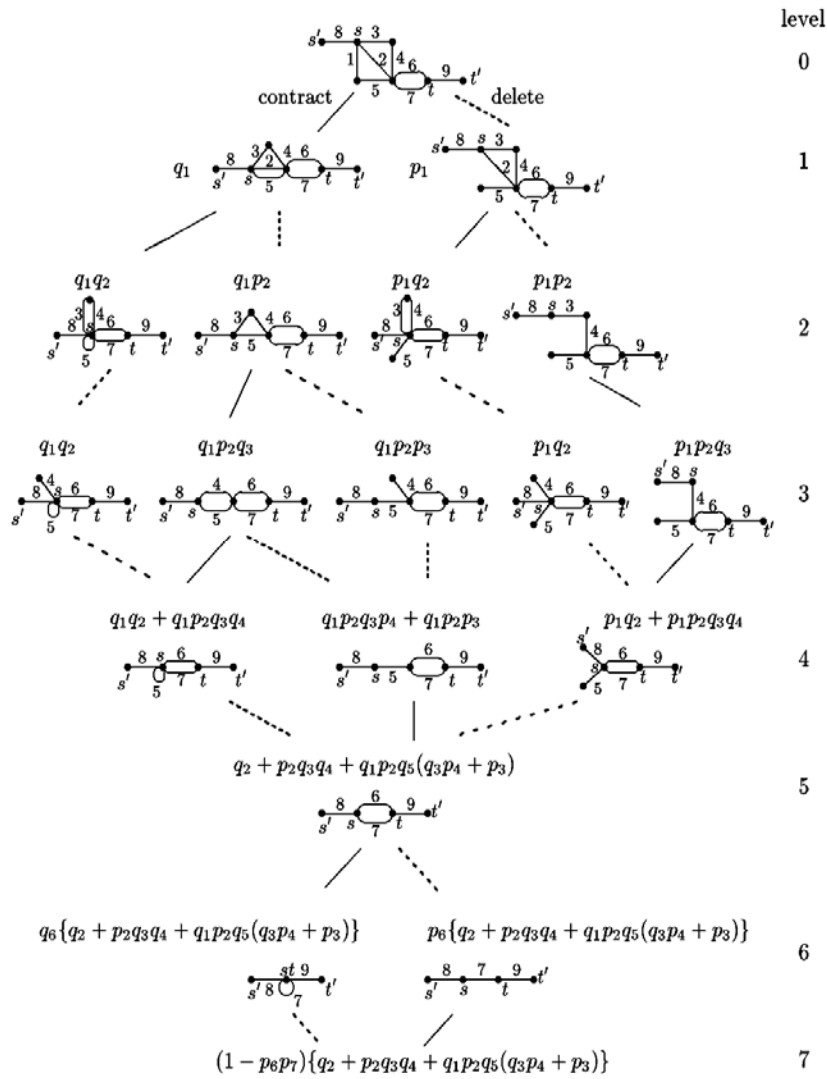


Рис. 1.7. Схема процесса расчета живучести двухполюсной СИС

Этот компонент все еще неактивен в $G' \setminus e$ и G' / e . Тогда справедлива следующая формула:

$$R(G'|s', t') = R(G' \setminus e|s', t') = R(G' / e|s', t'). \quad (1.11)$$

В случае двухполюсной СИС, так как существуют неактивные ребра, вся структура ациклического графа для случая многополюсной СИС не может использоваться в данном виде расчета. В отличие от случая многополюсной СИС, каждый маршрут от корня до стока не соответствует простому маршруту, однозначно соединяющему s и t , хотя множество ограниченных ребер для каждого маршрута включает такой маршрут (например, на рис. 1.6 $\{e_2, e_6\}$ – один из простых маршрутов, соединяющих s и t). Однако существуют два соответственных множества, состоящих из ограниченных ребер $\{e_1, e_2, e_6\}$ и $\{e_2, e_6\}$. Может ли такой ациклический граф быть составлен, является открытым вопросом (Неизвестно, как составить БСПР, которая бы прямо подсчитывала количество простых маршрутов) [10, 14]. Если такой ациклический граф может быть составлен, мы решим эту задачу более просто.

Размер ациклического графа зависит от расположения ребер. Правильное расположение ребер для уменьшения размера ациклического графа зависит от существования малых разделителей [57]. Для планарных графов такое расположение ребер существует согласно теореме планарных разделителей [7]. Суммируя все вышеизложенное и используя результат [14], мы получаем следующее утверждение.

Утверждение 6. Как живучесть многополюсной СИС, так и живучесть двухполюсной СИС с общей вероятностью удаления ребра может быть рассчитана с помощью действий, пропорциональных размеру ациклического графа процесса расчета. Для планарного графа с n вершинами она может быть рассчитана для $O(2^{O(\sqrt{n})})$ периода времени.

Наконец, живучесть графа ориентированной СИС имеет связь с гридоидами (greedoids) и их полиномом Тутте (см. [16]).

Верхний предел живучести СИС. Расчет полинома Тутте для полного графа. Прямой метод рекурсивного расчета полинома Тутте для полного графа был предложен Аннаном (Annan) [57]. Он заключается в следующем. Рассмотрим граф $U_{m,r}$, полученный из K_m добавлением новой вершины v и соединением ее с каждой вершиной K_m с помощью r кратных ребер. По определению, K_n изоморфен $U_{n-1,1}$:

$$\begin{aligned} T(U_{m,r}, x, y) &= \\ &= \sum_{i=1}^m \binom{m}{i} (y^{r-1} + y^{r-2} + \dots + 1)^i y^{\binom{i}{2}} T(U_{m-i,i}, x, y) + (x-1)T(U_{m-1,1}, x, y). \end{aligned} \quad (1.12)$$

Эта формула соответствует рекурсивному применению формулы удаления/контракции ко всем ребрам, примыкающим к v и объединяющим все изоморфные миноры. Например, применим формулу удаления/контракции к $n - 1$ ребрам, которые инцидентны вершине K_n . Как и в предыдущей процедурной модели, если мы объединим изоморфные миноры с одинаковым расположением ребер, то получим $2^{(n-1)} - n + 1$ неизоморфных миноров. Однако эта формула предполагает дальнейшее объединение изоморфных миноров с различным расположением ребер. В этом случае мы получаем, что есть только $n - 1$ неизоморфных миноров.

Заметим, что равенство $T(K_n, x, y) = T(U_{n-1, 1}, x, y)$ получаем из расчета всех $T(U_{j, k}, x, y)$, таких, что $j + k < n - 1$. По определению, для полного графа K_n наивысшая степень в x меньше, чем n , а в y меньше, чем n^2 . Тогда существует максимум $O(n^3)$ термов. $T(G, 1, 1)$ означает количество остовных деревьев, и для K_n это равняется n^{n-2} . Таким образом, каждый коэффициент можно записать с помощью максимум $O(n \log n)$ битов.

Функция живучести для полного графа. Применяя вышеупомянутую формулу, мы можем эффективно рассчитать функцию живучести полного графа с одинаковой вероятностью удаления ребер.

Утверждение 7.

$$R(U_{m,r}, p) = \sum_{i=1}^m \binom{m}{i} (1-p^r)^i p^{r(m-i)} R(U_{m-i,i}, p), \quad (1.13)$$

где $R(U_{0,r}, p) = 1$.

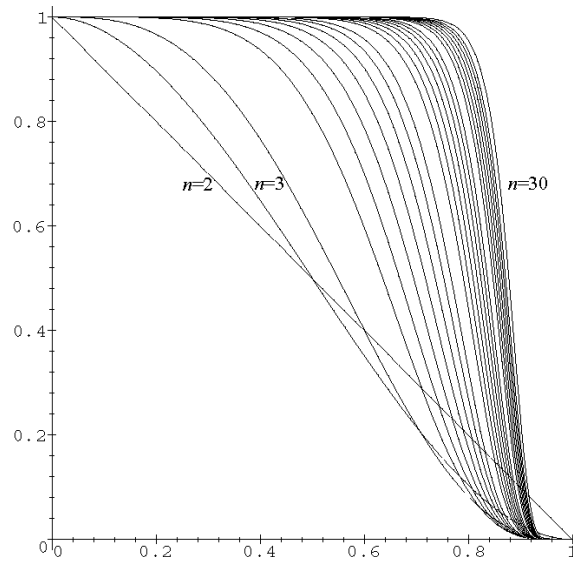
Обоснование утверждения 7 см. в прил. 6.

Практические результаты расчета приведены в табл. 1.1 и на рис. 1.8. На рис. 1.8 каждая кривая представляет собой верхний предел для других простых связанных графов с таким же количеством вершин. Например, рис. 1.9 отражает случай $k \times k$ решетчатых (или сетчатых) графов.

1.1. Полином живучести $R(K_n, p)$

n	Полином живучести $R(K_n, p)$
2	$-p + 1$
3	$2p^3 - 3p^2 + 1$
4	$-6p^6 + 12p^5 - 3p^4 - 4p^3 + 1$
5	$24p^{10} - 60p^9 + 30p^8 + 20p^7 - 10p^6 - 5p^4 + 1$
6	$-120p^{15} + 360p^{14} - 270p^{13} - 90p^{12} + 120p^{11} + 20p^9 - 15p^8 - 6p^5 + 1$
7	$720p^{21} - 2520p^{20} + 2520p^{19} + 210p^{18} - 1260p^{17} + 210p^{16} - 70p^{15} + 210p^{14} - 35p^{12} + 42p^{11} - 21p^{10} - 7p^6 + 1$
8	$-5040p^{28} + 20160p^{27} - 25200p^{26} + 3360p^{25} + 12810p^{24} - 5040p^{23} - 1960p^{21} + 420p^{20} + 560p^{19} - 336p^{18} + 336p^{17} - 35p^{16} - 56p^{15} + 56p^{13} - 28p^{12} - 8p^7 + 1$
9	$40320p^{36} - 181440p^{35} + 272160p^{34} - 90720p^{33} - 128520p^{32} + 90720p^{31} - 2520p^{30} + 15120p^{29} - 11340p^{28} - 7000p^{27} + 5544p^{26} - 4536p^{25} + 1386p^{24} + 1008p^{23} - 504p^{21} + 378p^{20} - 84p^{18} + 72p^{15} - 36p^{14} - 9p^8 + 1$
10	$-362880p^{45} + 1814400p^{44} - 3175200p^{43} + 1663200p^{42} + 1247400p^{41} - 1489320p^{40} + 201600p^{39} - 75600p^{38} + 189000p^{37} + 65100p^{36} - 105840p^{35} + 60480p^{34} - 27930p^{33} - 11970p^{32} + 5040p^{31} + 5040p^{30} - 5040p^{29} + 1260p^{28} + 1680p^{27} - 126p^{25} - 930p^{24} + 720p^{23} - 120p^{21} + 90p^{17} - 45p^{16} - 10p^9 + 1$

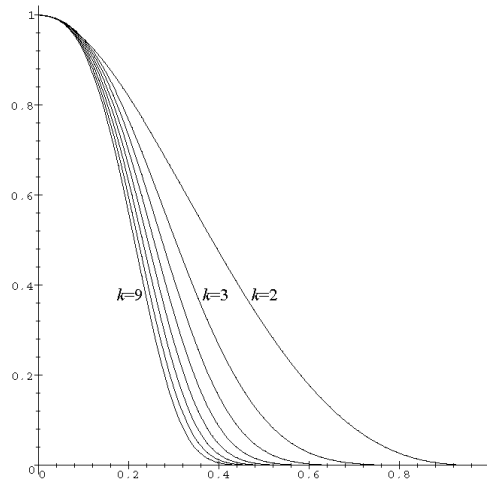
Живучесть $R(K_n, p)$



Вероятность удаления ребра p

Рис. 1.8. Диаграмма живучести для полинома $R(K_n, p)$ графа, содержащего n ребер ($n = 2, \dots, 30$)

Живучесть $R(L_{k \times k}, p)$



Вероятность удаления ребра p

Рис. 1.9. Диаграмма живучести для полинома $R(L_{k \times k}, p)$ графа, содержащего k ребер и n вершин (сетчатый граф) ($n = 2, \dots, 30; k = 2, \dots, 9$)

1.2. Полином живучести $R(L_{k \times k}, p)$

k	Полином живучести $R(L_{k \times k}, p)$
2	$-3p^4 + 8p^3 - 6p^2 + 1$
3	$79p^{12} - 560p^{11} + 1668p^{10} - 2656p^9 + 2331p^8 - 960p^7 + 96p^5 + 21p^4 - 16p^3 - 4p^2 + 1$
4	$-17\,493p^{24} + 232\,144p^{23} - 1\,409\,764p^{22} + 5\,168\,576p^{21} - 12\,693\,232p^{20} + 21\,854\,512p^{19} - 26\,726\,036p^{18} + 22\,824\,576p^{17} - 12\,739\,373p^{16} + 3\,710\,880p^{15} + 139\,672p^{14} - 37\,0176p^{13} - 35\,464p^{12} + 63\,968p^{11} + 5912p^{10} - 7808p^9 - 1791p^8 + 656p^7 + 204p^6 + 64p^5 - 8p^4 - 16p^3 - 4p^2 + 1$
5	$32\,126\,211p^{40} - 681\,809\,240p^{39} + 6\,852\,471\,548p^{38} - 43\,322\,118\,652p^{37} + 192\,968\,405\,711p^{36} - 642\,590\,690\,400p^{35} + 16\,559\,334\,579\,666p^{34} - 3\,370\,276\,114\,636p^{33} + 5\,476\,061\,558\,391p^{32} - 7\,122\,774\,813\,980p^{31} + 7\,375\,859\,530\,466p^{30} - 5\,981\,426\,876\,044p^{29} + 3\,667\,377\,815\,630p^{28} - 1\,573\,096\,624\,396p^{27} + 375\,423\,772\,810p^{26} + 9\,584\,416\,484p^{25} + 26\,112\,103\,320p^{24} - 6\,268\,146\,140p^{23} + 8\,011\,274\,210p^{22} - 1\,051\,500\,660p^{21} - 575\,028\,980p^{20} - 53\,196\,700p^{19} + 139\,031\,550p^{18} - 2\,265\,380p^{17} - 10\,705\,120p^{16} - 3\,593\,556p^{15} + 1\,357\,510p^{14} + 394\,172p^{13} + 35\,042p^{12} - 49\,636p^{11} - 10\,290p^{10} - 2036p^9 + 1021p^8 + 164p^7 + 250p^5 + 64p^4 - 11p^3 - 20p^2 - 4p + 1$

1.4. ВЫЧИСЛЕНИЕ ЖИВУЧЕСТИ СЕТЕВЫХ ИНФОРМАЦИОННЫХ СИСТЕМ С ИСПОЛЬЗОВАНИЕМ ЭЛЕМЕНТОВ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

Точный расчет общей живучести СИС представляет собой NP-сложную задачу, затраты на решение которой возрастают экспоненциально с ростом числа узлов и связей СИС. Из-за громоздкости расчета общей живучести для СИС больших и средних размеров в качестве альтернативного используется метод моделирования Монте-Карло для оценки живучести СИС в верхних и нижних пределах.

Предлагается другая альтернатива для оценки живучести СИС – прогностическая модель искусственной нейросети. Нейронные СИС конструируются, тренируются и обосновываются с использованием альтернативной топологии, цель которой – общая надежность верхних связей и живучесть конкретной цепи. Используется иерархический подход, т.е. за общей нейронной сетью, закрывающей конструкцию всех СИС из соображений живучести, следует специальная нейронная СИС, придающая особую живучесть конструкции СИС. Представлены результаты решения для задачи из десяти узлов с использованием подхода перекрестной оценки групп.

Живучесть и цена – это два главных соображения при расчете коммуникационных СИС, особенно при расчете опорных телекоммуникационных СИС, глобальных и местных СИС, а также СИС передачи данных на промышленных объектах. Если узлы (станции, терминалы или компьютерные центры) СИС фиксированные, тогда главными конструктивными решениями должны стать выбор типа и маршрутизация связей (кабелей и линий) СИС с целью обеспечения должной надежности при ограниченных расходах. Задача предполагает следующие допущения:

- 1) задается размещение каждого узла СИС;
- 2) узлы достаточно надежны;
- 3) стоимость узла и живучесть фиксированы и известны;
- 4) каждое звено двунаправлено;
- 5) СИС не содержит избыточных звеньев;
- 6) звенья либо рабочие, либо поврежденные;
- 7) повреждения звеньев независимы;
- 8) ремонт не рассматривается.

Математически проблема может быть описана следующим образом:

$$\min Z(x) = \sum_{i=1}^{N-1} \sum_{j=i+1}^N c_{ij} x_{ij} + \delta (c_{\max} (R(x) - R_0))^2, \quad (1.14)$$

где
$$\delta = \begin{cases} 0, & \text{если } R(x) \geq R_0; \\ 1, & \text{если } R(x) < R_0; \end{cases} \quad (1.15)$$

N – количество узлов; (i, j) – связь между узлом i и j ; x_{ij} – переменная принятия решения, $x_{ij} \in \{0, 1\}$; x – топология связей вида $\{x_{11}, x_{12}, x_{13}, \dots, x_{ij}, \dots, x_{N,N-1}\}$; $R(x)$ – живучесть СИС; R_0 – требование к живучести СИС (минимальное значение живучести); Z – целевая функция; c_{ij} – стоимость связи (i, j) ; c_{\max} – максимальное значение c_{ij} .

Термин $R(x) - R_0$ исключает те СИС, которые не удовлетворяют требованиям минимальной живучести, и направляет поиски на ряд пригодных СИС.

Проблема конструирования СИС изучалась в литературе как численными методами (обычно сочетанием методов вершин и ребер) [58], так и эвристическими [3, 59 – 64]. Одной из особенностей этих методов является то, что живучесть должна рассчитываться для каждой из выбранных конструкций СИС, а часто это бывают тысячи и миллионы конструкций. Таким образом, предстоит найти альтернативу расчету общей живучести СИС для СИС реальных размеров.

Проблема конструирования СИС особенно осложняется, когда речь идет о расчете общей живучести СИС, основанном на предположении, что все узлы могут связываться друг с другом. Это равно постоянной доступности, когда время миссии не берется в расчет. Сложность возникает потому, что расчет живучести СИС представляет собой NP-сложную задачу, т.е. сложность расчета возрастает с ростом самой СИС [64]. Именно поэтому для реальных СИС точный расчет живучести не всегда практичен. Стохастический метод моделирования Монте-Карло может дать достаточно точное представление о живучести СИС [65, 66], однако моделирование должно выполняться несколько раз, чтобы обеспечить надежность оценки. Отсюда следует, что моделирование при расчете живучести СИС (т.е. коммуникационных систем) тоже требует значительных усилий.

Нейронные СИС возникли благодаря силе, гибкости и надежности человеческого мозга. Это компьютерные математические аналоги основных биологических компонентов человеческого мозга – нейронов, синапсов и дендритов. Искусственные нейронные СИС (далее ИНС) состоят из многих простейших вычислительных элементов (суммирующих блоков – нейронов и весовых соединений – весов), которые могут работать параллельно или последовательно (рис. 1.10). Нейронная СИС начинается с произвольного состояния и «обучается» с использованием повторяющейся обработки тренируемого действия, т.е. ряда вводов с целевым выводом. Обучение происходит потому, что вычисляемая погрешность между вводом и целевым выводом используется для настройки весовых синапсов ИНС. Это повторяется до полной минимизации погрешности или до тех пор, пока веса уже больше не изменяются. В этом случае ИНС натренирована и веса фиксируются. Натренированная ИНС может быть использована для новых вводов с целью решения задач оценки или классификации.

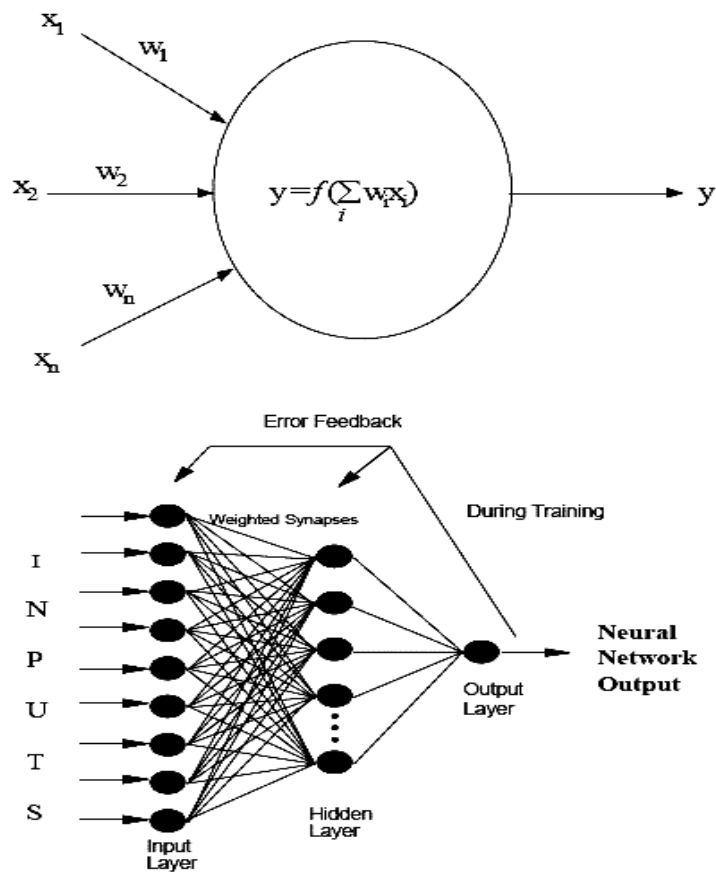


Рис. 1.10. Типичные компоненты и структура ИНС

Поскольку исходным прототипом стал человеческий мозг, ИНС можно рассматривать как статистику, так как имеется много реальных и важных параллелей между областью статистики и областью ИНС [245, 246]. Процесс тренировки ИНС с использованием набора данных сходен с вычислением векторно-значимой статистики с применением того же набора данных.

Так же, как коэффициент регрессии уравнения (т.е. наклоны и пересечения) вычисляется через минимизирование квадрата погрешности (RMSE) для набора данных, так и веса ИНС определяются через минимизирование погрешности для набора данных. Но между ИНС и статистикой имеется важное отличие. У ИНС имеется много свободных параметров (т.е. весовых соединений). ИНС с пятью входами, промежуточным скрытым слоем из пяти нейронов и одним выходом имеет 36 обучаемых весов, где простое умножение линейной регрессии дает шесть (пять наклонов и одно прерывание). ИНС может также свободно иметь избыточные параметры. Но есть опасность и чрезмерной подгонки модели ИНС [67]. Чрезмерно подогнанная ИНС будет очень зависеть от набора данных, использованных для ее создания, и будет слабо отражать внутреннюю взаимосвязь (распределение). В силу этого оценка ИНС через ее тренировку посредством набора данных не используется.

Важной характеристикой ИНС при определенных условиях является их способность быть универсальными аппроксиматорами [68 – 70]. Это означает, что смещение, связанное с выбором функциональной формы, как это делается при анализе регрессии, когда выбирается линейная зависимость, исключается. Это очень важное преимущество по сравнению с традиционными прогностическими моделями, поскольку связь между топологией СИС и живучестью имеет очень нелинейный характер, обусловленный важными и сложными взаимодействиями между звеньями.

ИНС разрабатываются или обучаются на основе общей живучести ряда возможных топологий СИС и надежности звеньев для данного числа узлов. Полученная в результате ИНС используется для оценки живучести СИС как функция от надежности звена и полученной оптимальной конфигурации. Таким образом, можно проводить многочисленные оценки живучести СИС без вычисления живучести для каждого из проектов. Неудобством в использовании ИНС в качестве критерия оценки живучести является то, что прогнозирование живучести – это оценка, которая может быть изменена в зависимости от адекватности ИНС. Сходный подход был использован для расчета последовательно-параллельных систем с учетом цены и живучести. Этот процесс описан в работе [71], но он фундаментально отличается, так как последовательно-параллельные системы достаточно легко рассчитываются аналитически.

Одним из важных способов оценки эффективности сочетания двух подходов: ИНС и оптимизации – является рассмотрение числа расчетов живучести, необходимых для обучения и оценки ИНС, в сравнении с числом расчетов, сэкономленных за счет ее использования. В [71] для решения первой последовательно-параллельной задачи расчета потребовалось 9600 вычислений живучести при обучении и оценке ИНС, что затем сэкономило 50 000 вычислений живучести при одной оптимизации. Если единственная оптимизация одной задачи расчета потребовала определенных усилий, тогда возможное сокращение в пять раз расчетов живучести не столь существенно, если учесть, что экономия была достигнута в ущерб точности оценки живучести.

Однако при рассмотрении связанных задач становится ясно, что подход имеет определенные достоинства. Для решения второй задачи потребовалась та же ИНС, поэтому не потребовалось дополнительных расчетов живучести и было сэкономлено 50 000 расчетов на каждой оптимизации. В общем ИНС-подход сократил на порядки расчеты живучести. С ростом числа

расчетов достоинства выбранного метода становятся все более очевидными (поскольку ИНС были сконструированы как универсальные аппроксиматоры для последовательно-параллельных систем). И наконец, «цена» первоначального тренировочного набора становится несущественной, если та же ИНС используется многократно для решения дополнительных расчетных задач. Этот же вывод справедлив и для расчетов живучести СИС. С помощью ИНС можно оценить живучесть любой топологии соединений, любого набора звеньев, любого набора определенного числа узлов.

Была выбрана процедурная модель обучения нейросети снизу [72], в обратном порядке (см. параграф 2.1), ввиду ее большой аппроксимирующей способности и возможности использования как в случаях двоичного, так и в случаях непрерывного ввода сигнала. Проблема изучалась там, где живучесть важна для каждого отдельного звена. Такое допущение распространено в работах, посвященных расчетам СИС [3, 62 – 66, 73]. Целью этого являлась разработка такой ИНС, которая смогла бы иметь различные степени живучести различных звеньев, хотя это и затрудняет задачу общей оценки. Количество узлов для данной ИНС было заданным. Входные данные в ИНС были следующими.

1. Архитектура СИС изображена серией двоичных переменных (x_{ij}). Длина строки 0 и 1 равна $\frac{N(N-1)}{2}$, где N – количество узлов в СИС.
2. Живучесть ребра.
3. Вычисленная по методу авторов [74] максимальная связность.

Вычисление верхних пределов связности производилось по следующей формуле:

$$R(x) \leq 1 - \left[\sum_{i=1}^N \left(\prod_{k \in E_i} (1 - p_{k_i}) \right) \prod_{j=1}^{i-1} \left(1 - \frac{\prod_{k \in E_j} (1 - p_{k_j})}{(1 - p_{k_i})} \right) \right], \quad (1.16)$$

где p – живучесть заданной связи; E – набор связей, связанных с заданной вершиной.

Расчеты верхних пределов связности потребовали определенных усилий, но это значительно повысило точность оценки ИНС. Без верхних пределов связности, связей, служащих как ввод, ошибки, повторенные в последующих секциях ИНС, практически удваиваются. Вывод ИНС расценивался как общая живучесть СИС. В процессе обучения и оценки желаемая живучесть СИС рассматривалась как точная величина, вычисляемая с использованием метода «обратного отсчета ошибки» [75].

Для исследования предложенного в работе подхода были выбраны узлы десятого размера. Использовались звенья с живучестью 0,80; 0,85; 0,90; 0,95 и 0,99. Был произвольно генерирован набор ИНС с 750 вариантами топологии (получилось, что каждая СИС формировала дерево с минимальным ветвлением, т.е. $R(x) > 0$) с 150 измерениями живучести каждого звена (максимальное количество топологий, обсчитываемое этой моделью ИНС 5×2^{45} или $1,7 \times 10^{14}$, таким образом, 750 – очень небольшая СИС). Верхние связи каждой СИС и реальная живучесть рассчитывались как ввод и целевой вывод, соответственно. После предварительных экспериментов архитектура СИС была представлена 47 вводами (45 возможными ребрами, живучестью ребер и верхней связностью), 47 скрытыми нейронами в одном скрытом слое и единым выходом. Набор данных делился с использованием метода пятикратной перекрестной проверки, таким образом, тренировалось пять оценок ИНС и одно окончательное использование ИНС. Пять оценок ИНС задействовали 4/5 набора данных, а оставшуюся 1/5 использовали для тестирования, при котором для каждой оценки ИНС заменялся тестовый набор. При окончательном использовании ИНС обучалась с применением всех 750 членов набора данных, и окончательная оценка выводилась с использованием перекрестной проверки ИНС. (см. полное описание процедуры перекрестной проверки ИНС в [76].)

Также использовалась вторая стратегия для СИС с высокой степенью живучести. Поскольку большая часть топологии реальных СИС обладает высокой живучестью, важно, чтобы оценка живучести была достаточно точной, когда $R(x) > 0,90$. Если, как описано выше, живучесть первой ИНС оценивалась как 0,90 или выше, то топология СИС, живучесть звеньев и верхние связи служили вводом для второй ИНС, как показано на рис. 1.11.

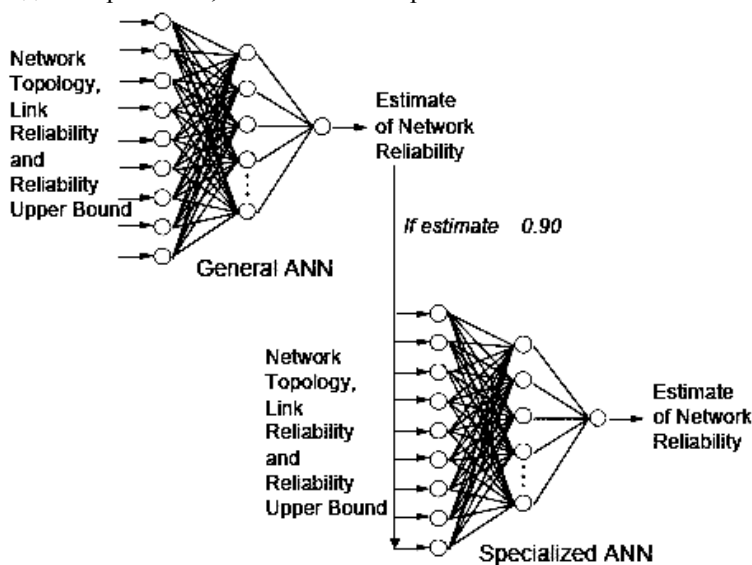


Рис. 1.11. Иерархия общей ИНС (General ANN) и специализированной ИНС (Specialized ANN)

Эта СИС обучалась на 250 произвольно выбранных топологиях с использованием той же самой живучести пяти звеньев, которые имеют степени живучести СИС от 0,90 или выше. Как для общих ИНС, было произведено равное количество наблюдений (50) для живучести каждого звена в наборе данных. Так же, как и в общих ИНС, использовался метод пятикратной перекрестной проверки при обучении и оценке ИНС. Архитектура ИНС была идентична первой СИС. Была сделана попытка использовать оценку живучести первой СИС как ввод для специализированной ИНС, но это не улучшило возможную работу специализированной СИС. Описание процедурной модели расчета см. в параграфе 2.2.

Перекрестная проверка среднеквадратичной ошибки (root mean squared error, RMSE) для заданной ИНС вычислялась по формуле:

$$RMSE = \sqrt{\frac{1}{750} \sum_{g=1}^5 \sum_{h=1}^{150} (y_{(g-1)150+h} - \hat{f}[T_{(g)}, x_{(g-1)150+h}])^2}, \quad (1.17)$$

где g – индекс выходной группы; h – индекс вычислений в выходной группе, а пример $T_{(g)} = \{(x_1, y_1), (x_2, y_2), \dots, (x_{(g-1)150}, y_{(g-1)150}), (x_{(g-1)151}, y_{(g-1)151}), \dots, (x_{750}, y_{750})\}$ использовался для конструирования ИНС $\hat{f}[T_{(g)}, x_{(g-1)150+h}]$.

В табл. 1.3 и 1.4 представлены результаты пятикратной проверки среднеквадратичной ошибки (RMSE) для общей и специализированной ИНС, соответственно. Можно заметить, что оценки ИНС существенно улучшаются в верхних пределах связности.

Таблица 1.3

Набор	RMSE-обучение	RMSE-тестирование	Верхний предел RMSE-связности
Набор № 1	0,03672	0,04260	0,08875
Набор № 2	0,03073	0,05004	0,08954
Набор № 3	0,03444	0,03067	0,07158
Набор № 4	0,03123	0,05666	0,07312
Набор № 5	0,03173	0,05131	0,08800
Среднее	0,03297	0,04626	0,08220

Таблица 1.4

Набор	RMSE-обучение	RMSE-тестирование	Верхний предел RMSE-связности
Набор № 1	0,00664	0,00688	0,01232
Набор № 2	0,00583	0,01271	0,01371
Набор № 3	0,00630	0,00892	0,00908
Набор № 4	0,00629	0,00795	0,00927
Набор № 5	0,00555	0,01125	0,01598
Среднее	0,00612	0,00954	0,01207

Более того, ошибки для специализированных СИС намного меньше, чем для общих ИНС, допуская, что оценки живучести более точные для тех топологий, которые считаются лучшими.

На рис. 1.12 показан пример одной из пятикратных оценок реальной живучести в сравнении с оценкой ИНС по тестовому набору, а на рис. 1.13 показано то же самое для специализированной ИНС.

Можно видеть, что прогнозы для ИНС объективны и достаточно точны. Там, где общая ИНС менее точна (при $R(x) > 0,90$), специализированная ИНС демонстрирует лучшие результаты работы. Подтверждается лучшая работа ИНС в верхних связях вместе с точностью оценки реальной живучести ИНС (рис. 1.14).

Погрешность использования общей ИНС равна 0,036, а специализированной составляет 0,007. Они могут быть как положительными, так и отрицательными, поскольку ИНС – это объективная оценка, а погрешность для верхних связей всегда положительна.

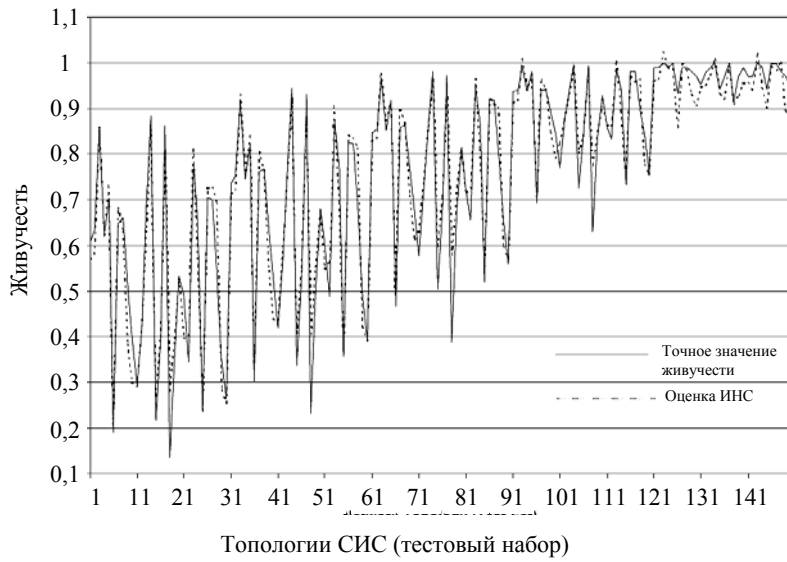


Рис. 1.12. Оценка живучести, вычисленной с помощью ИНС, в сравнении с реальной живучестью СИС на пятом сложении

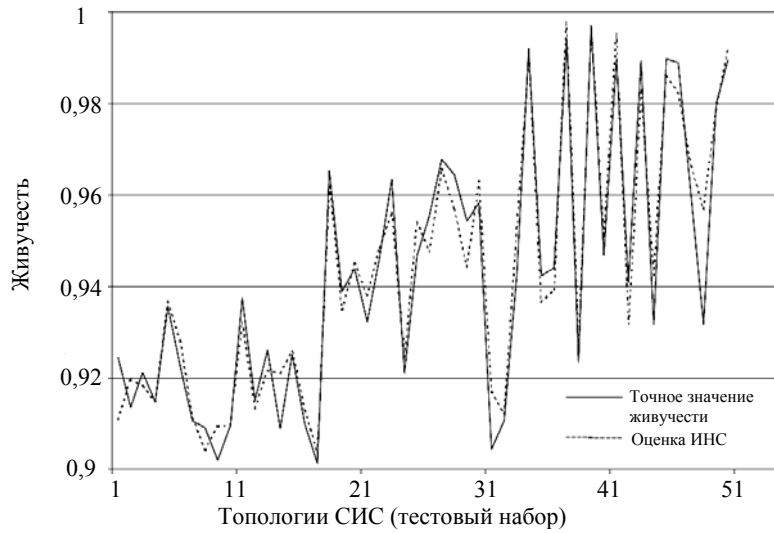


Рис. 1.13. Оценка живучести специализированной ИНС в сравнении с реальной живучестью СИС на пятом сложении

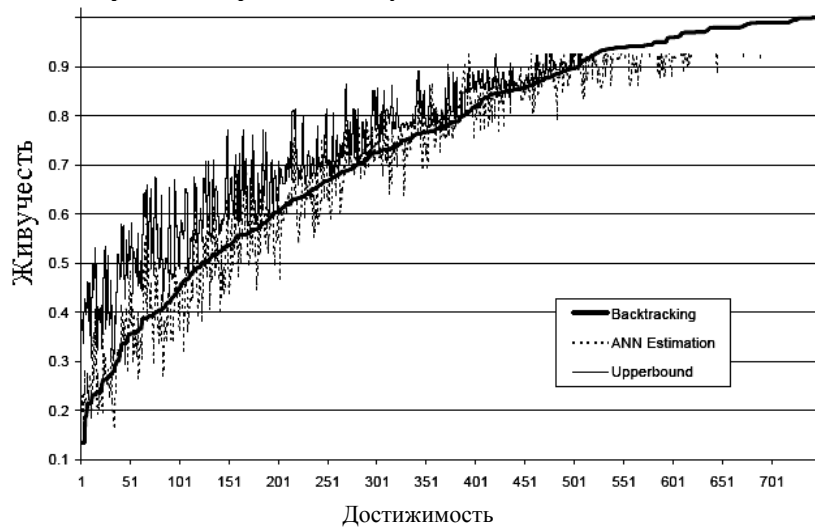


Рис. 1.14. Сравнение достижения результатов текущей живучести (backtracking), верхнего предела живучести СИС и вычисленных результатов основной ИНС

Принимая во внимание относительно малый объем тренировочного набора (всего 150 наблюдений степени живучести общей ИНС и только 50 наблюдений для живучести каждого звена специализированной ИНС), становится очевидным со-

крашение объема вычислений при данном подходе. ИНС-подход, следовательно, может быть использован для решения проблем расчета любой СИС из 10 узлов с пятью звеньями живучести. Полагая, что каждая задача расчета живучести СИС из десяти узлов (т.е. с определенной надежностью звеньев) имеет поле поиска $3,5 \cdot 10^{13}$ при процедуре оптимизации, которая исследует только малую часть возможных конструкций, потребуются миллионы расчетов живучести. Малое количество топологий СИС, необходимых для тренировки и оценки ИНС, – 150 для общей и 50 для специализированной. Увеличение размера тренировочного и оценочного наборов, конечно, существенно повысят точность оценки ИНС, а его сокращение приведет к понижению надежности оценки.

ИНС-подход к оценке общей живучести СИС оправдал себя. Используя только малую часть возможных топологий для решения задачи проектирования СИС из 10 узлов, удалось обучить и оценить как общую ИНС, так и специализированную ИНС. Соответствующее использование ИНС в процессе оптимизации проектирования СИС в значительной мере сокращает объем необходимых для этого вычислений. Предложенный подход рекомендует использование ИНС для оценки в процессе оптимизации проектирования любых СИС, а затем для точного расчета живучести СИС с оптимальной конструкцией или для нескольких лучших проектов СИС. Это помогает значительно сократить затраты усилий на вычисления живучести СИС при использовании прогрессивного метода оптимизации ее расчетов. Вероятно, что сведение подхода, основанного на использовании ИНС, только к живучести одного звена еще больше увеличит надежность оценки, но это приведет к значительной потере гибкости в процессе проектирования. Соответственно, увеличение ИНС до звеньев с разной степенью живучести в одном проекте значительно повысит гибкость при проектировании, но усложнит задачу оценки ИНС.

1.5. ПОТОКОВАЯ МОДЕЛЬ ЖИВУЧЕСТИ СЕТЕВОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ

1.5.1. ИССЛЕДОВАНИЕ ЖИВУЧЕСТИ СТОХАСТИЧЕСКОЙ СЕТЕВОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ

Телекоммуникационные технологии построения систем передачи информации как самостоятельное понятие возникли в середине XX в. К числу факторов, оказавших определяющее воздействие на развитие телекоммуникационных технологий, в первую очередь следует отнести развитие микроэлектронной индустрии и связанное с этим развитие вычислительной техники, а также успехи последнего времени в технологии оптоволоконных систем. Можно утверждать, что все региональные сети передачи данных (РСПД) построены с использованием только оптоволоконных технологий. Но, хотя данные технологии и являются оптимальными для таких СИС, они не являются абсолютно надежными, как и любое другое оборудование, и под влиянием каких-либо факторов или внешних воздействий (выход из строя оборудования, обрыв канала, огневое поражение узла при военных действиях, различные стихийные бедствия, носящие глобальный характер) эффективность работы таких СИС может сильно снижаться. В данном разделе рассматривается задача повышения эффективности СИС в условиях возможности воздействия неблагоприятных факторов (НФ), уменьшающих пропускную способность ребер физического графа СИС, учитывая неинформированность о связности графа и степени поражения его вершин и ребер.

СИС основана на сети массового обслуживания, и потому в ее функционировании всегда присутствует элемент неопределенности, порожденный случайным характером формирования продуктового потока. Дополнительную неопределенность вносит и функционирование входящих в СИС многих структурных элементов (СЭСИС). Случайный отказ одного из них приводит к перераспределению потоков между каналами связи: поток из элемента нижнего уровня (транзитного узла) пересылается в элемент более высокого уровня (управляющий узел), и, если канал уже работает на пределе пропускной способности, он блокируется для данного потока вследствие перегрузки. Серия нарушений передачи потоков приобретает лавинообразную форму, и вся СИС оказывается заблокированной. Из-за воздействия НФ физический граф непредсказуемым образом разбивается на неизвестное количество фрагментов, о параметрах которых можно высказывать только вероятностные соображения.

Предположим, что построена СИС, определяемая связным физическим детерминированным графом. В таком случае всегда существует путь между двумя произвольными вершинами этого графа. Известны критерии, позволяющие судить о связности графа. Но под воздействием НФ неизвестной плотности и силы [31] часть вершин и ребер графа (СЭСИС) потеряли свои функциональные способности (оказались полностью поражены) и граф распался на неизвестное число компонентов, о которых также нельзя создать определенного суждения. Проанализировав вышеизложенное, можно прийти к выводу, что проблема живучести есть проблема связности стохастического графа.

При решении задач анализа стохастического графа определяют [31]:

- 1) вероятность распада исходного графа на r компонентов; как правило, отыскивается вероятность того, что стохастический граф связан, P ($\rho = 1$);
- 2) вероятность p существования ребра (или вершины);
- 3) вероятность принадлежности двух вершин одной компоненте;
- 4) верхняя и нижняя границы вероятности существования графа, ребра которого существуют с вероятностью p .

Структура случайного графа описывается множеством вершин $\{V_0, V_n\}$ и множеством элементарных ситуаций $\{\varepsilon_{i,j}\}$, $i, j = \overline{1, n}$, где $\varepsilon_{i,j}$ определяет событие, состоящее в наличии или отсутствии связи между вершинами V_i и V_j через ребро (V_i, V_j) . Определяется вероятность $p(\varepsilon_{i,j})$ (или распределение вероятностей, если случайные события не являются независимыми). Если события независимы и равновероятны, граф называется несмещенным, в противном случае – смещенным [31].

Аналитические результаты получены лишь для ограниченного числа графов (содержащих до 30 вершин), относящихся преимущественно к логическому типу. Для больших графов численные отношения трудно формализуемы, тогда исследуемый граф заменяется бесконечным, асимптотические свойства которого оказываются близкими к точному решению [31, 40]. Для физических графов чаще используют метод Монте-Карло [77], в котором природа некоего, не поддающегося непосред-

ственному наблюдению процесса исследуется с помощью другого случайного процесса, действие которого можно наблюдать.

Создаваемая СИС должна быть максимально живучей, однако общего решения, т.е. такого математического или даже логического выражения, из которого, варьируя численные значения параметров, можно получить частные решения для разных графов, не существует [78]. Приемлемые значения можно получить для однородных симметрических структур, где граф имеет форму решетки, в которой каждая вершина соединена с ближайшими соседними (рис. 1.15) [32].

Для решетчатых структур общая живучесть определяется по работам [7 – 17].

В этом случае граф определяется степенью вершины d и правилом соединения одной вершины с соседними.

В начальном состоянии граф предполагается связным, т.е. не имеет изолированных вершин. Это означает, что каждой вершине инцидентно по крайней мере одно ребро.

Пусть Q_i представляет собой событие, состоящее в том, что не существует поврежденных ребер, инцидентных V_i . Объединение событий $\{Q_i\}$, $i = \overline{1, n}$, есть событие, что одна вершина графа не имеет поврежденных ребер. Дополнительным событием поэтому служит следующее: каждая вершина имеет, по крайней мере, одно существующее инцидентное ребро. Таким образом:

$$P(\rho = 1) \leq 1 - P\{Q_i\}. \quad (1.18)$$

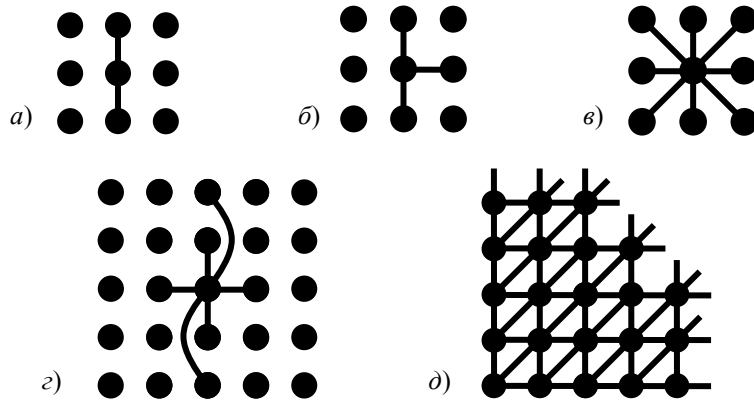


Рис. 1.15. Решетчатая структура графа:

$$a - d = 4; \quad b - d = 5; \quad c - d = 6; \quad d - d = 6$$

Пусть $d(i)$ есть степень V_i и $q = 1 - p$ есть вероятность разрушения ребра. Очевидно, $P\{Q_i\} = q^{d(i)}$ (теорема о произведении вероятностей одновременно происходящих независимых событий).

Условием, что рассматриваемые числа реализуются в виде графа, является отношение:

$$\sum_{i=1}^n d(i) = 2m, \quad (1.19)$$

откуда следует:

$$\langle d \rangle = \frac{2m}{n}. \quad (1.20)$$

Используя дисперсионные соотношения для случайной величины $d(i)$:

$$S^2 = \frac{1}{n-1} \sum_{i=1}^n (d(i) - \langle d \rangle)^2, \quad (1.21)$$

получают верхнюю границу вероятности связности графа

$$P(\rho = 1) \leq 1 - nq^d \left[1 - \frac{n-1}{2} q^{d(1-\frac{1}{n})} \right]. \quad (1.22)$$

Нет необходимости искать возможные значения P , так как на практике можно ограничиться вполне приемлемым значением $P(\rho = 1) \geq \frac{3}{4}$ [18, 79]. Для таких значений P множитель в квадратных скобках из (1.22) можно заменить значением 0,5 (или 1/2). Выражение (1.22) приобретает вид:

$$P(\rho = 1) \leq 1 - \frac{n}{2} q^d, \quad d \ll n. \quad (1.23)$$

Выражение (1.23) и есть критерий живучести стохастического однородного графа с заданными n , d , q . Если задано $P(\rho = 1)$, то

$$d \geq \frac{1}{\ln q} \left[\ln(1-P) - \ln \frac{n}{2} \right] - \quad (1.24)$$

необходимое условие для степени вершин, чтобы граф был связным, т.е. из каждой вершины в среднем должно исходить d ребер при заданных q и n .

Исследуем изменения структуры СИС при внезапном воздействии на СИС НФ. Граф может быть постоянным или изменяться со временем, детерминированным или стохастическим (для конечного пользователя системы граф после воздействия НФ, с точки зрения теории вероятностей, будет всегда случайным).

Поставленная задача решается при следующих условиях:

1. Каждая вершина имеет в среднем d исходящих ребер.
2. Каждое ребро, инцидентное вершине V_i , также инцидентно вершине V_j с вероятностью $\frac{1}{(n-1)}$, $i, j = \overline{1, n}$.
3. Все вершины и ребра идентичны.
4. Воздействие НФ направлено в некоторую случайно выбранную подобласть Δ , тогда вероятность поражения равна Δ / D , где D – площадь, занимаемая СИС.
5. Частота возникновения последствий воздействия НФ составляет $\eta(t)$ единиц на единицу площади.
6. Воздействие НФ на СИС одномоментное.
7. Вершина разрушается в том случае, если подвергается воздействию НФ мощностью не менее K_S^b .
8. Ребро разрушается в том случае, если подвергается воздействию НФ мощностью не менее K_S^p .
9. Восстановительные работы на СИС не проводятся.

Если $\eta(t)$ велико, то ожидаемое число разрушенных вершин (или ребер) будет соответствовать вероятности того, что любая данная вершина может быть разрушена.

Пусть $g_k^b(\eta)$ и $g_k^p(\eta)$ обозначают ожидаемые части вершин и ребер, которые подверглись воздействию НФ с частотой возникновения последствий η . Вероятность того, что некоторая вершина k_s не будет разрушена, равна

$$\sum_{k=0}^{k_s-1} g_k^b(\eta). \quad (1.25)$$

Вероятность того, что некоторое ребро k_l не будет разрушено, равна соответственно

$$\sum_{k=0}^{k_l-1} g_k^p(\eta). \quad (1.26)$$

Пусть π^b и π^p – вероятности того, что воздействие НФ разрушает данную вершину и ребро, тогда, согласно распределению Пуассона для вероятностей:

$$g_k^b(\eta) = e^{-\pi^b \eta} \frac{(\pi^b \eta)^k}{k!}; \quad (1.27)$$

$$g_k^p(\eta) = e^{-\pi^p \eta} \frac{(\pi^p \eta)^k}{k!}. \quad (1.28)$$

Предположим, что после удара осталось в среднем α неповрежденных ребер, направленных из V_1 , причем V_1 считается неразрушенной.

Для графов, удовлетворяющих условию (1.23), существует соотношение [32]:

$$\beta = 1 - e^{-\alpha\beta} \quad \text{или} \quad \alpha = \frac{-\ln(1-\beta)}{\beta}; \quad (1.29)$$

$$P\{Q_i\} = q^d. \quad (1.30)$$

Следовательно, учитывая выражения (1.25) – (1.30):

$$\beta = 1 - \exp \left\{ -d \left[\sum_{k=0}^{k_s-1} g_k^b(\eta) \right] \left[\sum_{k=0}^{k_l-1} g_k^p(\eta) \right] \beta \right\}. \quad (1.31)$$

Таким образом, выражение (1.31) определяет среднее количество вершин, не разрушенных воздействием НФ, а

$\sum_{k=0}^{k_s-1} g_k^b(\eta)$ – среднее число вершин первоначального графа, которые могут установить связь с другими после воздействия

НФ, и выражение (2.31) можно использовать для определения коэффициента живучести.

Компонента α , учитывая (1.25), (1.26), (1.29), будет иметь вид:

$$d \left[\sum_{k=0}^{k_s-1} g_k^b(\eta) \right] \left[\sum_{k=0}^{k_l-1} g_k^p(\eta) \right] = \frac{-\ln(1-\beta)}{\beta}. \quad (1.32)$$

Задав $\beta \geq \beta_0$ как коэффициент живучести:

$$d \left[\sum g_k^b \right] \left[\sum g_k^p \right] \geq \frac{-\ln(1-\beta_0)}{\beta_0}$$

и учитывая выражения (1.27) – (1.29), получим:

$$d \left[\sum_k \frac{(\pi^b \eta)^k}{k!} \right] \left[\sum_k \frac{(\pi^p \eta)^k}{k!} \right] \geq e^{(\pi^b + \pi^p) \eta} \left(\frac{-\ln(1-\beta_0)}{\beta_0} \right). \quad (1.33)$$

Принимая во внимание, что k_s и k_l – целые числа, ограниченные некоторыми предельными значениями, находим минимальное значение d .

Например, сконструируем граф СИС с большим количеством вершин так, чтобы из вершины, выбранной случайно, можно было установить связь после нападения с 90 % ($\beta = 0,9$) неповрежденных вершин. При этом считаем, что ребра графа неуязвимы, а вершины подвержены разрушению. Плотность огня составила 100 ударов/км². Вероятность поражения конкретной вершины $\frac{\Delta}{D} = \frac{0,05 \text{ км}^2}{1 \text{ км}^2} = 0,05$. $k_s = 1$.

$$\beta = 1 - \exp \left\{ -d \sum_{k=0}^{k_s-1} \frac{5^k}{k!} e^{-5\beta} \right\};$$

$$d \sum_{k=0}^{k_s-1} \frac{5^k}{k!} \geq e^5 \frac{-\ln 0,1}{0,9} = 380.$$

Таким образом, граф, содержащий 380 вершин, может выдержать один удар, после которого можно установить связь с 90 % оставшихся после удара вершин. Подсчитаем количество оставшихся вершин:

$$\sum_{k=1}^{k_s-1} g_k^l(\eta) = e^{-0,05} \frac{0,05 \cdot 100}{1} = 0,034.$$

Таким образом, от начальных 380 вершин останется только 3 % – 12 вершин. Общий поток упадет более чем в 30 раз, СИС станет определенно недопустимой. Или, выражаясь по-другому, СИС будет уничтожена.

Стоит заметить, что расчет вероятностей не отражает истинной картины [80], адекватное представление дает распределение потоков, т.е. мультипоток, образующийся после удара. При подобном расчете удобнее пользоваться не вероятностями, а определять среднее значение вероятностных величин.

1.5.2. ОПИСАНИЕ ПОТОКОВОЙ МОДЕЛИ СЕТЕВОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ

Многопродуктовая потоковая СИС служит математической моделью реальных распределенных СИС, объединяющих в своем составе множество узлов. Для таких СИС важны вопросы, связанные с принятием решений по использованию имеющихся сетевых ресурсов, их распределением между СЭСИС, т.е. с анализом возможностей улучшения работы СИС за счет рационального перераспределения потоков. В условиях дефицита ресурсов СИС возникает ухудшение качества связи, выраженное в отказах в обслуживании, потерях, задержках и т.д.

Требования пользователей к пропускной способности ребер физического графа СИС считаются известными, однако это допущение не выполняется при повреждении СИС, тогда возникает проблема выбора приемлемого распределения потоков в случае недопустимости СИС. Приходится искать решение, использующее все ресурсы СИС, пока они могут быть использованы хотя бы одной тяготеющей парой. Подобное распределение потоков, максимально обеспечивающее требования всех тяготеющих пар, называется нормативным. Оно является таким решением задачи распределения потоков в СИС, которое не дает возможности одним тяготеющим парам улучшить обеспеченность своих требований за счет других, менее обеспеченных, и кроме того, использует все способы увеличения обеспеченности требований тяготеющих пар за счет ресурсов СИС.

Критерий допустимости потоковой СИС задается условием $\Theta_0 \geq 0$, которое гарантирует существование допустимого распределения потоков, такого, что соответствующий вектор мультипотока будет не меньше вектора заданных требований.

В случае недопустимости СИС $\Theta_0 < 0$ возникает трудно разрешимая проблема перераспределения потоков по информационной сети (так как распределенные СИС всегда имеют «узкие места» – ребра физического графа СИС, пропускная способность которых не позволяет увеличивать поток выше определенного значения), различных для разных тяготеющих пар. Нормативно распределенный поток позволяет осуществить распределение в соответствии с уровнем обеспеченности различных тяготеющих пар оптимальным образом.

Информационная потоковая СИС $S = (V, R, P)$ задается множествами $V = \{v_1, v_2, \dots, v_n\}$ – узлов СИС, $R = \{r_1, r_2, \dots, r_l\} \subset V \times V$ – ребер физического графа СИС G и $P = \{p_1, p_2, \dots, p_m\} \subset V \times V$ – тяготеющих пар или ребер логического графа СИС \bar{G} . Соответствующие индексные множества обозначим: $N = \{1, \dots, n\}$, $E = \{1, \dots, e\}$, $M = \{1, \dots, m\}$, следовательно: $V = \{v_j\}_{j \in N}$, $R = \{r_k\}_{k \in E}$, $P = \{p_i\}_{i \in M}$.

Все ребра не ориентированы, прямым направлением потока будем считать направление от вершины с меньшим номером к вершине с большим. Каждое ребро r_k физического графа СИС G будем представлять ориентированными дугами с

номера k и $k + l$ для прямого и обратного направлений. Для любой вершины $v \in V$ обозначим через $S(v)$ множество индексов выходящих из нее дуг, а через $T(v)$ – множество индексов входящих.

Для каждой i -й тяготеющей пары введем обозначение $p_i = (v_{s_i}, v_{t_i})$, где $s_i < t_i$ и вершина v_{s_i} называется источником, а v_{t_i} – стоком i -го вида продукта. В случае ориентированного ребра логического графа вершины источник/сток определяются согласно ориентировке.

Указанная структура СИС может быть представлена с помощью матрицы инцидентий «дуги – вершины» физического графа СИС G : $A = \{a_{k,j}\}$ размера $(2e \times n)$:

$$a_{k,j} = \begin{cases} 1, & \text{если } k \in S(v_j); \\ -1, & \text{если } k \in T(v_j); \\ 0 & \text{в остальных случаях,} \end{cases} \quad (1.34)$$

и матрицы связей логического графа СИС \bar{G} $B = \{b_{i,j}\}$ размера $(m \times n)$:

$$b_{i,j} = \begin{cases} 1, & \text{если } v_j = v_{s_i}; \\ -1, & \text{если } v_j = v_{t_i}; \\ 0 & \text{в остальных случаях.} \end{cases} \quad (1.35)$$

Тем самым однозначно задаются значения z_i потока между источником и стоком для каждой тяготеющей пары p_i в зависимости от распределения f потоков по ребрам физического графа G .

Введем матричную переменную $f = \{f_{i,k}\}$ размера $(m \times 2e)$. Каждый элемент $f_{i,k}$ обозначает количество потока i -й тяготеющей пары по ребру r_k в прямом направлении для $k \in E$ или по ребру r_{k-l} в обратном направлении, если $k > e$, $f_{i,k} \geq 0$.

В транзитных узлах выполняются условия неразрывности потока, что приводит к соотношению:

$$\sum_{k \in S(v)} f_{i,k} - \sum_{k \in T(v)} f_{i,k} = \begin{cases} z_i, & \text{если } v = v_{s_i}; \\ -z_i, & \text{если } v = v_{t_i}; \\ 0 & \text{в остальных случаях,} \end{cases} \quad (1.36)$$

где $z_i \geq 0$ – величина входного потока, проходящего по СИС от источника к стоку p_i при распределении потоков f .

В матричной форме получаем для $Z = Z(f)$ систему уравнений:

$$f_i A = z_i B, \quad i \in M,$$

нижний индекс у матрицы обозначает соответствующую вектор-строку.

Вектор $Z = Z(f) = (z_1, z_2, \dots, z_m)$, определяемый вектором распределения потоков f , является совокупностью потоков между всеми тяготеющими парами p_i и называется мультипоток Z .

Обозначим через $y_k = \sum_i (f_{i,k} + f_{i,(k+l)})$ общий поток по ребру r_k в соответствии с распределением потоков f .

Каждому ребру r_k припишем некоторое число $C_k \geq 0$, называемое пропускной способностью ребра r_k и измеряемое в условных единицах потока. Вектор $C = (c_1, c_2, \dots, c_l)$ будем считать фиксированным и известным исследователю.

Вектор C задает следующие ограничения на распределение потоков по СИС:

$$\sum_{i=1}^m (f_{i,k} + f_{i,(k+l)}) \leq C_k. \quad (1.37)$$

Обозначим:

$F(c)$ – множество возможных распределений потоков f ;

$L(c)$ – множество возможных мультипоток z ;

$X(c) = \{x\} = (f_1, \dots, f_m, Z)$ – множество всех возможных распределений потоков f и мультипоток Z .

В информационной потоковой СИС предполагается заданным вектор d требований или заявок тяготеющих пар p_i на величины потоков между источником и стоком, т.е. всем ребрам p_i логического графа приписаны числа $d_i \geq 0$, измеряемые в условных единицах потока, которые требуется пропустить по данному логическому ребру информационной потоковой СИС (так называемому информационному направлению). Если вектор d известен, ставится задача о допустимости СИС для указанного вектора требований, т.е. выполнения условия:

$$d \in L(c).$$

Это определяет такое распределение потоков $F(c)$, на котором достигается вектор требований:

$$d = Z(f),$$

что формально понимается как $d = Z$.

Соответствующее распределение потоков f , допустимое для вектора d , будем обозначать $f[d]$. Отметим, что такое распределение не единственное. Вектор d может быть и не известен точно, например, задается лишь множество его значений D , такое, как:

$$D = \left\{ d \mid \sum_{i=1}^m d_i = d_0 \right\}.$$

Возникают две различные постановки задачи о допустимости:

1) гарантированная – $D \subseteq L(c)$;

2) слабая (допустимость хотя бы одного вектора $d \in D$). При такой постановке задачи допустимо только одно какое-то распределение:

$$D \cap L(c) \neq \emptyset.$$

Определение критерия допустимости потоковой СИС $\Theta_0(d)$ по отношению к вектору требований d описано в параграфе 2.2.

1.5.3. ОРГАНИЗАЦИЯ НОРМАТИВНОГО ПОТОКА

Большие СИС имеют, как правило, несколько узких мест, ребер физического графа, пропускная способность которых не позволяет осуществить увеличение проходящего через них потока, которые различны для различных тяготеющих пар p_i . Следовательно, в любой произвольно взятой СИС будет несколько уровней максиминной обеспеченности потоковых требований Θ .

Пусть воздействие НФ приложено к нескольким ребрам, составляющим наиболее «узкий» участок графа. Поток z^0 между соответствующими этим ребрам тяготеющими парами p_i уменьшится, что понизит уровень максиминной обеспеченности потоковых требований Θ :

$$\max_{(f^0, z^0) \in X_0(c, d)} \min_{i \in M} z_i^0(d) d_i^{-1} = \Theta_0 < 1. \quad (1.38)$$

Обозначим $M_0 = M_0(d)$ – множество индексов i этих пар, $M_0 \subseteq M$.

$$M_0 = M_0(d) = \{i \in M \mid Z_i^0(d) d_i^{-1} = \Theta_0\}.$$

Тяготеющие пары p_i с индексом из M_0 будем называть имеющими нулевой уровень максиминной обеспеченности требований, подразумевая под этим уровнем значение Θ_0 .

Продолжим поиск следующего, 1-го уровня максиминной обеспеченности потоковых требований:

$$\begin{aligned} \Theta_1 &= \max_{(f^1, Z^1) \in X_1(c, d)} \min_{i \in M \setminus M_0} z_i^1(d) d_i^{-1}; \\ M_1 &= \{i \in M \setminus M_0 \mid Z_i^1 d_i^{-1} = \Theta_1\}; \\ &\dots \\ \Theta_l &= \max_{(f^l, Z^l) \in X_l(c, d)} \min_{i \in M \setminus \bigcup_{j=0}^l M_j} z_i^l(d) d_i^{-1}; \\ M_l &= \{i \in M \setminus \bigcup_{j=0}^l M_j \mid Z_i^l d_i^{-1} = \Theta_l\}; \\ &\dots \\ \Theta_L &= \max_{(f^L, Z^L) \in X_L(c, d)} \min_{i \in M \setminus \bigcup_{j=0}^L M_j} z_i^L(d) d_i^{-1}; \\ M_L &= \{i \in M \setminus \bigcup_{j=0}^L M_j \mid Z_i^L d_i^{-1} = \Theta_L\}; \\ M_0 \cup M_1 \dots \cup M_l \dots \cup M_L &= M; \end{aligned}$$

Подобное L -распределение потоков (f^L, Z^L) будем называть нормативным. Всем нормативным распределениям соответствует единственный нормативный мультипоток $Z^L(d)$ с компонентами:

$$Z_i^L(d) = \begin{cases} \Theta_0 d_i; & i \in M_0; \\ \Theta_1 d_i; & i \in M_1; \\ \dots & \\ \Theta_l d_i; & i \in M_l; \\ \dots & \\ \Theta_L d_i; & i \in M_L. \end{cases} \quad (1.39)$$

Нормативный поток $Z^L(d)$ является *максимальным* элементом множества Z из достижимых мультипоток. Все тяготеющие пары в СИС оказываются по своему значению («положению») в равных условиях, т.е. получают одинаковое значение обеспеченности потоковых требований.

Полученный в процессе определения $Z^L(d)$ набор $(\Theta_0, M_0, \dots, \Theta_L, M_L)$ дает достаточно информативную характеристику качества обслуживания СИС в целом, безотносительно к обеспеченности потоковых требований конкретных тяготеющих пар, что иллюстрируется ступенчатой диаграммой (рис. 1.16).

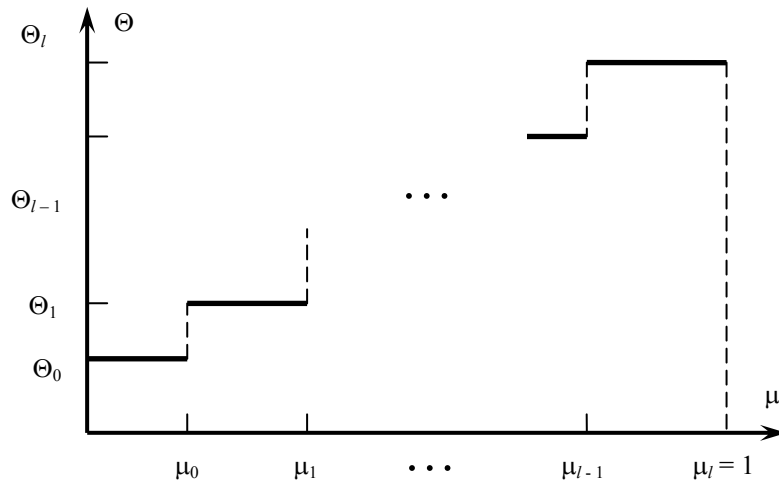


Рис. 1.16. Диаграмма обеспеченности требований

По вертикальной оси отложены значения $\Theta_l, l = \overline{0, l}$, а по горизонтали – суммарные величины требований, которые могут быть обеспечены на уровне выше l -го, приведенные к сумме всех потоковых требований СИС:

$$\mu_l = \sum_{i \in \bigcup_{j=0}^l M_j} \left(\frac{d_i}{\sum_{i \in M} d_i} \right). \quad (1.40)$$

Точки $(\mu_0, \Theta_0), \dots, (\mu_l, \Theta_l)$ соединены между собой с помощью ступенчатых линий, дающих диаграмму обеспеченности потоковых требований. Чем больше компонента Θ_l , тем лучше положение тяготеющей пары p_i по сравнению с остальными, это определяется структурой физического и логического графов СИС.

Таким образом, вектор Θ является решением задачи нормативного анализа СИС – анализа того, насколько сеть способна обеспечить требования на установление потока между вершинами графа.

Площадь под функцией $\min \{1, \Theta(\mu)\}$ равна той части всех требований тяготеющих пар СИС, которая может быть обеспечена при нормативном распределении потока.

Численное значение этой площади и есть показатель живучести Q СИС после воздействия на нее НФ силой γ .

$$Q = \sum_{k=0}^S \Theta_k (\mu_k - \mu_{k-1}); \quad \mu_{-1} = 0, \quad (1.41)$$

где S – число уровней организации потоковых требований.

$R = 1 - Q$ в таком случае следует именовать показателем уязвимости СИС.

Приведенное значение живучести Q не является гарантированным (так как оно зависит от распределения воздействия НФ по ребрам).

Перебрав все возможные комбинации распределения, находят наименьшее значение Q – в данном случае уже гарантированное.

Практически достаточно одного построения диаграммы, чтобы получить исчерпывающее представление о поведении СИС при воздействии на нее внешних НФ. Варьируя γ от 0 до 1, получают полную картину поражения СИС для всех случаев и, главное, для нескольких классов различных топологий физического графа СИС, что важно знать при синтезе СИС.

Любая точка на диаграмме означает, что доля μ всех требований обеспечена не более, чем $\Theta \cdot 100\%$ [55, 76, 78, 81 – 84].

2. ПРОЦЕДУРНОЕ ОБЕСПЕЧЕНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ ОЦЕНКИ ЖИВУЧЕСТИ СЕТЕВЫХ СТРУКТУР

В главе 1 мы рассмотрели структуру (рис 1.1) и аналитическое обеспечение информационной системы оценки живучести СИС, включающее в себя функциональные модели ИС и ее блоков и модулей (рис. 1.2), аналитические модели выбора критериев живучести, аналитические модели расчета живучести с помощью полинома Тутте (параграф 1.3), модели с элементами искусственного интеллекта (нейросетевой модели, параграф 1.4), а также модели МП-сети (поточковой модели, параграф 1.5).

В главе 2 рассмотрим процедурное обеспечение ИСЖС, в том числе процедурные модели оценки живучести СИС, структуру и процедурные модели блока анализа исходных данных, включая модель генерации правил, состоящую из компоненты фаззификации (перевода реальных данных о системе в четком виде к нечеткому виду), компоненты анализа и выделения правил, а также компоненты очистки правил, и разработанную логико-лингвистическую модель блока анализа исходных данных. По этой модели была разработана процедурная модель.

Также были рассмотрены процедурные модели оценки живучести СИС, основанные на аналитическом обеспечении, предложенном в параграфах 1.2 – 1.5, определены размерности и топологии графов, оптимально подходящих для каждой процедурной модели.

Кроме того была предложена структура и описана функциональность модуля АТС, предложены процедурная модель анализа СИС на основе модели МП-сети, комбинированная задача выбора пропускных способностей и распределения потоков, задачи нахождения кратчайших путей и расчета объемов суммарной передачи информации.

В главе 1 нами были рассмотрены процедуры построения графика уязвимостей СИС с помощью нахождения гарантированного критерия живучести, и построены на их основе процедурные модели анализа уязвимости и синтеза СИС с повышенной живучестью по критерию удовлетворения сети запросов на организацию потоков.

Для начала расчета живучести заданной СИС необходимо извлечь набор параметров из СЗ и выбрать процедурную модель расчета. Этим занимается блок анализа исходных данных.

2.1. ОПИСАНИЕ БЛОКА АНАЛИЗА СЕТЕВОЙ СТРУКТУРЫ

В системе знаний (СЗ) данные о СИС хранятся в виде реальных значений (пропускная способность каналов, размерность и топология графа, информация о возможном воздействии НФ на систему и т.д.), работа с которыми (например, выбор топологии, от которого зависит выбор процедурной модели расчета) аналитическими методами сложна. Решение класса подобных задач было предложено в работе [77] с помощью лингвистических переменных и нечетких множеств и в дальнейшем продолжено в работах [85, 86]. Для создания логико-лингвистической модели необходимо задать набор некоторых правил, содержащих в своем составе логические термы. Так как логико-лингвистические модели работают с набором нечетких множеств, реальные переменные из БД № 1 – № N СЗ (четкие значения) нужно привести к нечеткому виду. Данная процедура называется процедурой фаззификации (fuzzification, от англ. fuzzy – нечеткий) и описана в работах авторов [86]. После этого из набора нечетких значений необходимо составить некоторые правила, исходя из которых и будет определяться выбор процедурной модели расчета СИС. Для этого рассмотрим процедурную модель составления правил, содержащую следующие компоненты.

1. Компонент фаззификации. Данный компонент процедурной модели работает с четкими значениями из БД СЗ. В теории нечетких множеств нечеткое подмножество предметной области U описывается функцией принадлежности вида:

$$\mu_V(V): U \rightarrow [0, 1], \quad V \subset U, ,$$

представляющее степень принадлежности $\mu \in U$. Нечеткая лингвистическая переменная V является атрибутом, домен которого содержит лингвистические значения для нечетких подмножеств [87]. Реальные значения переводятся в лингвистические переменные, например, размерность графа СИС в лингвистических переменных может быть записана как: Размерность. Количество_вершин. Малое (S), Размерность. Количество_вершин. Среднее (M) и Размерность. Количество_вершин. Большое (L). Функция принадлежности для данных нечетких множеств представлена на рис. 2.1. Пределы каждого терма определены с использованием гладкой гистограммы четких значений по работе авторов [88].

2. Компонент анализа. Данный компонент используется для анализа уровня доверительной вероятности (P_c) конъюнкции различных значений в БД. Значения в заданной БД делятся на N атрибутов предсказания и один целевой атрибут (класс). Каждый атрибут описан переменной $A_i (i = 1, 2, \dots, N)$ и различными вероятностными значениями

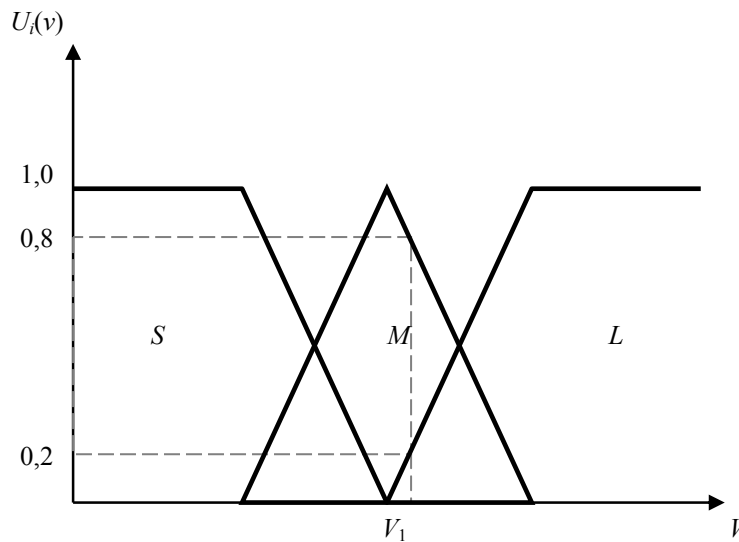


Рис. 2.1. Перекрытие функций принадлежности

m_i для атрибута $A - V_{ij}$ ($j=1, 2, \dots, m_i$), каждый целевой атрибут делится на K классов и описан переменной $class_k$ ($k=1, 2, \dots, K$). Глубина уровня поиска возможных значений описана переменной $level_l$ ($1 \leq l \leq N-1$). Таким образом, компонента анализа выделяет конъюнкции значения (значений), удовлетворяющего условию $P_c = 1$. Поиск начинается с пустого набора данных $test_set$ и множества S . Каждый раз конкретное значение V_{ij} добавляется в множество S и отвечает условной вероятности, вычисляется $P(S|class_k)$.

Таким образом, условная вероятность P_c для множества S и $class_k$ представлена термом $P(S|class_k)$.

Если $P(S|class_k) = 1$, тогда создаем правило, включающее значение множества S , принадлежащее также $class_k$ и не создающее конъюнкции с остальными оставшимися элементами множества S . Таким образом, мы сокращаем время поиска в процедурной модели.

Если $P(S|class_k) \neq 1$, тогда модифицируем S , добавляя другое значение из оставшихся атрибутов, и проверяем условную вероятность P_c .

Добавление новых значений в S ограничено количеством условий в правиле: $\leq N-1$.

3. Компонент очистки. Данный компонент «очищает» выделенные правила в течение трех уровней.

1) На первом уровне происходит удаление избыточных (повторяющихся) правил.

2) На втором уровне происходит суммирование правил, состоящих из одинаковых атрибутов, но содержащих различные значения.

Например:

(IF $A1$ is $v11$ THEN $Class1$) AND (IF $A1$ is $v13$ THEN $Class1$),

Суммирующее правило будет иметь следующий вид:

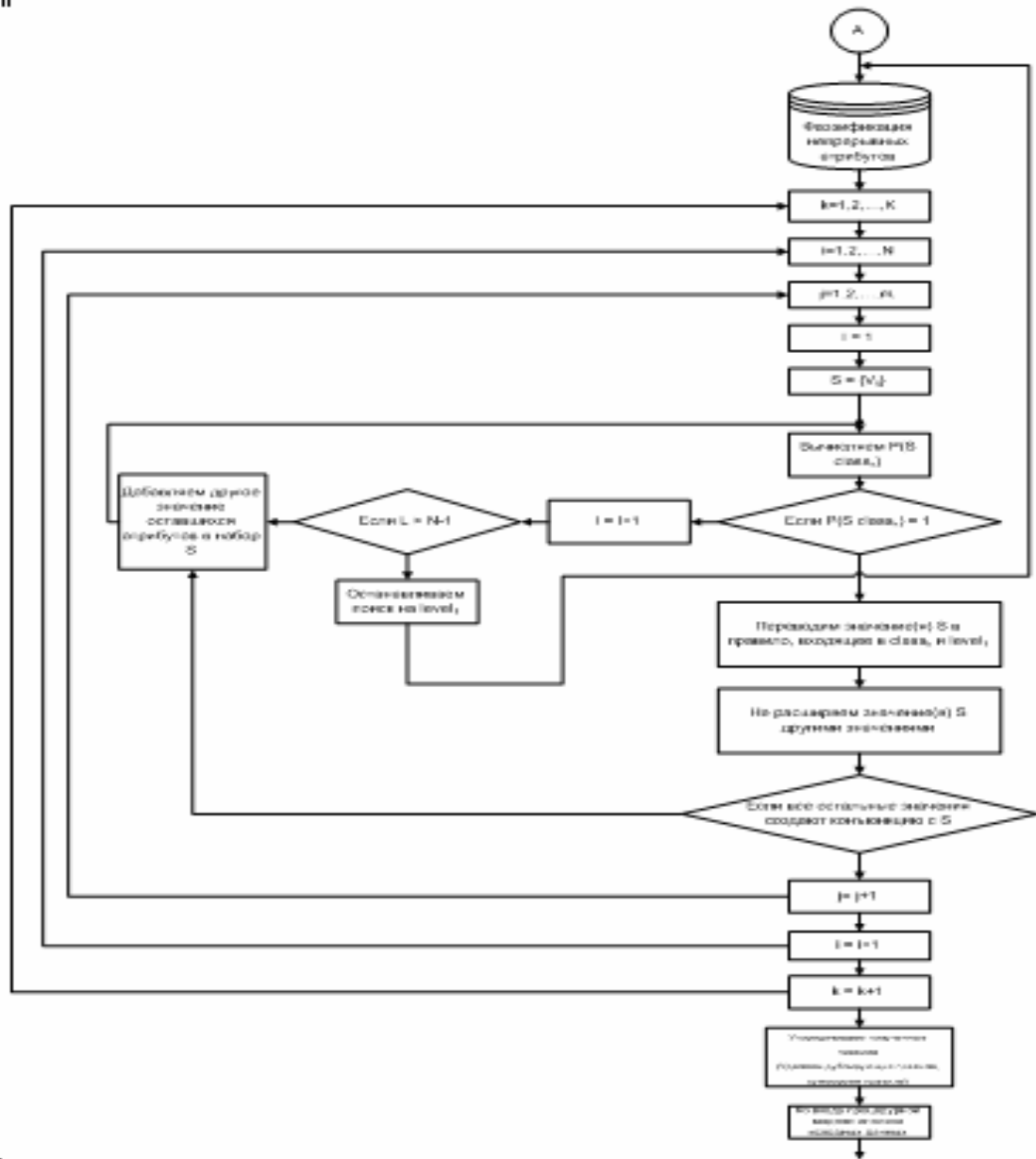


Рис. 2.2. Процедурная модель генерации правил для блока анализа исходных данных

IF $A1$ is $v11 \vee v13$ THEN $Class1$

3) На третьем уровне отбрасываются правила, уровень точности которых ниже специфичного граничного уровня точности. Это необходимо по следующим причинам:

– Поиск на всем множестве конъюнкций реальных значений в БД СЗ может потребовать значительных ресурсов и времени.

– Если процедурная модель выполняет поиск по нижней границе уровня доверительной вероятности P_c , количество индуцированных правил может быть огромным, и соответственно комбинаторно сложным для дальнейших вычислений.

– Если поиск процедурной моделью ведется на высоком уровне точности и отбрасывает значения правил с низким уровнем точности, то полученный набор правил может быть бесполезен для предсказания класса скрытых наборов и создавать обобщения.

Схема процедурной модели индукции правил для блока анализа исходных данных представлена на рис. 2.2.

С помощью данной процедурной модели был создан набор следующих правил:

Топологии (графа СИС):

1. Радиальная.
2. Радиально-кольцевая.

3. Сетчатая.
4. 2-полюсная полная.
5. Древовидная.
6. Гибридная.

Размерность (графа СИС):

Количество вершин:

1. Малое.
2. Среднее. Большое.

Количество ребер:

1. Малое.
2. Среднее.
3. Большое.

Диаметр (графа СИС):

1. Малый.
2. Средний.
3. Большой.

Стоимость:

1. Низкая.
2. Средняя.
3. Высокая.

Продукт (вид продукта или тяготеющие пары):

1. Известен.
2. Неизвестен.

Вероятность (удаления ребра в графе):

1. Низкая.
2. Средняя.
3. Высокая.

Расчет живучести СИС проводится по следующим моделям:

1. Поточковая.
2. Полиномиальная.
3. Нейросетевая.

Процедурная модель блока анализа исходных данных и выбора модели расчета будет иметь следующий вид:

Начало

Задаем размерность графа;

Вычисляем диаметр;

Если **топология** = гибридная и/или **размерность** = много_вершин и много_ребер или много_вершин и средне_ребер или средне_вершин и много_ребер и диаметр != малый или средний и продукт = неизвестно и/или стоимость = известна или частично_известна и вероятность = неизвестна то **использовать модель** = 3 (нейросеть)

Иначе

Если **топология** = древовидная и/или радиальная или радиально-кольцевая и **размерность** = много_вершин и много_ребер или много_вершин и средне_ребер или средне_вершин и много_ребер и диаметр = малый или средний и продукт = известно и/или стоимость = известна или частично_известна или неизвестна и вероятность = неизвестна то **использовать модель** = 1 (поточковая)

Иначе

Если **топология** != гибридная или древовидная или радиальная или радиально-кольцевая и **размерность** = любая(1-6) и диаметр = любой(1-3) и продукт = неизвестно и стоимость = неизвестно и вероятность = известно то **использовать модель** = 2(полиномиальная) иначе **использовать модель** = 3(нейросеть);

Конец.

Другим образом процедурную модель можно записать:

IF топология = гибридная **AND** размерность.количество_вершин = Большое **AND** размерность.количество_ребер = Большое **AND** Диаметр = Большой **AND** Продукт = неизвестно **AND** стоимость != низкая **AND** вероятность = Низкая **THEN** использовать_модель = 3 (Нейросетевая)

ELSE

IF топология = древовидная **OR** топология = радиальная **OR** топология = радиально-кольцевая **AND** размерность.количество_вершин = Среднее **AND** размерность.количество_ребер = Среднее **AND** Диаметр = Малый **OR** Диаметр = Средний **AND** Продукт = известно **AND** стоимость = низкая **AND** вероятность = Низкая **THEN** использовать_модель = 1 (Поточковая)

ELSE

IF топология != древовидная **OR** топология != радиальная **OR** топология != радиально-кольцевая **AND** размерность.количество_вершин != Большое **AND** размерность.количество_ребер != Большое **AND** Диаметр = Малый **OR** Диаметр = Средний **AND** Продукт = Неизвестно **AND** стоимость = Низкая **AND** Вероятность = Высокая **OR** Вероятность = Средняя **THEN** использовать_модель = 2 (Полиномиальная)

ELSE

2.2. ПРОЦЕДУРНЫЕ МОДЕЛИ ОЦЕНКИ ЖИВУЧЕСТИ СЕТЕВЫХ ИНФОРМАЦИОННЫХ СИСТЕМ

После выбора процедурной модели, по которой должен вестись расчет, параметры исследуемой СИС дефаззифицируются (defuzzification) [260 – 264] и передаются в одну из трех описанных ниже процедурных моделей расчета живучести СИС.

Полиномиальная модель вычисления живучести сетевых информационных систем (модель № 1). Как было показано в параграфе 1.4, путем разделения изоморфных миноров расширительное дерево видоизменяется в корневой ациклический граф с одним источником (корнем) и одним стоком (ребра ориентированы от корня к стоку). Например, рис. 1.7 отражает расчет живучести многополюсной СИС для заданного графа. На этой схеме p_i обозначает вероятность удаления ребра e_i , а $q_i = 1 - p_i$. Этот ациклический граф тесно связан с БСПР (бинарной схемой принятия решений: средство работы с булевой функцией [9]), репрезентирующей все остовные деревья заданного графа. Это потому, что, как отмечалось выше, каждый маршрут однозначно соответствует остовному дереву графа.

Исходя из этого, живучесть многополюсной СИС в этом общем случае может быть рассчитана по следующей процедурной модели:

$R(\text{источник}) := 1$; {примечание: источник – это однозначно определяемый узел на нулевом уровне};

$R(\text{сток})$ – это $R(G)$.

для $i :=$ от 1 до t

начало

для всех узлов u на уровне i делаем $R(u) := 0$;

для каждого узла v на уровне $(i - 1)$

начало

если v имеет ребро к потомку u , соответствующее копелле e , тогда $R(u) := R(u) + (1 - p(e))R(v)$;

если v имеет ребро к потомку u , соответствующее петле e , тогда $R(u) := R(u) + R(v)$;

если v имеет ребро к потомку u , соответствующее удалению e и ребро к потомку w , соответствующее контракции e , тогда $R(u) := R(u) + p(e)R(v)$; $R(w) := R(w) + (1 - p(e))R(v)$;

конец;

конец;

Графическая схема процедурной модели процесса вычисления живучести многополюсной СИС изображена на рис. 2.3.

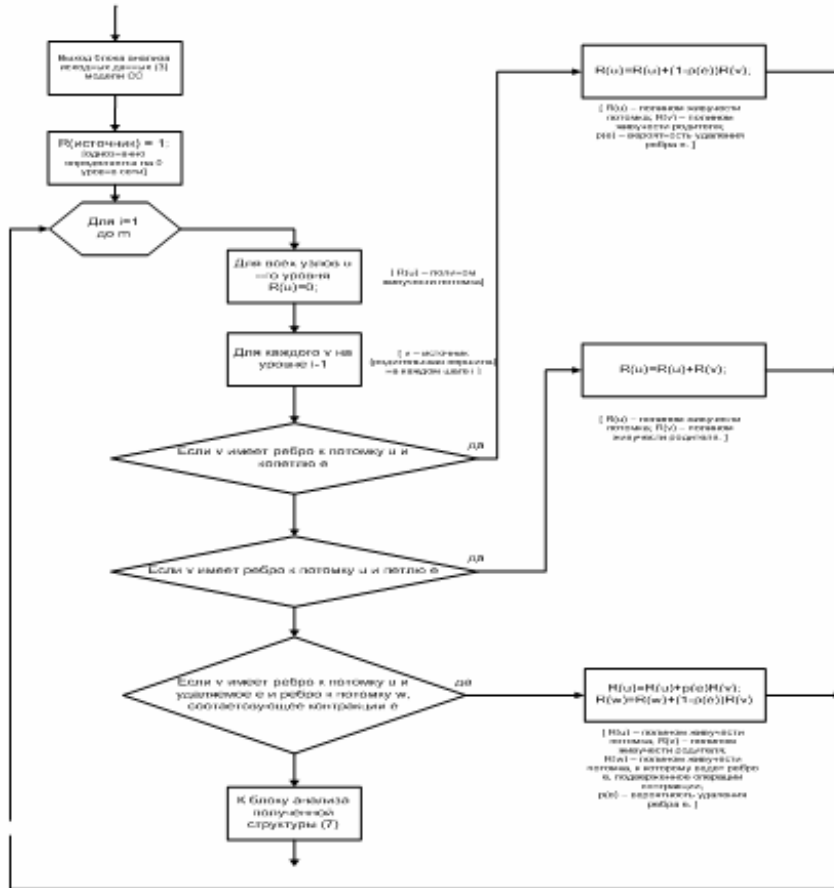


Рис. 2.3. Процедурная модель процесса вычисления полнума общей живучести СИС

Для проверки работоспособности данной модели была использована методика авторов [15], расчет живучести производился на графе с 14 вершинами и $\binom{14}{2} = 91$ ребром (полный граф), а также на планарном (решетчатом) графе 12×12 , содержащим 144 вершины и 264 ребра. Теоретически с помощью данной процедурной модели можно вычислить живучесть графа СИС, содержащего до 1000 ребер, но в этом случае процесс будет слишком длительным и ресурсоемким, поэтому для графов, содержащих более 500 ребер целесообразнее использовать модель № 2 (процедурную модель вычисления живучести СИС с использованием элементов искусственного интеллекта).

Процедурная модель процесса вычисления живучести СИС с использованием элементов искусственного интеллекта. Процедурная модель процесса вычисления общей живучести СИС с помощью элементов искусственного интеллекта (рис. 2.4), аналитическое обеспечение которой было описано нами ранее в параграфе 1.5, имеет следующий вид:

//Инициализация

1. Помечаем все связи как «свободные», создаем пустой стек связей;
2. Генерируем модифицированный срез:
 - a. Находим набор «свободных» связей, которые в сумме с неактивными связями образуют сетевой срез;
 - b. Помечаем все связи, найденные на шаге 2a неактивными, помещаем в стек связей;
 - c. Полученный на шаге 2b стек представляет собою модифицированный срез сети, вероятности связей, помещенных в этот стек складываем с нарастающей суммой;

//Процедура поиска ошибки снизу (backtracking procedure)

3. Если стек пустой, переходим к выходу из процедуры, иначе
 - a. Выбираем связь из начала стека;
 - b. Если выбранная связь неактивна и образует spanning tree активных связей, будучи активной, помечаем выбранную связь «свободной» и переходим к шагу 2b;
 - c. Если выбранная связь неактивна и, будучи активированной, не удовлетворяет условию 3b, помечаем выбранную связь как активную, помещаем в конец стека и переходим к шагу 2;
 - d. Если выбранная связь активная, помечаем ее как «свободная» и переходим к шагу 3a.

Для проверки работоспособности данной процедурной модели была использована методика, предложенная авторами [3, 63 – 70].

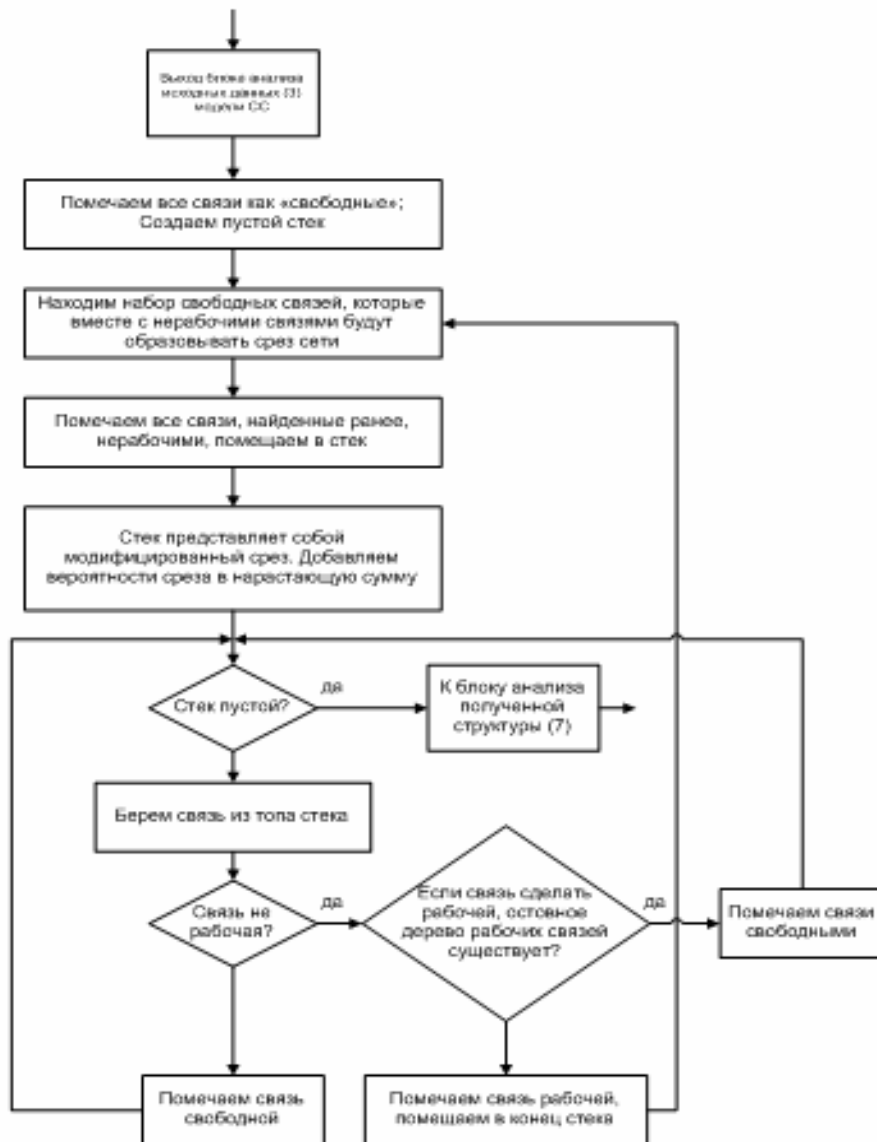


Рис. 2.4. Процедурная модель модуля оценки живучести СИС с использованием элементов искусственного интеллекта

Данная процедурная модель является достаточно универсальной для вычисления общей живучести СИС, граф которой может иметь произвольную топологию и очень большое количество вершин и ребер (см. параграф 1.5). Однако для графов малого диаметра ($d < 4$) и содержащих в своем составе менее 20 вершин количество циклов вычислений для данной модели является избыточным. Из данной ситуации есть два выхода: либо вводить в модель механизм выявления и отсеивания «ненужных» вычислений, либо использовать более простую в плане вычислений процедурную модель, такую, как потоковая (модель МП-сеть). Также следует отметить, что процедура дефаззификации для данной модели проводилась после проведения расчета живучести СИС, так как данная модель работает с нечеткими значениями, предварительная дефаззификация не требуется.

Потоковая модель (модель № 3). Процедурная модель определения критерия допустимости $\Theta_0(d)$ (или, как ее еще называют, потоковая процедурная модель оценки функциональной живучести системы) основана на аналитическом обеспечении, описанном в параграфе 1.6, и выглядит следующим образом (рис. 2.5).

1. Создать какой-либо поток f и, соответственно, мультипоток $Z(f)$.
2. Перебрать все ребра графа G , определить для каждого ребра z_i, d_i .
3. Исходя из условия допустимости $Z \geq d$, составить отношения $\frac{z_i}{d_i}$. Если все эти значения будут больше единицы,

СИС допустима.

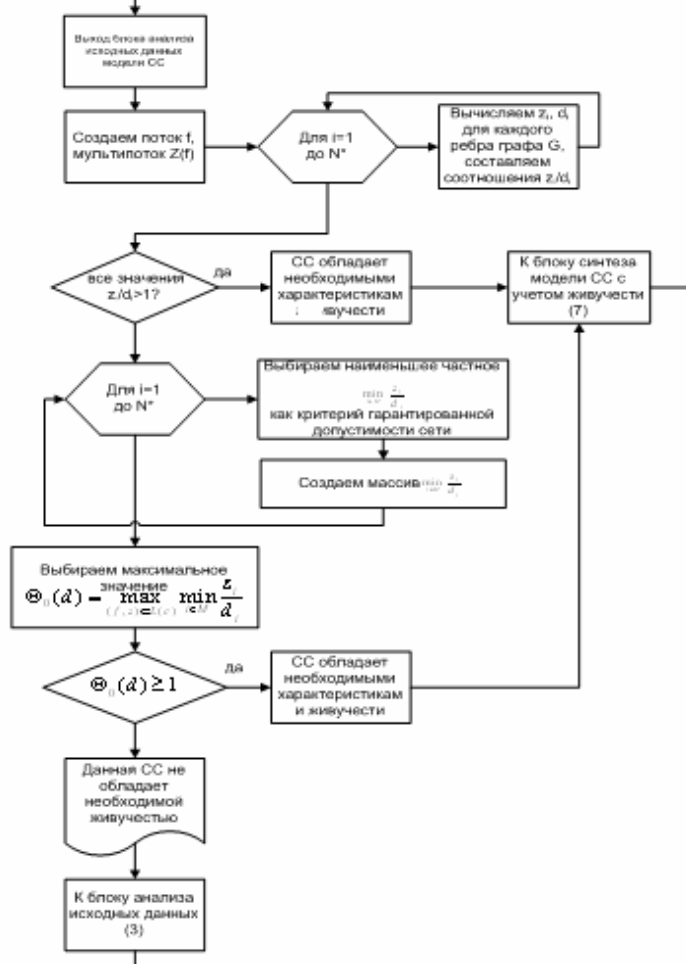
4. Наименьшее значение частного $\min_{i \in M} \frac{z_i}{d_i}$ является критерием гарантированной допустимости СИС.

5. Создать массив $\min_{i \in M} \frac{z_i}{d_i}$. Из полученного массива выбрать максимальное значение $\frac{z_i}{d_i}$.

$$\Theta_0(c, d) = \max_{(f, z) \in L(c)} \min_{i \in M} \frac{z_i}{d_i}.$$

Если $\Theta_0(d) \geq 1$ – СИС гарантированно допустима; если $\Theta_0(d) < 1$ – СИС недопустима.

Итак, даже в случае недопустимости СИС в целом, соответствующее перераспределение потоков может сделать ее хотя бы слабо допустимой. Но ресурсов СИС может сильно нехватать для обеспечения требований тяготеющих пар, и тогда возникает вопрос о конкуренции или компромиссе между ними: либо постановка в очередь, либо отказ в обслуживании. Но сложная структура СИС позволяет выбрать из конкурентных распределений потоков максимально обеспечивающее требования всех тяготеющих пар без дискриминации какой-либо из них.



N – общее количество ребер

Рис. 2.5. Процедурная модель оценки допустимости СИС

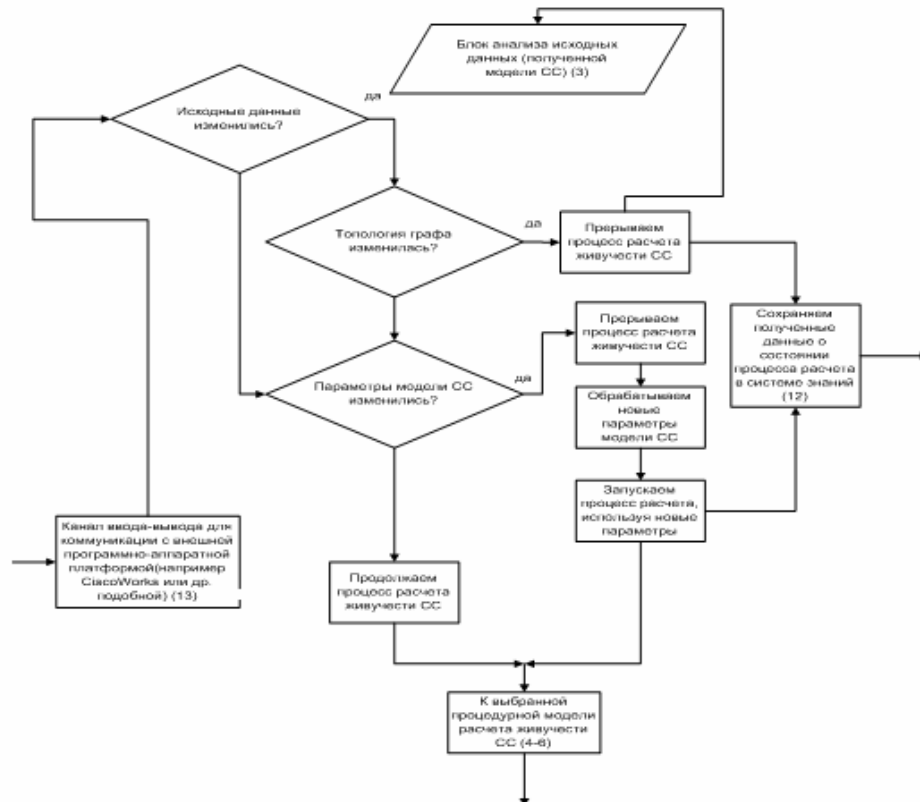


Рис. 2.6. Процедурная модель модуля АТС

Такое распределение потоков было впервые предложено в работе [89] и названо нормативным (описание организации нормативного потока относится к аналитическому обеспечению ИСЖС и описано в п. 1.6.3).

Модуль анализа текущего состояния (АТС). Модуль анализа текущего состояния СИС предназначен для контроля состояния процесса расчета живучести СИС в условиях быстро меняющихся параметров модели (другими словами, осуществляет контроль параметров расчета модели живучести СИС «на лету», в реальном времени). Данный модуль соединен по интерфейсному каналу с исследуемой (анализируемой) СИС через какую-либо программно-аппаратную платформу (например, CiscoWorks или любую другую). Модуль АТС различает три типа данных, поступающих через данный интерфейс:

1. критическое состояние изменения параметров;
2. незначительное состояние изменения параметров СИС;
3. контрольное состояние параметров СИС.

При значительном изменении параметров СИС (таких, как, например, топология или размерность графа исследуемой СИС) происходит прерывание процесса расчета, сохранение контрольного состояния СИС на момент останова в СЗ, полная перепараметризация модели СИС, повторный выбор процедурной модели расчета живучести и запуск процесса расчета с новыми глобальными параметрами (рис. 2.6).

При незначительном изменении параметров (таких как изменение пропускной способности ребра, изменение количества передаваемых потоков по СИС и т.д., когда не нарушается топология графа СИС) процесс расчета также прерывается с записью состояния в СЗ, но новые параметры модели обрабатываются без выбора процедуры расчета живучести, а ведутся по уже выбранной.

Наконец, контрольное состояние системы вводится в модуль АТС исключительно для создания точек возврата в процессе расчета живучести СИС, так как сам процесс расчета не прерывается, а приостанавливается, данные о состоянии системы записываются в СЗ, далее процесс расчета продолжается.

Если процесс расчета живучести СИС завершен, на модуль АТС передается контрольное состояние результатов расчета и данные передаются в блок анализа и синтеза структуры СИС.

2.3. АНАЛИЗ СЕТЕВОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ НА ОСНОВЕ МОДЕЛИ МП-СЕТИ

Имеем СИС на базе сети передачи данных (СПД) с коммутацией пакетов (кадров) (рис. 2.7).

Постановка задачи анализа. Задана СИС с коммутацией пакетов (кадров), задана матрица входящих потоков H , СИС задана в виде орграфа $G = (X, E)$, где $X = \{X_j\}$ – множество узлов связи (УС), $j = 1, n$, E – множество ребер, $E = \{(r, s)\}$ –

множество каналов связи (КС), \forall КС (r, s) характеризуется своей длиной – l_{rs} (км), пропускной способностью – μ_{rs} (бит/с), (пакет/с), стоимостью – $C_{rs}(l_{rs}, \mu_{rs})$.

Для СИС с заданной структурой, где

$$l_{rs} = \text{const} \Rightarrow C_{rs} = C_{rs}(\mu_{rs}).$$

Матрица входящих требований:

$$H = \|h_{ij}\|, \quad i, j = \overline{1, n} \text{ (пакетов/с)},$$

где h_{ij} – интенсивность потока, который необходимо передать от i до j .

Здесь по каналам связи протекают потоки f_{rs} , вектор многопродуктового потока (МП) $F = [f_{rs}]$, при этом $f_{rs} \leq \mu_{rs}, \forall (r, s) \in E$.

Рассмотрим основные характеристики СИС:

- 1) задержка между смежной парой узлов t_{rs} ;
- 2) T_{ij} – средняя задержка между заданной парой узлов;
- 3) T_{cp} – общая средняя задержка в СИС между произвольной парой узлов;
- 4) $C_{\Sigma} = \sum_{(r,s) \in E} C_{rs}(\mu_{rs})$ – суммарная стоимость СИС.

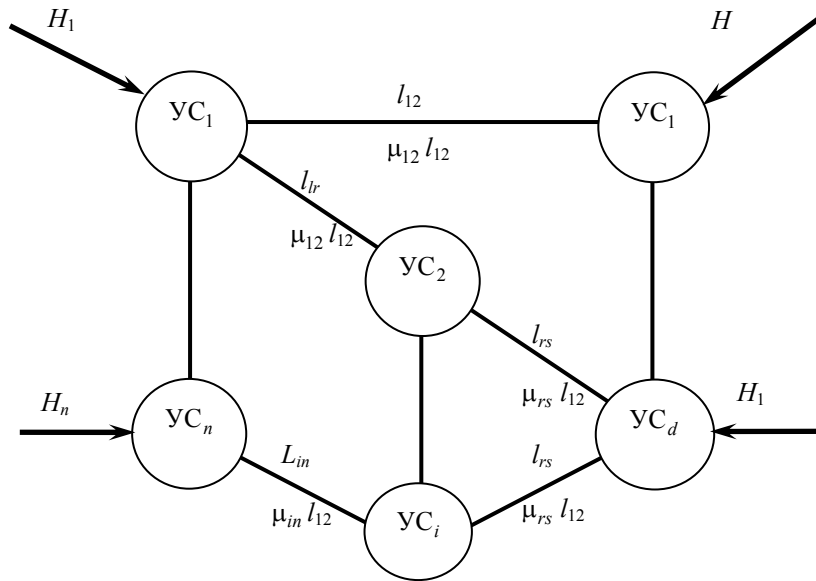


Рис. 2.7. Функциональная схема СИС с коммутацией пакетов (кадров)

Введем основные допущения по СИС (предложены Л. Клейенроком):

1. Входящие потоки – пуассоновские. Если входящий поток с интенсивностью l_i , то время задержки между соседними входящими требованиями распределено:

$$p(t_{\text{пак}} \leq \tau) = 1 - e^{-\lambda_i \tau}, \quad \text{где} \quad \bar{T}_{\text{пост}}^i = \frac{l_i}{H_i}. \quad (2.1)$$

2. Задержкой в узлах пренебрегаем, считаем, что они имеют буфер неограниченной емкости.

3. Обслуживание в каналах – показательное с параметром μ_{rs} – интенсивность обслуживания.

4. Выполняется гипотеза независимости (по Клейенроку): «Предположим, что времена обслуживания одного и того же пакета в различных каналах являются статистически независимыми случайными величинами».

Это позволяет вывести выражение для средней задержки доставки пакета по СИС в целом.

Рассмотрим задержку в одном канале (рис. 2.8) в соответствии с принятыми допущениями:

$$p(t_{\text{обсл. } r,s} = \mu_{rs} e^{-\mu_{rs} t}. \quad (2.2)$$

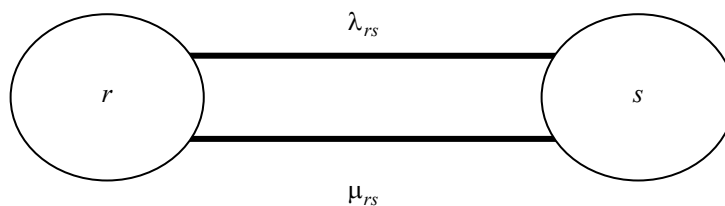


Рис. 2.8. Схема задержки в одном канале

Так как канал обслуживания рассматривается как система $(M | M | l)$, то

$$\bar{t}_{rs} = \bar{t}_{\text{ож}(r,s)} + \bar{t}_{\text{обсл}(r,s)} = \frac{1}{\mu_{rs} - \lambda_{rs}} = \frac{1}{\mu_{rs}(1 - \rho_{rs})}, \quad (2.3)$$

где ρ – загрузка канала.

Допустимый канал состоит из нескольких транзитных участков и необходимо определить задержку в доставке на этом маршруте. Очевидно:

$$\bar{T}_{ij} = \sum_{(r,s) \in M_{ij}} \bar{t}_{rs}, \quad (2.4)$$

где t_{rs} – задержка в КС (r, s) ; M_{ij} – маршрут, соединяющий узел i и j .

Если поток идет по нескольким маршрутам от i до j , то λ_{ij} разделяется на составляющие (рис. 2.9):

$$\lambda_{ij} = \{\lambda_{ij}^1, \dots, \lambda_{ij}^k\};$$

$$\lambda_{ij} = \sum_{k=1}^k \lambda_{ij}^k,$$

где λ_{ij}^k – доля потока, протекающая по k -му маршруту.

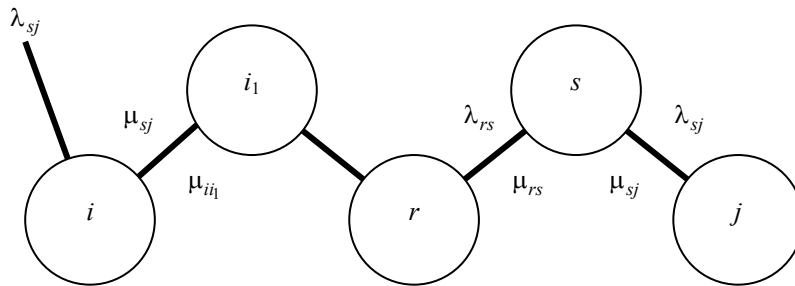


Рис. 2.9. Пример разделения потока на составляющие

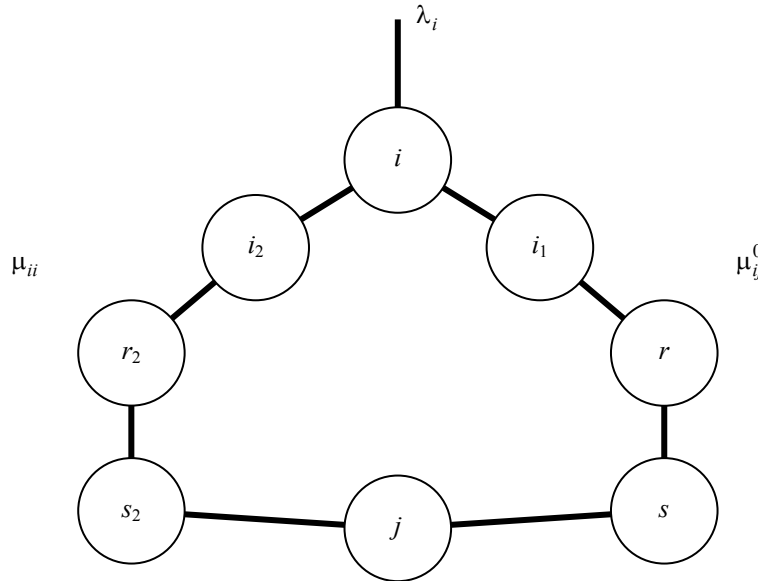


Рис. 2.10. Пример разделения потока на составляющие

$$\bar{T}_{ij} = \sum_{k=1}^k p_{ij}^k \bar{T}_{ij}^k, \quad (2.5)$$

где p_{ij}^k – вероятность выбора k -го маршрута M_{ij}^k ; T_{ij}^k – задержка на k -м маршруте; $\bar{T}_{ij} = \frac{1}{\lambda_{ij}} \sum_{k=1}^k \lambda_{ij}^k T_{ij}^k$. Подставляя T_{ij} из формулы (2.1), получим:

$$\bar{T}_{ij} = \frac{1}{\lambda_{ij}} \sum_{k=1}^k \sum_{(r,s) \in M_{ij}^k} \frac{\lambda_{ij}^k}{\mu_{rs} - \lambda_{ij}^k}, \quad (2.6)$$

$\lambda_{rs} = \lambda_{ij}^k, \forall (r, s) \in M_{ij}^k$, если маршруты не пересекаются.

Найдем среднюю задержку между произвольной парой узлов, продолжая аналогичным образом:

$$\bar{T}_{cp} = \frac{1}{h_{\Sigma}} \sum_{(r,s) \in E} \frac{f_{rs}}{\mu_{rs} - f_{rs}}, \quad (2.7)$$

где $h_{\Sigma} = \sum_i \sum_j h_{ij}$ – суммарная величина входного потока; f_{rs} – суммарный поток, протекающий по КС (r, s) ,

$$f_{rs} = \sum_i \sum_j \lambda_{rs}^{i,j}, \quad (2.8)$$

$\lambda_{rs}^{i,j}$ – составляющая глобального потока, передаваемого от узла i до j по каналу (r, s) .

Задачи анализа:

1. Выбор пропускных способностей КС (ВПС).
2. Распределение потоков (РП) в СИС.
3. Комбинированная задача (ВПС РП).

2.3.1. ЗАДАЧА ВЫБОРА ПРОПУСКНЫХ СПОСОБНОСТЕЙ

Постановка задачи. Заданы структура СИС в виде ориентированного графа: $G = (X, E)$, распределение потоков в КС, $\{f_{rs}\}, (r, s) \in E$, набор пропускных способностей КС – $D = \{d_1, \dots, d_k\}$ и удельных стоимостей каналов в зависимости от типа $C = \{C_1, C_2, \dots, C_k\}$ на единицу длины.

Требуется найти такие пропускные способности каналов $\{\mu_{rs}\}$ всех КС, для которых

$$C_{\Sigma} = \sum_{(r,s) \in E} C_{rs}(\mu_{rs}) \rightarrow \min(1) \quad (2.9)$$

и средняя задержка T_{cp} не будет превышать заданной величины $T_{зад}$:

$$T_{cp} = \frac{1}{h_{\Sigma}} \sum_{(r,s)} \frac{f_{rs}}{\mu_{rs} - f_{rs}} \leq T_{зад} \quad (2.10)$$

и $\mu_{rs} > f_{rs}, (r, s) \in E$.

Решение. В случае, если μ_{rs} – непрерывные переменные, можно применить метод множителей Лагранжа, составить функцию Лагранжа и решить задачу строго аналитически.

$$L(\mu_{rs}, \lambda) = \sum_{(r,s)} C_{rs}(\mu_{rs}) + \lambda \left(\frac{1}{h_{\Sigma}} \sum_{(r,s)} \frac{f_{rs}}{\mu_{rs} - f_{rs}} - T_{зад} \right);$$

$$\frac{dL(\mu_{rs}, \lambda)}{d\mu_{rs}} = 0;$$

$$\frac{dL}{d\lambda} = \frac{1}{h_{\Sigma}} \sum_{(r,s)} \frac{f_{rs}}{\mu_{rs} - f_{rs}} - T_{зад} = 0. \quad (2.11)$$

Решим систему (2.11) относительно неизвестных μ_{rs} .

Можно найти численное решение из системы (2.3) $\Rightarrow \{\mu_{rs}^0\}$.

Если стоимости являются линейными:

$$C_{rs}(\mu_{rs}) = C_{rs}^0 + C_{rs}^1 \mu_{rs}, \quad (2.12)$$

то получаем следующее решение для оптимальных пропускных способностей:

$$\mu_{rs}^0 = f_{rs} + \frac{f_{rs}}{h_{\Sigma} T_{зад}} \left(\frac{\sum_{k,l} \sqrt{C_{k,l}^1} f_{kl}}{\sqrt{C_{rs}^1} f_{rs}} \right). \quad (2.13)$$

Недостаток. Такой подход применим, если пропускные способности являются непрерывными и статистически-линейными функциями.

В случае, если μ_{rs} – дискретные, $\mu_{rs} \in D = \{d_1, \dots, d_k\}$, Клейенрок предлагает округлить (2.13) до ближайшего члена из этого ряда. Имеем задачу нелинейного дискретного программирования.

$$D = \{2400, 4800, 9600, 14400, 19200, \dots\}.$$

2.3.2. ПРОЦЕДУРНАЯ МОДЕЛЬ ВЫБОРА ДИСКРЕТНЫХ ПРОПУСКНЫХ СПОСОБНОСТЕЙ

Для дискретного случая задача решается по-другому, например методом ПАВ.

Пусть требуется: $C_{\Sigma} = \sum_{(r,s)} C_{rs}(\mu_{rs}) \rightarrow \min$ при условии

$$T_{cp} = \frac{1}{h_{\Sigma}} \sum_{(r,s)} \frac{f_{rs}}{\mu_{rs} - f_{rs}} \leq T_{зад}, \quad (2.14)$$

где $\mu_{rs} \in D = \{d_1, d_2, \dots, d_k\}$

Метод ПАВ состоит из процедур отсева W_1 и W_2 .

Процедура W_1 : Определим диапазон возможных значений по каждому каналу (r, s) :

$$M_{rs}^0 = \{\mu_{rs(\min)}^0, \mu_{rs(\max)}^0\},$$

где $\mu_{rs(\min)}^0 : \frac{1}{h_{\Sigma}} \frac{f_{rs}}{(\mu_{rs} - f_{rs})} = T_{зад}$.

Начинается процедура отсева по каналу связи (r, s) :

$$\frac{\mu_{rs}}{h_{\Sigma}(\mu_{rs} - f_{rs})} > T_{зад} - \frac{1}{h_{\Sigma}} \sum_{(k,e) \neq (r,s)} \frac{f_{KC}}{\mu_{KC}^{\max} - f_{RC}}. \quad (2.15)$$

Отсеиваются $\mu_{rs} \leq \bar{\mu}_{rs}$, где $\bar{\mu}_{rs}$ – наибольшее значение, при котором начинает выполняться условие (2.15).

W_1 повторяется для всех КС $\forall (r, s)$, и при этом отсеиваются нижние значения пропускных способностей. Получили новое множество вариантов:

$$M_{rs}^1 = \{\mu_{rs(\min)}^1, \dots, \mu_{rs(\max)}^0\}.$$

Переходим к W_2 – отсев по значениям целевой функции. Отсеиваем верхние значения μ_{rs} .

Задаемся некоторым начальным порогом:

$$C_1^* = \frac{1}{2} (C_{\Sigma_{\min}} + C_{\Sigma_{\max}}).$$

Начинаем отсев. Отсеивается все, что выше C_1^* , т.е. $C_{\Sigma}(\{\mu_{rs}\}) > C_1^*$.

Условие отсева значений ПС для КС (r, s) :

$$\mu_{rs} : C_{rs}(\mu_{rs}) > C_1^* - \sum_{\forall (k,l) \neq (r,s)} C_{kl}(\mu_{kl}^{\min}). \quad (2.16)$$

Отсеиваются все значения $\mu_{rs} > \bar{\mu}_{rs}$, где $\bar{\mu}_{rs}$ – минимальное значение, при котором начинает выполняться (2.5).

Если отсев произошел, то переходим на процедуру W_1 , если нет, то вновь проходим W_2 и сужаем множество вариантов.

Выбираем новый порог отсева C_2^* , согласно:

- 1) $C_2^* = \frac{1}{2} (C_1^* + C_{\Sigma_{\min}})$ – для сужения, если отсева нет;
- 2) $C_2^* = \frac{1}{2} (C_1^* + C_{\Sigma_{\max}})$ – для расширения, если все отсеивается.

W_1 и W_2 повторяются многократно, до тех пор, пока не получаем сокращенное множество вариантов $M_{r,s}^k$, где количество каналов не превышает 1-2. Далее выбор оптимального решения осуществляется простым перебором.

2.3.3. КОМБИНИРОВАННАЯ ЗАДАЧА ВЫБОРА ПРОПУСКНЫХ СПОСОБНОСТЕЙ И РАСПРЕДЕЛЕНИЯ ПОТОКОВ. ПОСТАНОВКА ЗАДАЧИ

Задано: структура СИС $G = (X, E)$, матрица требований $H = \|h_{ij}\|$, набор пропускных способностей каналов связи $D = \{d_1, d_2, \dots, d_k\}$ и соответствующих удельных стоимостей каналов связи $C = \{c_1, c_2, \dots, c_k\}$. Необходимо найти такое распределение потоков $\Lambda^0 = [\lambda_{rs}^0]$ и соответствующие пропускные способности всех каналов связи $\{\mu_{rs}^0\}$, при которых

$$C_{\Sigma} = \sum_{(r,s) \in E} C_{rs}^{\text{неп}}(\mu_{rs}^0) \rightarrow \min \quad (2.17)$$

при ограничении

$$P_{CD} = \prod_{r,s \in E} \frac{\mu_{rs} k_{\Gamma} - \frac{\lambda_{rs}}{n_{rs}}}{\mu_{rs} k_{\Gamma} - \frac{\lambda_{rs}}{n_{rs}} + v_y \frac{\lambda_{rs}}{\Lambda_{\Sigma}}} \geq P_{зад}^{CD} \quad (2.18)$$

и условия, что $\Lambda = [\lambda_{rs}]$ – многопродуктовый поток, совместимый с матрицей $H = \|h_{ij}\|$.

2.3.4. ОБОБЩЕННАЯ ЗАДАЧА ВЫЧИСЛЕНИЯ ПРОПУСКНЫХ СПОСОБНОСТЕЙ И РАСПРЕДЕЛЕНИЯ ПОТОКОВ

Заданы:

1. Условные стоимости передачи одного пакета (кадра) информации C'_{ij} .
2. Матрица требований в передаче информации $\|H\|$.

Процедурная модель решения ВПС РП использует многократное решение комбинации задач ВПС РП. Она состоит из начального этапа и конечного числа однотипных итераций.

Начальный этап:

1. В качестве метрики $L^0 = [l_{rs}^0]$ используем длины всех каналов связи.
2. Определяем кратчайшие пути в СИС с выбранной метрикой π_{ij}^{\min} и находим начальный поток $\Lambda^0 = [\lambda_{rs}^0]$ по кратчайшим путям.
3. Решаем задачу ВПС по критерию $\min C_{\Sigma}$ при ограничении $P_{CD}(\{\mu_{rs}\}) \geq P_{зад}^{CD}$ и находим начальные пропускные способности всех каналов связи $\{\mu_{rs}^0\}$.

Переходим к 1-й итерации.

Пусть уже проведено k итераций и найден некоторый поток в СИС $\Lambda(k) = [\lambda_{rs}(k)]$ и соответствующие ему пропускные способности всех каналов $\mu_{rs}(k)$, $(r, s) \in E$.

$(k+1)$ -я итерация.

1. Решаем задачу РП при заданных пропускных способностях $\{\mu_{rs}(k)\}$ и находим такое распределение потоков $\Lambda^0(k+1) = [\lambda_{rs}^0(k+1)]$, которое обеспечивает $\max P_{CD}(\lambda_{rs}^0(k+1))$.
2. При новом распределении потоков $\Lambda^0(k+1)$ решаем задачу ВПС по критерию $\min C_{\Sigma}(\{\mu_{rs}(k+1)\})$ при ограничении $P_{CD}(\{\mu_{rs}(k+1)\}) \geq P_{зад}^{CD}$ и находим новые оптимальные значения пропускных способностей всех каналов связи $\{\mu_{rs}^0(k+1)\}$.
3. Вычисляем новое значение критерия $C_{\Sigma}(k+1) = \sum C_{rs}(\mu_{rs}(k+1), l_{rs})$. Сравниваем $C_{\Sigma}(k+1)$ и $C_{\Sigma}(k)$. Если $|C_{\Sigma}(k) - C_{\Sigma}(k+1)| > E$, то переходим к шагу 1 $(k+1)$ -й итерации, иначе наступает конец работы процедурной модели. Распределение потоков $\Lambda(k+1) = [\lambda_{rs}(k+1)]$ и пропускные способности $\mu_{rs}(k+1)$, $(r, s) \in E$ найдены.

2.3.5. ПРОЦЕДУРНАЯ МОДЕЛЬ НАХОЖДЕНИЯ КРАТЧАЙШИХ ПУТЕЙ И РАСЧЕТА ОБЪЕМОВ СУММАРНОЙ ПЕРЕДАЧИ ИНФОРМАЦИИ (ТРАФИКОВ)

В процессе работы процедурных моделей ВПС РП и нахождения максимального потока в СИС при отказах многократно используется процедурная модель нахождения кратчайших путей и расчета информационных потоков (трафиков) в многосвязной СИС.

Пусть имеем многосвязную СИС, которая задана в виде графа $G(X, E)$, где $X = \{x_j\}$ – множество вершин графа; $E = \{(i, j)\}$ – множество его ребер. Будем считать, что при нескольких маршрутах передачи информации от x_i к x_j она всегда передается по пути минимальной стоимости.

Необходимо рассчитать параметры СИС $G(X, E)$: такие маршруты передачи информации и информационные потоки в каждом канале f_{rs} , $(r, s) \in E$, при которых общая стоимость информации будет минимальной, при этом задана матрица требований $H = \|h_{ij}\|$.

Исходные данные:

Матрица смежности $B = \|b_{ij}\|$, где:

$$b_{ij} = \begin{cases} 1, & \text{если } x_i \text{ смежна с } x_j; \\ 0 & \text{в другом случае.} \end{cases}$$

Матрица удельных стоимостей передачи информации $C = \|c_{ij}\|$.

В процессе работы процедурной модели используются такие рабочие матрицы:

$A = \|a_{ij}\|$ – матрица удельных стоимостей передачи информации из x_i в x_j по пути минимальной стоимости π_{ij}^{\min} (если $b_{ij} = 1$, то $a_{ij} = c_{ij}$), $M = \|m_{ij}\|$ – матрица минимальных маршрутов π_{ij}^{\min} по СИС, где $m_{ij} = r$ – вершина, смежная с x_j на пути π_{ij}^{\min} , $F = \|f_{ij}\|$ – искомая матрица информационных потоков (или трафиков) в СИС, где f_{ij} – суммарный трафик, который передается по каналу связи (i, j) .

Описание процедурной модели. Процедурная модель состоит из конечного числа однотипных итераций, в ходе которых строят и уточняют матрицы $A(k), M(k)$.

1-я итерация.

1. Определяем матрицу $A(1) = \|a_{ij}\|$: $a_{ij}(1) = \begin{cases} c_{ij}, & \text{если } b_{ij} = 1; \\ \infty & \text{в другом случае.} \end{cases}$

2. Заполняем матрицу маршрутов $M(1)$ в соответствии с соотношением $m_{ij}(1) = \begin{cases} i, & \text{если } b_{ij} = 1; \\ 0 & \text{в другом случае.} \end{cases}$

Таким образом, в результате 1-й итерации рассчитаны характеристики СИС для маршрутов длины 1.

2-я итерация.

Определяем минимальные маршруты длиной $l = 2$ и рассчитываем $A(2), M(2)$.

1. Просматриваем i -ю строку матрицы $A(1)$, $i = 1, 2, \dots$ и определяем

$$a_{ij}(2) = \min_{s \neq j} \{a_{is}(1) + a_{sj}(1)\} < a_{ij}(1). \quad (2.19)$$

2. Если $a_{ij}(1) = a_{ij}(2)$, то $m_{ij}(2) = m_{ij}(1)$, и, приравняв $j = j + 1$, переходим к рассмотрению следующего элемента i -й строки.

Если

$$a_{ij}(2) = \min_{s \neq j} \{a_{is}(1) + a_{sj}(1)\} < a_{ij}(1),$$

то меняем элемент m_{ij} в матрице M :

$$m_{ij}(2) = m_{sj}(1). \quad (2.20)$$

3. Меняем i от 1 до n и выполняем шаг 2 со всеми строками матрицы $A(1)$.

4. Если $A(2) = A(1)$ и $M(2) = M(1)$ и в матрице нет пустых элементов, то наступает конец работы процедурной модели, иначе переходим к следующей итерации.

Пусть проведено $(k - 1)$ итераций, в ходе которых построены матрицы $A(k - 1), M(k - 1)$.

В ходе выполнения k -й итерации строим $A(k), M(k)$, где $M(k)$ – маршрутная матрица, которая содержит минимальные пути длиной не более $2(k - 1)$ (по количеству каналов).

1. Просматриваем i -ю строку матрицы $A(k - 1)$ и определяем $a_{ij}(k)$:

$$a_{ij}(k) = \min_{s \neq j} \{a_{ij}(k - 1), a_{is}(k - 1) + a_{sj}(k - 1)\}. \quad (2.21)$$

2. Если $a_{ij}(k) = a_{ij}(k - 1)$, то $m_{ij}(k) = m_{ij}(k - 1)$, приравняв $j = j + 1$, переходим к рассмотрению следующего элемента i -й строки.

Если

$$a_{ij}(k) = \min_{s \neq j} \{a_{is}(k - 1) + a_{sj}(k - 1)\} = a_{is'}(k - 1) + a_{s'j}(k - 1) < a_{ij}(k - 1),$$

то меняем элемент m_{ij} в маршрутной матрице $M(k - 1)$ $m_{ij}(k) = m_{s'j}(k - 1)$. Меняя i от 1 до n , шаг 2 повторяем со всеми строками матрицы $A(k - 1)$.

3. Если $A(k) = A(k - 1)$ и $M(k) = M(k - 1)$ и в матрице $M(k)$ нет пустых элементов, то конец работы процедурной модели, матрица $M(k) = M$ – матрица кратчайших путей, а $A(k) = A$ – соответствующая ей матрица длин этих путей.

Иначе переходим к $k + 1$ -й итерации.

Итак, допустим, что построена маршрутная матрица M . Покажем, как при помощи ее можно найти кратчайший путь π_{ij}^{\min} между произвольной парой узлов (i, j) .

1. Рассмотрим элемент m_{ij} . Если $m_{ij} = i$, то конец, $\pi_{ij}^{\min} = \{i - j\}$. Допустим, что $m_{ij} = s_1$.

2. Находим в матрице M элемент m_{is_1} . Пусть он равняется $m_{is_2} = s_2$.

3. Если $s_2 = i$, то конец работы процедурной модели, искомый путь $\pi_{ij}^{\min} = \{i - s_1 - j\}$. Иначе переходим к шагу 2, приравняв $s_1 = s_2$.

Допустим, что через k итераций получен элемент $m_{is_k} = i$. Тогда кратчайший путь $\pi_{ij}^{\min} = \{i - s_k - s_{k-1} - \dots - s_2 - s_1 - j\}$.

Процедурная модель построения потока $\{f_{ij}\}$ через ребра графа G на содержательном уровне имеет следующую структуру:

1. Записывается матрица требований $H = \|h_{ij}\|$.

2. Составляется матрица смежности $B = \|b_{ij}\|$.

3. Составляется матрица $A(1) = \|a_{ij}\|$ кратчайших расстояний между соседствующими узлами, которая является вектором «длин» ребер, записанная в матричной форме, при этом $a_{ij} = a_{ji}$.

4. Составляется матрица кратчайших маршрутов $M(1)$. Это та же матрица смежности B , но теперь указывается конкретный номер соседнего узла. На следующем этапе разыскивается расстояние кратчайших путей между вершинами, не являющимися смежными, но соединенными логическими связями.

5. Составляется матрица $A(2)$ кратчайших расстояний, элементы которой представляют суммы «длин» ребер, по которым проходит маршрут.

6. Составляется матрица $M(2)$ кратчайших маршрутов.

На этом построение кратчайших маршрутов заканчивается, так как все маршруты использованы (диаметр графа = 2).

Организуются потоки требований d между тяготеющими парами. Это и есть мультипоток $Z = \{z_{ij}\}$, так как для мультипотока следует принять $Z = d$.

В данном случае вектору Z приписываются 2 индекса, так как логические дуги двунаправлены, вершина v_i является источником потока z_{ij} и стоком z_{ji} от узла v_j (в соответствии с матрицей требований $\|h_{ij}\|$).

Сформируем потоки f_{ij} по соответствующим ребрам физического графа.

Между вершинами (v_1, v_2) , т.е. по ребру 1 – 2 проходит поток требований, равный 13 в прямом направлении и 6 – в обратном, по тому же ребру проходит поток $z_{13} = 11 + 5 = 16$ и поток $z_{24} = 14 + 4 = 18$. Таким образом, через ребро 1 – 2 проходит суммарный поток $f_1 = z_{12} + z_{13} + z_{24} = 53$.

Также разыскиваются потоки по остальным физическим ребрам.

Вектор пропускной способности y_{ij} рассчитывается по формуле (2.9):

$$y_{ij} = f_{ij} + k_{ij}f_{ij}; \quad k_{ij} < 1. \quad (2.22)$$

Теперь можно рассчитать матрицу информационных потоков по кратчайшим путям. Процедурная модель расчета потоков состоит из $k = n(n - 1)$ итераций.

1-я итерация.

$$k = 0, \quad F(0) = [f_{rs}(0)] = 0, \quad (r, s) \in E.$$

1. Выбираем 1-й элемент h_{12} из матрицы H .

2. Находим кратчайший путь π_{12}^{\min} по матрице M .

3. Тогда $f_{rs}(1) = \begin{cases} f_{rs}(0) + h_{12}, & \text{если } (r, s) \in \pi_{12}^{\min}; \\ f_{rs}(0) & \text{в другом случае.} \end{cases}$

Пусть уже проведено $(k - 1)$ итераций и определен вектор $F(k - 1)$.

k -я итерация.

1. Из матрицы H выбираем следующее требование, еще не распределенное по СИС. Пусть это $h_{i_k j_k}$.

2. Определяем кратчайший маршрут $\pi_{i_k j_k}^{\min}$.

3. Пересчитываем величину суммарного трафика на каналах маршрута $\pi_{i_k j_k}^{\min}$:

$$f_{rs}(k) = \begin{cases} f_{rs}(k-1) + h_{i_k j_k}, & \text{если } (r, s) \in \pi_{i_k j_k}^{\min}; \\ f_{rs}(k-1) & \text{в другом случае.} \end{cases}$$

4. Проверка условия $k = n(n - 1)$. Если да, то конец работы процедурной модели, иначе $k = k + 1$ и переход на шаг 1 $(k + 1)$ -й итерации.

Пример нахождения кратчайших путей и расчета объемов суммарной передачи информации (трафиков) приведен в прил. 6, а пример выделения мультипотока z по заданному вектору потока f – в прил. 7.

2.4. ПОСТРОЕНИЕ ГРАФИКА УЯЗВИМОСТИ

Гарантированный критерий живучести получают, положив показатель $\Theta_0(c, d) \equiv 1$. Избыточная проводимость отсутствует: $y = d$, т.е. исследуется мультипоток. Такая СИС при любом воздействии внешних НФ становится недопустимой, $\Theta_0 < 1$. Но в данном случае мы определяем не степень допустимости: проводимый анализ позволяет найти в СИС слабые участки, на которых сетевая структура наиболее уязвима.

Пример нахождения подобных участков описан в прил. 8 и иллюстрирует случай простейшей однопродуктовой СИС, функционирующей в условиях жесткого расписания (как, например, СИС теплоснабжения или железнодорожного сообщения).

Коммуникационные многопродуктовые СИС в отличие от примера всегда функционируют в режиме неопределенности, прежде всего, обоснованные компонентами вектора потоковых требований. Диспетчеру сетевой структуры вектор требований d известен с точностью до множества:

$$d \in D, \quad \text{где } D = \{d \mid d_{\min} \leq d \leq d_{\max}\}$$

или

$$D = \left\{ d \mid \sum_{i=1}^m d_i = d_0 \right\}.$$

Значение нулевого уровня максиминной обеспеченности тяготеющих пар оценивается $\Theta_0(d^{\min})$ сверху и $\Theta_0(d^{\max})$ снизу. Между этими экстремальными диаграммами допустимости располагается множество ступенчатых функций для различных распределений потоков, причем может оказаться, что $\Theta_0(d^{\max}) > \Theta_0^k(d^{\min})$, что дополнительно усложняет картину.

Рассмотрим сегмент СИС, топология которого близка к треугольной (рис. 2.11), где физический и логический графы потоковой СИС совпадают, вектор пропускной способности $C = (10, 15, 20)$, а для вектора d имеются две вероятности:

$$d(i) = \begin{cases} d^{\max} = (20, 20, 20); \\ d^{\min} = (20, 14, 20). \end{cases}$$

В первом случае по ребрам 1 – 2 необходимо пропустить поток в 40 единиц, тогда как в соответствии с вектором пропускной способности можно осуществить максимальный поток в 25 единиц, что дает:

$$\Theta_0(d^{\max}) = \frac{10+15}{20+20} = 0,625.$$

Для ребер 2 – 3 получаем:

$$\Theta_1(d^{\max}) = \frac{15+20}{40} = 0,875;$$

$$\mu_0 = \frac{10+15}{10+15+20} = 0,56.$$

Для минимального потока (20, 14, 20) получаем (рис. 2.12):

$$\Theta_0(d^{\min}) = 0,735; \quad \Theta_1(d^{\min}) = 1,03.$$

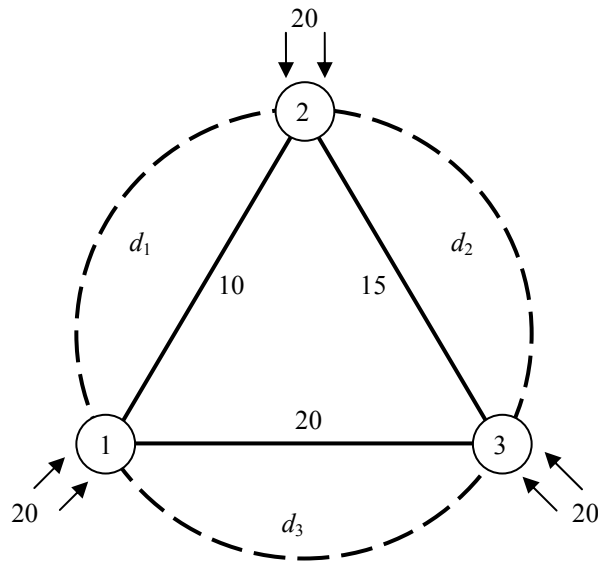


Рис. 2.11. Схема треугольного сегмента СИС

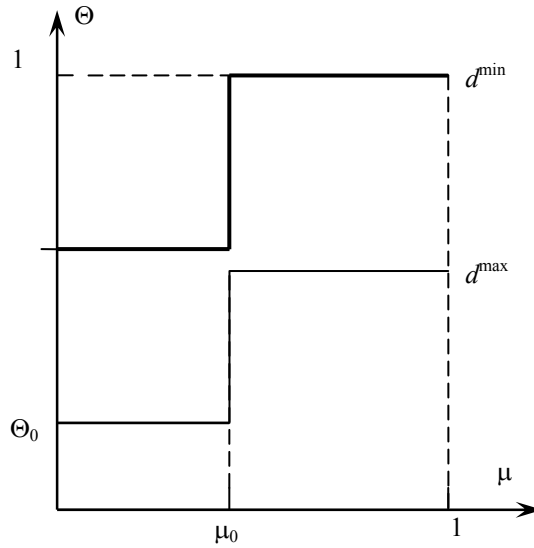


Рис. 2.12. Диаграмма критерия допустимости для графа треугольного сегмента СИС

Таким образом, для максимального потока $f^{\max} = (20, 20, 20)$ СИС недопустима: $\Theta < 1$. Для минимального потока $f^{\min} = (20, 14, 20)$ СИС слабо допустима, так как существуют ребра графа, где одновременно $\Theta < 1$ и $\Theta > 1$.

Диаграммы, подобные рис. 2.11, легко строятся для тривиальных небольших графов. Следует помнить, что возможность построения огибающей кривой по нижним «ступенькам» для произвольных графов не доказана. Не существует и единого численного метода для построения такой диаграммы [90].

Точно не известен также и вектор пропускной способности C . В случае гарантийной допустимости СИС нижний предел вектора пропускной способности определяется из соотношения $\Theta \geq 1$.

$$d = z = C;$$

$$d \in \mathfrak{Z}(c); \quad d = f(z).$$

Локальное поражение СИС в результате воздействия НФ приводит к уменьшению вектора пропускной способности физического графа и, следовательно, вектора требований d .

СИС, функционировавшая в режиме $\Theta(c, d) = 1$, становится недопустимой в жесткой постановке, однако остается слабо допустимой с неизвестным вектором требований d .

Предположим, что относительно вектора d известно лишь некоторое множество $D = \left\{ d \left| \sum_{i=1}^m d_i = d_0 \right. \right\}$ (данное соотношение в дальнейшем будет использовано при расчетах показателей живучести СИС).

При слабом поражении СИС гарантированная допустимость может сохраниться. Это условие записывается следующим образом:

$$D \subseteq L(c),$$

т.е. для $\forall d \in D$ найдется такое значение $z \in L(c)$, что $z \geq d$.

Это так называемая жесткая постановка допустимости. При значительном поражении может сохраниться слабая допустимость: $\exists z \in L(c); z \geq d; \forall d \in D$, т.е. найдется хоть один такой вектор мультипотокa z , который пройдет через СИС.

Рассмотрим случай двух тяготеющих пар, связанных ребром r_k , с пропускной способностью c_k . Множеству $L(c)$ будет соответствовать треугольник $c_k \cup c_k$ (см. рис. 2.13). Уменьшение пропускной способности c_k приведет к сдвигу отрезка $c_k c_k$ в направлении начала координат (т.е. к «0»). Множество D – это отрезок, соединяющий точки d_1 и d_2

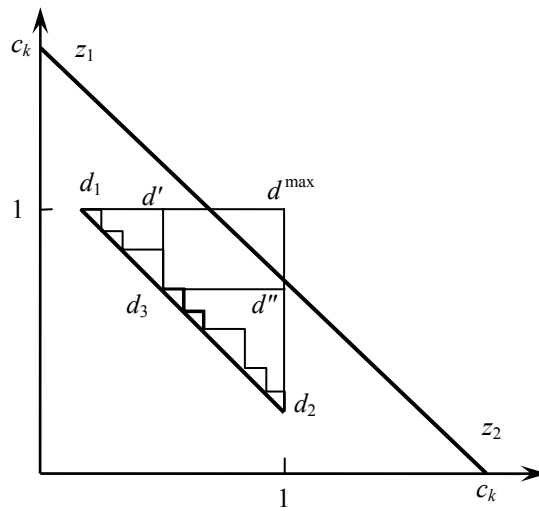


Рис. 2.13. Графическое отображение

(выполняется условие $D \subseteq L(c)$). Но существует значение d^{\max} , не укладывающееся в треугольник $c_k \cup c_k$, т.е. СИС не допустима. Однако, если поток запустить не по пути (v_1, v_2) , а найти обходной путь (v_1, v_3, v_2) , то соответствующие значения максимальных потоков d' и d'' удовлетворяют условию $D \subseteq L(c)$ – обеспечивается слабая допустимость: удовлетворяет условию не любой поток, но специально подобранный (рис. 2.13).

Может случиться, что таких специально подобранных путей для потоков будет несколько (мелкая «лесенка»). Подобная процедура обеспечивает живучесть СИС даже при значительном поражении ее пропускной способности.

2.5. ПРОЦЕДУРНАЯ МОДЕЛЬ АНАЛИЗА УЯЗВИМОСТИ СЕТЕВОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ

Рассмотренная ранее задача анализа СИС с неточно известными потоком требований и пропускной способностью ребер физического графа находит свое дальнейшее развитие в исследовании уязвимости СИС, возникающей под воздействием НФ стихийного или целенаправленного характера.

При подобном воздействии повреждается конкретный прототип «физического ребра» графа – медный кабель, оптическое волокно и т.д., с помощью которого осуществляется транспортировка потребляемого продукта. Пропускная способность C канала связи уменьшается, что ведет к уменьшению проходящего по ребру логического потока требований d .

Потери пропускной способности приводят к необходимости перераспределения потоков по оставшимся каналам, конкретная локализация места воздействия НФ и распределения его мощности по ребрам считается неизвестным. В результате гарантированная оценка функциональных возможностей СИС – ее живучесть – предполагает поиск наихудшего для сетевой структуры распределения воздействия НФ, т.е. поиск варианта уменьшения пропускной способности ребер, приводящего к максимальному ущербу функционирования СИС, что и служит задачей анализа уязвимости СИС.

Скалярной характеристикой качества функционирования СИС является мера ее допустимости. Наименьшее гарантированное значение уровня обеспеченности потока требований с учетом неопределенности дает следующая формула:

$$\Theta_0^{r*} = \Theta_0^{r*}(c, d) = \min_{c \in C} \max_{(f, z) \in X(c)} \min_{i \in M} \frac{z_i}{d_i}, \quad (2.23)$$

где C – множество векторов c в пределах заданной неопределенности.

$$C[\gamma] = \left\{ c > \bar{0} \mid c \leq c^0, \sum_{k=1}^e c_k = (1-\gamma) \sum_{k=1}^e c_k^0 \right\}, \quad \gamma \in [0, 1], \quad (2.24)$$

где c^0 – изначальный вектор пропускной способности ребер физического графа СИС; $C[\gamma]$ – множество векторов пропускной способности СИС после воздействия на нее НФ мощностью γ (где $\gamma=0$ означает отсутствие воздействия НФ, $\gamma=1$ – полное разрушение СИС), следовательно, наибольшее гарантированное значение уровня обеспеченности потоковых требований:

$$\Theta_0^*[\gamma] = \Theta_0^*[\gamma, d] = \min_{c \in C[\gamma]} \max_{(f, z) \in X(c)} \min_{i \in M} \frac{z_i}{d_i}; \quad \gamma \in [0, 1], \quad X = \mathfrak{Z} \cup f. \quad (2.25)$$

Предлагаемый подход к оценке поражающего воздействия НФ весьма условен. Предполагается, что воздействию подвергается вся СИС целиком, что применимо для небольшого (локализованного на ограниченной территории) физического графа, тогда как для распределенной СИС (например, магистральной СПД, WAN) трудно осуществимо.

Процедурную модель поиска гарантированного уровня допустимости СИС проиллюстрируем на примере исследования живучести радиальной иерархической потоковой СИС (рис. 2.14).

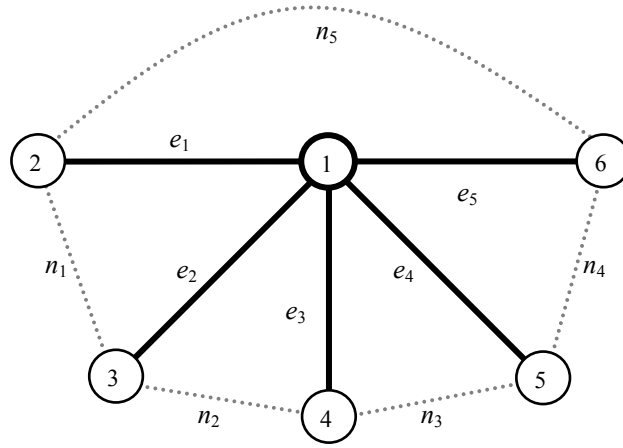


Рис. 2.14. Схема радиальной иерархической СИС

Логический граф – граф тяготений – имеет радиальную топологию. Все тяготеющие пары p_i заданы $(v_0, v_i), i \in M, M = \{1, 2, \dots, m\}$ с общим источником v_0 в узле 1. Управляющий центр находится в узле 1 (v_0), периферические узлы 2 – 6 – v_i . Физический граф G СИС повторяет ее логическую структуру. $G = (V, E)$, где $V = \{v_0, v_1, v_2, \dots, v_m\}$, $E = \{e_0, e_1, e_2, \dots, e_m\}$, $e_i = (v_0, v_i)$.

Подобный граф обладает плохими характеристиками живучести, поэтому наряду с ним рассматривается «укрепленный граф» $\bar{G} = (V, \bar{E})$ на том же множестве вершин v_i , но с удвоенным числом ребер: $\bar{E} = E \cup E^\circ$, где $E^\circ = \{e_{m+1}, e_{m+2}, \dots, e_{2m}\}$, $e_{m+i} = (v_i, v_{i+1}), i < m$, $e_{2m} = (v_m, v_1)$, т.е. все узлы нижнего уровня связаны дополнительной кольцевой структурой, с целью дублирования сообщений в случае потери связи между v_0 и v_i . В отсутствие НФ считаем, что набор данных по кольцевой структуре не передается, ребра E° – резервные.

$d = \{d_1, \dots, d_i\}$ – вектор требований на передачу потока между центром и подчиненными узлами, связанный физическими ребрами e_k , имеющими проводимость y_k .

Предположим, что логические ребра и ребра графа G ориентированы от центра к подчиненным узлам; ребра графа \bar{G} – двунаправлены. Введем на графе \bar{G} два типа потоковых переменных $f_{k-m}^{i+} \geq 0$ и $f_{k-m}^{i-} \geq 0$, для потоков, направленных против и по часовой стрелке, $k > m$; на радиальных ребрах $k \leq m$, потоковые переменные обозначим как $f_k^i \geq 0$.

При таких обозначениях условие неразрывности потоков (рис. 2.15) записывается следующим образом:

$$f_k^i = f_k^{i+} - f_{k-1}^{i+} + f_{k-1}^{i-} - f_k^{i-}; \quad \forall k \neq i; \quad f_i^{i+} = 0; \quad f_{i-1}^{i-} = 0. \quad (2.26)$$

Составим потоковую переменную:

$$z_i = \sum_{k \in M} f_k^i; \quad k > i. \quad (2.27)$$

Что собою представляет эта переменная?

$$\begin{aligned} \sum_{k \neq i} f_k^i &= \sum_{k=i+1}^{m+i-1} f_k^i = \sum_{k=i+1}^{m+i-1} [(f_k^{i+} - f_{k-1}^{i+}) + (f_{k-1}^{i-} - f_k^{i-})] = \\ &= f_{m+i-1}^{i+} - f_i^{i+} + f_i^{i-} - f_{m+i-1}^{i-} = f_{i-1}^{i+} + f_i^{i-}. \end{aligned} \quad (2.28)$$

Это означает: если поток по ребру r_i отсутствует (условие $k \neq i$), то к узлу v_i протекают потоки от узлов v_{i+1} и v_{i-1} , проходящие по ребрам r_{i+1} и r_{i-1} . Если ребро r_i цело, потоки f_{i-}^{i+}, f_i^{i-} отсутствуют и к узлу v_i идет поток f_i^i .

Таким образом, $z_i = f_i^i + f_i^{i-} + f_{i-1}^{i+}$ – это поток от центра к i -му подчиненному, т.е. является составляющей вектора мультипотока Z .

Набор потоковых переменных $f_k^i, f_k^{i-}, f_k^{i+}$ задает f -распределение потоков по СИС.

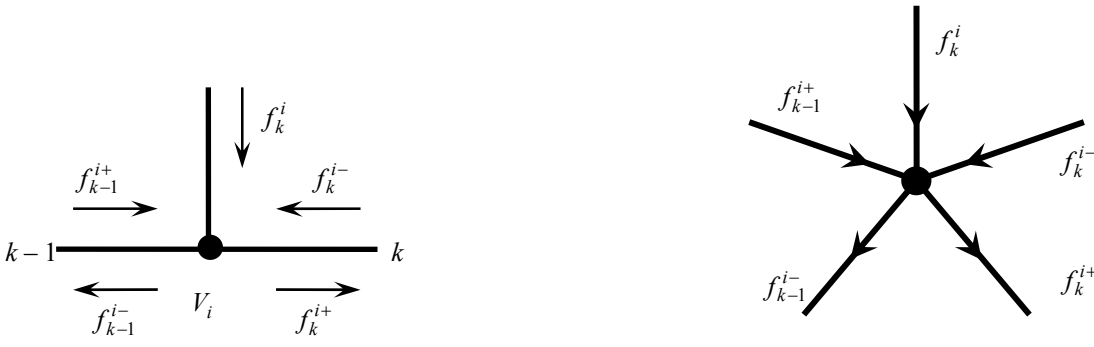


Рис. 2.15. Условие неразрывности потоков

$Z = z(f) = (z_1, \dots, z_m)$ – мультипоток, задаваемый распределением f .

Вектор $d = (d_1, \dots, d_m)$ определяет требование к мультипотoku:

$$Z = d .$$

Это условие допустимости СИС выполняется при ограничении: суммарный поток по ребру не должен превышать пропускную способность ребра:

$$\begin{aligned} \sum_{k \in M} f_k^i &\leq y_k \quad \forall e_k \in M; \\ \sum_{i \in M} (f_k^{i-} + f_k^{i+}) &\leq y_{k+m} \quad \forall e_{k+m} \in E^0. \end{aligned} \quad (2.29)$$

Множество всех мультипотоков $Z(f)$, удовлетворяющих этому условию, называют множеством допустимости СИС с вектором y пропускной способности физических ребер и обозначают $L(y)$.

$$\sum_{i \in M} Z_i = \sum_{i \in M} \sum_{i \in M} f_k^i \leq \sum_{k=1}^m y_k, \quad (2.30)$$

т.е. сумма величин потоков ограничена суммарной пропускной способностью лишь радиальных ребер.

Пусть исходная СИС, отображаемая графом G , выбрана оптимальным образом: все требуемые потоки проходят и избыточной пропускной способности не создается: $\Theta_0(c, d) = 1$. При любых повреждениях СИС перестает быть допустимой для вектора d . Даже при достижении равенства $\sum Z_i = \sum y_k$ при малейшем уменьшении y_k эффективность сильно уменьшается, так как она определяется не суммой величин потоков, а минимумом этих величин.

Введем для любого распределения потока f величину $\min_{i \in M} \frac{Z_i(f)}{d_i}$ – значение уровня обеспеченности потоковых требований. Мерой эффективности функционирования СИС будем считать:

$$\Theta_0(y) = \max_{z \in \mathfrak{Z}(y)} \min_{i \in M} \frac{Z_i}{d_i}. \quad (2.31)$$

Пусть $\gamma \in (0, 1)$ – параметр, характеризующий мощность поражающего воздействия и показывающий, какая часть суммарной пропускной способности ребер может оказаться пораженной. Этому значению γ соответствует множество возможных значений вектора пропускной способности радиальных ребер:

$$Y_\gamma(c) = \left\{ y \left| \sum_{i=1}^m y_i = (1-\gamma) \sum_{i=1}^m c_i \right. \right\}; \quad y \leq c. \quad (2.32)$$

Рассматривается действие на радиальные ребра, кольцевая структура считается неуязвимой.

Для каждого значения γ определяем $\Theta_\gamma^r(c) = \min_{y \in Y_\gamma(c)} \Theta_0(y) = \min_{y \in Y_\gamma(c)} \max_{z \in \mathfrak{Z}(y)} \min_{i \in M} \frac{Z_i}{d_i}$ – гарантированное значение максимума обеспеченности потоковых требований. Если $\Theta_0(y) = 1$, то $\Theta_\gamma^r(c) = 1 - \gamma$.

Это значение показателя живучести реализуется только при равномерном распределении по ребрам СИС мощности воздействия НФ. В остальных случаях этого не происходит. Так, если все воздействие происходит на одно ребро из m ребер, то уже при $\gamma = \frac{1}{m}$, $\Theta_{1/m}^r = 0$, т.е. такой вариант СИС становится недопустимым.

Наличие резервных пропускных способностей позволяет избежать подобных ситуаций, так как в модель всегда можно добавить пропускные способности кольцевой структуры:

$$t = \{t_1, \dots, t_m\}, \quad \text{тогда} \quad \Theta_0(y, t) = \left(\sum_{k \in M} y_k \right) \left| \sum_{i \in M} d_i \right. \quad (2.33)$$

Введение резервной кольцевой структуры в модель не восстанавливает исходного потока требований, так как величина потока $\gamma \sum d_i$ утрачена. Следует также отметить, что реальное значение показателя живучести всегда меньше $1 - \gamma$.

В результате при любых $\gamma < 1$ получают

$$\Theta_\gamma^r(d, t) = 1 - \gamma. \quad (2.34)$$

Величину пропускной способности кольцевой структуры (достаточный ресурс) можно оценить сверху:

$$t^0 \leq \sum_{i \in M} \frac{d_i}{4}. \quad (2.35)$$

Обоснование данного утверждения приводится в прил. 9.

Рассмотрим, каким образом перераспределяются потоки после воздействия НФ мощностью 20 % в СИС радиально-кольцевой топологии.

Исходный граф $G = G_1 + G_2$ – объединение радиальной и кольцевой топологий (рис. 2.16). Пропускная способность кольцевой структуры $\sum_{k=4}^6 c_k = \sum_{i=1}^3 d_i = 30$; $c_k = 10$. Логический граф совпадает с графом на рис. 2.16, б).

Исходная пропускная способность радиальной структуры $C_k^0 = 120$; пропускная способность после воздействия НФ – $C_k = 96$. При равномерном распределении воздействия НФ по ребрам живучесть СИС $\Theta_0(d, c) = 1 - \gamma = 0,8$. Распределение пропускных способностей для этого случая $C_k = (24, 40, 32)$ показано на рис. 2.16, а).

Неравномерное распределение:

1. Все воздействие НФ направлено и распределяется по ребрам 1 – 3 (рис. 2.17, б). Для осуществления равномерного распределения потоков ребро 2 должно пропустить 10 единиц потока, 50 – 40, который, проходя по ребрам кольцевой структуры, разделяется на две слагаемых в 4,4 и 5,6 единиц, эти два потока, складываясь с остальными, проходящими к узлам 1 и 3 (19,6 и 25,4) по радиальным ребрам, создают потоки равномерного распределения удара по СИС.

2. Воздействие НФ происходит по ребру 2 (рис. 2.17, в).

3. Воздействие НФ направлено на ребро 1 (рис. 2.17, з). Наихудшее состояние СИС, так как подверженное воздействию НФ ребро пропускает лишь 6 единиц потока. Ребро 3 пропускает в кольцевую структуру 8 единиц потока, ребра 2 – 10, которые и дополняют поток, приходящий в узел 1 до необходимых 24 единиц, но при этом достигается предел в 10 единиц пропускной способности ребра кольцевой структуры.

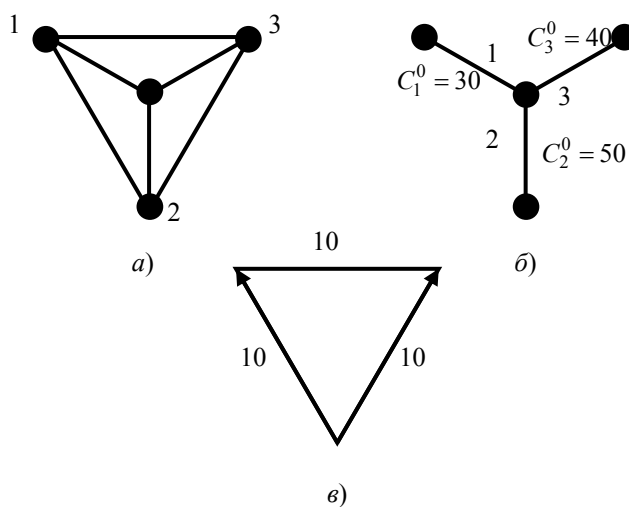


Рис. 2.16. Иллюстрация к примеру

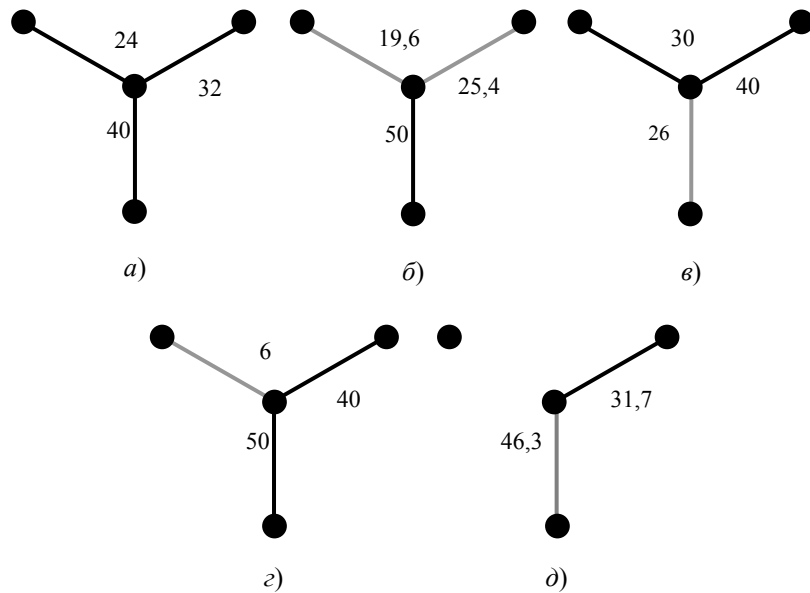


Рис. 2.17. Распределение пропускных способностей после воздействия НФ

3. Воздействие НФ силой $\gamma = 0,3$ осуществлено по ребру 1 (рис. 2.17, д), разрушает его и частично поражает остальные радиальные ребра, общий поток падает до значения 84 единицы, к узлу 1 приходит только 12 единиц потока, $\Theta_0(\gamma) = 0$.

4. При разрушении всех радиальных ребер связность графа не нарушается, но потоки уничтожаются полностью, так как связи с центральным узлом (управляющим потоками в иерархической СИС) уничтожаются полностью, узел оказывается изолированным.

2.6. ПОИСК ГАРАНТИРОВАННОГО УРОВНЯ ДОПУСТИМОСТИ СЕТЕВОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ

$$\Theta_0^{r*}$$

В пределах измененного суммарного вектора пропускной способности для заданного значения γ мощности удара $(1-\gamma)c_0$ необходимо создать несколько графов с различным распределением пропускных способностей ребер, для каждого из них вычислить значение Θ и из этого массива выбрать наименьшее. Это и будет $\Theta_0^{r*}[\gamma]$.

$$\Theta_0^{r*}[\gamma](d) = \min_{c \in C[\gamma]} \max_{(t,z) \in X(c)} \min_{i \in M} \frac{z_i}{d_i}. \quad (2.36)$$

Значение γ определяют по отношению $\frac{C[\gamma]}{C_0}$; суммарные значения пропускной способности часто удается определить легче, чем распределение по ребрам.

Случай $\Theta_0^{r*}[\gamma] > 1$ возможен при $\gamma \ll 1$, при $\Theta_0^{r*}[\gamma] < 1$ найдется такое распределение мощности удара, что не удастся обеспечить необходимый уровень потоковых требований, т.е. СИС становится недопустимой.

Исследуем случай, когда $\Theta_0(c, d) = 1$.

В исходной СИС все требования обеспечены, но хотя бы для одной тяготеющей пары – без запаса по пропускной способности ребра. Тогда $\Theta_0^{r*}[\gamma] \leq 1 - \gamma$, т.е. ни для одной тяготеющей пары p_i невозможно гарантированно рассчитывать на большее чем $1 - \gamma$ обеспечение потоковых требований.

Поиск $\Theta_0^{r*}[\gamma]$ – сложная многокритериальная задача, в конечном итоге сводящаяся к перебору физических ребер графа, подвергшихся удару, с целью определения канала с наименьшей пропускной способностью C .

Приведем пример расчета вектора пропускной способности графа, подвергшегося удару мощностью γ .

$$C[\gamma] = \arg \min_{c \in C[\gamma]} \max_{(t,z) \in X(c)} \min_{i \in M} \frac{z_i}{d_i}; \quad \gamma \in [0, 1]; \quad (2.37)$$

$$C[\gamma] = \{C_k\}, \quad \text{где} \quad \sum_{k=1}^l C_k = (1-\gamma) \sum_{k=1}^l C_k^0, \quad (2.38)$$

где $C^0 = \{C_k^0\}$ – вектор пропускной способности до удара ($\gamma = 0$); $\sum_{k=1}^l C_k^0 = C_0$ – начальное суммарное значение пропускной способности СИС.

Упорядочим ребра физического графа СИС по возрастанию C_k^0 . Построим дерево поиска $c \in C[\gamma]$ для различных γ . В

корневой вершине считаем $c = c^0$. Для каждой вершины 1-го яруса $k_1 \in E$ вычисляем

$$C'_{k_1} = C_{k_1}^0 - \gamma c_0.$$

Если $C'_{k_1} \geq 0$, то полагаем $C_{k_1} = C'_{k_1}$ и закрываем вершину как конечную, в противном случае присваиваем $C_{k_1} = 0$ и переходим на следующий ярус. Обозначим через E_1 множество незакрытых на первом ярусе вершин. Для каждой $K_1 \in E_1$ строим вершины второго яруса $K_2 \in E \setminus E_1$ и вычисляем для них

$$C'_{k_2} = C_{k_2}^0 - (\gamma c_0 - C_{k_1}^0).$$

В случае $C'_{k_2} \geq 0$ полагаем $C_{k_2} = C'_{k_2}$ и закрываем вершину как конечную, в противном случае присваиваем $C_{k_2} = 0$. Обозначим через E_2 множество незакрытых вершин второго яруса, выходящих из K_1 . Для всех $K_1 \in E_1$ и всех $K_2 \in E_2$ определяем одинаковые множества $\{K_1, K_2\}$ и закрываем вершины K_2 , соответствующие большим номерам K_1 в повторяющейся паре.

Переобозначим через $E'_2(K_1)$ множество оставшихся незакрытых вершин второго яруса выходящих из K_1 . Дальнейшее рассмотрение проведем в общем виде для $t = 2, 3, \dots$

Пусть $E'_t(K_1, \dots, K_{t-1})$ – множество незакрытых вершин t -го яруса, выходящих из вершины K_{t-1} ($t-1$ -го яруса, расположенной на ветви (K_1, \dots, K_{t-1}) дерева поиска. Тогда для любой $K_t \in E'_t(K_1, \dots, K_{t-1})$ строим вершины $(t+1)$ -го яруса $K_{t+1} \in E \setminus \{K_1, \dots, K_t\}$ и вычисляем для них

$$C'_{k_{t+1}} = C_{k_{t+1}}^0 - \left(\gamma c_0 - \sum_{s=1}^t C_{k_s}^0 \right).$$

Те вершины K_{t+1} , для которых $C'_{k_{t+1}} \geq 0$, закрываем как конечные и присваиваем им значения $C_{k_{t+1}} = C'_{k_{t+1}}$. Для всех остальных K_{t+1} полагаем $C_{k_{t+1}} = 0$. Обозначим через $E_{t+1}(K_1, \dots, K_t)$ множество незакрытых вершин $(t+1)$ -го яруса, исходящих из вершины K_1 , на ветви (K_1, \dots, K_t) . Теперь закроем те вершины $K_{t+1} \in E_{t+1}(K_1, \dots, K_t)$, для которых $K_{t+1} < K_t$. Последнее по построению эквивалентно тому, что мы выявили для всех $K_{t+1} \in E_{t+1}(K_1, \dots, K_t) \forall K_1 \in E'(K_1, \dots, K_t) \dots \forall K_2 \in E'_2(K_1)$ и $\forall K_1 \in E_1$ повторяющиеся множества вершин (K_1, \dots, K_{t+1}) , т.е. ветви из одинаковых компонент, но в разном порядке, и из всех таких ветвей оставим по одной, в которой ребра упорядочены по возрастанию номеров (а значит, и величин C_k^0). Множество оставшихся незакрытыми вершин $(t+1)$ -го яруса для каждой ветви (K_1, \dots, K_t) (состоящей из вершин, не закрытых на предыдущих ярусах) обозначим через $E'_{t+1}(K_1, \dots, K_t)$.

Перейдем к следующему ярусу.

Из условия на рассматриваемые γ в некоторый момент все вершины яруса окажутся незакрытыми. Тогда остается перебрать множество конечных вершин. Каждая из них характеризуется тем, что на ветви, приводящей к этой вершине, всем ребрам K_s , кроме, быть может, последнего, присвоены нулевые C_k . Присвоим оставшимся ребрам (не принадлежащим ветви) значения их исходной пропускной способности – $C_k = C_k^0$.

Получим вектор C для данной ветви. Процедура построения закончена.

Таких вершин может оказаться слишком много, и вычислить $\Theta_0(c, d)$ для всех построенных C не реально в существующих информационных сетях больших размеров. В вычислениях приходится ограничиваться компактными СИС и строить решение, разыскивая какие-либо аналогии, объединяя, например, несколько ребер в одно и перенося источники потоков на другие вершины.

2.7. СИНТЕЗ СЕТЕВОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ С ГАРАНТИЕЙ ЖИВУЧЕСТИ. ПОСТРОЕНИЕ ПО КРИТЕРИЮ ДОПУСТИМОСТИ

Формально задача проектирования глобальной СИС сводится к отысканию минимума функционала приведенной стоимости

$$C(U, \Omega, Y) \rightarrow \min$$

при наличии ограничений на вероятностные, временные, структурные характеристики СИС

$$V_i(U, \Omega, Y) \leq V_i^0$$

и требования принадлежности множества вариантов архитектуры СИС Q , удовлетворяющих указанным ограничениям, к области технически реализуемых решений:

$$Q(U, \Omega, Y) \in Q^0.$$

U – векторная величина, отражающая параметры сетевой нагрузки, включая интенсивности потоков сообщений между каждой парой узлов коммутации, распределение длин сообщений, приоритетность потоков данных и т.д.

Ω – вектор совокупности параметров технических средств, включая производительность узла коммутации (УК) и каналообразующей аппаратуры, надежность технических средств, достоверность передачи информации, живучесть всей СИС.

Y – вектор, отражающий параметры логической структуры СИС и допустимость ее.

Решение этой общей задачи проектирования достигается созданием комплекса математических и процедурных моделей СИС. Задача будет решена только в том случае, когда отдельные методы и модели будут объединены на основе системного подхода к проектированию.

Невозможность достичь решения в рамках одной конкретной математической задачи приводит к необходимости использования процедуры, основанной на декомпозиции, что позволяет решить задачу большой размерности в виде логической суммы решений задач меньшей размерности.

Декомпозиция на структурном уровне означает, что проектирование СИС сводится к независимому проектированию ряда подсетей при условии совпадения или близости оптимальных решений частной и общей задачи.

Таким образом, в основу проектирования СИС положены следующие общие принципы:

- интерактивности (симбиоз технических условий и интуиции конструктора, так как факты и ограничения проектирования и анализа СИС трудно формализуемы, поэтому математические методы и процедурные модели – в том числе оптимизации – объединяются с опытом и интуицией проектировщика);

- независимости многоуровневого моделирования;

- принцип адаптивности и развития, смысл которого состоит в том, что создание и анализ СИС – непрерывный (часто многолетний) процесс, в результате которого в архитектуру СИС вносятся изменения, вызванные природными, техногенными, политическими или коммерческими событиями и соображениями, т.е. сама СИС непрерывно развивается.

Принятые ранее решения не должны вступать в противоречия с вновь появляющимися требованиями. Непременным условием является требование двухсвязности, когда каждая тяготеющая пара источник – сток продукта связана по крайней мере двумя путями, не имеющими общих узлов и каналов, что значительно повышает живучесть СИС [31].

В том случае, когда полученные на определенном этапе проектирования параметры подсети не удовлетворяют требованиям технического задания, осуществляется повторный расчет на всех предыдущих этапах с изменением соответствующих ограничений.

Наиболее простым решением представляется простое перечисление графов, удовлетворяющих поставленным условиям, но имеющих разные топологии: различное число вершин, ребер, соединяющих вершины, порядок соединения вершин ребрами.

Граф, отвечающий функционалам C , V , Q , и будет искомым, но следует помнить, что каждый из векторов U , Ω , Y является многомерным, тогда становится ясно, что число вариаций слишком велико, чтобы можно было произвести вычисления в реальном времени.

Умелой формализацией условий построения можно существенно уменьшить время перечислений, но сама формализация не менее трудная задача.

Логическим условиям построения СИС более всего отвечает вектор Y . Исходя из этого, можно упростить задачу, учитывая лишь требования к информационным потокам, проходящим по СИС, а также ограничения в пропускной способности, налагаемые на поток. Для получения оптимального значения Θ_Y^r – максимальной величины показателя гарантированной живучести СИС, необходимо построить СИС с максимально возможной пропускной способностью:

$$\sum C_k \rightarrow \max .$$

Рассмотрим процедурную модель синтеза СИС с гарантией живучести.

Задается множество $V = \{v_1, \dots, v_n\}$ вершин – точек входа и выхода потоков, которые выполняют также роль транзитных узлов. На этом множестве задаются графы:

- тяготений, дуги которых p_1, \dots, p_m связывают попарно пункты входа/выхода, указывая информационные направления;

- физический, или граф СИС $G = (V, E)$, ребра которого e_1, \dots, e_k соответствуют реальным каналам связи.

Логические ребра $p_i = (v_{s_i}, v_{t_i})$; $i \in M$; $M = [1, m]$ называют также тяготеющими парами. Каждому физическому ребру e_k формально соответствует пара противоположно направленных дуг g_k и g_{K+k} , при этом для любой вершины $v \in V$ определено множество $T(v)$ и $S(v)$ индексов входящих и исходящих дуг. Обозначим через $f\{f_{ij}\}$ матрицу распределения потоков по дугам СИС. В транзитных узлах выполняется условие сохранения потока:

$$\sum_{j \in S(v)} f_{ij} = \sum_{j \in T(v)} f_{ij} . \quad (2.39)$$

Введем $Z_i(f) = \sum_{j \in S(v)} f_{ij} = \sum_{j \in T(v)} f_{ij}$ – количество потока i -го вида продукта, пропускаемого по СИС в соответствии с известным распределением потоков f . Вектор $Z(f) = \{Z_1(f), \dots, Z_m(f)\}$ называется мультипотокотом и характеризует эффективность функционирования СИС. Таким образом, задачу об эффективности распределения потоков в СИС можно поставить как многокритериальную с вектором критериев $Z(f)$ и ограничением

$$\sum_{i=1}^m (f_{ik} + f_{i(k+K)}) \leq y_k, \quad e_k \in E, \quad (2.40)$$

учитывающим пропускную способность ребер e_k графа G .

Решением задачи будем полагать нахождение максимального значения мультипотока Z . Процедурная модель построения будет выглядеть следующим образом.

Шаг 1. В соответствии с матрицей требований $\|d\|$ образуем мультипоток Z и выберем $\min_{i \in M} Z_i d_i^{-1}$.

Шаг 2. Проведем многократно изменение потока требований (как это происходит в реальной СИС массового обслуживания), образуя тем самым множество мультипотока $L(Z)$, $Z \in L$.

Шаг 3. В каждом из этих мультипотоков выберем $\min_{i \in M} Z_i d_i^{-1}$, а из полученного массива выберем $\max_{i \in M} \min Z_i d_i^{-1}$, тем самым получив предельно-возможные значения мультипотока, допустимого для искомой СИС.

Шаг 4. Сложим все составляющие мультипотока Z_{ij} , проходящие по ребру r_j , тем самым получим общий поток f_j , проходящий через ребро j :

$$\sum_j Z_{ij} = f_j.$$

Шаг 5. Вычислим пропускную способность ребра в соответствии с формулой, предложенной в работе [90]:

$$y_j = f_j + \frac{f_j}{\sum dT_{\text{зад}}} \left[\sum_{k,l} \frac{\sqrt{c_{kl}^1 f_{kl}}}{\sqrt{c_{kl} f_{kl}}} \right]. \quad (2.41)$$

Тем самым будет определена максимальная пропускная способность каждого ребра $\max C_k$.

Шаг 6. Выберем максимальные значения уровней обеспеченности потоковых требований Θ :

$$\Theta_1^r = \min Z_i d_i^{-1}; \quad \Theta_2^r = \max \min Z_i d_i^{-1}.$$

Шаг 7. Применим к СИС воздействие НФ мощностью γ , в результате чего исходная пропускная способность уменьшится в соответствии с соотношением:

$$\sum_k y_k = (1-\gamma) \sum C_k.$$

Шаг 8. y_k – текущее значение пропускной способности ребра C_k . Перебирая ребра графа G , определяем степень поражения каждого ребра в наихудшем варианте, выбираем разрез с наименьшей пропускной способностью.

Шаг 9. Определяем величину гарантированного уровня обеспечения потоковых требований:

$$\Theta_\gamma^r = \min_{y \in Y_\gamma(c)} \max_{z \in L(y)} \min_{i \in M} \frac{Z_i}{d_i}.$$

Это и будет значение показателя гарантированной живучести СИС, представленной графом G .

Шаг 10. Составляем для пораженного графа диаграмму уязвимости.

Шаг 11. Наиболее уязвимые участки СИС укрепляем, вводя новые ребра или объединяя вершины, что приводит к дополнительному увеличению пропускной способности.

Шаг 12. Определяем конечную пропускную способность как

$$C = \arg \max_{c \in C} \min_{y \in Y_\gamma(c)} \max_{z \in L(y)} \min_{i \in M} \frac{Z_i}{d_i}.$$

Шаг 13. Если существует такая возможность, увеличиваем пропускную способность ребер в отношении $C_{\max} = \frac{c}{1-\gamma}$, что создает резерв пропускной способности СИС. Успех подобного построения в значительной степени определяется тем, насколько удачно выбрана топология исходного графа, в противном случае шаги 1 – 13 повторяются несколько раз либо используются иные из предложенных выше процедурных моделей. Графическое отображение процедурной модели приведено на рис. 2.18.

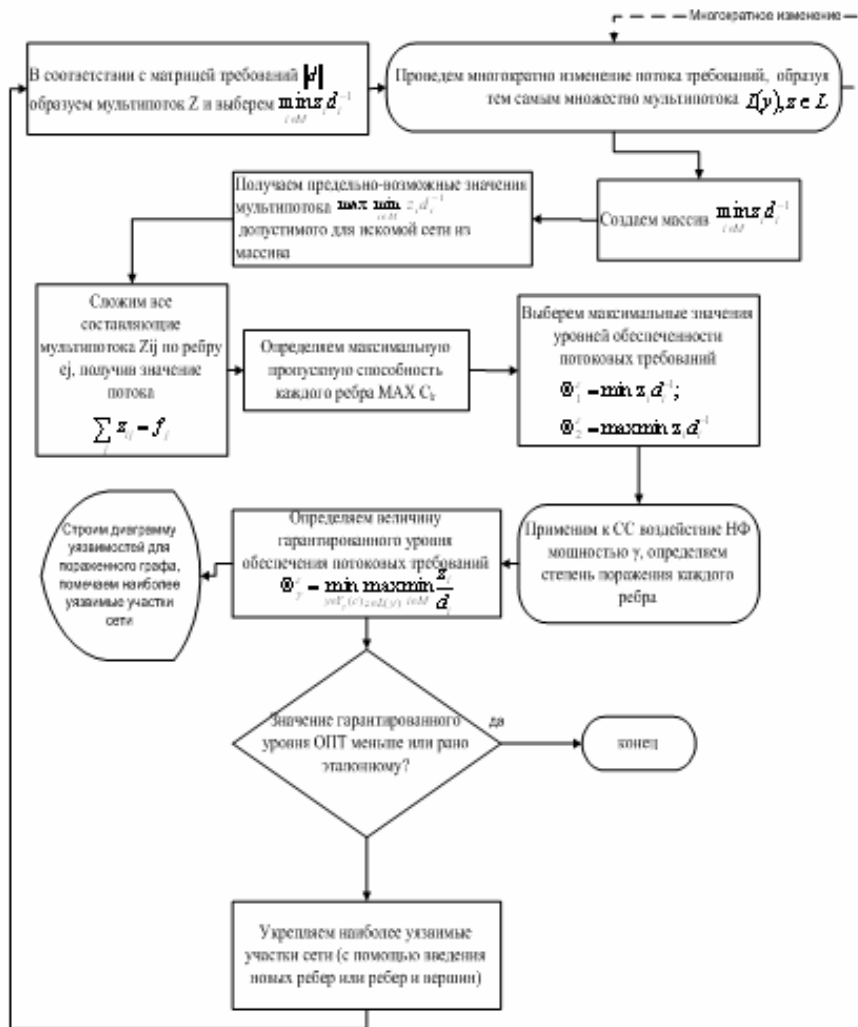


Рис. 2.18. Процедурная модель синтеза СИС с гарантией живучести по критерию допустимости

Пример сравнительного анализа уязвимости СИС (на примере трехпродуктовой потоковой СИС) с различными физическими графами приведен в прил. 10.

3. ПОСТРОЕНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ ОЦЕНКИ ЖИВУЧЕСТИ СЕТЕВЫХ СТРУКТУР

В главах 1 и 2 данной работы было описано аналитическое и процедурное обеспечения информационной системы оценки живучести сетевых информационных систем (ИСЖС). В главе 3 мы рассмотрим реализацию данных обеспечений в виде программно-аппаратного комплекса, а также различные виды обеспечения (таких, как информационное, общее и специальное программное обеспечения), необходимые для реализации данной ИС.

При построении информационной системы оценки живучести сетевых информационных систем (ИСЖС) решались следующие задачи.

- Разработка методики проектирования ИСЖС.
- Разработка и адаптация процедурного и аналитического обеспечений, а также пакета прикладных программ, основанных на данных процедурном и аналитическом обеспечениях.
- Разработка модели, структуры и выбор состава информационного обеспечения, которое должно включать в себя всю необходимую информацию для каждой стадии проектирования.
- Разработка диалогового интерфейса с пользователем в удобной для него форме.

- Разработка структуры и выбор состава технического обеспечения, которые заключаются в выборе технических средств, обеспечивающих автоматизированное получение проектных решений.

- Разработка структуры и выбор состава программного обеспечения, которые представляют собой пакет программ, управляющих работой технических средств и выполняющих проектные процедуры.

Использование при проектировании СИС ИСЖС позволяет:

- 1) намного сократить время расчета;
- 2) найти оптимальные параметры исследуемой (или проектируемой) СИС;
- 3) специалисту проводить анализ информации на любой из стадий проектирования;
- 4) существенно сократить время на оформление и вывод документации, а также повысить ее качество;
- 5) проводить численные эксперименты на ЭВМ с последующей визуализацией их результатов;
- 6) хранить большое количество вариантов параметров в памяти ЭВМ и быстро выбирать из них нужные по определенным критериям на всех стадиях проектирования при расчете оптимальных параметров, при создании документации.

Разработанная ИСЖС является инструментом для проектирования сетевых информационных систем (в том числе транспортных сетей) для телекоммуникационной, энергетической, транспортной и других отраслей промышленности.

3.1. ОБЩЕЕ ОПИСАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ ОЦЕНКИ ЖИВУЧЕСТИ СЕТЕВЫХ ИНФОРМАЦИОННЫХ СИСТЕМ

Качество и современность исследуемого (проектируемого) объекта должны обеспечиваться возможностью постоянного пополнения имеющихся баз данных, используемых при проектировании. Как открытая и развивающаяся система ИСЖС предоставляет пользователю средства и возможность изменения отдельных ее компонентов. Процесс проектирования довольно сложен и многообразен, а потому средства автоматизированного проектирования объединяют в подсистемы, ориентированные на выполнение определенных совокупностей проектных процедур.

Любая информационная система (ИС) состоит из подсистем, которые являются ее неотъемлемой частью. Подсистема ИС представляет собой законченный функциональный блок, имеющий свой неповторимый комплекс средств (программное, лингвистическое, методическое, техническое или другие виды обеспечения) и выполняющий определенные функции, возложенные на него разработчиком системы, такие, как получение законченных проектных решений, подготовка проектной документации, визуальное представление результатов работы подсистемы расчета и оптимизации и т.п. Следует отметить, что результат работы одной подсистемы является входными данными для другой подсистемы ИС.

По назначению подсистемы ИС подразделяют на проектирующие и обслуживающие. К проектирующим относят такие подсистемы, в которых выполняются проектные процедуры и операции. Примером могут служить подсистемы технологического проектирования, конструкторского проектирования и т.д. В отличие от проектирующих подсистем, предназначенных в основном для расчета, обслуживающие подсистемы предназначены для поддержания работоспособности проектирующих и расчетных подсистем. К ним относят подсистемы ввода и коррекции информации, подготовки проектной документации, графическую подсистему, поисковую и другие.

Реально ИСЖС представляет собой комплекс технических средств (КТС), размещенный на нескольких автоматизированных рабочих местах (АРМ), соединенных в локальную вычислительную сеть. На одном АРМ возможно совмещение нескольких подсистем, что уменьшает количество используемой вычислительной техники.

В результате анализа для данной системной области было выбрано устройство КТС, основанное на двух АРМ и сервере баз данных, как один из оптимальных вариантов, отвечающих нуждам небольшого конструкторского (или, к примеру, аналитического) отдела предприятия связи. Использование одной-двух единиц вычислительной техники не рационально в силу того, что данные машины будут слишком загружены как расчетами, так и поддержанием работоспособности самой системы, т.е. ее обслуживанием.

Разделение функциональных обязанностей по АРМ следующее: одна из станций будет являться информационно-обслуживающей, т.е. будет заниматься вопросами ввода исходных данных, вывода проектной документации и информационным обслуживанием процесса расчета и оптимизации проектируемого объекта, а вторая станция будет являться расчетно-проектирующей и будет отвечать за расчет проектируемой СИС и нахождение оптимальных параметров.

3.2. ОПИСАНИЕ ИНФОРМАЦИОННОГО ОБЕСПЕЧЕНИЯ

Как отмечалось выше, ИСЖС при своей работе оперирует большими объемами информации. На каждом этапе функционирования ей необходимы разнородные данные, такие, как логико-лингвистические переменные, нечеткие переменные, а также четкие (реальные) значения. Для того чтобы среди огромного числа данных можно было легко ориентироваться: находить необходимую информацию, изменять, записывать или удалять ее, используются специальные хранилища информации – базы данных (БД).

Информационное обеспечение ИСЖС включает в себя базы данных:

- 1) готовых проектных решений;
- 2) «рабочую» базу данных, создаваемую для текущей задачи;
- 3) форматов текстовой и графической документации;
- 4) базу знаний поисковой подсистемы.

База данных готовых проектных решений (см. рис. 3.1) содержит информацию о проектах, полученных или проанализированных ранее с помощью ИСЖС, и представляет собой совокупность исходных данных и соответствующих проектных решений. База данных содержит следующие таблицы:

- таблицу данных об аппаратном обеспечении узла;
- таблицу данных об аппаратном обеспечении канала связи между вершинами (узлами);
- таблицу топологий графа СИС;

- таблицу сгенерированных правил для использования в логико-лингвистических моделях;
- таблицу ребер;
- таблицу информационных потоков в СИС;
- таблицу параметров вершин логического и физического графов СИС;
- таблицу параметров ребер физического и логического графов СИС;
- таблицу видов и параметров неблагоприятных факторов;
- таблицу проекта СИС;
- таблицу уязвимостей графа и ребер СИС;
- таблицу статистической информации о действиях пользователя.

Структура рабочей базы данных копирует структуру базы данных готовых проектных решений с той лишь разницей, что в таблицах не размещена полная техническая документация, а лишь дана ссылка на документацию в базе готовых проектов.

База данных форматов текстовой и графической документации содержит информацию о форматах документов и представляет собой таблицу, которая включает в себя следующие поля:

- название шаблона;
- описание содержащейся в шаблоне информации;
- имя файла (или сам файл), содержащего шаблон.

База данных подсистемы поиска представляет собой набор срезов (views), а также следующих таблиц:

- таблицу введенных поисковых запросов по остальным БД;
- таблицу индексов срезов, созданных на основании запросов;
- служебные таблицы для индексации БД.

Для того чтобы создавать, наполнять, обновлять и удалять базы данных используются специальные программные средства – системы управления базами данных (СУБД).

В настоящее время существует несколько типов баз данных. Это локальные базы данных, файл-серверные базы данных и базы данных типа клиент-сервер. *Локальная база данных*, все ее части размещаются на одном компьютере пользователя. Если к одной БД обращаются несколько пользователей одновременно, то каждый пользовательский компьютер должен иметь свою копию локальной БД. Эффективная работа современного офиса невозможна без использования сотрудниками общей базы данных. Общность БД достигается ее установкой в локальной сети на специально для этого выделенный компьютер (сервер) при одновременном доступе к серверу компьютеров пользователей (клиентов).

До недавнего времени для организации коллективных БД использовалась, как правило, архитектура файл-сервер (рис. 3.1), в которой организация управления БД целиком ложилась на клиентов, а сама БД представляла собой набор файлов в одном или нескольких каталогах на сетевом сервере.

Данные об аппаратном обеспечении узла

Наименование	Тип	Размер в байтах
Идентификатор записи (PK)	Счетчик	4
Наименование оборудования	Текстовый	255
Процессорный контроллер	Текстовый	50
Количество портов	Числовой	4
Количество потоков	Числовой	4
Процессорный контроллер типа 1	Текстовый	255
Процессорный контроллер типа 2	Текстовый	255
Процессорный контроллер типа 3	Текстовый	255
Процессорный контроллер типа 4	Текстовый	255
Тайминги T1	Массив	1024
Тайминги T2	Массив	1024
Тайминги T3	Массив	1024
Тайминги T4	Массив	1024
Полная техническая документация	Массив	До 2 Гб
Стимость за шт.	Числовой	8

1

T

Данные об аппаратном обеспечении канала

Наименование	Тип	Размер в байтах
Идентификатор записи (PK)	Счетчик	4
Тип кабеля	Текстовый	50
Производитель/Наименование кабеля	Текстовый	255
Минимальный радиус изгиба	Числовой	4
Максимальная нагрузка	Числовой	4
Размеры внешние	Массив	512
Размеры упаковки	Массив	512
Длина волны 1(нм)	Числовой	8
Коэффициент широкополосности 1	Числовой	8
Коэффициент затухания 1	Числовой	8
Длина волны 2(нм)	Числовой	8
Коэффициент широкополосности 2	Числовой	8
Коэффициент затухания 2	Числовой	8
Тайминги T1	Массив	1024
Тайминги T2	Массив	1024
Тайминги T3	Массив	1024
Тайминги T4	Массив	1024
Тайминги T5	Массив	1024
Полная техническая документация	Массив	До 2 Гб
Стоимость за ед. изм.	Числовой	8

2

Таблица физического графа СИС

Наименование	Тип	Размер в байтах
Идентификатор (PK)	Счетчик	4
Наименование графа	Текстовый	255
Вершины графа	Массив	2048
Ребра графа	Массив	2048

3

4

5

Таблица логического графа СИС

Наименование	Тип	Размер в байтах
Идентификатор (PK)	Счетчик	4
Наименование графа	Текстовый	255
Вершины графа	Массив	2048
Ребра графа	Массив	2048

6

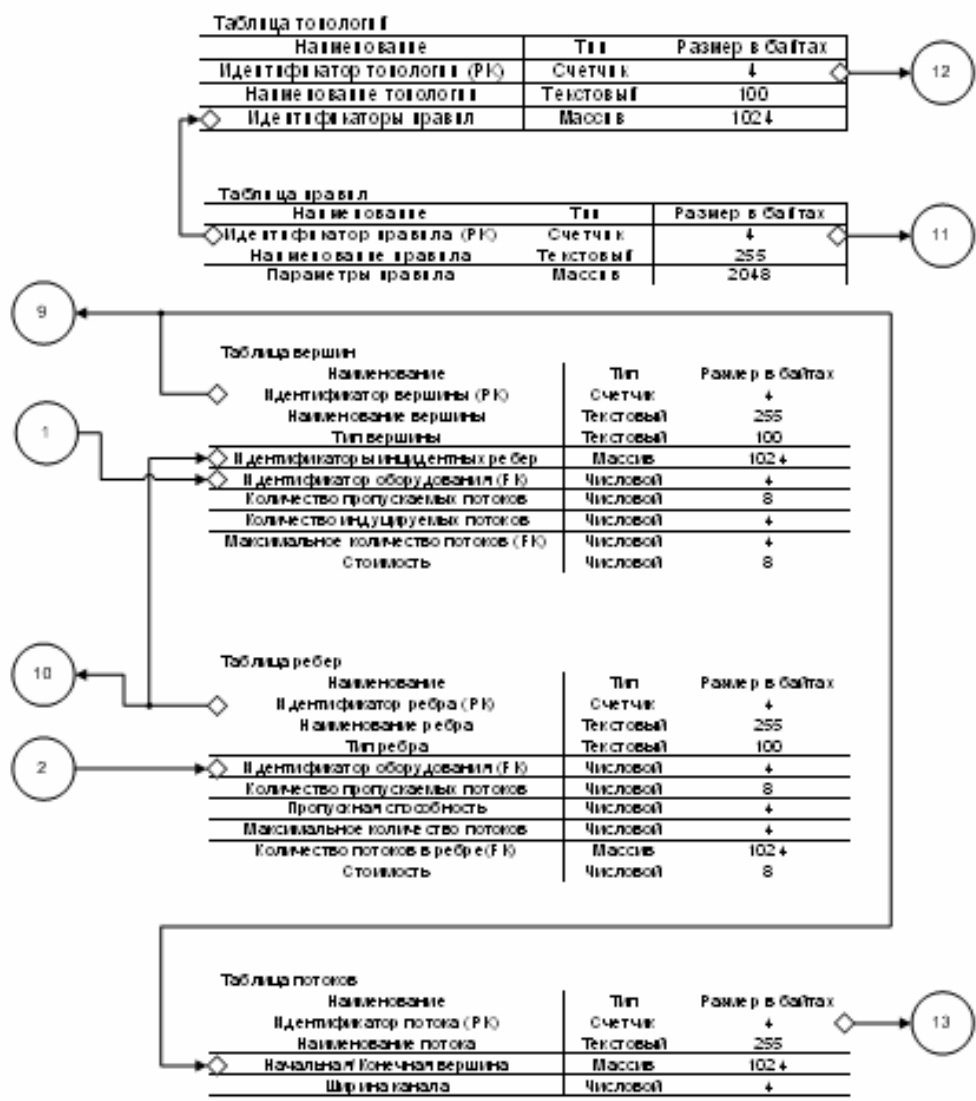
7

8

9

10

Рис. 3.1. Дatalogическая модель базы данных готовых проектных решений информационной системы оценки живучести сетевых информационных систем



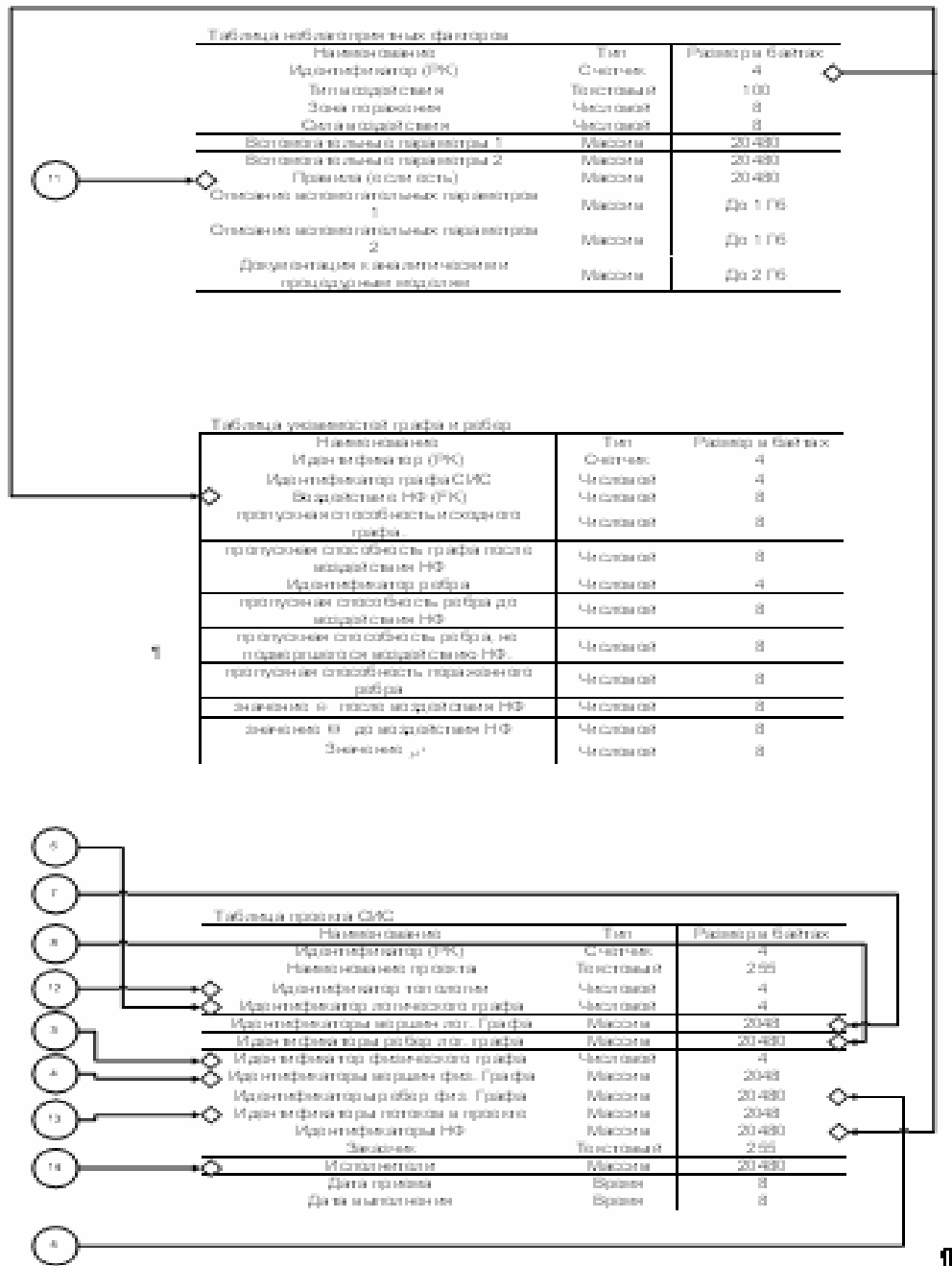


Рис. 3.1. Окончание

В ходе эксплуатации таких систем были выделены общие недостатки архитектуры файл-сервер [91], которые состоят в следующем:

- вся тяжесть вычислительных работ ложится на компьютер клиента; например, если в результате запроса клиент должен получить две записи из таблицы объемом в 100 000 записей, все 100 000 записей будут скопированы с файл-сервера на клиентский компьютер; в результате возрастает нагрузка сети и увеличиваются требования к аппаратным мощностям пользовательского компьютера;
- потребности в постоянном увеличении вычислительных мощностей клиентского компьютера обуславливаются постоянно возрастающим объемом накапливаемой и обрабатываемой информации;
- поскольку БД представляет собой набор файлов на сетевом сервере, доступ к таблицам регулируется только сетевой операционной системой, что делает базы данных незащищенными от случайного или намеренного искажения хранящейся в них информации;
- недостаточно развитый аппарат транзакций локальных СУБД служит потенциальным источником ошибок как при одновременном внесении изменений в одну и ту же запись, так и при реализации отката результатов серии объединенных по смыслу в единое целое операций над БД, когда некоторые из них завершились успешно, а некоторые – нет; это может нарушать ссылочную и смысловую целостность БД.

Перечисленных недостатков во многом лишены БД, построенные по архитектуре клиент-сервер (рис. 3.3), характерной особенностью которой является перенос вычислительной нагрузки на сервер БД и максимальная разгрузка клиента от вы-

числительной работы, связанной с обслуживанием очередей и запросов, а также существенное укрепление безопасности данных как от злонамеренных, так и просто ошибочных изменений.

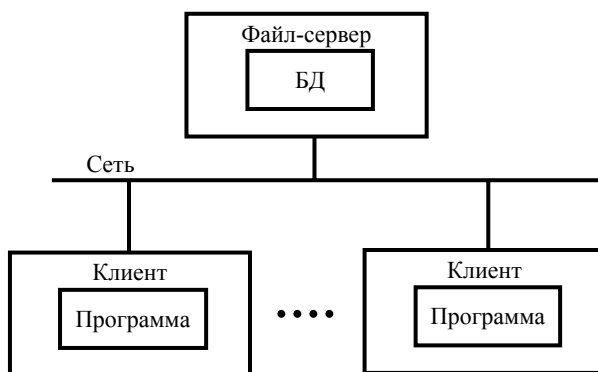


Рис. 3.2. Файл-серверная база данных

Как и в архитектуре файл-сервер, БД в этом случае помещается на сетевом сервере, однако программа клиента лишена возможности прямого доступа к базе данных. Доступ к БД регулируется специальной программой – сервером БД.

Взаимодействие сервера и клиента реализуется с помощью SQL-запросов, которые формирует и отправляет серверу клиент. Сервер, приняв запрос, выполняет его и возвращает результат клиенту. В клиентском приложении в основном осуществляется интерпретация полученных от сервера данных, реализация пользовательского интерфейса. Преимущества архитектуры клиент-сервер:

- большинство вычислительных процессов происходит на сервере, что снижает требования к вычислительным мощностям компьютера клиента; увеличение вычислительной мощности одного сервера эквивалентно одновременному увеличению мощности всех клиентских мест;
- снижается сетевой трафик за счет посылки сервером клиенту только тех данных, которые он запрашивал;
- БД на сервере представляет собой, как правило, единый файл, в котором содержатся таблицы, ограничения целостности и другие компоненты БД. Испортить такую БД значительно труднее: существенно увеличивается защищенность от ввода неправильных значений, поскольку сервер БД проводит автоматическую проверку соответствия вводимых значений наложенным ограничениям, кроме того, сервер отслеживает уровни доступа для каждого пользователя и блокирует попытки выполнения неразрешенных для пользователя действий; все это позволяет говорить о значительно более высоком уровне обеспечения безопасности БД, ссылочной и смысловой целостности информации;

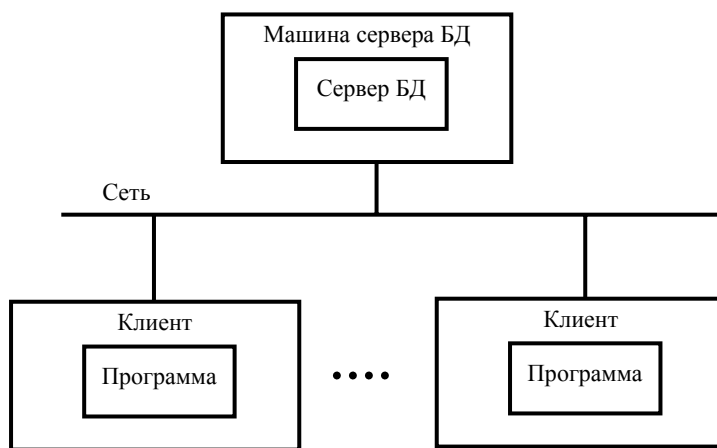


Рис. 3.3. Клиент-серверная база данных

- сервер реализует управление транзакциями и предотвращает попытки одновременного изменения одних и тех же данных; различные уровни изоляции транзакций позволяют определить поведение сервера в этом случае;
- база данных, а также ее части не обязательно должны размещаться на одной физической машине, а на нескольких. Эта технология получила название «распределенные базы данных».

Все это позволяет считать использование архитектуры клиент-сервер в ИСЖС наиболее удобным и целесообразным.

3.3. ОПИСАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Программное обеспечение (ПО) – это совокупность программ, необходимых для нормального функционирования ИС. Это обеспечение принято подразделять на общее и специальное программные обеспечения.

3.3.1. ОБЩЕЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

К общему программному обеспечению относятся операционная система, СУБД, различные табличные и графические процессоры.

Операционная система – это комплекс программ, предназначенных для организации процесса обработки данных, ввода-вывода и управления данными, распределения имеющихся ресурсов. В проекте в качестве операционной системы (ОС) вы-

брано семейство ОС Microsoft Windows 2000. Семейство Microsoft Windows 2000 – это семейство многозадачных многопользовательских операционных систем с вытесняющей многозадачностью, состоящее из разновидностей Professional и Server, предназначенных соответственно для работы в качестве рабочей станции и сервера. Эти ОС обеспечивают возможность работы с новейшими программными средствами и поддерживают различные виды приложений, которые необходимы на этапе разработки ИСЖС и ее использования. Кроме того, данное семейство ОС имеет мощную встроенную поддержку различных сетевых архитектур с гибкой системой администрирования и управления ресурсами пользователей, уровнями доступа, возможностью квотирования дискового пространства и т.д. В качестве альтернативной ОС может использоваться также любая ОС, разработанная на основе ядра ОС Unix (*NIX-образная система, например RedHat Linux/Mandrake Linux, Solaris, HP UNIX и т.д.).

В качестве СУБД и утилит к ней в данном проекте предполагается использовать комплекс ПО, входящего в поставку ORACLE Enterprise Edition [92]. Данная СУБД поддерживает модель «клиент-сервер» и обеспечивает многопользовательский доступ к данным. Можно выделить следующие важные компоненты системы:

- Oracle Database 10g – первая в мире база данных, разработанная специально для работы в сетях распределенных вычислений. Поставляется в четырех различных редакциях: Oracle Database 10g Standard Edition One, Oracle Database 10g Standard Edition (SE), Oracle Database 10g Enterprise Edition (EE), Oracle Database 10g Personal Edition.
- Комплекс Oracle Applications (Oracle E-Business Suite) включает в себя полный набор решений, необходимых для автоматизации управления современным предприятием.
- Oracle Application Server – первый на рынке сервер приложений, позволяющий упростить управление приложениями, выполняемыми в распределенной вычислительной среде.
- Oracle Collaboration Suite – это интегрированный набор приложений для работы с почтой, файлами, голосовой почтой и факсимильными сообщениями, беспроводной работы с почтой.
- Oracle Enterprise Manager – первый в отрасли продукт, предназначенный для управления Oracle Enterprise Grids – сетями распределенных вычислений предприятия.
- Продукты Oracle Data Hub обеспечивают централизованную синхронизацию информации, поступающей из всех систем вашего предприятия, и предоставляют вам точный, согласованный, всесторонний взгляд на данные любых используемых приложений.

Таким образом, комплект приложений данной СУБД обеспечивает гибкое управление и администрирование баз данных. Кроме того, данный тип СУБД используется ведущими компаниями в различных отраслях промышленности для создания ERP, CRM и ACP (автоматизированных расчетных систем, или биллинговых систем), включая таких лидеров мирового рынка как IBM, Hewlett-Packard, Schlumberger Sema, Amdox и TelesensKSCS, что облегчает интеграцию ИСЖС и данных систем. Также следует отметить, что СУБД от ORACLE чрезвычайно легко настраиваются для работы в составе кластерных и распределенных систем, а также сетей хранения данных (Storage Area Networks, SAN), а также сетевых хранилищ данных (Network-attached Storage, NAS) [93].

Для вывода и обработки проектной документации СИС предполагается использовать текстовый процессор Microsoft Word 2000 из состава Microsoft Office 2002. Данный процессор поддерживает возможность написания различных скриптов (набора команд), обеспечивающих работу шаблонов выходной проектной документации. Альтернативой данному текстовому процессору может служить, например, Office Writer любой версии, выпущенной позднее 2002 г., входящий в состав Open Office компании Sun Microsystems и пакет установочных программ *NIX-образных ОС.

Для поиска в СЗ и наполнения базы данных пользователем в ручном режиме используется программа навигации и просмотра информации в сети Интернет – Internet Explorer 5.5 (или более поздней версии) или альтернативная система (Mozilla Firefox, Opera 8.0 и т.д.)

3.3.2. СПЕЦИАЛЬНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Специальное программное обеспечение (СПО) ИСЖС состоит из текстов прикладных программ, которые предназначены для решения поставленной задачи, т.е. реализации аналитического и процедурного обеспечений.

Для разработки СПО необходимо выбрать язык программирования. Выбранный язык должен удовлетворять следующим требованиям: удобство организации ввода-вывода, высокое быстродействие, возможность реализации поставленных задач, наличие графических средств, поддержка выбранной операционной системы.

Исходя из всего вышесказанного, была выбрана среда визуального проектирования Borland Delphi 8 [94] на основе языка программирования высокого уровня Object Pascal. Стоит отметить, что на основе данной среды визуального проектирования также разработана Borland Kylix для *NIX-образных систем, что позволяет выполнять кроссплатформенное портирование ПО, разработанного в среде Delphi для семейства ОС Windows для *NIX-образных систем, в среду Kylix и наоборот.

Для ИСЖС в рамках данной работы было разработано следующее СПО:

- программа-ядро ИСЖС, связывающее между собой все компоненты и модули СПО и обеспечивающее связь этих компонент с СЗ;
- программный модуль model.mmfile, отвечающий за параметры моделирования исследуемой СИС, а также за создание графической модели СИС и подготовку графической информации для модуля пользовательского интерфейса;
- программный модуль ui.mmfile, отвечающий за внешний вид графического интерфейса, а также за функции взаимодействия пользователя и ИС;
- программные модули «блок анализа исходных данных» ie.mmfile и parser.mmfile, от которых зависит выбор процедурной модели расчета живучести СИС, процессы фазификации и дефазификации параметров модели СИС и построения лингвистических термов и правил;
- программный модуль ATC (CSE.mmfile);
- подключаемые программные модули Polynomial.mpfile, ANN.mpfile и Flow.mpfile, ответственные соответственно за процесс расчета живучести СИС с использованием полиномиальной, нейросетевой или потоковой модели;

- редактор базы данных.

Общий вид интерфейса ИСЖС моделирования СИС представлен на рис. 3.4.

Редактор базы данных предназначен для ввода, редактирования и удаления значений в базе данных ИСЖС в процессе проектирования или оценки живучести СИС (рис. 3.5), а также для поиска информации в СЗ и передачи найденных параметров СИС в ИСЖС (рис. 3.6).

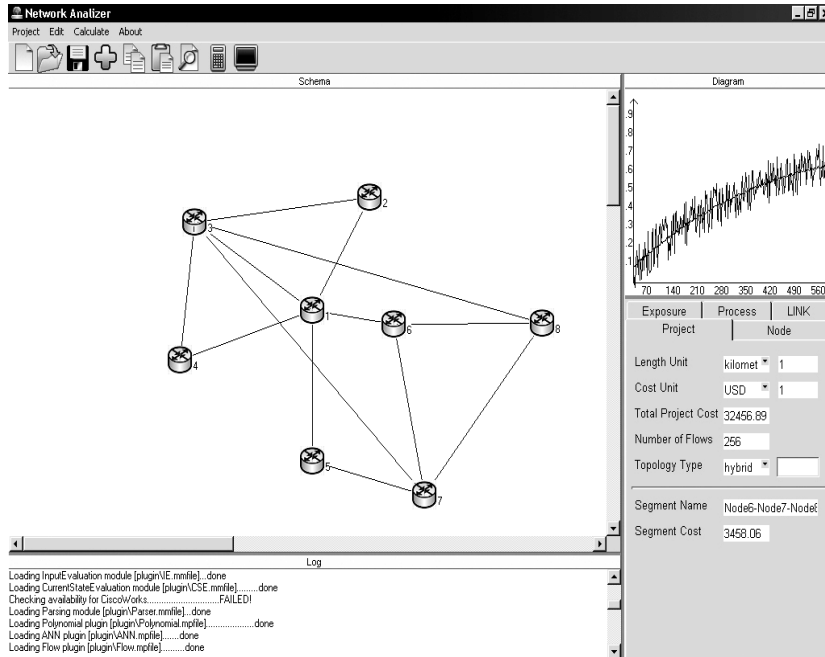


Рис. 3.4. Общий вид интерфейса ИСЖС

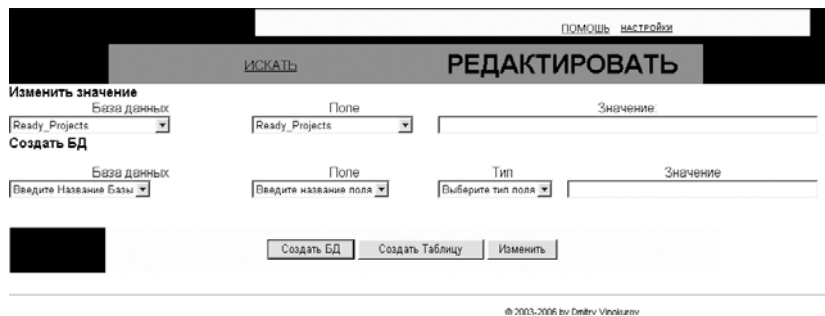


Рис. 3.5. Интерфейс редактора баз данных

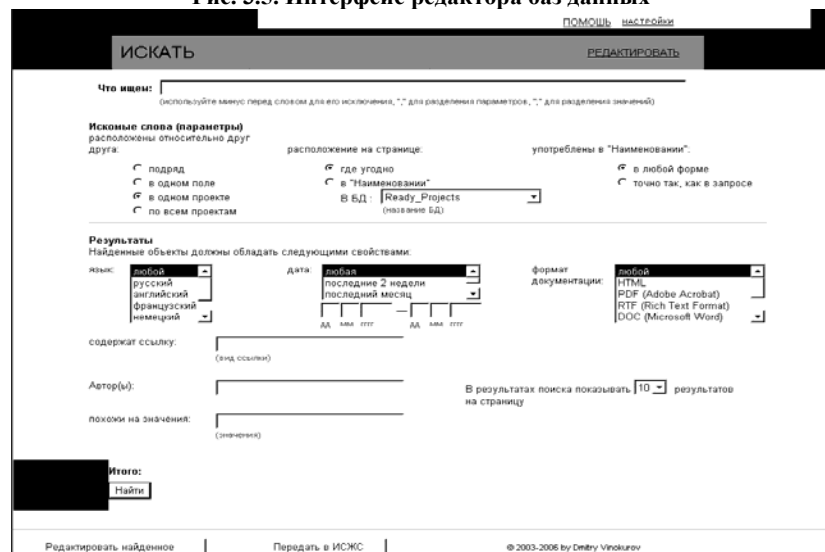


Рис. 3.6. Поискный интерфейс редактора баз данных

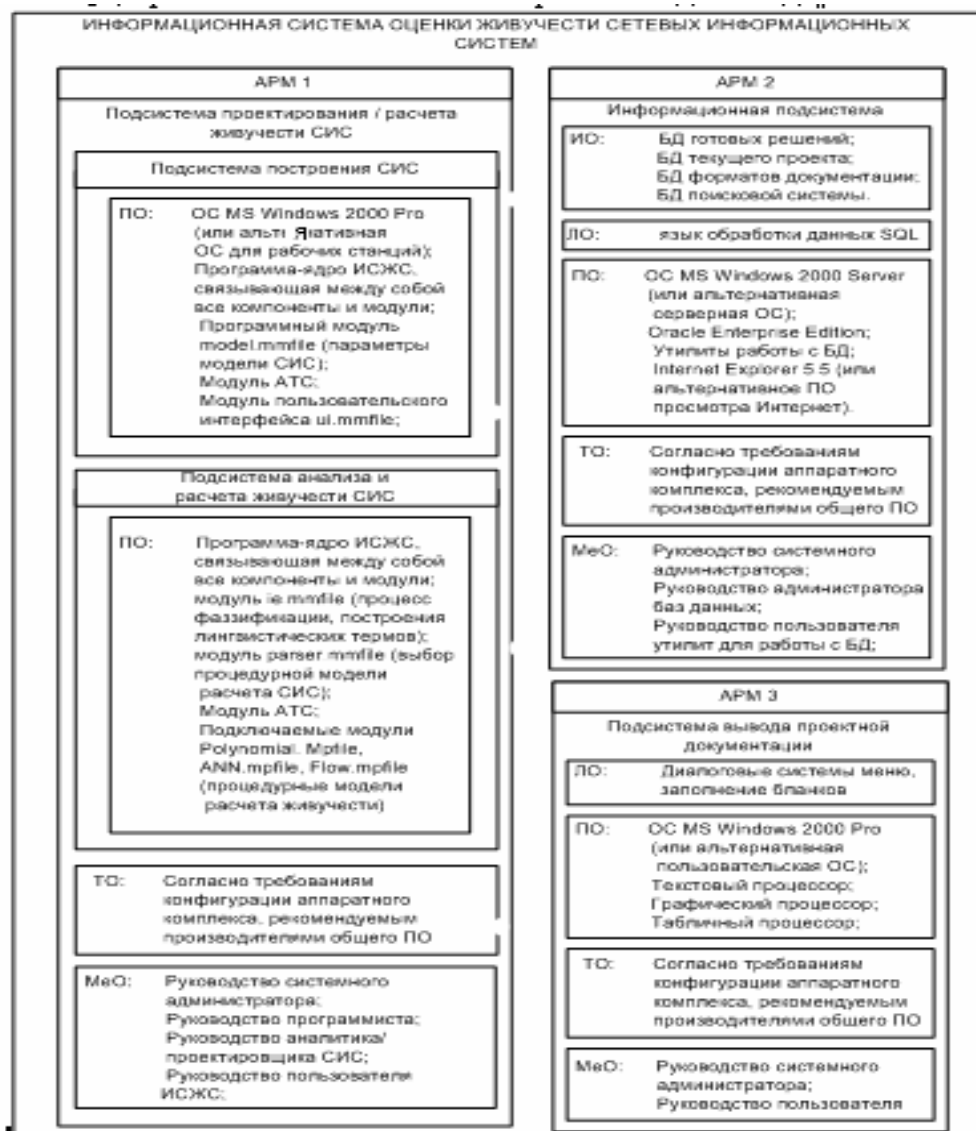
3.3.3. ОПИСАНИЕ ЛИНГВИСТИЧЕСКОГО ОБЕСПЕЧЕНИЯ

Лингвистическое обеспечение – совокупность языков, используемых в ИС для представления информации об анализируемых или проектируемых объектах, процессе и средствах анализа или проектирования, которой обмениваются люди с

ЭВМ и между собой в процессе выполнения операций проектирования, исследования и анализа. Лингвистическое обеспечение представляет собой набор информационно-поисковых (используются в обслуживающих подсистемах), вспомогательных языков и языков проектирования, обеспечивающих взаимодействие пользователя и ИС.

Взаимодействие пользователя с подсистемами осуществляется посредством специализированного языка взаимодействия разрабатываемой ИС с пользователем, который организован в виде диалога с пользователем. Он включает в себя следующие виды диалога:

- «Окно»;
- «Меню»;
- «Заполнение бланков»



В состав лингвистического обеспечения любой системы, независимо от функционального предназначения, должны входить Язык Определения Данных (ЯОД) и Язык Манипулирования Данными (ЯМД). В состав лингвистического обеспечения данной ИС введен язык SQL, представляющий собой интеграцию ЯОД и ЯМД.

Рис. 3.7. Структурная схема обеспечения ИСЖС

SQL (Structured Query Language) имеет мощные средства как по определению данных (семейство операторов create), так и по манипулированию данными (операторы select, drop, update). Кроме того, общесистемное программное обеспечение позволяет создавать и выполнять SQL-сценарии для определения и модификации данных.

Следует отметить, что для работы с SQL пользователь должен обладать довольно высокой квалификацией и иметь представление о внутренней структуре базы данных. Следовательно, в целях снижения профессиональных требований к операторам подсистем необходимо, чтобы SQL-запросы формировались пользователем «неявно» во время работы с при-

кладным ПО. SQL в «чистом» виде должен использоваться администратором баз данных, имеющим высокую квалификацию.

Структурная схема обеспечения ИСЖС показана на рис. 3.7.

ЗАКЛЮЧЕНИЕ

Сетевая информационная система (СИС) представляет собой многоуровневую иерархическую структуру, включающую в себя множество узлов, связанных между собой определенным образом. Такой конструкции присуще свойство уязвимости, которая определяется тем, что за счет многочисленных узлов и связей между ними (учитывая, что нормальное функционирование нескольких узлов иерархической сети возможно только при нормальном функционировании одного основного узла, называемого управляющим) нередко проявляется «каскадный эффект», когда сбой в одном каком-либо месте провоцирует перегрузки и выход из строя многих других элементов СИС.

Проектирование новых ИС и развитие уже существующих связано с проблематикой принятия решений по использованию имеющихся сетевых структур: управлению потоками, распределению ресурсов между узлами. Перечисленные проблемы тесно связаны с задачей определения связности и живучести существующей или проектируемой ИС. Для рассматриваемых систем характерно наличие не только объективной, но и субъективной неопределенности, когда некоторые параметры системы известны отдельным пользователям, но не известны ЛПР (лицу, принимающему решения) или другим пользователям. Ответственность за принятые решения обязывает аккуратно разграничить неопределенные и случайные неконтролируемые факторы: случайность должна быть теоретически обоснована (и подтверждена результатами применения статистических методов), имеющаяся информация о функциях распределения, используемых случайных величинах должна быть указана явно. Взаимная зависимость элементов СИС приводит к немарковости случайных процессов, протекающих в них.

Проблеме оценки живучести СИС посвящен ряд работ (А.Г. Додонов, М.Г. Кузнецова, В.М. Вишневский, Д.Л. Белоцерковский,

Ю.Е. Мельников, Ж.С. Сарыпбеков, Ю.Е. Малашенко, С.Л. Colbourn, К. Sekine, Н. Imai, S. Tani, А.Е. Smith и др.), разработаны аналитические модели, адекватно описывающие процесс расчета живучести СИС, тем не менее, в настоящее время актуальной является задача разработки аналитического описания, обобщающего полученные ранее результаты и позволяющего не только осуществить разработку новых методов проектирования и анализа СИС, но и ставить и решать задачи расчета живучести СИС большой размерности и сложной структуры.

Успешное решение проблемы обеспечения живучести сложных СИС во многом зависит от типов используемых методов оценки живучести, поскольку они адекватны моделируемым объектам и обеспечивают получение точных оценок о реальном состоянии системы.

В результате разработано процедурное обеспечение ИСЖС, в том числе структура блока анализа исходных данных, включающая в себя процедурную модель генерации правил, состоящую из компоненты фазификации (перевода реальных данных о системе в четком виде к нечеткому виду), компоненты анализа и выделения правил, а также компоненты очистки правил, и логико-лингвистическая модель блока анализа исходных данных. По этой модели была разработана процедурная модель.

Рассмотрены процедурные модели оценки живучести СИС, основанные на аналитическом обеспечении, предложенном в параграфах 1.2 – 1.5, определены размерности и топологии графов, оптимально подходящих для каждой процедурной модели.

Кроме того, была предложена структура и описана функциональность модуля АТС, предложены процедурная модель анализа СИС на основе модели МП-сети, комбинированная задача выбора пропускных способностей и распределения потоков, задачи нахождения кратчайших путей и расчета объемов суммарной передачи информации.

При построении информационной системы оценки живучести сетевых информационных систем (ИСЖС) были решены следующие задачи:

- Разработана методика проектирования ИСЖС.
- Разработано процедурное и аналитическое обеспечение (см. главу 1, 3), а также пакет прикладных программ, основанных на данном процедурном и аналитическом обеспечении.
- Разработаны модели, структуры выбора состава информационного обеспечения, которое должно включать в себя всю необходимую информацию для каждой стадии проектирования.
- Разработан диалоговый интерфейс с пользователем в удобной для последнего форме.
- Разработана структура выбора состава технического обеспечения, которое заключается в выборе технических средств, обеспечивающих автоматизированное получение проектных решений.
- Разработана структура и выбран состав программного обеспечения, которое представляет собой пакет программ, управляющих работой технических средств и выполняющих проектные процедуры.

Использование при проектировании СИС ИСЖС позволяет:

- 1) намного сократить время расчета;
- 2) найти оптимальные параметры исследуемой (или проектируемой) СИС;
- 3) специалисту проводить анализ информации на любой из стадий проектирования;
- 4) существенно сократить время на оформление и вывод документации, а также повысить ее качество;
- 5) проводить численные эксперименты на ЭВМ с последующей визуализацией их результатов;
- 6) хранить большое количество вариантов параметров в памяти ЭВМ и быстро выбирать из них нужный по определенным критериям на всех стадиях проектирования при расчете оптимальных параметров, при создании документации.

Разработанная ИСЖС является инструментом для проектирования сетевых информационных систем (в том числе транспортных сетей) для телекоммуникационной, энергетической, транспортной и других отраслей промышленности.

СПИСОК ЛИТЕРАТУРЫ

1. Никитин, А.И. Отказоустойчивость распределенных систем / А.И. Никитин // Управляющие системы и машины. – 1987. – № 5. – С. 25 – 30.
2. Colbourn C.J. The Combinatorics of Network Reliability / C.J. Colbourn. – Oxford University Press, 1987.
3. Deeter, D.L. Heuristic optimization of network design considering all-terminal reliability / D.L. Deeter, A. E. Smith // Proceedings of the Reliability and Maintainability Symposium. – 1997. – P. 194 – 199.
4. Малашенко, Ю.Е. Модели неопределенности в многопользовательских сетях / Ю.Е. Малашенко, Н.М. Новикова. – М. : Едиториал УРСС, 1999. – 160 с.
5. Itai, A. The complexity of finding maximum disjoint paths with length constraints / A. Itai, Y. Perl, Y. Shiloach // Networks. – 1982. – V. 12, № 3. – P. 277 – 286.
6. Chen, C.-K.E. The generalised diameter of a graph / C.-K.E. Chen, R.S. Garfinkel // Networks. – 1982. – V. 12, № 3. – P. 335 – 340.
7. Bjorner, A. Introduction to Greedoids / A. Bjorner, G.M. Ziegler // Matroid Applications : Encyclopedia of Mathematics and Its Applications. – Cambridge University Press, 1992. – V. 26. – P. 284 – 357.
8. Bryant, R.E. Graph-Based Algorithms for Boolean Function Manipulation / R.E. Bryant // IEEE Transactions on Computers. – 1986. – V. C-35. – P. 677 – 691.
9. Colbourn, C.J. The Combinatorics of Network Reliability / C.J. Colbourn. – Oxford University Press, 1987.
10. Network Reliability: Experiments with a Symbolic Algebra Environment / D.D. Harms, M. Kraetzl, C.J. Colbourn, J.S. Devitt. – CRC Press, Inc., 1995.
11. Lipton, R.J. A Separator Theorem for Planar Graphs / R.J. Lipton, R.E. Tarjan // SIAM J. on Appl. Math. – 1979. – V. 36, № 2. – P. 177 – 189.
12. Oxley, J.G. Matroid Theory / J.G. Oxley. – Oxford Science Publications, 1992.
13. Provan, J.S. The Complexity of Reliability Computations in Planar and Acyclic Graphs / J.S. Provan // SIAM Journal on Computing. – 1986. – V. 15, № 3. – P. 694 – 702.
14. Sekine, K. Computing the Tutte Polynomial of a Graph of Moderate Size / K. Sekine, H. Imai, S. Tani // Proceedings of the 6th International Symposium on Algorithms and Computation (ISAAC'95), Lecture Notes in Computer Science. – 1995. – V. 1004. – P. 224 – 233.
15. Tani, S. An Extended Framework of Ordered Binary Decision Diagrams for Combinatorial Graph Problems / S. Tani. – Master's thesis, University of Tokyo, 1995.
16. Tutte, W.T. A Contribution to the Theory of Chromatic Polynomials / W.T. Tutte // Canadian Journal of Mathematics. – 1954. – V. 6. – P. 80 – 91.
17. Welsh, D.J.A. Complexity: Knots, Colourings and Counting / D.J.A. Welsh // London Mathematical Society Lecture Note Series. – Cambridge University Press, 1993. – Vol. 186.
18. Мельников, Ю.Е. Модель комплексной оценки и обеспечения живучести распределенных информационно-вычислительных систем / Ю.Е. Мельников, Ж.С. Сарыпбеков // Материалы II Всесоюзной науч.-техн. конф. – М., 1988.
19. Graphs and interconnection networks: diameter and vulnerability / J.C. Bermond, J. Bond, M. Paoli, C. Peyrat // London Mathematical Society Lecture. Note Ser. – 1983. – № 82. – P. 1 – 30.
20. Задачи оптимизации иерархических структур / В.Т. Дементьев, А.И. Ерзин, Р.М. Ларин, Ю.В. Шамордин. – Новосибирск : Изд-во Новосибирского ун-та, 1996. – 200 с.
21. Кривулец, В.Г. Что такое теория связности и живучести транспортных сетей? / В.Г. Кривулец, В.П. Полесский // Информационные процессы. – 2001. – Т. 1, № 2. – С. 199 – 203.
22. Вишнеvский, В.М. Принципы построения единой системы продажи и бронирования билетов на транспорте / В.М. Вишнеvский, А.В. Ризов, Е.В. Федотов // ВКСС connect. – 2001. – № 1. – С. 80 – 85.
23. Ogryczack, W. EDINET – A Network Editor for Transshipment Problems with Facility Location, in Computer Science and Operations Research / W. Ogryczack, K. Studzinski, K. Zorychta // New Developments in their Interfaces. – U.K. : Pergamon Press, 1992.
24. Pirkul, H. The Maximal Covering Location Problem with Capacities on Total Workload / H. Pirkul, D. Schilling // Management Science. – 1991. – 37. – P. 233 – 248.
25. Pirkul, H. Hierarchical Concentrator Location Problem / H. Pirkul, S. Narasimhan // Computer Communications. – 1992. – 15, 3. – P. 185 – 191.
26. Pirkul, H. Locating Concentrators in Centralized Computer Networks / H. Pirkul, V. Nagarajan // Annals of Operations Research. – 1992. – 36. – P. 247 – 262.
27. Pracht, W.E. Model Visualization: Graphical Support for DSS Problem Structuring and Knowledge Organization / W.E. Pracht // Decision Support Systems. – 1990. – 6. – P. 13 – 27.
28. Ramamurthy, K. User Characteristics – DSS Effectiveness Linkage: An Empirical Assessment / K. Ramamurthy, W.R. King, G. Premkumar // International Journal of Man-Machine Studies. – 1992. – 36. – P. 469 – 505.
29. Steiger, D. Functional Description of a Graph-Based Interface for Network Modeling (GIN), in Computer Science and Operations Research / D. Steiger, R. Sharda, B. Leclaire // New Developments in their Interfaces. – U.K. : Pergamon Press, 1992.
30. Tan, J.K.H. The Effectiveness of Graphical Presentation for Information Extraction: A Cumulative Experimental Approach / J.K.H. Tan, I. Benbasat // Decision Sciences. – 1993. – 24 : 1. – P. 167 – 191.
31. Фрэнк, Г. Сети, связь и потоки : пер. с англ. / Г. Фрэнк, И. Фриш ; под ред. Д.А. Поспелова. – М. : Связь, 1978. – 448 с.
32. Додонов, А.Г. Введение в теорию живучести вычислительных систем / А.Г. Додонов, М.Г. Кузнецова, Е.С. Горбачик. – Киев : Наукова думка, 1990. – 184 с.

33. Вишнеvский, В.М. Теоретические основы проектирования компьютерных сетей / В.М. Вишнеvский. – М. : Техно-сфера, 2003. – 512 с.
34. Иьуду, К.А. Теория надежности и живучести вычислительных машин / К.А. Иьуду. – М. : МАИ, 1978. – 53 с.
35. Иьуду, К.А. Математические модели отказоустойчивых вычислительных систем / К.А. Иьуду, С.А. Кривошеков. – М. : МАИ, 1989. – 144 с.
36. Takashi, I. Timedomain analysis of characteristics of the human operator in a simple manual control system / I. Takashi, K. Hidefuma // *Fac. Eng. Univers.* – 1973. – V. 32, № 2. – P. 419 – 430.
37. Крапивин, В.Ф. О теории живучести сложных систем / В.Ф. Крапивин. – М. : Наука, 1978. – 248 с.
38. Малашенко, Ю.Е. Оперативная корректировка запасов и потоков энергоресурсов в энергетических комплексах при внешних возмущениях : справочник по общим моделям анализа и синтеза надежности систем энергетики / Ю.Е. Малашенко ; под ред. Ю.Н. Руденко. – М. : Энергоатомиздат, 1994. – С. 435 – 447.
39. Алгоритмы и программы решения задач на графах и сетях / М.И. Нечепуренко, В.К. Попков, С.М. Майнагашев, С.Б. Кауль и др. – Новосибирск : Наука : Сиб. отд-ние, 1990.
40. Bellmore, M. Optimal defense of multicommodity networks / M. Bellmore, H.D. Ratliff // *Management Science.* – 1971. – V. 18, № 4. – P. 174 – 185.
41. Frederickson, G.N. Approximation algorithms for several graph augmentation problems / G.N. Frederickson, J.J. Joseph // *SIAM Journal on Computing.* – 1981. – V. 10, № 2. – P. 270 – 283.
42. Habib, M. A construction method for minimally k-edge connected graphs / M. Habib, B. Peroche // *Combinatorics.* – 1980. – V. 79. – Part 2. – P. 199 – 204.
43. Gusfield, D. Optimal mixed graph augmentation / D. Gusfield // *SIAM Journal on Computing.* – 1987. – V. 16, № 4. – P. 599 – 613.
44. Boesch, F.T. Lower bounds on the vulnerability of a graph / F.T. Boesch // *Networks.* – 1972. – V. 2. – P. 329 – 340.
45. Deniel, J.W. On perturbations in systems of linear inequalities / J.W. Deniel // *SIAM J. of Numerical Analysis.* – 1973. – № 10. – P. 299 – 307.
46. Plesnik, J. The complexity of designing a network with minimum diameter / J. Plesnik // *Networks.* – 1981. – V. II. – P. 77 – 85.
47. Shoone, A.A. Diameter increase caused by edge deletion / A.A. Shoone, H.L. Bodlaender, J. van Leewen // *J. of Graph Theory.* – 1987. – V. II, № 3. – P. 409 – 427.
48. Harary, F. Conditional connectivity / F. Harary // *Networks.* – 1983. – V. 13, № 3. – P. 347 – 357.
49. Cutting and partitioning a graph after a fixed pattern / M. Yannakakis, P.C. Kannelakis, S.S. Cosmadakis, C.N. Papadimitriou // *Lecture Notes on Comput. Science.* – 1983. – V. 154. – P. 712 – 722.
50. Nash-Williams, C. Edge-disjoint spanning trees of finite graphs / C. Nash-Williams // *Journal of London Mathematical Society.* – 1961. – V. 36. – P. 445 – 450.
51. Tutte, W.T. On the problem of decomposing a graph into n connected factors / W.T. Tutte // *J. of London Mathematical Society.* – 1961. – V. 36. – P. 221 – 230.
52. Gusfield, D. Fast algorithms for bipartite network flow / D. Gusfield, Ch. Martel, D. Fernandes-Baco // *SIAM Journal on Computing.* – 1987. – V. 16, № 2. – P. 237 – 251.
53. Chung, F.R.K. Diameters of communication networks / F.R.K. Chung // *Proceedings of symposia in applied mathematics.* Providence, R.I. – 1986. – V. 34. – P. 1 – 18.
54. Multicommodity flow approach to assignment of circuits in case of failure in a communication network / Y. Ishiyama, Y. Ishizaki, S. Sasabe, N. Yoshida // *Survey of mathematical programming. Proceedings of the 9th international mathematical programming symposium.* – 1976, 1979. – V. 3. – P. 195 – 209.
55. Математические постановки задач восстановления и обеспечения живучести для многопродуктовых сетей / М.Р. Давидсон, Ю.Е. Малашенко, Н.М. Новикова и др. – М. : ВЦ РАН, 1993.
56. Мину, М. Математическое программирование / М. Мину. – М. : Наука, 1990.
57. Annan, J.D. The Complexity of Counting Problems / J.D. Annan. – University of Oxford, 1994.
58. Geman, S. Neural networks and the bias/variance dilemma / S. Geman, E. Bienenstock, R. Doursat // *Neural Computation.* – 1992. – V. 4. – P. 1 – 58.
59. Aggarwal, K.K. Topological layout of links for optimizing the overall reliability in a computer communication system / K.K. Aggarwal, Y.C. Chopra, J.S. Bajwa // *Microelectronics and Reliability.* – 1982. – V. 22. – P. 347 – 351.
60. Cavers, J.K. Cutset manipulations for communication network reliability estimation / J.K. Cavers // *IEEE Transactions on Communications.* – 1975. – V. 23. – P. 569 – 575.
61. Network topology for maximizing the terminal reliability in a computer communication network / Y.C. Chopra, B.S. Sohi, R.K. Tiwari, K.K. Aggarwal // *Microelectronics & Reliability.* – 1984. – V. 24. – P. 911 – 913.
62. Coit, D.W. Solving the redundancy allocation problem using a combined neural network / genetic algorithm approach / D.W. Coit, A.E. Smith // *Computers and Operations Research.* – 1996. – V. 23. – P. 515 – 526.
63. Jan, R.-H. Design of reliable networks / R.-H. Jan. // *Computers and Operations Research.* – 1993. – V. 20. – P. 25 – 34.
64. Dengiz, B. Local search genetic algorithm for optimal design of reliable networks / B. Dengiz, F. Altiparmak, A.E. Smith // *IEEE Transactions on Evolutionary Computation.* – 1997. – V. 1. – P. 179 – 188.
65. Deeter, D.L. Economic design of reliable networks / D.L. Deeter, A.E. Smith // *HE Transactions.* – 1998. – V. 30.
66. Konak, A. An improved general upperbound for all-terminal network reliability / A. Konak, A.E. Smith // *HE Transactions.* – 1998. – V. 30.
67. Dengiz, B. Efficient optimization of all-terminal reliable networks using an evolutionary approach / B. Dengiz, F. Altiparmak, A.E. Smith // *IEEE Transactions on Reliability.* – 1997. – V. 46. – P. 18 – 26.

68. Funahashi, K. On the approximate realization of continuous mappings by neural networks / K. Funahashi // *Neural Networks*. – 1989. – V. 2. – P. 183 – 192.
69. Konak, A. A general upperbound for all-terminal network reliability and its uses / A. Konak, A.E. Smith // *Proceedings of the Industrial Engineering Research Conference*. – Banff, Canada, 1998. – May. – CD-Rom format.
70. Cheng, B. Neural networks: A review from a statistical perspective / B. Cheng, D.M. Titterton // *Statistical Science*. – 1994. – V. 9. – P. 2 – 54.
71. Jan, R.-H. Topological optimization of a communication network subject to a reliability constraint / R.-H. Jan, F.-J. Hwang, S.-T. Chen // *IEEE Transactions on Reliability*. – 1993. – V. 42. – P. 63 – 70.
72. Garey, M.R. *Computers and Intractability: A Guide to the Theory of NP-Completeness* / M.R. Garey, D.S. Johnson. – San Francisco, 1979.
73. Ball, M. Backtracking algorithms for network reliability analysis / M. Ball, R.M. Van Slyke // *Annals of Discrete Mathematics*. – 1977. – V. 1. – P. 49 – 64.
74. Aggarwal, K.K. Reliability evaluation in computer-communication networks / K.K. Aggarwal, S. Rai // *IEEE Transactions on Reliability*. – 1981. – V. R-30. – P. 32 – 35.
75. Hornik, K. Multilayer feedforward networks are universal approximators / K. Hornik, M. Stinchcombe, H. White // *Neural Networks*. – 1989. – V. 2. – P. 359 – 366.
76. Iri, M. On the extension of the maximum-flow minimum-cut theorem to multicommodity flows / M. Iri // *J. Oper. Res. Soc. Japan*. – 1971. – V. 13. – P. 129 – 135.
77. Заде, Л.А. Понятие лингвистической переменной и его применение к принятию приближенных решений / Л.А. Заде. – М. : Мир, 1976. – 165 с.
78. Малашенко, Ю.Е. Поточковые задачи анализа уязвимости многопродуктовых сетей / Ю.Е. Малашенко, Н.М. Новикова. – М. : ВЦ АН СССР, 1989.
79. Карманов, В.Г. Моделирование в исследовании операций / В.Г. Карманов, В.В. Федоров. – М. : Твема, 1996.
80. Пападимитриу, Х. Комбинаторная оптимизация / Х. Пападимитриу, К. Стайглиц. – М. : Мир, 1985.
81. Onago, K. A multicommodity flow theorem / K. Onago // *Electronics Commun.* – 1970. – V. 53, № 7. – P. 16 – 22.
82. Biswas, J. Two flow routing algorithms for the maximum concurrent flow problem / J. Biswas, D.W. Matula // *Fall Joint Comput. Conf., Dallas, Tex., 1986, Nov. 2 – 6. – Proc. Washington, D.C., 1986.* – P. 629 – 636.
83. Faster approximation algorithms for the unit capacity concurrent flow problem with applications to routing and finding sparse cuts / P. Klein, S. Plotkin, C. Stein, E. Tardos // *SIAM Journal on Computing*. – 1994. – V. 23, № 3. – P. 466 – 487.
84. Малашенко, Ю.Е. Математические модели анализа потоковых сетевых систем / Ю.Е. Малашенко. – М. : ВЦ АН СССР, 1993.
85. Методы робастного, нейро-нечеткого и адаптивного управления : учебник / под ред. Н.Д. Егупова. – М. : Изд-во МГТУ им. Н.Э. Баумана, 2001. – 744 с.
86. Integration of quantitative and qualitative knowledge for online decision support / A.E. Elalfy, A.F. ElGamal, M.H. Haggag, M.E. Elallmi // *IJCIS*. – 2003. – V. 3, № 1.
87. Petridis, V. Clustering and Classification in Structured Data Domains Using Fuzzy Lattice Neurocomputing (FLN) / V. Petridis, V.G. Kaburlasos // *IEEE Transactions on Knowledge and Data Engineering*. – 2001. – V. 13, № 2.
88. Wang, J.H. Histogram-Based Fuzzy Filter For Image Restoration / J.H. Wang, Liu Wen-Jeng, Lin Lian-Da // *IEEE Trans On Systems, Man, and Cybernetics*. – 2002. – V. 32, № 2. – P. 230 – 238.
89. Bellmore, M. Multicommodity disconnecting sets / M. Bellmore, H.J. Greenberg, J.J. Jarvis // *Management Science*. – 1970. – V. 16, № 6. – P. 427 – 433.
90. Нефедов, В.Н. Аппроксимация множества оптимальных альтернативных решений / В.Н. Нефедов // *Новые задачи оптимизации авиационных систем*. – М. : Изд-во МАИ, 1989.
91. Парус-Предприятие. Архитектура «файл-сервер» // <http://parus.optilink.ru/soft/default.asp?id=2008> – Компьютерное агентство БАС, 1999.
92. Oracle Database 10g Product Family (PDF) // http://www.oracle.com/technology/products/database/oracle10g/pdf/twp_general_10gdb_product_family.pdf. – 2006.
93. SGI Products: SGI InfiniteStorage 4000 // <http://www.sgi.com/products/storage/tech/4000.html>. – Copyright © SGI, 1993 – 2006.
94. Borland: The Leading Provider of Technology for Software Applications // <http://www.borland.com>. – Copyright © Borland Software Corporation, 1994 – 2006.
95. Gupta, R. *Problems in Communication Network Design and Location: Planning; New Solution Procedures* / R. Gupta. – The Ohio State University, 1996.
96. Богуславский, Л.Б. Моделирование основных параметров функционирования компьютерной сети Государственной Думы / Л.Б. Богуславский, В.И. Дрожжинов, А.И. Ляхов // *Вычислительные сети-95 : тез. докл. Междунар. конф. и шк.* – М. : Изд-во Науч. совета РАН по комплексной проблеме «Кибернетика», 1995. – С. 120 – 124.
97. Воробейчикова, О.А. Векторный минимум со связанными ограничениями / О.А. Воробейчикова, Н.М. Новикова // *Вестник МГУ. Вычислительная математика и кибернетика*. – 1996. – № 4. – С. 45 – 48.
98. Давидсон, М.Р. Об условной устойчивости крайних точек многогранного множества / М.Р. Давидсон // *Журнал вычислительной математики и математической физики*. – 1994. – Т. 34, № 1. – С. 29 – 43.

**УСЛОВИЕ СУЩЕСТВОВАНИЯ ФИЗИЧЕСКОГО ГРАФА
СЕТЕВОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ**

$n = 6, d = (2, 4, 3, 4, 3, 2); d < n - 1 = 5$ (условие выполняется).

- 2) Соединяем вершину 1 с вершинами 2 и 3. Степень вершины 1 исчерпана.
- 3) Соединяем вершину 2 с вершинами 1, 3, 4, 5. Степень вершины 2 исчерпана.
- 4) Соединяем вершину 3 с вершинами 1, 2, 4. Степень вершины 3 исчерпана.
- 5) Соединяем вершину 4 с вершинами 2, 3, 5, 6. Степень вершины 4 исчерпана. Соединить вершину 4 с вершиной 1 невозможно по п. 1.
- 6) Соединяем вершину 5 с вершинами 2, 4, 6. Степени вершин 5 и 6 исчерпаны, граф построен.

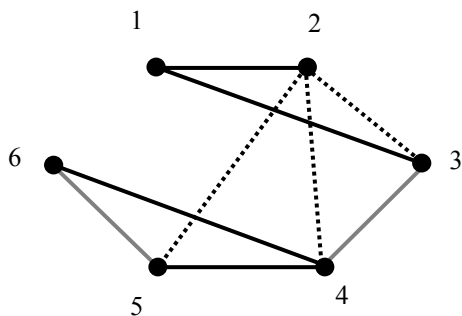


Рис. III.1. Физический граф СИС

**ФУНКЦИОНИРОВАНИЕ СЕТЕВОЙ ИНФОРМАЦИОННОЙ
СИСТЕМЫ. ДЕТЕРМИНИРОВАННАЯ МОДЕЛЬ**

СИС условно делят на три категории: глобальные СИС (WAN), городские СИС (MAN) и локальные СИС (LAN) [33]. Логическая, программная и физическая структуры WAN и MAN практически совпадают, отличие состоит лишь в охвате территории и скоростях передачи информации. В механизме передачи информации глобальных и локальных СИС имеются существенные различия (WAN используют соединение до начала передачи пакетов данных, в локальных сетях пакеты посылаются без подтверждения готовности получателя к обмену).

В базовой СИС используют различные методы коммутации: коммутацию каналов, сообщений и пакетов (фреймов).

При коммутации каналов для связи двух абонентов СИС организуется составной канал, состоящий из отдельных транзитных участков, закрепленных за парой абонентов на все время сеанса связи. После окончания сеанса составной канал распадается на исходные отдельные участки, которые могут быть использованы для организации других составных каналов. Недостатком данного типа коммуникации является невозможность использования временно закрепленных за конкретным соединением участков для связи между другой парой абонентов. Коммутация каналов особенно неэффективна для диалогового режима передачи коротких порций информации и значительным временным интервалом между ними.

При коммутации сообщений текст передается от одного узла коммутации (УК) к другому, пока не достигнет адресата. Сообщение таким образом занимает в каждый момент времени только небольшой участок канала связи (КС) на пути между источником и адресатом или ожидает в очереди освобождения канала в каком-либо УК.

Коммутация пакетов совпадает с коммутацией сообщений, кроме того, сообщение разбивается на части, называемые пакетами. Пакеты нумеруются, снабжаются адресом и независимо передаются по СИС, эффективно используя пропускную способность КС. Конечным адресатом предусматривается сборка пакетов в соответствии с нумерацией, так как в силу независимой передачи пакеты могут поступать не в той последовательности, в которой были отправлены.

Управление сетью осуществляется с помощью процедур, называемых протоколами. Широкое распространение получила семиуровневая модель, определяющая направление разработки протоколов [95].

Совокупность действий на первых четырех уровнях можно иллюстрировать примером функционирования двухполюсной многоканальной линии связи. Считаем заданными:

- Входящий поток запросов (количество запросов за единицу времени λ).
- Время обслуживания запросов (интенсивность потока обслуживания μ).
- Количество и типы обслуживающих устройств (количество каналов n).
- Дисциплину обслуживания (порядок обработки запросов в системе, например FIFO: first in – first out, первым пришел – первым передан).

Предполагается, что число запросов, поступивших за некоторый интервал времени, зависит только от длины интервала, но не зависит от расположения этого интервала на временной оси; будущее случайного процесса определяется только его состоянием в настоящий момент и не зависит от предшествующей истории. Одновременное поступление двух и более запросов исключается. Общий поток информации рассматривается как суперпозиция большого числа равномерно малых независимых потоков. Распределение длин интервалов между моментами поступления запросов является показательным с функцией распределения $A(t) = 1 - e^{-\lambda t}$. Такие процессы называются марковскими случайными процессами [66 – 68].

Таким процессом определяется работа двухполюсной многоканальной СИС, в которой запрос, поступивший в момент, когда все каналы заняты, блокируется и «выбывает из игры» (рис. П2.1). Подобные системы символически изображаются кодом $M|M|n$, граф состояний имеет вид марковской системы. S_0 – все каналы свободны, S_1 – занят 1 канал, S_2 – занято 2 канала, S_n – заняты все каналы. Один и тот же поток заявок с интенсивностью λ переводит систему из состояния S_{i-1} в состояние S_i в направлении слева направо. Поток сообщений справа налево переводит систему из состояния S_i в состояние S_{i-1} потоком обслуживания интенсивностью μ . Для перехода из S_0 в S_1 безразлично, какой канал освободится, следовательно, этот переход имеет интенсивность

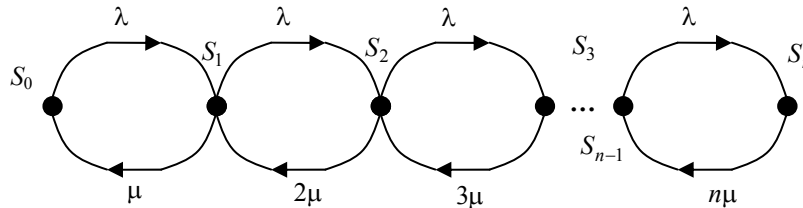


Рис. П2.1. Граф состояний

$i\mu$, для состояния S_i поток равен $i\mu$. После завершения пребывания процесса в состоянии i он переходит в состояние с вероятностью q_i , $0 < q_i < 1$, и в состояние $i + 1$ с вероятностью $p_i = 1 - q_i$.

Распределение вероятностей $p_i(t)$ удовлетворяет системе линейных дифференциальных уравнений:

$$\begin{cases} P'_0(t) = -\lambda P_0(t) + \mu P_1(t); \\ P'_i(t) = -\lambda P_{i-1}(t) - (\lambda + i\mu)P_i(t) + (i+1)\mu P_{i+1}(t). \end{cases} \quad (\text{П2.1})$$

Преобразованием Лапласа она переводится в систему линейных алгебраических уравнений для распределения π_i :

$$\begin{aligned} \lambda\pi_0 &= \mu\pi_1; \\ \pi_i &= \lambda\pi_{i-1} - (\lambda + i\mu)\pi_i + (i+1)\mu\pi_{i+1}, \end{aligned} \quad (\text{П2.2})$$

что влечет выполнение соответственно:

$$\begin{aligned} \lambda\pi_{i-1} &= i\mu\pi_i; \\ \pi_1 &= \rho\pi_0; \\ \pi_i &= i\rho\pi_{i-1}; \end{aligned} \quad (\text{П2.3})$$

$$\rho_0 = 1; \quad \rho = \frac{\lambda}{\mu} = \text{const}.$$

Условие стационарности

$$\sum_{i=0}^n \pi_i = 1.$$

Формула для вероятности π_0 следует из условия нормировки (уравнение Колмогорова) [96].

Исходя из равенств (П2.1) – (П2.3), стационарные вероятности π_i , $0 \leq i \leq n$, определяются следующим образом:

$$\pi_i = \frac{\rho^i}{i!} \sum_{j=0}^n \frac{\rho^j}{j!}, \quad \text{откуда} \quad \pi_0 = \left[\sum_{i=0}^n \frac{\rho^i}{i!} \right]^{-1}. \quad (\text{П2.4})$$

Вероятность того, что заявка будет принята

$$Q = 1 - \pi_n = 1 - \frac{\rho^n}{n!} \pi_0. \quad (\text{П2.5})$$

Умножая Q на λ , получаем интенсивность потока обслуженных системой заявок, разделив это произведение на μ , находим среднее число занятых каналов в системе:

$$\langle N \rangle = \rho \left(1 - \frac{\rho^n}{n!} \pi_0 \right). \quad (\text{П2.6})$$

Поскольку $n!$ возрастает быстрее, чем любая степень ρ , то при увеличении количества каналов n число принятых запросов приближается к 100 %, но одновременно приближается к постоянному пределу, равному ρ и количеству обслуживаемых каналов (рис. П2.2). Дальнейшее увеличение n ведет к возрастанию числа простаивающих каналов (хотя при этом возрастает качество обслуживания, так как $Q \rightarrow 1$).

Сообразясь со стоимостью канала и ценой каждого потерянного запроса, подбирают наиболее выгодные значения величин n и ρ (изначально полагаются)

$$n > \frac{\lambda}{\mu}, \quad \langle N \rangle = \frac{\lambda}{\mu}. \quad (\text{П2.7})$$

Следует также учитывать, что обработка принятого запроса требует конечного промежутка времени. При плотном потоке запросов может быть достигнут предел, при котором длительность интервалов между следующими друг за другом запросами окажется равной времени обработки сигнала, тогда все каналы окажутся одновременно заблокированными. Дальнейшее же увеличение числа каналов может быть лимитировано по техническим причинам, кроме того, это уменьшит надежность системы в целом.

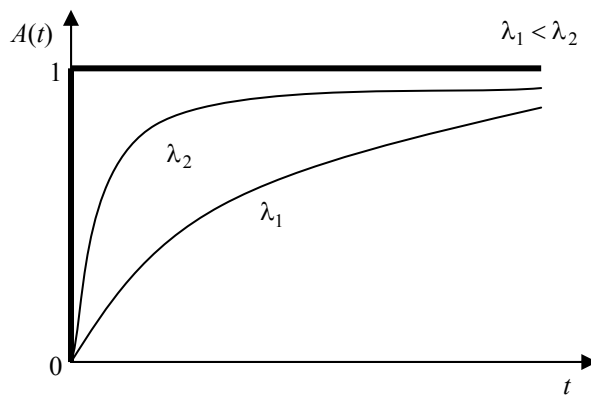


Рис. П2.2. Достижение предела до блокировки канала

Тогда поток заявок надо разбить на меньшие фрагменты, а управляющий УК заменить несколькими блоками с соответственно уменьшенным числом каналов. Не следует добиваться очень малых потоков, так как это сильно снижает коммерческую ценность системы. Для оптимизации работы необходимо учитывать динамику поступления запросов в течение суток, вводя или исключая дополнительно дежурные каналы, при этом исходный поток запросов разбивается на фрагменты с разным значением λ / μ .

Данный подход является морально устаревшим, так как не обеспечивает необходимой точности вычислений вследствие больших допущений. Для анализа сетевых структур на живучесть в настоящее время используют графовые модели потоковых СИС.

**ОБЩАЯ ЖИВУЧЕСТЬ СЕТЕВОЙ ИНФОРМАЦИОННОЙ
СИСТЕМЫ В ПОЛИНОМИАЛЬНОЙ ФОРМЕ.
ОБОСНОВАНИЕ УТВЕРЖДЕНИЯ 1**

Утверждение 1. Для ребра e

$$R(G) = \begin{cases} (1-p(e))R(G/e); \\ R(G \setminus e); \\ p(e)R(G \setminus e) + (1-p(e))R(G/e). \end{cases}$$

Обоснование. В случае, если ребро e – копетля, оно не должно удаляться, если граф G – связный, а также если ребро e – петля, то ее можно игнорировать. Когда ребро e не является ни петлей, ни копетлей, в случае удаления e , если $G \setminus e$ – связный, тогда граф G – тоже связный. Более того, ребро e не удаляется, если G/e – связный, тогда G – связный.

**РАСЧЕТ С ПОМОЩЬЮ ФОРМУЛЫ КОНТРАКЦИИ И
УДАЛЕНИЯ РЕБРА. ОБОСНОВАНИЕ УТВЕРЖДЕНИЯ 2**

Утверждение 2. Пусть H_1 и H_2 – два минора G с одинаковой совокупностью ребер \bar{E}_i . H_1 и H_2 изоморфны и имеют одинаковое расположение ребер, только если их разделения на уровне i являются идентичными.

Обоснование. По определению, каждая вершина фронта исключения V_i на уровне i инцидентна некоторым ребрам в \bar{E}_i . Сейчас H_1 и H_2 изоморфны с одинаковым расположением ребер. Вершины определяются однозначно своими инцидентными ребрами. Тогда разделения исключения H_1 и H_2 на уровне i идентичны [15].

**ОБЩАЯ ЖИВУЧЕСТЬ ДВУХПОЛОСНОЙ СЕТЕВОЙ
ИНФОРМАЦИОННОЙ СИСТЕМЫ.
ОБОСНОВАНИЕ УТВЕРЖДЕНИЯ 4**

Утверждение 4. Для ребра e , кроме вновь добавленных двух ребер:

$$R(G'|s',t') = \begin{cases} R(G' \setminus e|s',t'); \\ (1-p(e))R(G'/e|s',t'); \\ p(e)R(G' \setminus e|s',t') + (1-p(e))R(G'/e|s',t'). \end{cases}$$

Обоснование. Когда e – это петля или неактивное e нерелевантно в расчете вероятности $R(G'|s',t')$, мы можем привести ее к виду $R(G' \setminus e|s',t')$. Когда e – это активная копетля e , удаление e делает s' и t' несвязными. Оставшийся случай: e – активное и не является ни петлей, ни копетлей, совпадает с общим случаем живучести СИС.

**ФУНКЦИЯ ЖИВУЧЕСТИ ДЛЯ ПОЛНОГО ГРАФА.
ОБОСНОВАНИЕ УТВЕРЖДЕНИЯ 7**

Утверждение 7

$$R(U_{m,r}; p) = \sum_{i=1}^m \binom{m}{i} (1-p^r)^i p^{r(m-i)} R(U_{m-i,i}; p),$$

где $R(U_{0,r}; p) = 1$.

Обоснование. $U_{m,r}$ имеет $m+1$ вершины, обозначаемые v_1, \dots, v_m в клике и v_{m+1} для оставшейся вершины. И v_{m+1} , и любая другая вершина являются инцидентными вершинами r кратных ребер. Мы называем их пучком кратных ребер. Теперь применим формулу удаления/контракции для этих $m \times r$ ребер. Мы можем выбрать i пучков из m пучков и удалить все ребра оставшихся $(m-i)$ пучков. Вероятность, с которой эти $r(m-i)$ ребра удаляются, равна $p^{r(m-i)}$. Отметим, что если $U_{m,r}$ связный, мы должны выбрать, по крайней мере, один пучок, т.е. $1 < i < m$. Для каждого i пучков мы имеем, по крайней мере, одно ребро, которое не удаляется. Эта вероятность равна $(1-p^r)^i$. Мы контрактируем одно из таких ребер для каждого i пучков. Тогда оставшиеся ребра становятся петлями и могут быть удалены по формуле. Теперь мы получаем $U_{m-i,i}$ и продолжаем выполнять данную процедуру для этого графа.

Снова $R(K_n; p) = R(U_{n-1,1}; p)$, и мы получаем это расчетом всех $R(U_{j,k}; p)$, таких, что $j+k < n-1$. Наивысшая степень – $1/2n(n-1)$, а коэффициент – $(-1)^{n-1}(n-1)!$. $R(K_n; p)$ может быть разделен на $(1-p)^{n-1}$, и из-за этой факторизации каждый элемент имеет положительный знак в остающемся коэффициенте. Например, $R(K_4; p) = (1-p)^3(6p^3 + 6p^2 + 3p + 1)$.

**ПРИМЕР НАХОЖДЕНИЯ КРАТЧАЙШИХ ПУТЕЙ И
РАСЧЕТА ОБЪЕМОВ СУММАРНОЙ ПЕРЕДАЧИ
ИНФОРМАЦИИ (ТРАФИКОВ)**

Рассмотрим задачу нахождения кратчайших путей и потоков по кратчайшим путям для сегмента информационной потоковой СИС, приведенной на рис. П7.1. На ребрах (i, j) СИС указаны условные стоимости передачи C_{ij} , причем $C_{ij} = C_{ji}$. Матрица требований в передаче информации H приводится ниже.

$$H = \begin{vmatrix} x & 13 & 11 & 7 & 9 \\ 6 & x & 6 & 4 & 8 \\ 5 & 4 & x & 6 & 3 \\ 6 & 14 & 12 & x & 15 \\ 1 & 6 & 7 & 3 & x \end{vmatrix};$$

$$B = \begin{vmatrix} x & 1 & 0 & 1 & 0 \\ 1 & x & 1 & 0 & 0 \\ 0 & 1 & x & 1 & 1 \\ 1 & 0 & 1 & x & 1 \\ 0 & 0 & 1 & 1 & x \end{vmatrix}.$$

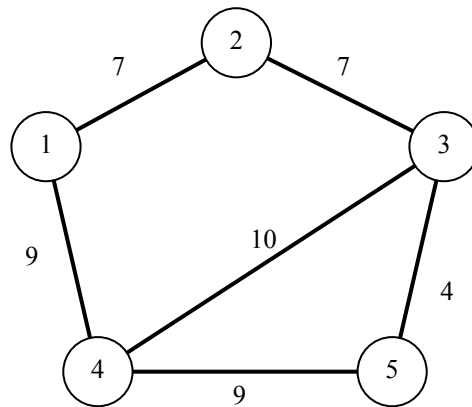


Рис. П7.1. Схема сегмента потоковой СИС

1 итерация.

Вычисляем матрицу длин кратчайших путей $A(1)$:

$$a_{ij}(1) = \begin{cases} C_{ij}, & \text{если } b_{ij} = 1; \\ \infty & \text{в противном случае.} \end{cases}$$

$$A(1) = \begin{vmatrix} x & 7 & \infty & 9 & \infty \\ 7 & x & 7 & \infty & \infty \\ \infty & 7 & x & 10 & 14 \\ 9 & \infty & 10 & x & 9 \\ \infty & \infty & 4 & 9 & x \end{vmatrix}.$$

Находим матрицу кратчайших маршрутов

$$a_{ij}(1) = \begin{cases} i, & \text{если } b_{ij} = 1; \\ \emptyset & \text{в противном случае.} \end{cases}$$

$$M(1) = \begin{vmatrix} x & 1 & 0 & 1 & 0 \\ 2 & x & 2 & 0 & 0 \\ 0 & 3 & x & 3 & 3 \\ 4 & 0 & 4 & x & 4 \\ 0 & 0 & 5 & 5 & x \end{vmatrix}.$$

2 итерация.

Вычисляем матрицу длин кратчайших путей $A(2)$:

$$a_{ij}(2) = \min_s \{a_{ij}(1); a_{is}(1) + a_{sj}(1)\};$$

$$A(2) = \begin{vmatrix} x & 7 & 14 & 9 & 18 \\ 7 & x & 7 & 16 & 11 \\ 14 & 7 & x & 10 & 4 \\ 9 & 16 & 10 & x & 9 \\ 18 & 11 & 4 & 9 & x \end{vmatrix}.$$

Находим соответствующую матрицу кратчайших маршрутов:

$$m_{ij}(2) = \begin{cases} m_{ij}(1), & \text{если } a_{ij}(2) = a_{ij}(1); \\ m_{s'j'}, & \text{если } a_{ij}(2) = a_{is'}(1) + a_{s'j}(1). \end{cases}$$

$$M(2) = \begin{vmatrix} x & 1 & 2 & 1 & 4 \\ 2 & x & 2 & 3 & 3 \\ 2 & 3 & x & 3 & 3 \\ 4 & 3 & 4 & x & 4 \\ 4 & 3 & 5 & 5 & x \end{vmatrix}.$$

3 итерация.

Вычисляем матрицу длин кратчайших путей $A(3)$ и соответствующую маршрутную матрицу $M(3)$ и убеждаемся, что $A(3) = A(2)$ и $M(3) = M(2)$. Следовательно, матрица $M(2) = M$ является искомой маршрутной матрицей (кратчайших путей), а $A(2) = A$ соответствующих длин кратчайших путей.

Покажем, как пользуясь матрицей M , найти кратчайший путь π_{15}^{\min} .

Рассматриваем элемент $m_{15} = 4$. Далее ищем элемент m_{14} , так как $m_{14} = 1$, то конец, $\pi_{15}^{\min} = \{1-4-5\}$, при этом $L(\pi_{15}^{\min}) = 18$.

Рассчитываем потоки по кратчайшим путям: $F^0 = [f_{rs}^0]$. Для этого полагаем $K = 0$, $F(0) = [f_{rs}(0)] = 0$.

1 итерация.

Выбираем элемент $h_{12} = 13$.

Находим $\pi_{12}^{\min} = \{1-2\}$ и вычисляем поток $F(1)$:

$$f_{rs}(1) = \begin{cases} f_{rs}(0), & \text{если } (r,s) \neq (1,2); \\ f_{rs}(0) + h_{12} = 13, & \text{если } (r,s) = (1,2), \end{cases}$$

$k = k + 1$ и переход к следующей итерации.

Остальные итерации выполняются аналогично. Спустя $k = n(n-1) = 20$ итераций, получим следующее распределение потоков по кратчайшим путям:

$$f_{12} = 53, \quad f_{23} = 40, \quad f_{14} = 41, \quad f_{34} = 18, \quad f_{35} = 24, \quad f_{45} = 28.$$

**ВЫДЕЛЕНИЕ МУЛЬТИПОТОКА z ПО
ЗАДАННОМУ ВЕКТОРУ ПОТОКА f**

Рассмотрим задачу, обратную показанной в прил. 6. Покажем, как выделить мультипоток z по заданному вектору потока f . Задан взвешенный параметризованный граф G , $n = 5$, $m = 6$ (рис. П8.1).

Все вершины графа являются не только источниками и стоками, но и транзитными узлами, коммутирующими компоненты мультипотока. Рассмотрим компоненты $z_{ij} = (i - j)$, $i \neq j$, проходящие через каждое из 6 ребер $[k, l]$:

- $[1 - 2] \rightarrow (1 - 2), (1 - 3), (4 - 2) \rightarrow [(13 + 6), (11 + 5), (14 + 4)];$
- $[2 - 3] \rightarrow (2 - 3), (1 - 3), (5 - 2) \rightarrow [(6 + 4), (11 + 5), (8 + 6)];$
- $[3 - 5] \rightarrow (3 - 5), (2 - 5) \rightarrow [(3 + 7), (8 + 6)];$
- $[3 - 4] \rightarrow (3 - 4) \rightarrow [(6 + 12)];$
- $[5 - 4] \rightarrow (5 - 4), (1 - 5) \rightarrow [(3 + 15), (9 + 1)];$
- $[4 - 1] \rightarrow (4 - 1), (1 - 5), (2 - 4) \rightarrow [(6 + 7), (9 + 1), (4 + 14)].$

Для ребра $[2 - 3]$ мы не используем маршрут $(4 - 3 - 2)$, так как он не является кратчайшим (его «длина» равна $10 + 7 = 17$), тогда как маршрут $(1 - 2 - 3)$ имеет длину, равную 14.

Также исключается маршрут $(4 - 5 - 3)$ для ребра $[3 - 5]$.

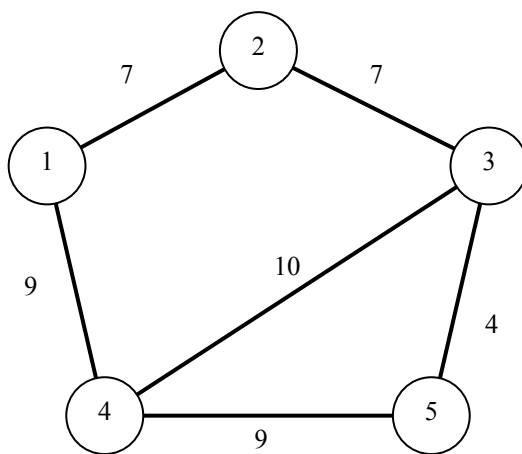


Рис. П8.1. Граф G , $n = 5$, $m = 6$

Пропустим по этим маршрутам потоки требований в соответствии с матрицей требований.

Маршрут $(1 - 2)$ содержит потоки требований 13; 6.

Маршрут $(1 - 3)$ содержит потоки требований 11; 5.

Маршрут $(4 - 2)$ содержит потоки требований 14; 4.

Все эти потоки проходят через ребро $[1 - 2]$, сумма их равна: $f_{12} = 19 + 16 + 18 = 53$. Аналогичные операции проделаем для остальных ребер. Полученный нами вектор потока $\{f_{ij}\}$ действительно определяет мультипоток $\{z_{ij}\}$, при этом значение $\Theta_0(c, d) \equiv 1$.

ПРИМЕР ПОСТРОЕНИЯ ГРАФИКА УЯЗВИМОСТИ

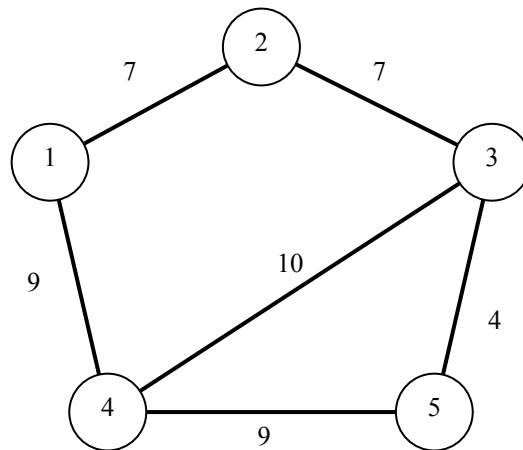


Рис. П9.1. Граф $G, n = 5, m = 6$

Пусть НФ воздействуют на два ребра (1 – 2) и (1 – 4) силой $\gamma = 0,2$, которые в данной СИС полагаем особо уязвимыми. Введем следующие обозначения и соответствующие им величины:

$c^0 = 204$ – пропускная способность исходного графа;

$c_\gamma^0 = 163$ – пропускная способность графа после воздействия НФ;

$c'_0 = 94$ – пропускная способность избранных ребер до воздействия НФ;

$c''_0 = 110$ – пропускная способность остальных ребер, не подвергшихся воздействию НФ;

$c'_\gamma = c_\gamma^0 - c''_0 = 53$ – пропускная способность пораженных ребер;

$$\Theta_\gamma = \frac{c'_\gamma}{c_0} = 0,56;$$

$\Theta_\gamma^k, k = 0,1,2$ – значение Θ после воздействия НФ;

$$\mu^k = \sum \frac{c^k}{c^0}.$$

В табл. П9.1 представлены значения для рассматриваемого примера.

Таблица П9.1

γ	c^0	c_γ^0	Ребра	c'_0	c''_0	c'_γ	Θ_γ	Θ_γ^k	μ^k
0,2	204	163	(1 – 2)	94	110	53	0,56	Θ_γ^2	1,00
			(1 – 4)	64	140	23	0,36	Θ_γ^1	0,54
			(2 – 3)	64	140	23	0,36	Θ_γ^1	0,54
			(3 – 5)	46	158	5	0,11	Θ_γ^0	0,23
			(5 – 4)	46	158	5	0,11	Θ_γ^0	0,23
0,2	204	163	(1 – 2)	94	110	12	0,13	Θ_γ^2	1,00
			(2 – 4)	64	140	-22	0		0,54
			(2 – 3)	64	140	-22	0		0,54
			(3 – 5)	46	158	-36	0		0,23
			(5 – 4)	46	158	-36	0		0,23
0,3	204	151	(1 – 2)	143	–	–	–	–	

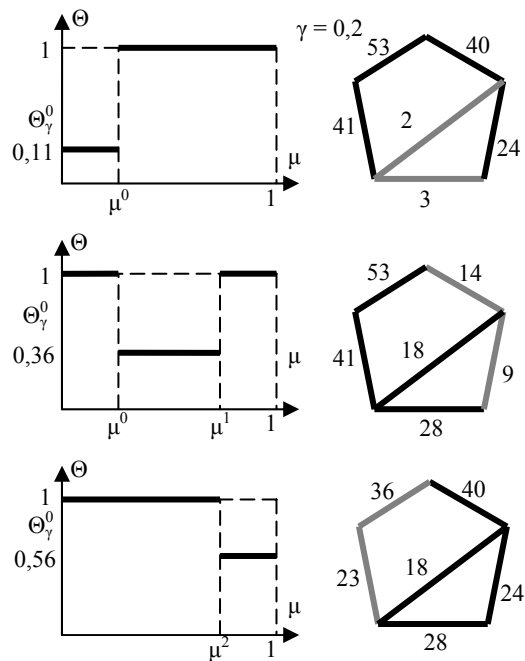


Рис. П9.2. Процесс построения графиков уязвимостей СИС

Как видно из рис. П9.2, при $\gamma = 0,2$ во всех случаях СИС является слабо допустимой. Наиболее уязвимыми оказались ребра с проводимостью 28 и 12. Воздействие НФ мощностью $\gamma = 0,4$ (рис. П9.3) разрушает все ребра, кроме (1 – 2), (1 – 4), эффективность СИС падает до 0,13. Воздействие НФ мощностью $\gamma = 0,3$, приложенное к каждому ребру, полностью разрушает его.

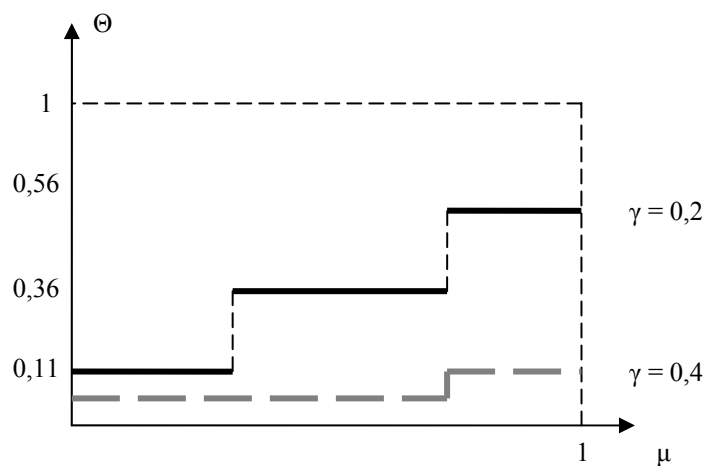


Рис. П9.3. График уязвимости СИС к примеру

**ОБОСНОВАНИЕ УТВЕРЖДЕНИЯ О ВЕРХНЕЙ ОЦЕНКЕ
ВЕЛИЧИНЫ ПРОПУСКНОЙ СПОСОБНОСТИ
ДОСТАТОЧНОГО РЕСУРСА**

$$\Theta_{\gamma}^r(d, t) = 1 - \gamma.$$

Величину пропускной способности кольцевой структуры (достаточный ресурс) можно оценить сверху:

$$t^0 \leq \sum_{i \in M} \frac{d_i}{4}.$$

Обоснование. Суммарное значение потерянного потока равно $\gamma \sum d_i$, следовательно, для обеспечения максимального уровня Θ_{γ}^r необходимо для каждой тяготеющей пары p_i пропустить поток $(1 - \gamma)d_i$.

По частично поврежденному ребру, пропускная способность которого окажется $y_r > (1 - \gamma)d_k$, нужное значение потока пройдет по радиальному ребру и еще останется резерв пропускной способности, что позволит к кольцевой структуре пройти потокам $\sum_{i \neq k} f_k^i \leq y_k - (1 - \gamma)d_k$, что обеспечит дополнительные потоки для других тяготеющих пар p_i с $y_i < (1 - \gamma)d_i$.

Суммарный поток, проходящий по кольцевой структуре, не превысит значения:

$$\sigma(y) = \sum_{k \in M} [y_k - (1 - \gamma)d_k].$$

Максимальное значение такого потока

$$\sigma^{\circ}[\gamma] = \gamma(1 - \gamma) \sum_{i \in M} d_i.$$

Максимального значения $(1 - \gamma)$ достигаем при $\gamma = \frac{1}{2}$, следовательно

$$\sigma^{\circ}[\gamma] = \frac{1}{4} \sum_{i \in M} d_i.$$

**ПРИМЕР СРАВНИТЕЛЬНОГО АНАЛИЗА УЯЗВИМОСТИ
ТРЕХПРОДУКТОВЫХ СЕТЕВЫХ ИНФОРМАЦИОННЫХ
СИСТЕМ С РАЗЛИЧНЫМИ ФИЗИЧЕСКИМИ ГРАФАМИ**

Рассмотрим и сравним три простейших примера СИС, которые, однако, позволяют продемонстрировать некоторые особенности введенных понятий.

На множестве из трех узлов построим три различные СИС. S_1 , S_2 и S_3 проставлены в соответствующих кружках, а цифры рядом с ребрами указывают на величину их исходной пропускной способности (рис. П11.1).

Пусть требования на передачу потоков, т.е. тяготения, в сетях S_1 , S_2 и S_3 являются одинаковыми и составляют: между узлами v_1 и v_2 $d_1 = 10$, между узлами v_1 и v_3 $d_2 = 20$ и, наконец, между v_2 и v_3 $d_3 = 30$ (граф тяготений совпадает с графом СИС S_2). Нетрудно убедиться, что данные требования можно обеспечить в каждой из СИС, причем $\Theta_0(c^0, d) = 1$, $M_0(c^0, d) = M$.

Далее на рис. П11.2 приведены различные варианты состояний СИС S_1 , S_2 и S_3 и построены диаграммы уязвимости для разрушений разной мощности: $\gamma = 0,2; 0,4; 0,5; 0,6$.

Напомним, что после разрушающего воздействия с параметром γ рассчитывается как $\sum_{k=1}^e c_k = (1 - \gamma)c_0$, где c_0 – исходная суммарная пропускная способность ребер, которая составляет для СИС S_1 : $c_{01} = 70$, для S_2 : $c_{02} = 60$ и для S_3 : $c_{03} = 120$.

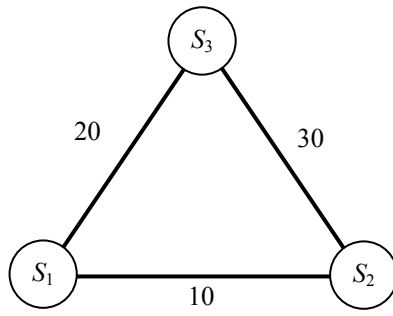
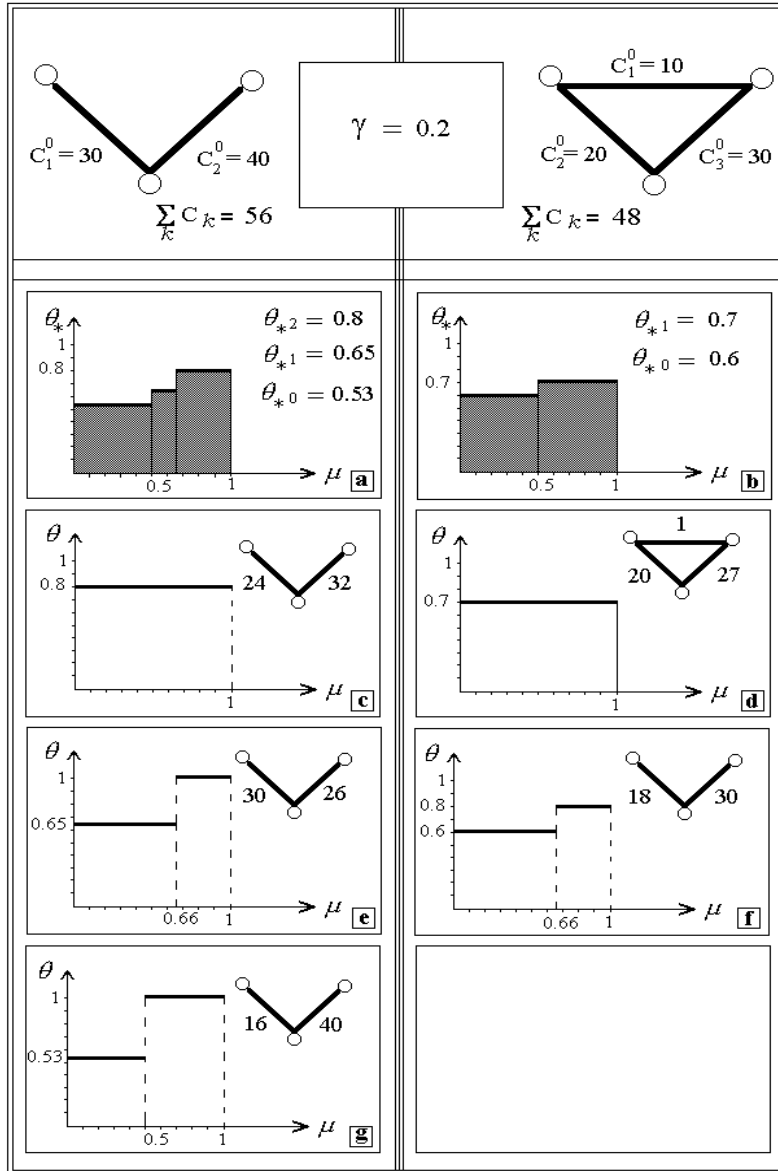
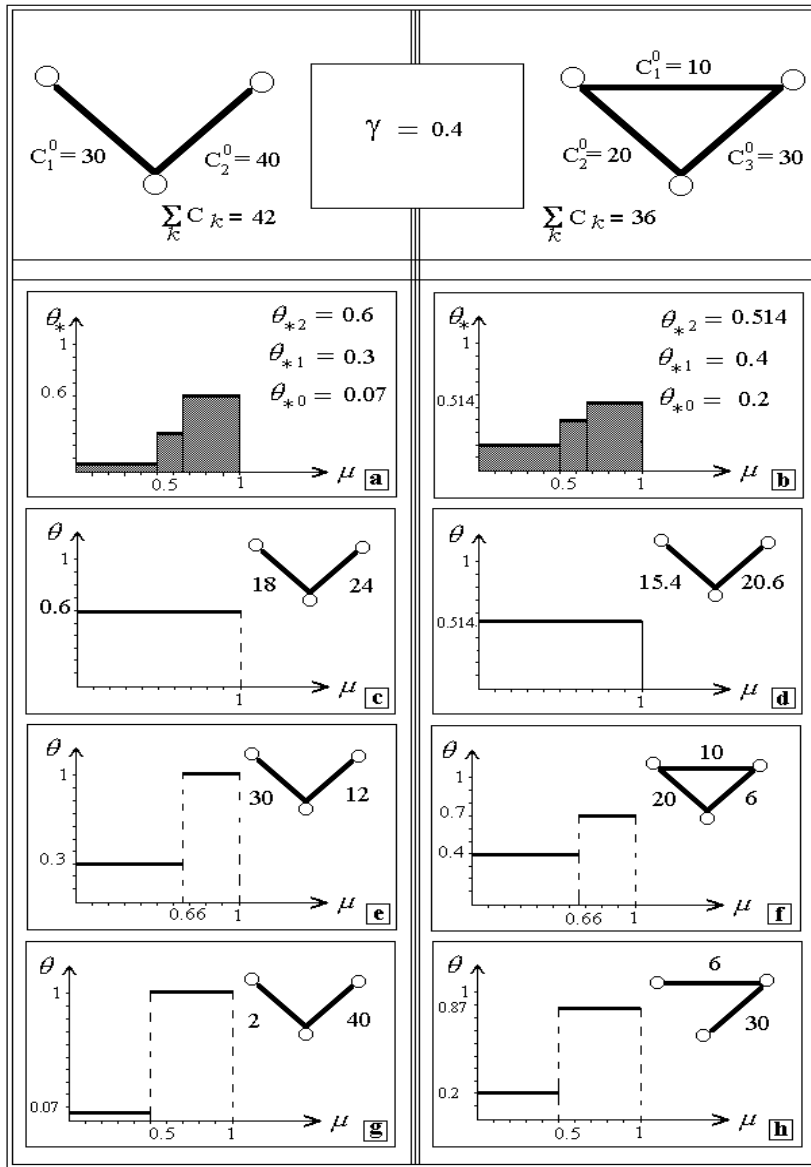


Рис. П11.1. Иллюстрация к примеру



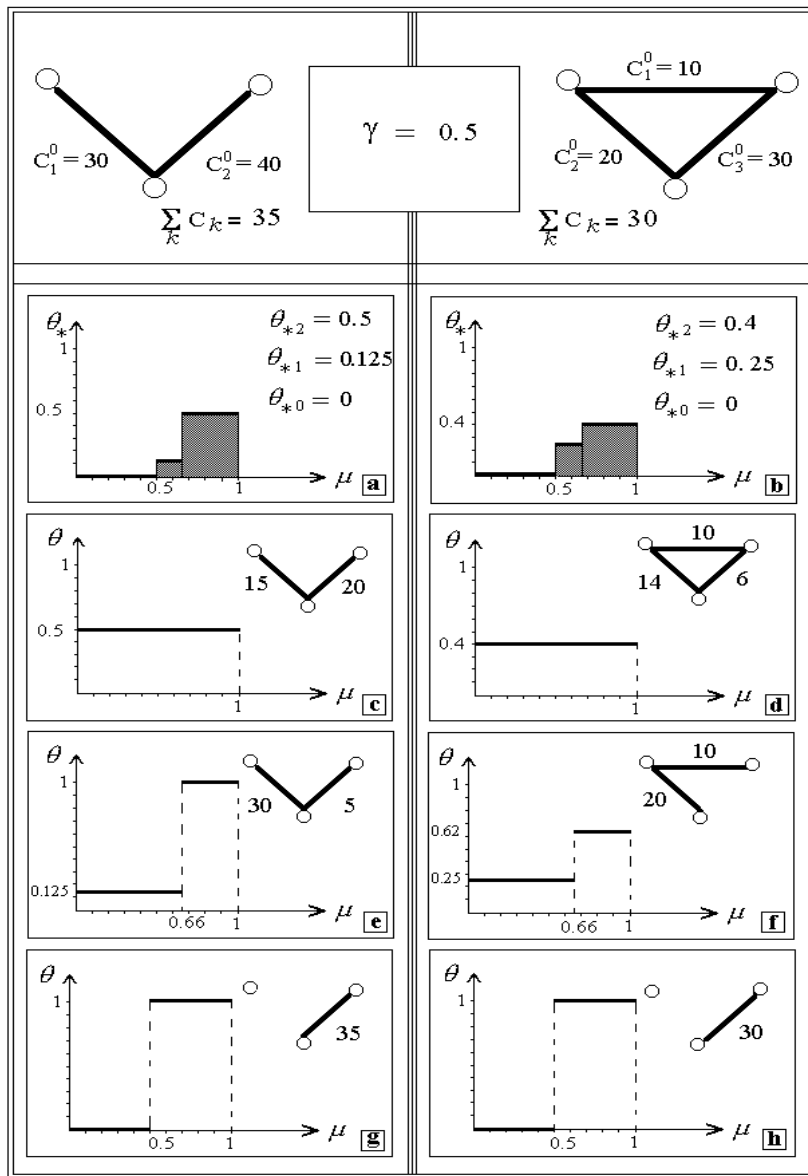
a)

Рис. П11.2. Различные варианты состояния СИС и диаграммы уязвимости



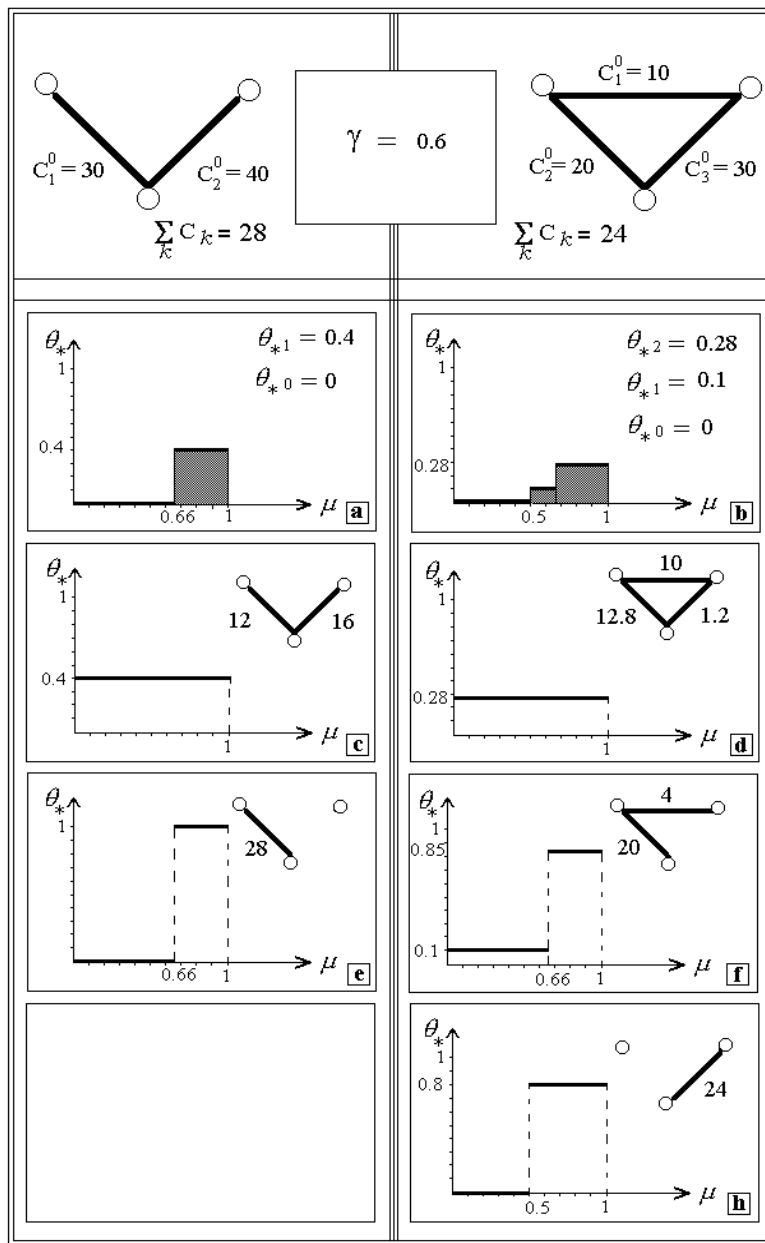
б)

Рис. П11.2. Продолжение



6)

Рис. П11.2. Продолжение



з)

Рис. П11.2. Окончание

Каждый из рис. П11.2 соответствует одному значению γ , все рисунки построены по одинаковому принципу, поэтому рассмотрим для примера только рис. П11.2, б. В верхней части рисунка приведено численное значение $\gamma = 0,4$. Указано, что левый столбец диаграмм относится к СИС S_1 , средний – к СИС S_2 , а правый — к СИС S_3 , далее приводятся соответствующие значения остаточной суммарной пропускной способности для СИС S_1 : $\sum_{k=1}^2 c_k = 42$, для S_2 : $\sum_{k=1}^3 c_k = 36$ и для S_3 : $\sum_{k=1}^3 c_k = 72$.

На рис. П11.2, б изображены результирующие диаграммы уязвимости, соответственно СИС S_1 , S_2 и S_3 . Здесь по оси ординат отложены значения Θ_{*i} , а по оси абсцисс – значения μ_i . Рядом приведены численные значения высот «ступенек» на данной диаграмме. Указанные значения являются точными гарантированными (нижними) оценками для обеспеченности соответствующей $(\mu_i - \mu_{i-1})$ -й части потоковых требований в сетях S_1 , S_2 и S_3 .

Диаграмма уязвимости (рис. П11.3) получена для СИС S_2 , для диаграмм обеспеченности требований (см. рис. П11.2) по всевозможным $c \in C[\gamma]$ – возмущениям с $\gamma = 0,4$. На рис. П11.2, а приведены в определенном смысле наихудшие варианты таких возмущений (составляющие множество $C^*(0,4)$ для рассматриваемой СИС).

Рассмотрим рис. П11.2. Справа – сверху изображена СИС, возникшая после возмущения. Пропускные способности ребер (определяются соответствующим c^*) указаны на рисунке. Считаем, что величины d_i тяготений – такие же, как в исходной СИС. При суперконкурентном распределении потоков будет получена диаграмма обеспеченности требований, изображенная на рисунке слева. По оси ординат (см. рис. П11.2) откладываются значения Θ_i (полученные из решения задач (1.4), (1.6) и т.п.), а по оси абсцисс – величины μ_i , которые вычисляются по формуле (1.9). Для СИС на рис. П11.3 значение

$\Theta_0(c, d) = 0,2$, и оно является наилучшим в том смысле, что при данном $\gamma = 0,4$ не существует такого возмущения (такого $c' \in C[\gamma]$), при котором $\Theta_0(c', d)$ окажется меньше, чем $0,2$. Аналогичным свойством обладают все СИС, изображенные на рис. П11.2 самыми нижними, они формируют нижние «ступеньки» диаграмм уязвимости и дают значения Θ_{*0} . Следующие ступеньки Θ_{*1} , Θ_{*2} и т.д. во всех рассмотренных примерах тоже задаются значениями Θ_0 .

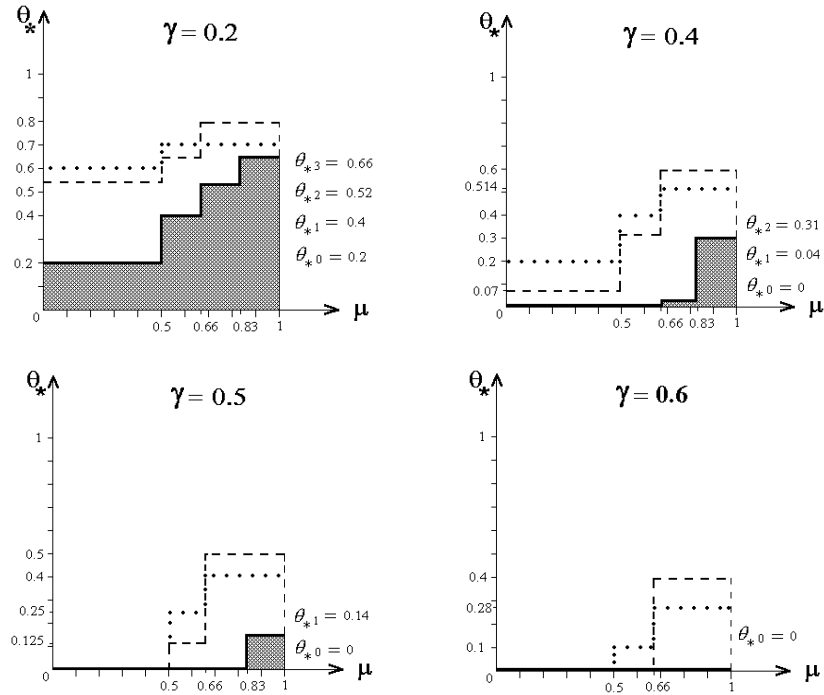


Рис. П11.3. Диаграммы уязвимостей графов СИС

При сравнении диаграмм уязвимости, приведенных на рис. П11.2, слева и в середине, обращают на себя внимание следующие особенности. Для СИС S_1 значения Θ_{*0} оказываются меньше или равными соответствующим значениям для СИС S_2 (т.е. левая крайняя ступенька диаграммы для СИС S_1 оказывается ниже или совпадает с соответствующей ступенькой для S_2). Естественно, при разрушении СИС со структурой типа «дерево» легче нарушаются связи между узлами, и поэтому при неравномерном (для тяготеющих пар) поражении ребер СИС удастся нанести больший ущерб наименее обеспеченным – см. нижние диаграммы на рисунках. С другой стороны, самая правая ступенька на диаграммах для СИС S_1 выше, чем на диаграммах для S_2 , т.е. структура «дерево» в каком-то смысле лучше (по критерию неуязвимости), чем полный граф. Последнее вызвано как особенностями СИС, так и предлагаемым правилом перераспределения потоков (выбором конкурентного решения).

Действительно, при разрушении СИС типа «дерево» несколько потоков приходится передавать по сильно разрушенному ребру СИС, но в другой части СИС могут сохраниться ребра с большой пропускной способностью, что позволит почти полностью удовлетворить требованиям оставшихся тяготеющих пар, т.е. получить (при суперконкурентном решении) относительно большие значения Θ_j . В СИС S_2 возможно равномерно (для тяготеющих пар) распределить оставшуюся пропускную способность после воздействия НФ и получить относительно большие значения Θ_0 . Но именно это более равномерное распределение приводит к тому, что последующие значения Θ_l (включая последнее Θ_L) не так сильно отличаются от Θ_0 . В результате, в случае когда противник выбирает удар с целью максимизировать ущерб одной – не важно какой – тяготеющей пары, СИС типа «дерево» для него предпочтительнее. А когда он пытается максимизировать ущерб каждой пары (справедливо также для стихийных бедствий), то уязвимей полный граф. Причем оптимальным для противника в этом случае будет неравномерное распределение мощности удара (в отличие от структуры типа «дерево»), требующее перераспределения потоков для уравнивания тяготеющих пар и применения «обходных» маршрутов, что приводит к менее эффективному, чем в исходной СИС, использованию оставшейся пропускной способности ребер.

Диаграммы на рис. П11.2 соответствуют стремлению противника равномерно понизить значения обеспеченности требований всех тяготеющих пар в СИС. В такой постановке – вполне обоснованной для противника (в роли которого, как уже подчеркивалось, может выступать и неопределенный фактор), не информированного об относительной важности тяготеющих пар или об их взаимозаменяемости, – уязвимость СИС S_1 (с физическим графом типа «дерево») меньше, чем уязвимость СИС S_2 (на полном физическом графе). Отметим также соотношение $\Theta_{*L} \leq 1 - \gamma$, приведенное в [97], т.е. на СИС S_1 реализовалась верхняя граница Θ_{*L} .

Таким образом, можно сказать, что полносвязная СИС является «равнопрочной» для всех тяготеющих пар в целом, но нельзя сказать, что она вообще «менее уязвима», поскольку в СИС типа «дерево» всегда находятся тяготеющие пары, для которых гарантирована более высокая (чем на СИС с полным графом) степень обеспечения их потоковых требований; боль-

шее значение Θ_{*L}^* показывает, что некоторая пара остается «менее уязвимой».

Указанный факт противоречит теоретико-графовым представлениям о живучести СИС, так как число реберной связности древовидного графа всегда меньше, чем полного. Конечно, модельные предположения, заложенные в основу теоретико-графовых постановок задач за противника («нападающего» – разрушающего СИС), не во всем близки к вышеупомянутой концепции равномерного подавления [98].

СИС S_3 дает еще один пример различия результатов применения теоретико-графового и потокового подходов к исследованию уязвимости и кроме того иллюстрирует противоречивость критериев уязвимости и стоимости при синтезе сетевых структур. Действительно, при решении задачи синтеза СИС зачастую используется (возможно, в качестве компоненты) радиальная структура, когда все узлы логического графа (источники и стоки) соединяют физическими ребрами с единственным транзитным узлом. Этот узел обычно располагают в так называемой «точке Штейнера», сумма расстояний от которой до остальных вершин минимальна. Подобная структура позволяет минимизировать затраты на прокладку линий связи без учета стоимости самих каналов связи и применяется, в частности, в теплосетях, при проектировании линий электропередачи и т.п. Тем не менее из сравнения приведенных рисунков очевидно, что СИС S_3 (с радиальной структурой) более уязвима, чем и S_1 , и S_2 , – ее диаграммы уязвимости ниже других при всех значениях γ , а удара с $\gamma = 0,6$ для нее уже достаточно, чтобы лишились связи все тяготеющие пары. Для удобства представим на одном рисунке диаграммы уязвимости всех трех рассмотренных вариантов физического графа СИС (см. рис. П11.3).

Выделим черным диаграмму для СИС с радиальной структурой, штриховой линией – древовидным графом и пунктирной – для полного графа.

Для обоих графов удаление одного ребра нарушает связность, а двух – разделяет все тяготеющие пары; минимальный разрез из разделяющих все пары и там и там имеет пропускную способность $30 + 40 = 70$. Таким образом, сетевая структура, создание которой может показаться более предпочтительным по экономическим соображениям, будет обладать худшими свойствами уязвимости, но чтобы это выяснить, придется провести их подробный анализ с учетом возможности перераспределения потоков. Последняя возможность не принимается во внимание теоретико-графовым подходом.

В целом следует отметить, что задачи вычисления на графах показателей, так или иначе характеризующих уязвимость СИС, в большинстве своем являются NP-трудными. Поэтому нет оснований предпочесть графовые подходы потоковым и с точки зрения возможностей практического счета.

ОГЛАВЛЕНИЕ

Введение	3
1. Аналитическое обеспечение информационной системы оценки живучести сетевых структур	4
1.1. Общее описание информационной системы оценки живучести сетевых структур	4
1.2. Критерий живучести графа. Максимизация живучести. Условие существования физического графа сетевой информационной системы	11
1.3. Вычисление общей живучести сетевой информационной системы в полиномиальной форме	17
1.4. Вычисление живучести сетевых информационных систем с использованием элементов искусственного интеллекта	28
1.5. Поточковая модель живучести сетевой информационной системы	38
1.5.1. Исследование живучести стохастической сетевой информационной системы	38
1.5.2. Описание поточковой модели сетевой информационной системы	44
1.5.3. Организация нормативного потока	48
2. Процедурное обеспечение информационной системы оценки живучести сетевых структур	52
2.1. Описание блока анализа сетевой структуры	53
2.2. Процедурные модели оценки живучести сетевых информационных систем	58
2.3. Анализ сетевой информационной системы на основе модели МП-сети	65
2.3.1. Задача выбора пропускных способностей	69
2.3.2. Процедурная модель выбора дискретных пропускных способностей	70
2.3.3. Комбинированная задача выбора пропускных способностей и распределения потоков. Постановка задачи	72
2.3.4. Обобщенная задача вычисления пропускных способностей и распределения потоков	72
2.3.5. Процедурная модель нахождения кратчайших путей и расчета объемов суммарной передачи информации (трафиков)	73
2.4. Построение графика уязвимости	77
2.5. Процедурная модель анализа уязвимости сетевой информационной системы	81
2.6. Поиск гарантированного уровня допустимости сетевой информационной системы $\Theta_0^{\Gamma^*}$	88
2.7. Синтез сетевой информационной системы с гарантией живучести. Построение по критерию допустимости	91
3. Построение информационной системы оценки живучести сетевых структур	97

3.1. Общее описание информационной системы оценки живучести сетевых информационных систем	98
3.2. Описание информационного обеспечения	99
3.3. Описание программного обеспечения	106
3.3.1. Общее программное обеспечение	106
3.3.2. Специальное программное обеспечение	108
3.3.3. Описание лингвистического обеспечения	110
Заключение	113
Список литературы	115
Приложения	122