ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

3(175) 2011

ТЕОРЕТИЧЕСКИЙ И ПРИКЛАДНОЙ НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

Издается с ноября 1995 г.

УЧРЕДИТЕЛЬ
Издательство "Новые технологии"

СОДЕРЖАНИЕ

Информация о журнале доступна по сети Internet по адресу http://www.**informika.ru/text/magaz/it/** или **http://novtex.ru/IT.**

Журнал включен в систему Российского индекса научного цитирования.

Журнал входит в Перечень научных журналов, в которых по рекомендации ВАК РФ должны быть опубликованы научные результаты диссертаций на соискание ученой степени доктора и кандидата наук.

УДК 621.317.75:519.2

Д. А. Перепелкин, ассистент,
Рязанский государственный
радиотехнический университет,
А. И. Перепелкин, канд. техн. наук, доц.,
Рязанский государственный университет
им. С. А. Есенина
E-mail: dmitryperepelkin@mail.ru

Алгоритм адаптивной ускоренной маршрутизации в условиях динамически изменяющихся нагрузок на линиях связи в корпоративной сети

Предложен алгоритм адаптивной ускоренной маршрутизации, повышающий эффективность функционирования корпоративных сетей в условиях динамических изменений нагрузок на линиях связи.

Ключевые слова: *адаптивная ускоренная маршрутизация, алгоритмы маршрутизации, динамические изменения, корпоративные сети*

Введение

Быстрый рост числа компьютерных сетей, успехи в развитии оптоволоконных и беспроводных средств связи сопровождаются непрерывной сменой сетевых технологий, направленной на повышение быстродействия и надежности сетей, возможностей интегрированной передачи данных, голоса и видеоинформации.

В современных корпоративных сетях обеспечение высокоскоростного и надежного обмена информацией между узлами сети при жестких требованиях к задержкам информации и увеличении числа узлов в сети является одной из важнейших проблем. Модификация структуры сети, включение в нее новых узлов и линий связи приводят к полному пересчету таблиц маршрутизации. Использование традиционных методов маршрутизации в этих условиях оказывается неэффективным. Разработка новых, более эффективных алгоритмов поиска оптимальных маршрутов позволяет повысить надежность и быстродействие передачи данных в корпоративных сетях.

Теоретический анализ

Выбор маршрутов в узлах связи телекоммуникационной системы проводится в соответствии с реализуемым алгоритмом маршрутизации.

В настоящее время наибольшее распространение получили алгоритмы адаптивной маршрутизации. Они обеспечивают автоматическое обновление таблиц маршрутизации после изменения конфигурации сети. Используя протоколы адаптивных алгоритмов, маршрутизаторы могут собирать информацию о топологии связей в сети и оперативно реагировать на все изменения конфигурации связей.

Анализ используемых в современных корпоративных сетях алгоритмов адаптивной маршрутизации показывает, что для построения таблиц маршрутизации используются два известных алгоритма — Беллмана—Форда с трудоемкостью порядка $O(N^3)$ и Дейкстры с трудоемкостью $O(N^2)$, где N — число маршрутизаторов в сети [1].

Применение этих алгоритмов в условиях динамического изменения нагрузок на линиях связи и увеличения числа узлов в современных корпоративных сетях является неэффективным вследствие высокой трудоемкости поиска оптимальных маршрутов и необходимости полного пересчета таблиц маршрутизации.

В последнее время разработаны новые алгоритмы маршрутизации, например алгоритм парных переходов [2], который позволяет уменьшить трудоемкость построения таблиц маршрутизации до значения $O(K \cdot N)$, где K — число парных переходов, при изменении нагрузки на линиях связи.

Алгоритм нахождения кратчайших путей на основе парных переходов характеризуется необходимостью расчета дополнительной информации. Поэтому объем данных, рассчитываемых на подготовительном этапе для обеспечения общности условий принимаемых решений, является избыточным. В данном алгоритме для того, чтобы обработать любое изменение веса некоторого ребра, необходимо присвоить этому весу граничные значения и провести поиск кратчайших путей для полученного графа.

Рассмотрим работу алгоритма парных переходов. Пусть корпоративная сеть представлена в виде неориентированного взвешенного связного графа $G = (V, E, W)$, где V — множество вершин, $\|V\| = N$;

E — множество ребер, $\|E\| = M$, W — множество весов ребер (рис. 1).

Пусть на графе G в некоторый момент времени уже решена задача поиска кратчайших путей до всех вершин множества $V_{\hat{s}} = V|v_s$ из начальной вершины v_s, т. е. построено дерево кратчайших путей с корнем в вершине v_s. Обозначим это дерево как T_g. На рис. 1 жирными линиями обозначено построенное дерево кратчайших путей.

Предположим, увеличился вес ребра, входящего в дерево кратчайших путей. Пусть изменилось значение веса ребра $e_{1,2}$, входящего в дерево кратчайших путей так, что новое значение $nw_{1,2} = 13 > w_{1,2} = 2$.

В этом случае для построения нового дерева кратчайших путей необходимо последовательно сделать четыре парных перехода:

$$e_{1,2}—e_{2,4};\ e_{2,5}—e_{4,5};\ e_{5,7}—e_{4,7};\ e_{5,8}—e_{6,8}.$$

Причем после каждого такого перехода требуется рассчитывать дополнительную информацию для осуществления следующего парного перехода.

После всех переходов граф G примет окончательный вид, показанный на рис. 2.

Общая трудоемкость алгоритма парных переходов для построения полного дерева кратчайших путей составляет $O(KN)$.

Недостатком алгоритма парных переходов является то, что в условиях динамически изменяющихся нагрузок на линиях связи в корпоративной сети после каждого парного перехода необходимо рассчитывать дополнительную информацию для того, чтобы определить оптимальный маршрут до других узлов в сети.

Разработка новых, более эффективных алгоритмов адаптивной маршрутизации позволяет уменьшить трудоемкость построения таблиц маршрутизации.

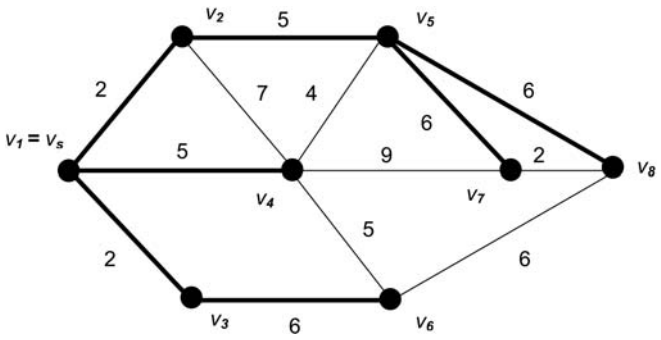


Рис. 1. Граф G корпоративной сети

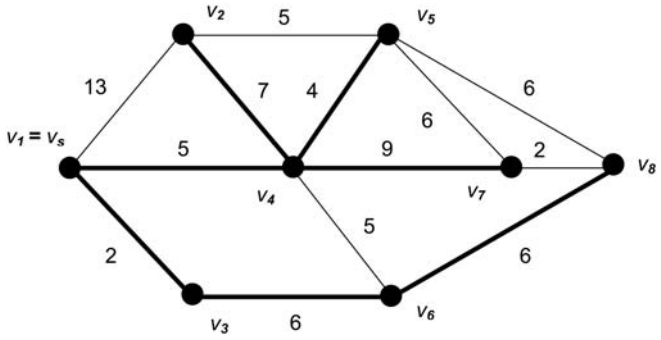


Рис. 2. Граф G после всех парных переходов

Разработка алгоритма

Для повышения эффективности функционирования корпоративных сетей предложен алгоритм адаптивной ускоренной маршрутизации на базе алгоритма парных переходов, который позволяет уменьшить трудоемкость построения таблиц маршрутизации до величины $O(N)$ по сравнению с известными алгоритмами с трудоемкостью $O(N^2)$ и $O(KN)$.

Обозначим $w_{i,j}$ — вес ребра, соединяющего вершины v_i и v_j; $nw_{i,j}$ — новое значение веса, полученное в результате изменения значения метрики линии связи. Вершина v_j располагается ниже по иерархии дерева кратчайших путей относительно v_i. Обозначим E_T множество ребер, каждый элемент которого входит, по крайней мере, в один кратчайший путь из начальной вершины, E_R — множество остальных ребер; $E_R \cup E_T = E$, $E_R \cap E_T = \varnothing$. Обозначим V_T — множество вершин, до которых найден кратчайший путь из начальной вершины; V_R — множество остальных вершин; $V_R \cup V_T = V$, $V_R \cap V_T = \varnothing$.

Будем называть V_k — путем R_k или совокупностью подмножества $V^{(Vk)} \subseteq V$ вершин, через которые проходит кратчайший путь до вершины v_k из исходной вершины v_s, и подмножества $E^{(Vk)} \subseteq E$ ребер, составляющих этот путь.

Назовем V_k — деревом T_k или совокупностью подмножества $V_T^{(Vk)} \subseteq V$, состоящего из всех вершин, кратчайшие пути до которых из исходной вершины содержат вершину v_k и подмножества $E_T^{(Vk)} \subseteq E$ ребер, составляющих эти пути после v_k при движении от вершины v_s.

Обозначим множество путей до вершины v_i из исходной вершины v_s через Π_i, где элемент множества $\pi_{i,k} \in \Pi_i$ будет множеством не повторяющихся ребер $e_{i,j} \in E$, образующих вместе путь, соединяющий v_s и v_i. Для всех $\pi_{i,k} \in \Pi_i$ поставим в соответствие некоторое число, равное сумме весов, входящих в него ребер, т. е. длину пути $d_{i,k} \in D_i$. На множестве Π_i задан селектор H, возвращающий кратчайший путь из множества Π_i. В том случае, если существует нескольких путей в Π_i с минимальной длиной, то выбирается один из них. Кратчайший путь до вершины v_i будем обозначать $\pi_i = H(\Pi_i)$, оценку длины $\pi_i - d_i$.

Для решения поставленной задачи сформулируем следующие теоремы.

Теорема 1. Если n$w_{i,j} > w_{i,j}$ и $e_{i,j} \in E_T$, то изменению могут подвергнуться кратчайшие пути и оценки их длин для вершин $V_T^{(Vj)}$.

Доказательство. Пусть увеличился вес ребра $e_{i,j} \in E_T$, которое входит, по крайней мере, в один кратчайший путь π_k, например в $\pi_{k,p}$. Вершины v_k, в кратчайшие пути до которых ребро $e_{i,j}$ не входит, будут составлять множество V_T вершин, кратчайшие пути до которых после изменения останутся прежними (не изменится последовательность ребер и значения кратчайших путей). Действительно, пусть существует кратчайший путь $\pi_k = \pi_{k,p}$ до вершины v_k, и известно, что ребро $e_{i,j}$ не входит в этот путь. Тогда увеличение веса этого ребра со значения $w_{i,j}$ до $nw_{i,j}$ не изменит маршрута этого пути и не повлияет на его значение $d_{k,p}$, т. е. $\pi_{k,p}$ и $d_{k,p}$, поскольку еще до увеличения веса рассматриваемого ребра включение этого ребра в кратчайший путь приводило к увеличению длины пути. Все вершины, не вошедшие во множество V_T, будут составлять множество V_R. Кратчайшие пути до вершин множества $v \in V_R$ станут "недействительными", т. е. невозможно будет без дополнительного расчета сказать, останутся они такими же или кратчайший путь до них не будет включать изменившееся ребро. Теорема доказана.

Теорема 2. Если $nw_{i,j} < w_{i,j}$ и $e_{i,j} \in E_T$, то без изменения останутся кратчайшие пути для вершин множества $v \in V_T^{(Vj)} \cup V^{(Vi)}$, а для вершин множества $V^{(Vi)}$ неизменными останутся и оценки длин кратчайших путей.

Доказательство. Пусть уменьшился вес ребра $e_{i,j} \in E_T$, входящего в кратчайший путь $\pi_k = \pi_{k,p}$ до вершины $v_k \in V$. Ребро $e_{i,j}$ после изменения также будет входить в кратчайший путь π_k до вершины v_k. Поскольку вес ребра $w_{i,j}$ изменился, то измениться должны длины всех путей $\pi_{t,r}$, в которые входит это ребро. Действительно, если ребро $e_{i,j}$ входит в какой-либо кратчайший путь и вес этого ребра уменьшается, то это изменение не потребует изменения кратчайшего пути $\pi_{k,p}$ (последовательности ребер) и длина пути $d_{k,p}$ изменится (уменьшится) на значение изменения веса ребра. Пути π_s, $v_s \notin V_T^{(Vj)} \cup V^{(Vi)}$ станут "недействительными", т. е. невозможно будет без дополнительного расчета сказать, останутся они такими же или кратчайший путь до них будет включать изменившееся ребро. Теорема доказана.

Теорема 3. Если $nw_{i,j} > w_{i,j}$ и $e_{i,j} \notin E_T$, то исходное дерево кратчайших путей и оценки длин путей всех вершин не изменятся.

Доказательство. Пусть ребро, не входящее ни в один кратчайший путь, увеличивает свой вес $w_{i,j}$, $e_{i,j} \in E_R$. Никаких изменений дерева кратчайших путей при этом не происходит. Действительно, пусть ребро $e_{i,j} \in E$ входит в путь $\pi_{k,p}$ до некоторой вершины v_k, который не является кратчайшим для v_i, $-\pi_{k,p} \neq \pi_k$, т. е. существует такой путь $\pi_{k,t} = \pi_k$, что $d_{k,p} > d_{k,t}$. Тогда после увеличения веса $w_{i,j}$ увеличится оценка длины $d_{k,p}$ и неравенство $d_{k,p} > d_{k,t}$ останется справедливым, т. е. кратчайший путь и его оценка до вершины v_k остается неизменной. Теорема доказана.

Теорема 4. Если $nw_{i,j} < w_{i,j}$ и $e_{i,j} \notin E_T$, то без изменения останутся кратчайшие пути и оценки их длин для вершин множества $V^{(Vi)}$.

Доказательство. Пусть уменьшился вес ребра $e_{i,j} \in E_R$, которое не входит ни в один кратчайший путь. Допустим, что это ребро входит в путь $\pi_{i,k} \neq \pi_i$ и $\pi_{j,p} \neq \pi_j$. Если изменившееся ребро $e_{i,j}$ не уменьшает оценок обеих инцидентных ему вершин v_i и v_j, т. е. $d_{i,k} \geqslant d_i$ и $d_{j,p} \geqslant d_j$, дерево кратчайших путей не изменится. Действительно, рассматриваемое ребро оказывает влияние прежде всего на инцидентные ему вершины множества V. Если включение ребра $e_{i,j}$ в дерево не уменьшает оценок пути d_i, d_j, то такое включение только увеличит оценки вершин. Так как существовавшие до изменения пути до этих вершин имели меньшую длину, то данное ребро не включается в дерево кратчайших путей. Если включение этого ребра приводит к уменьшению оценки какой-либо из инцидентных вершин, например v_i, то эта оценка $d_{i,k}$ будет оценкой кратчайшего пути до вершины v_i и ребро $e_{i,j}$ войдет в искомое дерево кратчайших путей. Это происходит в силу того, что после изменения не существует иного кратчайшего пути π_i до вершины v_i, кроме пути $\pi_{i,k}$, содержащего ребро $e_{i,j}$. Этот кратчайший путь $\pi_{i,k}$ не будет существовать, если не будет кратчайших путей до всех промежуточных вершин $v_p \in V^{(Vi)}$ этого пути. Кратчайшие пути до остальных вершин графа станут "недействительными", т. е. невозможно будет сказать, останутся они такими же или кратчайший путь до них будет включать изменившееся ребро. Теорема доказана.

Теорема 5. Если $nw_{i,j} > w_{i,j}$ и $e_{i,j} \in E_T$ и $nw_{i,j} > nw_{i,j}^t$ (точки вхождения в дерево), то изменению могут подвергнуться кратчайшие пути и оценки их длин для вершин $V_T^{(Vj)}$, и новые кратчайшие пути к этим вершинам будут проходить через ребра, состоящие в отношении парного перехода к ребрам этих вершин.

Доказательство. Пусть увеличился вес ребра $e_{i,j} \in E_T$, которое входит, по крайней мере, в один кратчайший путь π_k, например в $\pi_{k,p}$. Согласно теоремы 1 вершины v_k, в кратчайшие пути до ко-

торых ребро $e_{i,j}$ не входит, будут составлять множество V_T вершин, кратчайшие пути до которых после изменения останутся прежними (не изменится последовательность ребер и длина кратчайших путей). Для вершин $V_T^{(Vj)}$ среди парных переходов, соответствующих этим вершинам, будут находиться ребра, имеющие минимальную длину пути к этим вершинам. Теорема доказана.

Следствие. При увеличении веса ребра, входящего в дерево кратчайших путей для вершин $V_T^{(Vj)}$, маршрутная степень которых больше двух, новые кратчайшие пути будут проходить через ребра, состоящие в отношении парного перехода к ребрам, входящим в исходный граф.

Теорема 6. Если $nw_{i,j} < w_{i,j}$ и $e_{i,j} \notin E_T$ и новое значение $nw_{i,j} < nw_{i,j}^t$ (точки вхождения в дерево), то новые кратчайшие пути к вершинам множества $v \in V_T^{(Vj)} \cup V^{(Vi)}$ будут проходить через ребра, состоящие в отношении парного перехода к ребрам этих вершин.

Доказательство. Пусть уменьшился вес ребра $e_{i,j} \in E_R$, которое не входит ни в один кратчайший путь. Согласно с теоремой 4 без изменения останутся кратчайшие пути и оценки их длин для вершин множества $V^{(Vi)}$. Так как $nw_{i,j} < nw_{i,j}$, то включение этого ребра приводит к уменьшению оценки какой-либо из инцидентных вершин, например v_i, и эта оценка $d_{i,k}$ будет оценкой кратчайшего пути до вершины v_i и ребро $e_{i,j}$ войдет в искомое дерево кратчайших путей. Это происходит в силу того, что после изменения не существует иного кратчайшего пути π_i до вершины v_i, кроме пути $\pi_{i,k}$, содержащего ребро $e_{i,j}$. Этот кратчайший путь $\pi_{i,k}$ не будет существовать, если не будет кратчайших путей до всех промежуточных вершин $v_p \in V^{(Vi)}$ этого пути. Теорема доказана.

Следствие. При уменьшении веса ребра, не входящего в дерево кратчайших путей для вершин $V_T^{(Vj)} \cup V^{(Vi)}$, маршрутная степень которых больше двух, новые кратчайшие пути будут проходить через ребра, состоящие в отношении парного перехода к ребрам, входящим в исходный граф.

На основе сформулированных и доказанных выше теорем разработан алгоритм парных перестановок маршрутов в условиях динамических изменений нагрузок на линиях связи в корпоративной сети. Укрупненная схема алгоритма имеет следующий вид.

Шаг 1. Для вершины, являющейся листом дерева, осуществляется поиск всех парных переходов без ограничений. Эти списки для удобства дальнейшей работы привязываются к вершине, инцидентной рассматриваемому ребру и расположенной ниже по иерархии.

Шаг 2. Если вершина не является листом дерева, то вычисляются парные переходы для этой вершины и выбираются лучшие значения потенциалов парных переходов для потомков вершины и собственных парных переходов. Подобная процедура выполняется для формирования списков парных переходов в случае увеличения и уменьшения веса ребра.

Шаг 3. Для каждой вершины формируется полный список парных переходов. Число элементов в каждом из этих списков не превышает числа вершин графа. Такое решение позволяет отказаться от предварительной сортировки потенциалов или приращений для парных переходов без значительного усложнения алгоритма обработки изменения.

Шаг 4. Для каждой вершины формируется полный список возможных маршрутов, проходящий через ребра, состоящие в отношении парного перехода, включая и ребра, входящие в дерево кратчайших путей.

Шаг 5. Анализируя полученную используемым протоколом маршрутизации информацию, определить, произошло ли изменение метрики для какого-либо ребра:

а) если да — перейти к шагу 6;

б) иначе — к шагу 8.

Шаг 6. Используя список парных переходов, определить, требуется ли сделать парный переход:

а) если да — перейти к шагу 7;

б) иначе — к шагу 9.

Шаг 7. Для каждой вершины, у которой в список возможных маршрутов входит ребро с изменившейся метрикой, определить путь минимальной длины и поместить его в дерево кратчайших путей, тем самым построив новое дерево кратчайших путей на графе.

Шаг 8.

А. Передать пакеты по доступным эквивалентным маршрутам.

Б. Установить флаг передачи.

Шаг 9. Пересчитать точки вхождения в дерево и переформировать список маршрутов замены для каждой изменившейся вершины.

Шаг 10. Перейти к шагу 5.

Рассмотрим работу алгоритма парных перестановок маршрутов на примере графа G корпоративной сети, показанного на рис. 1, в котором уже решена задача поиска кратчайших путей и построено дерево оптимальных маршрутов. Дополнительно рассчитываем список парных переходов для каждого ребра, входящего в дерево оптимальных маршрутов. Определяем возможные маршруты замены из исходной вершины до каждой вершины в сети.

Вершина $V2$:

$\pi_{1,2}^{(1)} = \{e_{1,2}\} = 2$ — входит в дерево оптимальных маршрутов;

$\pi_{1,2}^{(2)} = \{e_{1,4}; e_{4,2}\} = 5 + 7 = 12$ — маршрут замены.

Вершина $V3$:

$\pi_{1,3}^{(1)} = \{e_{1,3}\} = 2$ — входит в дерево оптимальных маршрутов;

$\pi_{1,3}^{(2)} = \{e_{1,4}; e_{4,6}; e_{6,3}\} = 5 + 5 + 6 = 16$ — маршрут замены.

Вершина $V4$:

$\pi_{1,4}^{(1)} = \{e_{1,4}\} = 5$ — входит в дерево оптимальных маршрутов;

$\pi_{1,4}^{(2)} = \{e_{1,2}; e_{2,4}\} = 2 + 7 = 9$ — маршрут замены.

Вершина $V5$:

$\pi_{1,5}^{(1)} = \{e_{1,2}; e_{2,5}\} = 2 + 5 = 7$ — входит в дерево оптимальных маршрутов;

$\pi_{1,5}^{(2)} = \{e_{1,4}; e_{4,5}\} = 5 + 4 = 9$ — маршрут замены.

$\pi_{1,5}^{(3)} = \{e_{1,4}; e_{4,7}; e_{7,5}\} = 5 + 9 + 6 = 20$ — маршрут замены.

Вершина $V6$:

$\pi_{1,6}^{(1)} = \{e_{1,3}; e_{3,6}) = 2 + 6 = 8$ — входит в дерево оптимальных маршрутов;

$\pi_{1,6}^{(2)} = \{e_{1,4}; e_{4,6}\} = 5 + 5 = 10$ — маршрут замены.

Вершина $V7$:

$\pi_{1,7}^{(1)} = \{e_{1,2}; e_{2,5}; e_{5,7}\} = 2 + 5 + 6 = 13$ — входит в дерево оптимальных маршрутов;

$\pi_{1,7}^{(2)} = \{e_{1,4}; e_{4,7}\} = 5 + 9 = 14$ — маршрут замены.

Вершина $V8$:

$\pi_{1,8}^{(1)} = \{e_{1,2}; e_{2,5}; e_{5,8}\} = 2 + 5 + 6 = 13$ — входит в дерево оптимальных маршрутов;

$\pi_{1,8}^{(2)} = \{e_{1,2}; e_{2,5}; e_{5,7}; e_{7,8}\} = 2 + 5 + 6 + 2 = 15$ — маршрут замены;

$\pi_{1,8}^{(3)} = \{e_{1,3}; e_{3,6}; e_{6,8}\} = 2 + 6 + 6 = 14$ — маршрут замены.

Пусть произошло увеличение веса ребра, входящего в дерево кратчайших путей, например, изменилось значение веса ребра $e_{1,2}$, входящего в дерево кратчайших путей так, что новое значение $nw_{1,2} = 13 > w_{1,2} = 2$.

Работа алгоритма парных перестановок маршрутов основывается на том, что при изменении веса ребра, входящего в дерево кратчайших путей или веса ребра, находящегося в отношении парного перехода к ребру из дерева кратчайших путей, необходимо просмотреть списки оптимальных маршрутов и их маршрутов замены до каждой вершины, куда входит ребро, вес которого изменился.

В приведенном примере для этого необходимо просмотреть все оптимальные маршруты и их маршруты замены до каждой вершины и определить новые кратчайшие пути. До вершины $V3$, $V4$ и $V6$ оптимальные маршруты не изменятся. Изменению подвергнутся маршруты до следующих вершин:

Вершина $V2$:

Новый оптимальный маршрут

$\pi_{1,2}^{(2)} = \{e_{1,4}; e_{4,2}\} = 5 + 7 = 12$.

Вершина $V5$:

Новый оптимальный маршрут

$\pi_{1,5}^{(2)} = \{e_{1,4}; e_{4,5}\} = 5 + 4 = 9$.

Вершина $V7$:

Новый оптимальный маршрут

$\pi_{1,7}^{(2)} = \{e_{1,4}; e_{4,7}\} = 5 + 9 = 14$.

Вершина $V8$:

Новый оптимальный маршрут

$\pi_{1,8}^{(3)} = \{e_{1,3}; e_{3,6}; e_{6,8}\} = 2 + 6 + 6 = 14$.

В итоге, после всех изменений граф G корпоративной сети примет вид, показанный на рис. 2.

Работа составных частей алгоритма основывается на использовании доказанных выше теорем, следовательно, можно сделать вывод о корректности работы всего алгоритма в целом.

Используя сформулированные и доказанные теоремы, удается рассчитать дерево кратчайших путей за линейное время. Такой результат получается за счет использования дополнительной информации о возможных маршрутах замены.

Оценка трудоемкости алгоритма парных перестановок маршрутов показывает, что верхняя оценка трудоемкости составляет $\Omega(N)$ и нижняя оценка — $\Theta(N)$.

Таким образом, разработанный алгоритм парных перестановок маршрутов позволяет повысить эффективность функционирования корпоративных сетей за счет использования дополнительной информации о конфигурации сети.

Экспериментальная часть

На основе предложенного алгоритма парных перестановок маршрутов разработана программа имитационного моделирования процессов маршрутизации в корпоративных сетях. При разработке основное внимание уделялось корректности предлагаемого алгоритма и размерности решаемой задачи.

Целью исследования было оценить максимальное, минимальное и среднее значения размерности решаемой задачи. Исходный граф, ребро для изменения и приращения веса выбирались случайным образом. Для каждого испытания на множестве обработанных изменений выбиралось минимальное, максимальное и среднее значения размерности задачи, выраженные через число вершин, для которых необходим поиск кратчайшего пути. По этим значениям были построены графики. Для каждого эксперимента было найдено математическое ожидание и среднее квадратичное отклонение числа изменений. Для разработанного алгоритма определялось число фактически выполненных парных переходов. Были проведены исследования графов, состоящих из 10, 100 и 500 вершин.

В таблице приведены обобщенные статистические характеристики изменений для средней размерности задачи. В представленной таблице через СКО обозначено среднее квадратичное отклонение.

Число вершин графа	Min значение	Max значение	МО	СКО
10	0,23	0,7	0,4236	0,0992
100	0	0,61	0,0924	0,0862
500	9	1,18	0,0324	0,119

Исследование разработанного алгоритма парных перестановок маршрутов показало, что математическое ожидание (МО) числа изменений не превышает величины $N/2$, а его максимальное значение не превышает N.

На основе проведенных экспериментов можно сделать вывод, что предложенный алгоритм парных перестановок маршрутов является эффективным при поиске оптимальных маршрутов в корпоративных сетях.

Заключение

Разработанный алгоритм парных перестановок маршрутов позволяет повысить эффективность функционирования корпоративных сетей за счет уменьшения трудоемкости построения таблиц маршрутизации до значения порядка $O(N)$ в условиях динамически изменяющихся нагрузок и характеристик на линиях связи корпоративной сети.

Список литературы

1. **Олифер В. Г., Олифер Н. А.** Основы компьютерных сетей. — СПб.: Питер, 2009. — 352 с.
2. **Уваров Д. В., Перепелкин А. И.** Построение дерева кратчайших путей на основе данных о парных переходах // Системы управления и информационные технологии. 2004. № 4 (16). С. 93—96.

УДК 004.7, 519.7

Ю. А. Богоявленский, канд. техн. наук, зав. каф.,
e-mail: ybgv@cs.karelia.ru,
К. А. Кулаков,
канд. физ.-мат. наук, ст. преподаватель,
e-mail: kulakov@cs.karelia.ru,
Д. Ж. Корзун, канд. физ.-мат. наук, доц.,
e-mail: dkorzun@cs.karelia.ru,
Петрозаводский государственный университет

Линейные диофантовы модели восстановления соединений в сетях MPLS

Для решения задачи восстановления соединений в сетях MPLS предлагается использовать однородные линейные диофантовы системы специального вида и их базисы Гильберта как математический аппарат моделирования маршрутов. Такой подход позволяет уменьшить трудоемкость решения задачи поиска резервных маршрутов по сравнению с известной графовой моделью. Предлагаемая кумулятивная характеристика маршрута определяет его качество в зависимости от характеристик линий связи. Для реализации моделей используются авторские псевдополиномиальные алгоритмы, позволяющие за приемлемое время находить маршруты-кандидаты для сетей MPLS реальных размерностей.

Ключевые слова: *сети MPLS, восстановление соединений, линейные диофантовы модели, кумулятивная характеристика*

Введение

В работе [1] было отмечено возрастание в сети Интернет роли технологии MPLS (*Multiprotocol Label Switching*) [2]. В сетях MPLS актуальной является задача восстановления соединений. Для каждого соединения определяется маршрут — последовательность MPLS-маршрутизаторов. При выходе из строя элемента маршрута требуется восстановить соединение за гарантированное время, переключив его на другой маршрут.

Технология MPLS имеет общий механизм восстановления соединения, включающий задачу поиска маршрутов [3]. Для последней предложен алгоритм Short Leap Shared Protection (SLSP) [4], где топология сети моделируется графом. Простые циклы графа определяют маршруты-кандидаты, а для выбора оптимального решается задача целочисленного линейного программирования (ЦЛП). Способ построения подходящего множества циклов определяет размерность задачи ЦЛП.

Для сетей реальных размерностей такая задача ЦЛП является трудоемкой. В настоящей статье для замены SLSP-модели предлагается иерархия линейных диофантовых моделей на основе систем однородных неотрицательных линейных диофантовых уравнений (одНЛДУ) и их базисов Гильберта. Они уменьшают трудоемкость решения задачи ЦЛП за счет рационального построения маршрутов-кандидатов. Такие маршруты определяются базисом Гильберта (БГ), для нахождения которого из-

вестны псевдополиномиальные алгоритмы [5, 6] и их протестированные программные реализации [7]. Для оценки качества маршрутов-кандидатов нами предлагается кумулятивная характеристика, интегрирующая характеристики линий связи маршрута непосредственно в компонентах решений из БГ. Возможность практической реализации предлагаемых моделей подтверждается экспериментально на моделях с размерностями, соответствующими реальным сетям MPLS. Эти модели строятся с помощью алгоритмов генерации систем одНЛДУ с известным БГ [8].

В разделе 1 сформулированы задачи восстановления соединения и поиска маршрута восстановления. В разделе 2 представлен используемый математический аппарат. Далее предложены модели топологии (раздел 3), с фиксированным соединением (раздел 4) и с характеристиками линии связи (раздел 5). Кумулятивная характеристика качества маршрута введена в разделе 6. В разделе 7 предложена модель с множественной пересылкой. В разделе 8 обсуждается практическое применение линейных диофантовых моделей.

1. Восстановление соединений в сети MPLS

Маршрутизатор MPLS (далее — маршрутизатор) выполняет коммутацию на основе меток [2]. Каждый пакет снабжается стеком меток, который размещается между заголовками пакетов сетевого и транспортного уровней и определяет принадлежность пакета к классу эквивалентности пересылки. Пакеты соединения пересылаются по коммутируемому метками маршруту, состоящему из входного, транзитных и выходного маршрутизаторов. При пересылке пакета они извлекают, заменяют и/или добавляют метки для управления маршрутом.

Маршрутизация может быть последовательной или явной. В первом случае каждый транзитный маршрутизатор по метке выходного маршрутизатора локально выбирает следующий. Во втором случае входной маршрутизатор сразу задает маршрут в виде начального стека меток.

Разрывы соединений вызываются отказами маршрутизаторов или линий связи. Они распространены на практике, существенно снижают надежность сети и делают задачу восстановления соединений актуальной. Кроме собственно поиска маршрутов-кандидатов методы ее решения должны учитывать дополнительные критерии качества [9]. Например, некоторые маршруты могут не подходить из-за загруженности линий связи.

В общий механизм восстановления MPLS [3] входит задача поиска маршрута, которая вычислительно трудоемка вследствие большого числа возможных маршрутов. Она решается или непосредственно после разрыва соединения (исправление маршрута), или заранее — при установлении соединения (резервирование маршрута). Известны два базовых метода: поиск маршрута между транзитными маршрутизаторами; поиск маршрута между входным и выходным маршрутизаторами соединения.

В первом из перечисленных методов строится маршрут, локально обходящий разрыв. Обеспечивается максимальное сохранение предыдущего маршрута и быстрое переключение соединения. Среди недостатков выделяют ухудшение характеристик маршрута после нескольких восстановлений, так как оптимизация при выборе маршрута выполняется локально.

Во втором методе строится маршрут, не пересекающийся с исходным. Обеспечивается построение качественных маршрутов независимо от числа ранее выполненных восстановлений. Среди недостатков этого метода выделяют вычислительную трудоемкость, зависящую от размера сети.

Алгоритм SLSP комбинирует оба базовых метода поиска маршрута [4]. Исходный маршрут разбивается на частично перекрывающиеся домены с собственными входным и выходным маршрутизаторами. Для каждого домена строится один или более резервных маршрутов. При разрыве соединения поврежденный домен заменяется резервным маршрутом.

В алгоритме SLSP используется модель сети MPLS в виде графа $\Omega(N, L)$ с множествами узлов N и линий связи L. Для каждого входного маршрутизатора домена поиск маршрута состоит из трех этапов.

1. Построить множество CY простых циклов графа $\Omega(N, L)$. Считается, что топология сети подвержена слабым изменениям и CY можно использовать для построения нескольких маршрутов восстановления.

2. Для каждого домена s текущего маршрута r выбрать простые циклы $CY_{rs} \subset CY$, каждый полностью включает s. Они определяют маршруты-кандидаты (исключением из цикла текущего маршрута).

3. Из элементов CY_{rs} выбрать оптимальный в соответствии с заданными критериями качества на основе характеристик линий связи.

Среди недостатков отметим, что задача нахождения простых циклов (этапы 1 и 2) является NP-сложной [10]. Число циклов для CY может расти экспоненциально с увеличением размера сети [10], а они определяют множество неизвестных задач ЦЛП для выбора оптимального маршрута (этап 3). В общем случае задача ЦЛП является NP-сложной [11].

Для уменьшения числа маршрутов-кандидатов за счет рационального (качество маршрута) и эффективного (вычислительная трудоемкость) построения нами предлагаются линейные диофантовые модели, заменяющие исходную модель алгоритма SLSP.

2. Однородные системы неотрицательных линейных диофантовых уравнений

Рассматриваемый класс диофантовых систем определен в работе [12] на основе взаимосвязи с контекстно-свободными (КС) грамматиками. В статье используется их алгебраическое определение. Обзор линейных диофантовых уравнений общего вида приведен в работе [11].

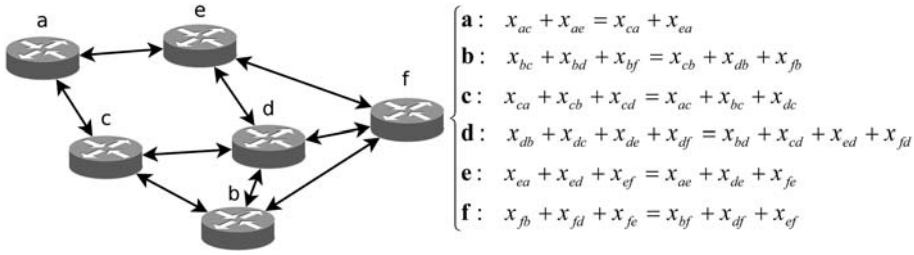


Рис. 1. Пример модели топологии сети

Введем индексное разбиение $I^{n,m} = \{I_0, I_1, ..., I_n\}$, где $I_k \subseteq N_m = \{1, ..., m\}$, $\bigcup_{k=0}^{n} I_k = N_m$, I_0 может быть пусто, $I_k \neq \varnothing$ для $k \neq 0$ и $I_k \cap I_j = \varnothing$ для $k \neq j$. Определим матрицу разбиения $E(I^{n,m}) \in \{0, 1\}^{n \times m}$: если $i \in I_k$, то $E_{ki} = 1$, иначе $E_{ki} = 0$. Столбцы E для $i \in I_0$ — нулевые, а для $i \in I_k$, $k \neq 0$, — стандартные единичные векторы e_k.

Ассоциированная (с КС-грамматикой) однородная система неотрицательных линейных диофантовых уравнений (одАНЛДУ) имеет вид

$$\sum_{i \in I_k} x_i = \sum_{i=1}^{m} a_{ki} x_i, \; k = 1, 2, ..., n, \qquad (1)$$

где n — число уравнений; m — число неизвестных; a_{ki} — элементы матрицы неотрицательных целочисленных коэффициентов, а компоненты решений принимают неотрицательные целые значения. В матричном виде система одАНЛДУ записывается как $E(I^{n,m})x = Ax$.

Ненулевое решение h системы (1) называется неразложимым, если оно не может быть представлено в виде суммы двух ненулевых решений этой же системы. Базисом Гильберта [11] называется множество всех неразложимых решений $H = \{h^{(1)}, ..., h^{(q)}\}$. Он единственен, конечен, а общее решение имеет вид $x = \sum_{s=1}^{q} \alpha_s h^{(s)}$ для произвольных $\alpha_1, \alpha_2, ..., \alpha_q \in Z_+$. Неразложимость решений эквивалентна минимальности, т. е. для любого решения x и произвольного базисного решения $h \in H$, $h \neq x$, неверно, что $x \leqslant h$ (покомпонентное сравнение векторов).

3. Модель топологии

Исходной в предлагаемой нами иерархии вложенных моделей выступает модель топологии сети MPLS. В алгоритме SLSP топология моделируется неориентированным графом $\Omega(N, L)$. Будем учитывать направленность пересылки, используя орграф $\Gamma(N, L)$, где N и L — множества маршрутизаторов и направленных линий связи соответственно.

На основе матрицы инцидентности $E(I^{n,m})$ – $E(J^{n,m})$ определим модель

$$\sum_{i \in I_k} x_i = \sum_{j \in J_k} x_j, \; I_k \cap J_k = \varnothing, \; k = 1, 2, ..., n, \qquad (2)$$

в которой маршрутизаторам соответствуют уравнения, линиям связи — неизвестные. Система (2) известна в теории потоков и определяет циркуляцию в орграфе [11, 13]. Дуга l выходит из вершины k, если $l \in I_k$, и входит в вершину j, если $l \in J_j$. Любое базисное решение h соответствует простому контуру в орграфе и наоборот, причем $h \in \{0, 1\}^m$ (значение 1 означает вхождение дуги в контур).

Пример сети MPLS и модели топологии представлены на рис. 1. Все возможные контуры для множества CY определяются БГ из 35 решений.

Модель (2) позволяет находить множество всех простых контуров CY (этап 1 в алгоритме SLSP), применяя алгоритмы нахождения БГ (НБГ). Если не требуется получения всех контуров, то находится часть БГ.

4. Модель с фиксированным соединением

Пусть исходный маршрут соединения r разбит на домены. Будем строить маршрут восстановления для домена $s = (u, v)$ с входным и выходным маршрутизаторами u и v. Для этого в орграфе $\Gamma(N, L)$ удалим все дуги, которые входят в u или выходят из v, удалим внутренние вершины и дуги исходного маршрута и добавим фиктивную дугу (v, u).

В полученном орграфе Γ_{rs}^{uv} любой контур, содержащий фиктивную дугу (v, u), определяет маршрут между u и v (получается исключением дуги (v, u) из контура). Таким образом, задача построения маршрутов-кандидатов сводится к нахождению простых контуров с дугой (v, u), т. е. в формуле (2) $x_{vu} = 1$. Они формируют множество CY_{rs}, которое получается на этапе 2 алгоритма SLSP без выполнения этапа 1. Как и в предыдущей модели, CY_{rs} может быть найдено с помощью алгоритма НБГ для модели (2).

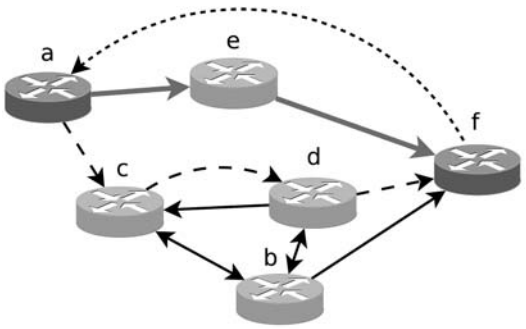
Пример сети MPLS, соответствующая модель с фиксированным соединением и фрагмент БГ пред-

ставлены на рис. 2. Текущий маршрут **a** → **e** → **f** выделен жирными линиями. БГ содержит девять решений, в четырех из которых $x_{fa} = 1$. Решение $h^{(4)}$: $x_{fa} = x_{ac} = x_{cd} = x_{df} = 1$ определяет маршрут-кандидат **a** → **c** → **d** → **f** (отмечен штриховыми линиями).

5. Модель с характеристиками линий связи

В соответствии с [9] будем считать, что значения характеристик линий связи принадлежат одному из небольшого числа классов, которые можно упорядочить по ухудшению, например, загруженность (слабая, средняя, высокая), приоритет (высокий, нормальный, низкий), число промежуточных устройств. Таким образом, компактно отражаются затраты на передачу данных по маршруту.

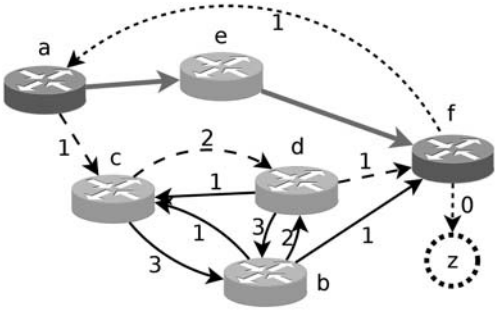
Выберем некоторую характеристику и дополним предыдущую модель с фиксированным соединением r и доменом $s = (u, v)$. Пусть каждому классу характеристики соответствует целое положительное число, указывающее порядок ухудшения. Нулевое значение указывает на отсутствие линии связи, единичное значение — типичная характеристика, значения два и более — коэффициент ухудшения при включении линии в маршрут. Пусть в орграфе топологии сети значения характеристик линий связи соответствуют весам дуг a_{ki} (дуга i входит в вершину k).



$$\begin{cases} \mathbf{a}: x_{ac} = x_{fa} \\ \mathbf{b}: x_{bc} + x_{bd} + x_{bf} = x_{cb} + x_{db} \\ \mathbf{c}: x_{cb} + x_{cd} = x_{ac} + x_{bc} + x_{dc} \\ \mathbf{d}: x_{db} + x_{dc} + x_{df} = x_{bd} + x_{cd} \\ \mathbf{f}: x_{fa} = x_{bf} + x_{df} \end{cases}$$

	x_{ac}	x_{fa}	x_{bc}	x_{bd}	x_{bf}	x_{cb}	x_{db}	x_{cd}	x_{dc}	x_{df}
$h^{(1)}$	1	1	0	0	1	0	1	1	0	0
$h^{(2)}$	1	1	0	1	0	1	0	0	0	1
$h^{(3)}$	1	1	0	0	1	1	0	0	0	0
$h^{(4)}$	1	1	0	0	0	0	0	1	0	1

Рис. 2. Пример модели с фиксированным соединением



$$\begin{cases} \mathbf{a}: x_{ac} = x_{fa} \\ \mathbf{b}: x_{bc} + x_{bd} + x_{bf} = 3x_{cb} + 3x_{db} \\ \mathbf{c}: x_{cb} + x_{cd} = x_{ac} + x_{bc} + x_{dc} \\ \mathbf{d}: x_{db} + x_{dc} + x_{df} = 2x_{bd} + 2x_{cd} \\ \mathbf{f}: x_{fa} + x_{fz} = x_{bf} + x_{df} \end{cases}$$

	x_{ac}	x_{fa}	x_{bc}	x_{bd}	x_{bf}	x_{cb}	x_{db}	x_{cd}	x_{dc}	x_{df}	x_{fz}
$h^{(1)}$	1	1	0	3	0	1	0	0	0	6	5
$h^{(2)}$	1	1	0	2	1	1	0	0	0	4	4
$h^{(3)}$	1	1	0	1	2	1	0	0	0	2	3
$h^{(4)}$	1	1	0	0	6	0	2	1	0	0	5
$h^{(5)}$	1	1	0	0	0	0	0	1	0	2	1
$h^{(6)}$	1	1	0	0	3	0	1	1	0	1	3
$h^{(7)}$	1	1	0	0	3	1	0	0	0	0	2

Рис. 3. Пример модели с характеристиками линий связи

Пусть орграф $\Gamma_{rs}^{uv}(A)$ с весами $A = (a_{ki})_{k \in N, i \in L}$ получен из Γ_{rs}^{uv} (см. раздел 4) добавлением фиктивной вершины z и дуги (v, z). Веса дуг (v, u) и (v, z) равны 1 и 0 соответственно. Рассмотрим соответствующую орграфу $\Gamma_{rs}^{uv}(A)$ систему одАНЛДУ, где x_i будут интерпретироваться как затраты на передачу данных по маршруту (оценка качества):

$$\begin{cases} x_{vu} + x_{vz} = \sum\limits_{i = J_v} a_{vi} x_i; \\ \sum\limits_{i \in I_w} x_i = \sum\limits_{i \in J_w} a_{wi} x_i,\ I_w \cap J_w = \varnothing, \\ w \in N \backslash \{v\}. \end{cases} \quad (3)$$

Пусть получено базисное решение x с $x_{vu} = 1$. Соответствующий маршрут-кандидат получается удалением дуг (v, u) и (v, z). В модели (3) величина x_{vz} определяет качественную характеристику маршрута как функцию от характеристик линий связи. Эта характеристика может быть использована как критерий отсева неэффективных маршрутов в реализующем диофантову модель (3) алгоритме НБГ на этапе 2 алгоритма SLSP. Тем самым упрощается решение задачи ЦЛП на этапе 3. Дальнейший анализ представлен в следующем разделе.

Пример сети MPLS, соответствующая модель с характеристиками линий связи и фрагмент БГ представлены на рис. 3. В ка-

честве характеристики выбрана нагрузка линий (1, 2 или 3). Маршрутизатор **b** находится в наиболее нагруженном участке сети ($a_{b,cb} = a_{b,db} = 3$), маршрутизатор **d** — в средненагруженном ($a_{d,cd} = a_{d,bd} = 2$), остальные линии связи работают с нормальной нагрузкой (вес дуг равен 1). БГ содержит 32 решения, для семи из которых $x_{fa} = 1$. Решение $h^{(5)}$: $x_{fa} = x_{ac} = x_{cd} = x_{fz} = 1$, $x_{df} = 2$ определяет маршрут-кандидат **a** → **c** → **d** → **f** с минимальным значением $x_{fz} = 1$.

6. Кумулятивная характеристика маршрута

Пусть маршрут из u в v в орграфе $\Gamma_{rs}^{uv}(A)$ состоит из вершин $u = w_0, w_1, ..., w_l = v$. Начиная с вершины u, рассмотрим вложенные маршруты (u, w_k), вычисляя их *кумулятивную характеристику* $C_u^{w_k}$ следующим образом. Полагаем $C_u^u = 1$, тогда величина $C_u^{w_k} = \sum_{i \notin I_{w_k}} a_{w_k i} x_i$ определяет затраты на маршруте (u, w_k). В силу модели (3) она должна распределяться по исходящим из w_k дугам маршрута: $C_u^{w_k} = \sum_{i \in I_{w_k}} x_i$. Таким образом, $x_i \geqslant 1$ для дуг $i \in I_{w_k}$ маршрута. Эти дуги входят в следующие вершины маршрута, распределяя затраты, накопленные для транзитной вершины w_k в дугах x_i. При достижении конечной вершины v значение C_u^v распределяется по дугам (v, u) и (v, z), что дает $C_u^v = x_{vu} + x_{vz} = \sum_{i \notin I_v} a_{vi} x_i$. Поскольку дуга x_{vu} единственная и $C_u^u = 1$, то $x_{vu} = 1$, а кумулятивная характеристика исходного маршрута равна $C_u^v = x_{vz} + 1$.

Описанный процесс можно представить в рекуррентной форме как

$$C_u^u = 1,\ C_u^{w_k} = \sum_{i \notin I_{w_k}} a_{w_k i} P_i(C_u^{w_i}),\ k = 1, ..., l,\ t < k,$$

причем $\sum_{i \in I_{w_t}} x_i = C_u^{w_i}$ и $P_i(C_u^{w_i}) = x_i$ определяют зависимость от качества предыдущих маршрутов, проходящих через дугу $i \in I_{w_t}$.

Таким образом, начиная с исходной вершины, в C_u^w накапливаются произведения линейных комбинаций характеристик вложенных маршрутов. Множители соответствуют вершинам маршрута, а линейные комбинации составляются по дугам, исходящим из этих вершин. В моделях (2) и (3) маршрут может состоять только из одного пути в орграфе (нет разветвлений), в этом случае $C_u^v = a_{w_1 i_1} a_{w_2 i_2} ... a_{w_v i_v}$. Общий случай представлен в разделе 7.

Если в маршруте присутствует неэффективная линия связи i (значение $a_{wi} > 1$), то оценка затрат x_i по дуге i увеличивается в a_{wi} раз. Это соответствует принципу "слабого звена", когда присутствие неэффективной линии связи приводит к существенному ухудшению качества маршрута. Таким образом, при реализации линейной диофантовой модели достаточно, чтобы алгоритм НБГ искал только маршруты, удовлетворяющие пороговому значению C_u^v. В силу свойства минимальности элементов базиса Гильберта (см. раздел 2) базисное решение x определяет маршрут с минимальным значением затрат x_i хотя бы для одной дуги i маршрута.

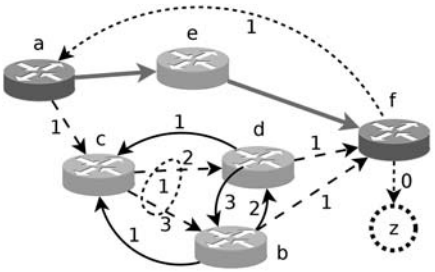
7. Модель с множественной пересылкой

Физический интерфейс маршрутизатора может представлять собой несколько логических линий связи, т. е. пересылка пакета выполняется сразу нескольким маршрутизаторам. Такой механизм называется множественной пересылкой [14], а маршрут объединяет все пути, которыми следуют пакеты.

Дополним модель (3) включением в матрицу A столбцов множественной пересылки (дополнительно к столбцам линий связи — одиночная пересылка). Если маршрутизатор w выполняет множественную пересылку к маршрутизаторам $y_1, y_2, ..., y_l$, то добавляется столбец $i = (w; y_1, y_2, ..., y_l)$, в котором ненулевыми будут только элементы, соответствующие $y_1, y_2, ..., y_l$. Их значения определяются характеристиками линий связи от w до $y_1, y_2, ..., y_l$.

Модель описывает маршруты из модели (3) и маршруты множественной пересылки. Первые соответствуют путям из u в v в орграфе, а последние — наборам путей из u в v. Если затраты на множественную пересылку существенно меньше затрат на одиночную пересылку, то соответствующей последней столбец из A можно исключить.

Если множественная пересылка не используется, то кумулятивная характеристика C_u^v равна произведению характеристик a_{ki} всех линий связи маршрута (см. раздел 6), а значит, порядок появления неэффективной линии связи не имеет значения. При использовании множественной пересылки затраты распределяются по исходящим линиям



$$\begin{cases} \mathbf{a}: x_{ac} = x_{fa} \\ \mathbf{b}: x_{bc} + x_{bd} + x_{bf} = 3x_{cb} + 3x_{db} + x_{(c;b,d)} \\ \mathbf{c}: x_{cb} + x_{cd} + x_{(c;b,d)} = x_{ac} + x_{bc} + x_{dc} \\ \mathbf{d}: x_{db} + x_{dc} + x_{df} = 2x_{bd} + 2x_{cd} + x_{(c;b,d)} \\ \mathbf{f}: x_{fa} + x_{fz} = x_{bf} + x_{df} \end{cases}$$

	x_{ac}	x_{fa}	x_{bc}	x_{bd}	x_{bf}	x_{cb}	x_{db}	$x_{(c;b,d)}$	x_{cd}	x_{dc}	x_{df}	x_{fz}
$h^{(1)}$	1	1	0	3	0	1	0	0	0	0	6	5
$h^{(2)}$	1	1	0	2	1	1	0	0	0	0	4	4
$h^{(3)}$	1	1	0	1	2	1	0	0	0	0	2	3
$h^{(4)}$	1	1	0	0	4	0	1	1	0	0	0	3
$h^{(5)}$	1	1	0	1	0	0	0	1	0	0	3	2
$h^{(6)}$	1	1	0	0	6	0	2	0	1	0	0	5
$h^{(7)}$	1	1	0	0	3	0	1	0	1	0	1	3
$h^{(8)}$	1	1	0	0	3	1	0	0	0	0	0	2
$h^{(9)}$	1	1	0	0	1	0	0	1	0	0	1	1
$h^{(10)}$	1	1	0	0	0	0	0	0	1	0	2	1

Рис. 4. Пример модели с множественной пересылкой

связи ($\sum_{i \in I_k} x_i$) и C_u^v зависит от места появления неэффективной линии связи в маршруте.

Пример сети MPLS, модель с множественной пересылкой и фрагмент БГ представлены на рис. 4. Маршрутизатор *c* допускает множественную пересылку (*c*; *b*, *d*) с характеристикой линии связи 1 (на рис. 4 дуги множественной пересылки обведены пунктирной линией). БГ содержит 39 решений, для 10 из которых $x_{fa} = 1$. Решение $h^{(9)}$: $x_{fa} = x_{ac} = x_{(c;\,b,\,d)} = x_{bf} = x_{df} = x_{fz} = 1$ дает маршрут-кандидат с множественной пересылкой $\mathbf{a} \to \mathbf{c} \begin{matrix} \to \mathbf{b} \to \\ \to \mathbf{d} \to \end{matrix} \mathbf{f}$ с минимальным значением кумулятивной характеристики $C_a^f = x_{fz} + 1 = 2$.

8. Аспекты применения моделей

Предлагаемые в статье линейные диофантовы модели сети MPLS предназначены для замены исходной модели алгоритма SLSP. Они позволяют реализовать этапы 1 и 2. После получения системы одАНЛДУ решается задача НБГ. При этом достаточно находить базисные решения с кумулятивной характеристикой маршрута, не превышающей порогового значения. Тем самым уменьшается трудоемкость этапа 3 в алгоритме SLSP.

Технология MPLS имеет встроенные механизмы получения значений характеристик линии связи [9], что позволяет получить необходимые коэффициенты модели.

Серьезным препятствием на пути применения диофантовых моделей на практике является трудоемкость задачи НБГ. Для систем одАНЛДУ известны псевдополиномиальные алгоритмы Syntactic [5] и TransSol [6]. В таблице представлены результаты экспериментального исследования потребления времени и памяти для линейных диофантовых систем (среднее значение и процентили Q_5 и Q_{95}), описывающих типичные по размеру сети MPLS ($500 \leqslant n$, $m \leqslant 1500$, $q \leqslant 5000$, $\|A\|_{l_\infty} \leqslant 500$, ЭВМ Intel Хеоп, 2.80 ГГц) [6]. В то же время построение маршрутов восстановления для таких сетей на основе исходной графовой модели алгоритма SLSP требует нескольких минут (ЭВМ SUN Ultra 80,4 × 450 МГц) [4].

Средние значения общего времени и памяти, потребляемых алгоритмами НБГ

Потребляемый ресурс	Алгоритм НБГ	
	Syntactic	TransSol
Память, Кбайт	15259 (12345—18482)	14280 (11800—15958)
Время решения, с	15,18 (11,41—19,75)	13,88 (9,83—17,61)

Линейные диофантовы модели могут использоваться и для других задач. Так, представляет интерес применение предложенной иерархии моделей для поиска основного маршрута соединения с учетом распределения нагрузки по всей сети — аналогично линейной диофантовой модели маршрутизации в одноранговых (P2P) сетях [15].

Заключение

Представленные в статье линейные диофантовы модели сетей MPLS позволяют уменьшить трудоемкость решения задачи восстановления соединения по сравнению с моделью, используемой в алгоритме Short Leap Shared Protection. Такой подход позволяет представить множество маршрутов-кандидатов в виде решений, принадлежащих БГ линейных диофантовых систем специального вида. Предложенная мера качества маршрута — кумулятивная характеристика — может быть использована для уменьшения числа исследуемых маршрутов-кандидатов, что уменьшает размерность задачи

ЦЛП поиска оптимального маршрута восстановления. С помощью вычислительного эксперимента показано, что авторские псевдополиномиальные алгоритмы нахождения базиса Гильберта позволяют решать задачу восстановления соединения за приемлемое на практике время.

Список литературы

1. **Васенин В. А., Жижченко А. Б.** Алгоритмическое и программное обеспечение Интернет следующего поколения // Информационное общество. 2005. № 1. С. 56—65.

2. **Rosen E., Viswanathan A., Callon R.** Multiprotocol Label Switching Architecture. RFC 3031 (Proposed Standard). 2001. URL: http://www.ietf.org/rfc/rfc3031.txt

3. **Sharma V., Hellstrand F.** Framework for Multi-Protocol Label Switching (MPLS)-based Recovery. RFC 3469 (Informational). 2003. URL: http://www.ietf.org/rfc/rfc3469.txt

4. **Но Р.-Н., Mouftah H. T.** Reconfiguration of spare capacity for MPLS-based recovery in the internet backbone networks // IEEE/ACM Trans. Netw. 2004. Vol. 12, N 1. P. 73—84.

5. **Корзун Д. Ж.** Syntactic Methods in Solving Linear Diophantine Equations // Тр. междунар. семинара Finnish Data Processing Week at the University of Petrozavodsk (FDPW'2004): Advances in Methods of Modern Information Technology. Петрозаводск: Изд-во ПетрГУ. 2005. Vol. 6. P. 151—156.

6. **Кулаков К. А., Корзун Д. Ж., Богоявленский Ю. А.** Итеративный алгоритм нахождения базиса Гильберта однородных линейных диофантовых систем, ассоциированных с контекстно-свободными грамматиками // Вестник Санкт-Петербургского университета. 2008. Сер. 10. Вып. 2. С. 73—84.

7. **Богоявленский Ю. А., Корзун Д. Ж., Кулаков К. А., Крышень М. А.** Проект Web-SynDic: Система удаленного решения линейных диофантовых уравнений в неотрицательных целых числах // Матер. междунар. конф. "Развитие вычислительной техники в России и странах бывшего СССР: история и перспективы". Т. 1. Петрозаводск: Изд-во ПетрГУ, 2006. С. 136—145.

8. **Кулаков К. А.** Генерация систем неотрицательных линейных диофантовых уравнений // Матер. междунар. конф. "Развитие вычислительной техники в России и странах бывшего СССР: история и перспективы". Т. 2. Петрозаводск: Изд-во ПетрГУ, 2006. С. 58—65.

9. **Wang, D., Li G.** Efficient distributed bandwidth management for MPLS fast reroute // IEEE/ACM Trans. Netw. 2008. Vol. 16, N 2. P. 486—495.

10. **Mateti P., Deo N.** On Algorithms for Enumerating All Circuits of a Graph // SIAM J. Comput. 1976. Vol. 5. N 1. P. 90—99.

11. **Схрейвер А.** Теория линейного и целочисленного программирования. М.: Мир, 1991. Т. 2. 342 с.

12. **Богоявленский Ю. А., Корзун Д. Ж.** Общий вид решения системы линейных диофантовых уравнений, ассоциированной с контекстно-свободной грамматикой // Тр. Петрозаводского государственного университета. Сер. "Прикладная математика и информатика". Вып. 6. Петрозаводск: Изд-во ПетрГУ, 1997. С. 79—94.

13. **Rockafellar R. T.** Network Flows and Monotropic Optimization. John Wiley and Sons, 1984. 616 p.

14. **Ooms D., Sales B., Livens W.** et al. Overview of IP Multicast in a Multi-Protocol Label Switching (MPLS) Environment. RFC 3353 (Informational). 2002. URL: http://www.ietf.org/rfc/rfc3353.txt

15. **Корзун Д. Ж., Гуртов А. В.** Использование линейных диофантовых уравнений для моделирования маршрутизации в самоорганизующихся сетях // Электросвязь. 2006. № 6. С. 34—38.

УДК 004.7

В. В. Наумова, д-р геол.-мин. наук, зав. лаб.,
e-mail: naumova@fegi.ru,
И. Н. Горячев, мл. науч. сотр.,
Дальневосточный геологический институт
Дальневосточного отделения РАН,
г. Владивосток

Разработка системы видеоконференцсвязи отделения наук о Земле РАН

Рассматриваются вопросы проектирования и разработки территориально распределенной Системы видеоконференцсвязи Отделения наук о Земле РАН. Предлагаемый проект основан на современном видении видеоконференцсвязи, которое заключается в создании единого поля коллективного взаимодействия территориально распределенных пользователей.

Ключевые слова: *информатика, современные информационные технологии, видеоконференцсвязь, системы видеоконференцсвязи РАН, интеграция систем видеоконференцсвязи РАН, виртуальные лаборатории, удаленный доступ к аналитическому оборудованию*

Территориальная разобщенность институтов Отделения наук о Земле РАН ставит задачи объединения территориально разрозненных научных сотрудников между собой для интеграции усилий при решении научных задач, для чего используются различные подходы и технологические решения.

Видеоконференцсвязь представляет собой одно из современных решений в этом направлении. По разным источникам 80...85 % информации человек воспринимает зрительно, поэтому видеоконференцсвязь оказывает неоценимую помощь человеку в жизни. В связи с этим применение видеоконференций в науке приносит огромную пользу. Общение с помощью видеоконференцсвязи, когда во время сеанса участники могут не только видеть и слышать друг друга, но и обмениваться данными и обрабатывать их в режиме реального времени, позволяет увеличить эффект восприятия информации до 90 %. По этой причине решения видеоконференцсвязи считаются мощными инструментами повышения эффективности научных исследований и представляют собой качественно новый уровень коммуникаций, объединяя технологические достижения в компьютерной области, телефонии и телевидении.

Системы видеоконференцсвязи РАН

В настоящее время в Российской академии наук существуют или разрабатываются следующие системы видеоконференцсвязи:

- Система видеоконференцсвязи РАН;
- Видеоконференцсвязь Уральского отделения РАН;
- Система видеоконференцсвязи Сибирского отделения РАН;
- Система видеоконференцсвязи Дальневосточного отделения РАН;
- Видеоконференцсвязь Отделения наук о Земле РАН.

Работы по созданию *Системы видеоконференцсвязи РАН* начались в 2009 г. Они ведутся в рамках Программы фундаментальных исследований Президиума РАН "Разработка фундаментальных основ создания научной распределенной информационно-вычислительной среды на основе технологий GRID". Организация-исполнитель — Межведомственный суперкомпьютерный центр РАН. В настоящее время проводятся работы по созданию базового узла видеоконференцсвязи РАН, а также комнат переговоров в здании Президиума РАН. Планируются работы по оснащению оборудованием видеоконференцсвязи конференц-зала Президиума РАН.

Видеоконференцсвязь Уральского отделения РАН (УрО РАН). В Институте математики и механики УрО РАН (ИММ УрО РАН) разработан ряд программных средств, связанных с технологиями передачи медиаданных через сеть Интернет, использующих различные кодеки [3]. Эти программные средства могут использовать различные кодеки для сжатия/распаковки медиаданных в реальном времени, включая разработанные в ИММ кодеки на базе стандарта MPEG-4 [4].

Созданные программные средства ориентированы на:

- проведение видеомостов и видеоконференций через сеть Интернет как открытых (общедоступных), так и закрытых (с персонифицированным доступом);
- организацию широковещательных видеопередач в сети Интернет с видеокамеры "в прямой эфир", в том числе — с применением беспроводного радиоканала (включая спутниковую связь);
- запись с видеокамеры для изготовления лазерного видеодиска доклада, лекции, концерта сразу же после окончания события (без традиционной длительной процедуры сжатия материала после съемки перед записью);
- создание библиотек "видео по запросу" со средствами просмотра записей через сеть Интернет (см., например, библиотеку на сайте http://webTV.uran.ru);
- создание в общественных местах необслуживаемых информационных мониторов (видеопанелей или "информационных киосков"), воспроизводящих по расписанию видеоматериалы, поступающие через сеть Интернет;
- обеспечение работы малых студий кабельного и эфирного телевещания, в том числе необслуживаемых передающих центров, сочетающих ретрансляцию центральных программ с местными программами и рекламными вставками, получаемыми через сеть Интернет;
- обеспечение "прямого эфира" для телекомпаний (включая двустороннюю связь) через радио- и проводные Интернет-каналы (включая передачу через спутниковую связь);
- создание "говорящих веб-страниц" (примером использования является сайт http://webTV.uran.ru);
- создание медиасети на основе разработанных Интернет-ретрансляторов и средств дистанционного управления ими, которая позволит сократить расходы на получение видеоматериалов за счет снижения трафика в сети при массовом (например, миллионы получателей) приеме.

Система видеоконференцсвязи Сибирского отделения РАН (СО РАН). Создана и введена в эксплуатацию первая очередь подсистемы видеоконференцсвязи СО РАН, обеспечивающая возможность регулярной трансляции в пределах СО РАН и далее общеобразовательных программ, значимых мероприятий СО РАН и отдельных его организаций, мероприятий местного и регионального уровней. Проведен рабочий семинар для специалистов из региональных научных центров СО РАН. Создан опорный узел подсистемы в Новосибирске, узел видеоконференцсвязи Президиума СО РАН, а также региональные узлы в Бурятском, Иркутском, Кемеровском, Красноярском, Омском, Томском, Тюменском и Якутском научных центрах СО РАН. Создан корпоративный медийный портал СО РАН, внутри которого будет осуществляться потоковое мультимедийное вещание. Выполнены работы по проектированию доукомплектования и расширения возможностей узла Президиума СО РАН и модернизации опорного узла. Работа выполнялась при поддержке Программы интеграционных фундаментальных исследований СО РАН (заказной проект № 3), программ Приборной комиссии СО РАН, программы государственной поддержки ведущих научных школ Российской Федерации (проект НШ-9886.2006.9) [5].

В рамках работ по Целевой программе Дальневосточного отделения РАН (ДВО РАН) "Информационно-телекоммуникационные ресурсы ДВО РАН" в 2006 г. построена *Система видеоконференцсвязи ДВО РАН* (СВКС ДВО РАН) [2].

При проектировании Системы были поставлены следующие основные задачи:

- реализация как передачи и приема видео- и аудиосигналов, так и возможность качественного показа графических изображений и презентаций;
- проведение видеоконференций между институтами и организациями Дальневосточного отделения РАН и высшими учебными заведениями Дальнего Востока, а также другими научными и образовательными организациями России и мира (двухсторонние, коллективные);
- организация прямой трансляции в сеть Интернет региональных, Всероссийских и международных конференций и мероприятий, проводимых Дальневосточным отделением РАН;
- возможность записи сеансов видеоконференцсвязи для последующей трансляции в сеть Интернет.

Важным принципиальным решением, которое принято при проектировании СВКС ДВО РАН, было оборудование конференц-залов институтов программно-аппаратными комплексами видеоконференцсвязи. Именно это решение дает возможность использовать Систему в научных целях.

В состав СВКС ДВО РАН входят следующие компоненты:

- устройство многоточечной связи MCU;
- программно-аппаратные комплексы видеоконференцсвязи, установленные в конференц-залах всех научных центров ДВО РАН в городах: Владивосток, Хабаровск, Магадан, Петропавловск-Камчатский, Благовещенск, Южно-Сахалинск;
- мобильный программно-аппаратный комплекс видеоконференцсвязи.

Оборудование базового узла Системы включает в себя сервер видеоконференцсвязи — Codian MCU-4210 (допускает до 20 точек соединения со скоростью до 2 Мбит/с) и IP VCR 2210 — устройство записи и трансляции. Системы видеоконференций, звукоусиления, видеопроекций — основные компоненты оснащения конференцзалов. Решения направлены на создание сбалансированного комплекса видео- и аудиокомпонентов для оперативной и комфортной работы. Терминальное оборудование залов включает в себя кодеки видеоконференцсвязи Polycom VSX 8400.

Проектирование Системы видеоконференцсвязи Отделения наук о Земле РАН

Все созданные и создаваемые сегодня в Российской академии наук системы видеоконференцсвязи в основном предназначены для использования в целях оптимизации управления. Поэтому в этих системах терминальные устройства видеоконференцсвязи располагаются в конференц-залах Президиумов РАН, Президиумов региональных научных центров, Президиумов научных центров, в кабинетах руководителей различных уровней. Однако современные технологии видеоконференцсвязи позволяют использовать видеоконференцсвязь не только для оптимизации управления, но и для решения научных и научно-организационных задач.

Современное видение видеоконференцсвязи заключается в том, что все голосовые и видеосистемы конференцсвязи развиваются встречными курсами, образуя, в конечном счете, единое поле коллективного взаимодействия сотрудников. Неважно, где находится участник конференции и какое абонентское устройство в данный момент имеется у него под рукой: концепция VC2 предполагает распространение возможностей конференцсвязи на любое пользовательское оборудование, в том числе ПК, стационарные и мобильные телефоны, индивидуальные и групповые видеоконференц-терминалы. Изменения касаются и методологии проведения конференций: от заранее планируемых сессий (всем участникам надлежит быть в определенное время на рабочих местах) — к сеансам связи "по требованию" в любое время, в любом месте и с любыми участниками.

Новая концепция и новые технологические возможности видеоконференцсвязи позволяют сформулировать новые задачи для обеспечения научных исследований, которые можно реализовать в настоящее время.

При проектировании Системы видеоконференцсвязи Отделения наук о Земле РАН решены следующие задачи:

- сформулированы основные задачи, которые должна решать Система;
- проведен анализ каналов связи выхода в сеть Интернет для центральных и региональных сетей РАН;
- проведен анализ терминального оборудования видеоконференцсвязи в институтах Отделения наук о Земле РАН;
- разработана топология и структура системы;
- предложены основные программно-аппаратные компоненты Системы;
- выделены этапы построения Системы, при этом сформулированы основные задачи каждого этапа.

Основные задачи Системы видеоконференцсвязи Отделения наук о Земле (ОНЗ) РАН:

- проведение многоточечных научных видеоконференций;
- осуществление активного доступа к видеоконференциям с персональных компьютеров научных сотрудников институтов ОНЗ РАН;
- запись и архивирование видеоконференций;
- трансляция конференций, проводимых институтами ОНЗ РАН в сеть Интернет;
- доступ в виртуальные территориально распределенные группы с ПК сотрудников;

- доступ удаленных клиентов аналитических центров институтов ОНЗ РАН к ПК аналитического оборудования, что дает новые возможности для организации взаимодействия аналитических центров с удаленными клиентами.

Анализ скоростей выхода в сеть Интернет центральных и региональных сетей РАН был необходим, поскольку каналы связей, по которым передается видеоинформация, должны быть достаточно скоростными, т. е. обладать высокой пропускной способностью.

Опорная телекоммуникационная сеть РАН в Московском регионе предоставляет доступ к вычислительным и информационным ресурсам институтов и учреждений РАН, а также уникальным научным установкам со скоростью 1...10 Гбит/с.

Пропускная способность каналов доступа в Интернет в Уральском отделении РАН: 30 Мбит/с для Екатеринбурга; скорость каналов Научные центры УрО РАН—Екатеринбург — 2...4 Мбит/с.

Сеть передачи данных Сибирского отделения РАН обеспечивает следующую пропускную способность каналов доступа в сеть Интернет: 10 Мбит/с — Барнаул, Кемерово и Тюмень; 80 Мбит/с — Иркутск; 40 Мбит/с — Красноярск; 500 Мбит/с — Новосибирск; 30 Мбит/с — Омск; 50 Мбит/с — Томск; 20 Мбит/с — Якутск.

Пропускная способность каналов доступа в сеть Интернет для региональных телекоммуникационных сетей научных центров Дальневосточного отделения РАН 70 Мбит/с во Владивостоке, 40 Мбит/с — в Хабаровске; 3 Мбит/с — в Благовещенске; 5 Мбит/с — в Южно-Сахалинске; 0,5...42 Мбит/с — в Петропавловске-Камчатском; 2 Мбит/с — в Биробиджане, Комсомольске-на-Амуре; 1,6 Мбит/с — в Магадане.

Существующее положение с видеоконференцсвязью в ОНЗ РАН

В настоящее время Отделение наук о Земле насчитывает в своем составе 72 института. В 11 институтах есть терминальные точки видеоконференцсвязи. Ниже представлен список этих институтов (в скобках спецификация оборудования видеоконференцсвязи).

1. г. Москва:
   - Институт физики Земли РАН (*Polycom VSX 6400 Presenter, Polycom V500 Presenter*);
   - Геофизический центр РАН (*Polycom VSX 7000*);
   - Международный институт теории прогноза землетрясений и мат. геофизики РАН (*Polycom VSX 6000*);
   - Институт проблем комплексного освоения недр РАН (*Polycom VSX 7000s*);
   - Геофизическая служба РАН, г. Обнинск (*Polycom VSX 7000s*).
2. Уральское отделение РАН: нет сведений.
3. Сибирское отделение РАН:
   - Институт угля и углехимии СО РАН, г. Кемерово (*ViewPoint 8000, Huawei*).
4. Дальневосточное отделение РАН:
   - Дальневосточный геологический институт ДВО РАН, г. Владивосток (*Polycom VSX 7000*);
   - Институт вулканологии и сейсмологии ДВО РАН, г. Петропавловск-Камчатский (*Polycom VSX 8000*);
   - Институт морской геологии и геофизики ДВО РАН, г. Южно-Сахалинск (*Polycom VSX 7000*);
   - Северо-Восточный комплексный научно-исследовательский институт ДВО РАН, г. Магадан (*Polycom VSX 8000*);
   - Институт геологии и природопользования ДВО РАН, г. Благовещенск (*Polycom VSX 8000*).

Топология и основные компоненты Системы видеоконференцсвязи ОНЗ

Нами предполагается, что Система видеоконференцсвязи ОНЗ РАН будет представлять собой территориально распределенную систему с серией базовых точек (не менее четырех), в которых будет установлено серверное оборудование Системы. Очевидно, что базовые точки должны быть установлены в институтах ОНЗ РАН, расположенных в Москве и в региональных Отделениях РАН, что позволит оптимизировать транспортные информационные потоки видеоконференцсвязи.

Для построения Системы видеоконференцсвязи ОНЗ РАН предложены следующие основные компоненты, включение которых в Систему позволит решить все поставленные задачи:

- устройства многоточечной связи и управления, установленные в базовых региональных узлах;
- программно-аппаратные комплексы видеоконференцсвязи в конференц-залах институтов, в комнатах переговоров и кабинетах руководителей;
- мобильные программно-аппаратные комплексы видеоконференцсвязи;
- программные комплексы видеоконференцсвязи, установленные на ПК сотрудников институтов и аналитических приборов институтов.

Для оборудования базовых узлов Системы нами предложены следующие технические решения:

- сервер видеоконференций — *Polycom RMX-2000* (способен поддерживать от 20 до 80 видеосоединений (портов) по протоколам SIP или H.323);
- устройство записи видеоконференций — *Polycom RSS 2000*;
- решение по организации и управлению видеоконференциями — *Polycom CMA 4000.*

Система управления Системой видеоконференцсвязи ОНЗ РАН

Для любой информационной сети можно сформулировать общий принцип: чем она больше, тем сложнее в управлении. Не являются исключением из этого правила и сети видеоконференцсвязи (ВКС). Более того, если в общие задачи управления сетями обычно не включается требование управления оборудованием, установленным на рабочих местах пользователей, то для сетей ВКС в настоящее время это становится одной из основных функций. Ситуация осложняется тем, что современная архитектура сетей ВКС требует высокой работоспособности от гетерогенных решений, построенных с использованием разнородных технологий на основе оборудования разных производителей [1]. Для обеспечения безопасности и повышения надежности вычислительных сетей используются технологии, получившие название управления сетями — наблюдение за функционированием, тестирование, предотвращение, выявление и устранение сбоев, обеспечение функционирования сетевых сервисов с задаваемым качеством обслуживания.

Применительно к сетям видеоконференцсвязи задачи, предусмотренные моделью управления, должны включать в себя следующие функции:

- обработка ошибок — обеспечение администратора сети необходимыми инструментами для обнаружения сбоев и отказов сетевых и терминальных устройств ВКС, определения их причин и принятия действий по восстановлению;
- управление конфигурацией — отслеживание и настройка конфигурации сетевого программного и аппаратного обеспечения (настройки и состояние отдельных сетевых устройств и сети в целом);
- учет — измерение использования и доступности сетевых ресурсов;
- управление производительностью — измерение производительности сети, сбор и анализ статистической информации о поведении сети для ее поддержания на приемлемом уровне как для оперативного управления, так и для планирования ее развития. Управление производительностью предоставляет возможность: получать уровень загрузки и ошибок сетевых устройств; обеспечивать соответствующий уровень производительности за счет необходимых сетевых ресурсов;
- управление безопасностью — контроль доступа к оборудованию и сетевым ресурсам (с ведением журналов доступа), предотвращение, обнаружение и пресечение несанкционированного доступа.

Использование системы управления серверами видеоконференцсвязи Polycom DMA позволит создать управляемую территориально распределенную Систему видеоконференцсвязи в ОНЗ РАН (рис. 1). Система позволит управлять загрузкой сети видеоконференцсвязи, распределяя соединения точек по нескольким серверам видеоконференцсвязи. Она предоставит единый интерфейс управления сетью серверов, упрощая администрирование, создание и поддержку распределенных конференций любого уровня сложности; использует имеющиеся в сети сервера видеоконференцсвязи для перенаправления вызовов в случае сбоя в одном из элементов сети. Мощная система управ-

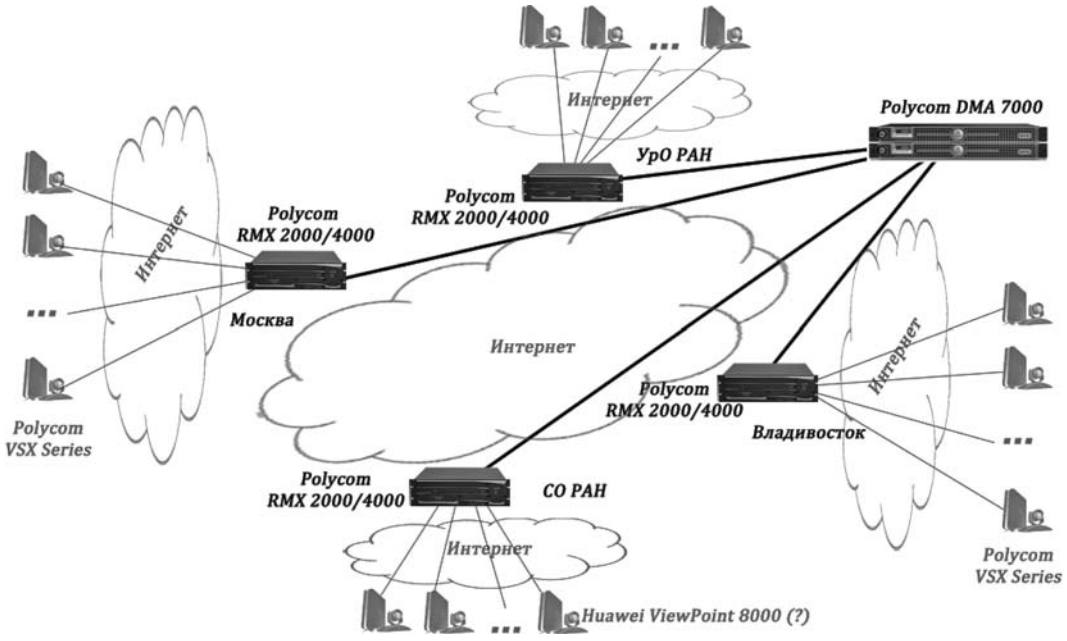


Рис. 1. Общая схема территориально распределенной Системы видеоконференцсвязи Отделения наук о Земле РАН

ления может избавить Систему видеоконференцсвязи ОНЗ РАН с многочисленными территориально распределенными институтами от нагромождения серверных устройств и необходимости каскадирования.

Система управления сетью Системы видеоконференцсвязи ОНЗ РАН

Используя систему управления сетью Системы видеоконференцсвязи Polycom CMA, можно обеспечить видеосвязью личные рабочие помещения, рабочие столы, конференц-залы и мобильные устройства с помощью единого масштабируемого приложения. Централизованно управляемые и распределяемые через Polycom CMA Server клиенты Polycom CMA Desktop являются частью той же среды, которая включает и системы класса Telepresence, и традиционные системы видеоконференцсвязи, создавая тем самым основу для получения мощного решения, охватывающего все виды и типы клиентского оборудования.

Программное обеспечение Polycom CMA Desktop — клиентское приложение для персональных компьютеров, обеспечивающее высококачественную видео- и голосовую связь, а также основанный на стандартах совместный доступ к информационным ресурсам. Простой и дружественный интерфейс CMA Desktop дает возможность корпоративному пользователю начать сеанс видеосвязи с коллегами в любом месте и в любое время, просто выбрав курсором нужный контакт и нажав кнопку мыши.

Используя возможности Polycom CMA, мы можем обеспечить решение следующих новых задач по организации:

- совместной работы в режиме реального видео территориально распределенных групп научных сотрудников;
- доступа удаленных клиентов аналитических центров институтов к операторам и экранам компьютеров аналитического оборудования, что даст новые возможности для организации взаимодействия аналитических центров с территориально удаленными научными сотрудниками.

Оснащение институтов ОНЗ РАН программно-аппаратными устройствами видеоконференцсвязи

Мы считаем, что одним из важных принципиальных решений при построении научных систем видеоконференцсвязи является оборудование конференц-залов институтов программно-аппаратными комплексами видеоконференцсвязи. Именно это решение дает возможность использовать Систему в научных целях. Системы видеоконференций, звукоусиления, видеопроекций — основные компоненты оснащения конференц-залов. Решения должны быть направлены на создание сбалансированного комплекса видео- и аудиокомпонентов для оперативной и комфортной работы. Терминальное оборудование залов для организации видеоконференцсвязи должно быть реализовано на базе технологической платформы, включающей в себя модуль кодека видеоконференцсвязи с подключенным к нему специализированным оборудованием. В качестве терминального оборудования видеоконференцсвязи предлагается кодек Polycom HDX 9000.

Стандартизация оборудования видеоконференцсвязи

Важным вопросом при построении Системы видеоконференцсвязи ОНЗ РАН является стандартизация оборудования. В результате проведенного тестирования Polycom выделил версии оборудования видеоконференцсвязи для обеспечения полной совместимости с системой управления сетью видеоконференцсвязи Polycom DMA. Ниже приведены полученные результаты тестирования.

Система Polycom DMA совместима со следующими версиями оборудования видеоконференцсвязи:

Polycom PathNavigator™ 7.00. 12;
Polycom ReadiManager™ SE200 3.00.06;
Polycom RMX 2000™ 4.0.0.78;
Polycom RMX 4000™ 5.0.0.45;
Polycom CMA™ 5000 4.0.1/4.1.0;
Polycom CMA Desktop 4.0.1/4.1.0;
Polycom HDX 9004 2.5.0.2/2.5.1;
Polycom HDX 9001 2.5.0.2/2.5.1;
Polycom HDX 8000 2.5.0.2/2.5.1;
Polycom HDX 4000 2.5.0.2/2.5.1;
Polycom VSX™ 8000 9.0.5;
Polycom VSX 7000 9.0.5;
Polycom VSX 6000 9.0.5;
Polycom VSX 5000 9.0.5;
Polycom VSX 3000 9.0.5;
Polycom VSX 500 9.0.5;
Polycom ViewStation® FX 6.0.5;
Polycom iPower™ 9000 6.2.1208;
Polycom PVX™ 8.0.2;
Polycom VS 7.5.4;
Polycom DST B5 V2.0;
Polycom DST K60 V2.0.1;
Tandberg 6000 MXP F7.2;
Tandberg 150 MXP L5.1;
Lifesize Team/Room/Express 4.0.11;
Sony PCS1 3.41;
Sony G50 2.70;
Sony XG80 2.02;
Aethra VegaStar Gold 6.00.00.0049;
Aethra X3 10.7.32;
Aethra X7 12.1.7.

Мы рекомендуем институтам Отделения наук о Земле РАН при покупке оборудования видеоконференцсвязи учитывать эту информацию, что в дальнейшем позволит построить в ОНЗ РАН корректно работающую территориально распределенную систему видеоконференцсвязи.

Нами предложены основные этапы построения Системы.

На первом этапе планируется создание двух базовых точек Системы:

- г. Москва, Геофизический центр РАН;
- г. Владивосток, Дальневосточный геологический институт ДВО РАН.

В дальнейшем планируется организация по крайней мере еще двух базовых узлов Системы (в Уральском и Сибирском отделениях РАН).

При внедрении видеоконференцсвязи в текущую деятельность ОНЗ РАН могут быть получены следующие основные результаты:

- Система видеоконференцсвязи ускорит принятие решений по ключевым вопросам, требующим присутствия всего руководящего состава ОНЗ РАН, а также существенно сократит финансовые затраты на их проезд в г. Москву.
- Ежемесячные заседания редколлегий научных журналов в режиме видеоконференцсвязи позволят повысить эффективность обсуждения статей членами редакционных коллегий.
- Повысится уровень научных конференций, проводимых в институтах ОНЗ РАН из-за полученной возможности проводить в режиме видеоконференцсвязи включение докладов из ведущих российских и мировых научных центров и университетов.
- У институтов ОНЗ, проводящих конференции, появится возможность трансляции этих конференций в сеть Интернет в реальном режиме времени, что будет способствовать повышению их уровня.
- У научных сотрудников из удаленных регионов России появится возможность защищать диссертационные работы в ведущих специализированных советах России в режиме видеоконференцсвязи.
- Появится возможность проведения постоянно действующих научных семинаров с участием научных сотрудников из различных регионов страны.
- Режим виртуальных лабораторий позволит более эффективно осуществлять работу по совместным научным проектам сотрудникам из территориально распределенных институтов.
- Удаленный доступ к дорогостоящему уникальному аналитическому оборудованию даст возможность его более эффективного использования.

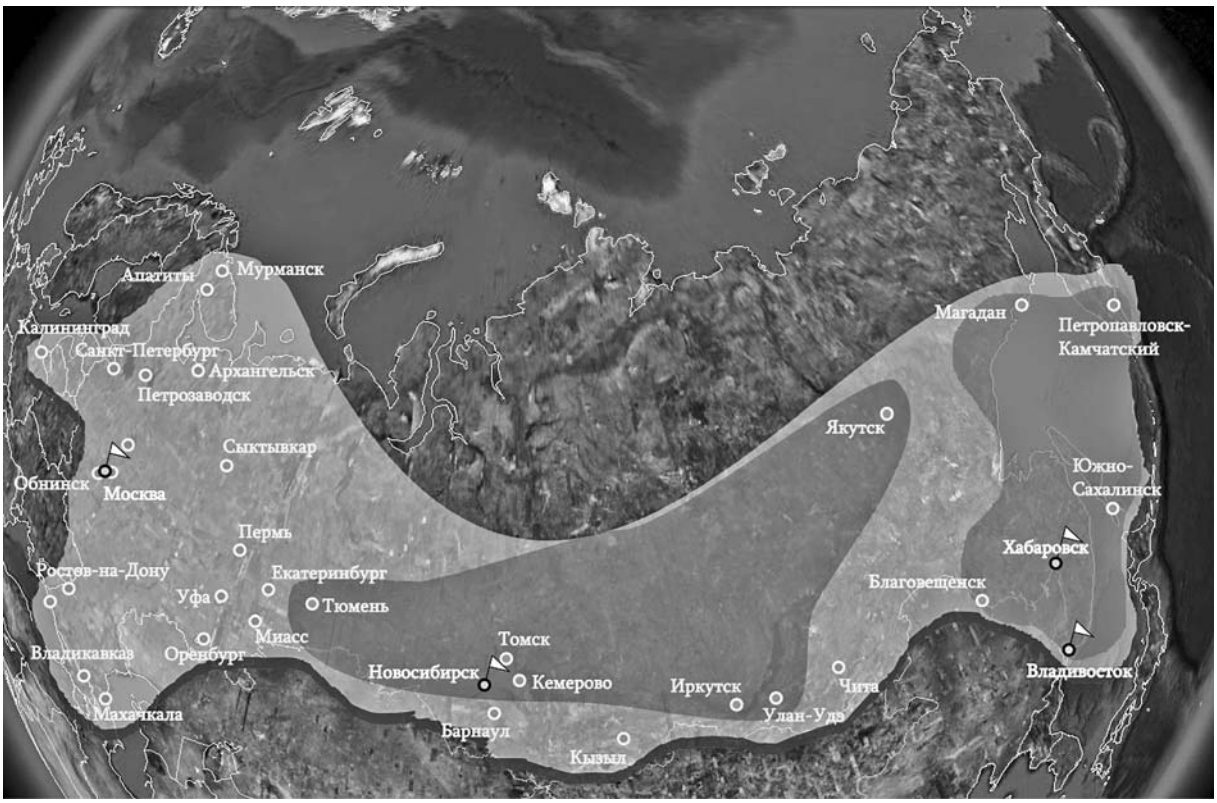


Рис. 2. Виртуальная интеграция систем видеоконференцсвязи РАН. Облаками различной интенсивности цвета (светло-серый — СВКС ОНЗ РАН, темно-серый — СО РАН и ДВО РАН) отмечены территориальные зоны действия систем видеоконференцсвязи ОНЗ РАН (проект), СО РАН и ДВО РАН. Флажками отмечены города РФ, в которых в настоящее время есть серверы видеоконференцсвязи РАН

В настоящее время появляется возможность интеграции всех существующих систем видеоконференцсвязи РАН. Концептуально интегрированная система видеоконференцсвязи РАН может быть сформирована в виде совокупности виртуальных облаков, каждое из которых связано с конкретным региональным отделением РАН или отделением наук РАН, например Отделением наук о Земле. Сами облака в силу своей виртуальности могут быть соединены в любой момент с другими или с группами облаков. Внутри облаков группы могут формироваться в соответствии с тематикой проводимой видеоконференции (или нескольких) в зависимости от уже имеющейся интеграции либо организуемой по необходимому запросу.

Технологически построить подобную интеграцию можно, используя тот же подход, который предлагается при создании Системы видеоконференцсвязи ОНЗ РАН, т. е. с использованием технологий Polycom DMA.

Таким образом, можно получить практически полное покрытие территории РФ виртуальной системой видеоконференцсвязи РАН, которая в настоящее время насчитывает не менее 30 клиентских терминалов в разных городах РФ от Москвы до Петропавловска-Камчатского и которая находится в состоянии активного развития (рис. 2).

Список литературы

1. **Виноградов М. В.** Современные методы и средства управления в сетях видеоконференцсвязи // Вестник связи. 2007. № 6. С. 81—85.
2. **Наумова В. В., Сорокин А. А., Горячев И. Н.** Видеоконференцсвязь —мультимедийный сервис Корпоративной сети Дальневосточного отделения РАН // Информационные технологии. 2009. № 4. С. 66—70.
3. **Прохоров В. В., Косарев В. А.** Программные средства передачи видеоаудиопотоков через Интернет. SIIS2002 // Первый региональный форум "Сибирская индустрия информационных систем", Новосибирск, 2002. URL: http://www-sbras.nsc.ru/win/telecom/forum_2002/prokhorov/prokhorov.htm
4. **Прохоров В. В.** и др. Многофункциональная система интернет-видеосвязи "VIPPHONE" // Тр. Всерос. науч. конф. "Научный сервис в сети Интернет", г. Новороссийск, сентябрь 2004 г. М.: Изд-во МГУ. 2004. С. 262—265.
5. **Шокин Ю. И.** и др. Создание корпоративной системы мультимедийных приложений в сети передачи данных Сибирского отделения РАН // Отчет о деятельности Института вычислительных технологий СО РАН в 2008 г. URL: http://www.ict.nsc.ru/sitepage.php?PageID=454

УДК 004.75; 004.942

Д. А. Сериков, аспирант,
Московский государственный университет
им. М. В. Ломоносова,
e-mail: serd@mexmat.net

Применение механизмов контроля насыщения для разделения ресурсов в распределенной вычислительной среде

Рассматривается подход к планированию ресурсов в распределенной вычислительной среде Grid, основанный на контроле насыщения (Congestion Control). Описывается дискретно-событийная модель процесса диспетчеризации задач на основе дисциплины планирования с контролем насыщения и результаты ее тестирования.

Ключевые слова: *Grid, планирование, Congestion Control*

Введение

Несмотря на быстрый рост производительности отдельных вычислительных установок [1] решить с их помощью целый ряд сложных научно-технических и практически значимых задач в настоящее время не представляется возможным. Одна из главных причин такого положения дел связана с недостатком необходимых для этого объемов вычислительных ресурсов[1], которыми даже такие сверхвысокопроизводительные установки не располагают. Вместе с тем, высокие темпы развития информационно-вычислительных и коммуникационных технологий, сетевой инфраструктуры на основе пакетных коммуникаций создают технические предпосылки для консолидации таких ресурсов. Методология построения подобных распределенных (в том числе географически) информационно-вычислительных комплексов, консолидирующих ресурсы различных организаций для решения сложных задач начала активно развиваться около 10—15 лет назад. В эти годы методологию Grid-вычислений применяли для решения многих научных задач, например, связанных с расшифровкой генома человека [2] или поиска внеземных цивилизаций [3]. В 1997 г. появился первый крупный проект, посвященный использованию ресурсов компьютеров обычных пользователей с помощью сети Интернет для решения исследователь-

[1] Под *ресурсом* понимается средство вычислительной установки (или нескольких), который она может использовать в процессе своей работы (процессорное время, оперативная и дисковая память и т. п.).

ских задач, требующих больших вычислительных ресурсов — проект distributed.net [4].

Согласно трем критериям классического определения Grid-системы, которые предложил один из основоположников данной методологии Ян Фостер [5], *Grid — открытая и стандартизованная компьютерная среда, которая обеспечивает децентрализованное разделение ресурсов и высококачественное обслуживание пользователей в рамках виртуальной организации*. Под виртуальной организацией понимается группа субъектов как отдельных лиц, так и структурных подразделений различных форм собственности, совместно использующих общие ресурсы. Открытость и стандартизация означает тот факт, что система должна строиться на основе стандартных, открытых протоколов и интерфейсов, позволяющих решать такие традиционные задачи, как аутентификация, авторизация, обнаружение ресурсов и управление доступом к ним. Кроме того, система должна координировать использование ресурсов при отсутствии централизованного управления ими. Здесь нужно подчеркнуть некоторое отличие от Grid-системы слабосвязанного кластера как группы компьютеров, объединенных высокоскоростными каналами связи и представляющей с точки зрения пользователя единый аппаратный ресурс. В случае такого кластера речь идет о компьютерах, изначально управляемых из единого центра, в случае Grid — об одноранговой сети с независимыми узлами, которые объединяются в виртуальные организации для решения общих задач. В этом смысле слабосвязанные кластеры не являются Grid-системами. Использование ресурсов в Grid должно осуществляться таким образом, чтобы обеспечивалась должная функциональность, высокое качество обслуживания потенциальных клиентов и защищенность. Обслуживание характеризуется доступностью ресурсов, надежностью работы системы в целом, временем отклика на запрос пользователя, пропускной способностью сетевых каналов и другими, подобными им атрибутами.

В силу многопользовательского характера Grid-инфраструктур необходимым условием, обеспечивающим высокое качество обслуживания, является наличие программного механизма планирования ресурсов (в рамках процесса диспетчеризации — автоматического распределения ресурсов при обслуживании запросов пользователей). Процесс планирования координирует разделение ресурсов между задачами пользователей. При правильной организации процесс диспетчеризации (и планирования) должен требовать от пользователя минимального участия, а именно — пользователь должен лишь запускать задачу и получать результат. Задачу, где и как будет исполняться приложение, диспетчер должен решить самостоятельно, без дополнительного вмешательства со стороны пользователя. Диспетчер является одним из основных компонентов любого Grid-комплекса, а его функциональность является фактором, в значительной степени определяющим производительность распределенных вычислений. По этой причине разработка эффективных алгоритмов планирования для диспетчеров Grid является актуальной задачей. В настоящей работе изложен алгоритм с контролем насыщения, который основан на одноименном алгоритме, который используется в сетевом протоколе TCP [6].

В качестве объекта для исследования алгоритмов планирования рассматривается программный комплекс GridWay [7], как наиболее распространенный в настоящее время диспетчер для Grid-систем. В качестве инфраструктурной основы исследований функциональных возможностей и эффективности механизмов планирования ресурсов для решения вычислительных задач на Grid-среде использовался экспериментальный Grid-полигон [8]. Данный полигон развернут на базе высокопроизводительных вычислительных систем Института механики МГУ (НИИ механики МГУ, Москва), Института проблем информационной безопасности МГУ (ИПИБ МГУ, Москва), Научно-образовательного Центра компьютерного моделирования и безопасных технологий МГУ (НОЦ КМиБТ МГУ, Москва) и Института вычислительной математики и математической геофизики Сибирского отделения РАН (ИВМиМГФ СО РАН, Новосибирск), объединенных с помощью сетей передачи данных МГУ—РАН.

1. Планирование с контролем насыщения

Данный алгоритм планирования основан на алгоритмах контроля насыщения, которые применяются в сетевом протоколе TCP (RFC 2581): алгоритме замедленного старта (Slow Start) и алгоритме предотвращения перегрузки (Congestion Avoidance). Далее подробнее рассмотрим каждый из этих алгоритмов уже применительно к вопросам планирования в Grid-среде.

1.1. Алгоритмы Slow Start и Congestion Avoidance

Для реализации рассматриваемых алгоритмов используются следующие переменные.

- *Размер окна насыщения* (*cwnd*) — это задаваемый диспетчером предел числа задач, которые он может передать на узел до получения подтверждения об их выполнении; начальное значение *cwnd* — 1.
- *Анонсируемое узлом окно* (*rwnd*), определяющее установленный узлом предел задач, которые могут находиться в его очереди. Планированием управляет меньшее из двух значений *cwnd* и *rwnd*.

- *Порог насыщения* (*ssthresh*) используется для определения момента времени, когда следует применять алгоритм замедленного старта или алгоритм предотвращения перегрузки в соответствии с представленными далее описаниями. Начальное значение порога насыщения *ssthresh* равно размеру анонсируемого окна *rwnd* и может быть уменьшено при возникновении насыщения. Алгоритм замедленного старта (Slow Start) используется в тех случаях, когда $cwnd < ssthresh$, а при $cwnd \geqslant ssthresh$ применяется алгоритм предотвращения перегрузки (Congestion Avoidance).
- *Тайм-аут RTO* — период времени, по прошествии которого в случае неполучения от узла подтверждений о выполнении задач, делается вывод о наступлении перегрузки (узел содержит в очереди больше задач, чем он может поставить на выполнение).

Начало доставки задач узлу с неизвестными условиями требует от диспетчера достаточно медленной проверки узла в целях определения доступной "емкости" для того, чтобы избежать перегрузки узла избыточным числом задач. Алгоритм Slow Start используется для решения этой задачи на начальном этапе планирования.

При замедленном старте диспетчер увеличивает размер окна *cwnd* на 1 для каждого успешного подтверждения со стороны узла, свидетельствующего о выполнении задачи. Замедленный старт завершается, когда размер окна насыщения *cwnd* превышает порог *ssthresh* (или становится равным этому порогу).

Предотвращение насыщения (Congestion Avoidance) продолжается до тех пор, пока насыщение наблюдается. Для обновления значений *cwnd* в процессе предотвращения перегрузки ведется подсчет новых задач, которые были выполнены. Когда число выполненных задач достигнет значения *cwnd*, размер окна *cwnd* увеличивается на 1.

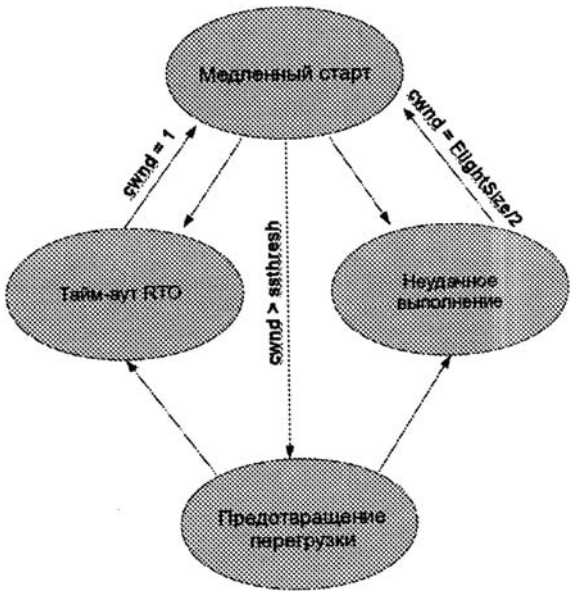


Рис. 1. Схема алгоритма

Когда диспетчер обнаруживает подтверждение выполнения задачи после возникновения тайм-аута *RTO*, для переменной *ssthresh* устанавливается значение

$$ssthresh = \max(FlightSize/2,\ 2).$$

FlightSize показывает число задач, которые запущены на узле (посланы, но не получено подтверждение их выполнения). Более того, при возникновении тайм-аута *RTO* размер окна насыщения *cwnd* устанавливается в значение 1. Следовательно, после тайм-аута *RTO* диспетчер использует замедленный старт для увеличения окна $cwnd = 1$ до нового значения *ssthresh*, после чего снова включается механизм предотвращения перегрузки.

При обнаружении неудачного выполнения задачи для *ssthresh* устанавливается значение

$$ssthresh = \max(FlightSize/2,\ 2),$$

а для *cwnd* значение

$$cwnd = \max(FlightSize/2,\ 1).$$

Схема алгоритма изображена на рис. 1.

1.2. Продолжение планирования после бездействия

При использовании описанных выше алгоритмов на Grid-среде возникает следующий вопрос. Диспетчер после продолжительного бездействия может передать на узел *cwnd* задач, хотя за это время состояние Grid-среды может измениться.

Для разрешения этой ситуации используется замедленный старт для продолжения процесса планирования после сравнительно долгого простоя. Суть реализующего его механизма в том, что когда диспетчер не получает задач в течение времени, превышающего двойной тайм-аут *RTO*, размер окна насыщения *cwnd* уменьшается до значения 1 перед началом планирования. Таким образом, диспетчер устанавливает перед началом планирования для окна *cwnd* значение, равное 1, если диспетчер не отправлял задачи в течение времени, превышающего двойной тайм-аут *RTO*.

1.3. Тайм-аут RTO

В условиях объективно происходящих в Grid-среде постоянных изменений и, как следствие, потенциально возможном различном времени выполнения задач значение тайм-аута *RTO* должно динамически изменяться. Далее представлен алгоритм определения значения тайм-аута *RTO* [6], который необходим в текущем состоянии вычислительной среды.

После получения подтверждения о выполнении задачи измеряется значение *RTT*, равное сумме времени, затраченного на транспортировку задачи, времени ожидания задачи в очереди на узле и времени выполнения задачи. После этого рассчитывается значение *SRTT*, как

$$SRTT = (ALPHA \cdot SRTT) + ((1 - ALPHA)RTT)$$

и на основе этого рассчитывается тайм-аут *RTO*

$$RTO = \\ = \min[UBOUND, \max[LBOUND, (BETA \cdot SRTT)]],$$

где *UBOUND* задает верхний предел значения тайм-аута (например 1 ч), *LBOUND* — нижний предел (например 1 мин), *ALPHA* — весовой фактор (например 0,9), а *BETA* — коэффициент вариаций задержки (например 2,0).

1.4. Результаты тестирования алгоритмов планирования

В таблице представлены сравнительные результаты тестирования различных алгоритмов планирования, в качестве тестовой задачи использовалась задача построения функциональной зависимости по экспериментальным данным.

Алгоритм	Время исполнения, мин
Используемый в GridWay (1600 подзадач по 1 с)	209, 245, 170
Используемый в GridWay (160 подзадач по 15—20 мин)	196
С контролем насыщения (1600 подзадач по 1 с)	82, 80, 83
С контролем насыщения (160 подзадач по 15—20 мин)	199

Результаты тестирования показали, что алгоритм планирования с контролем насыщения лучше справляется с легкими (короткими по времени выполнения) задачами, чем стандартный алгоритм, используемый в GridWay. В этом случае ускорение по времени выполнения составило около 2,5 раз. С "толстыми" (длительными по времени выполнения) задачами алгоритм справился за такое же время, как и стандартный алгоритм GridWay.

2. Имитационное моделирование алгоритма с контролем насыщения

Один из кластеров, входящих в состав Grid-полигона, а именно кластер Института вычислительной математики и математической геофизики Сибирского отделения РАН sscc.grid.pp.ru в настройках локальной политики планирования, имеет ограничение в пять запущенных задач от одного пользователя. Это ограничение никак не регистрируется информационной службой полигона MDS (Monitoring and Discovery System), соответственно диспетчер этой информацией не обладает. Как показали результаты тестирования, алгоритм с контролем насыщения хорошо подходит для среды, где информация о ресурсах не является полной для принятия эффективного решения по планированию. Соответственно, исследование возможностей применения этого алгоритма к подобным ограничениям представляет отдельный интерес. На рис. 2 представлен график зависимости от времени числа задач, запущенных на узле sscc.grid.pp.ru. Как следует из графика, число задач не превышает 6 (в то время как информационная служба полигона MDS показывает 9 свободных узлов), что является признаком того, что алгоритм адаптируется под состояние узла.

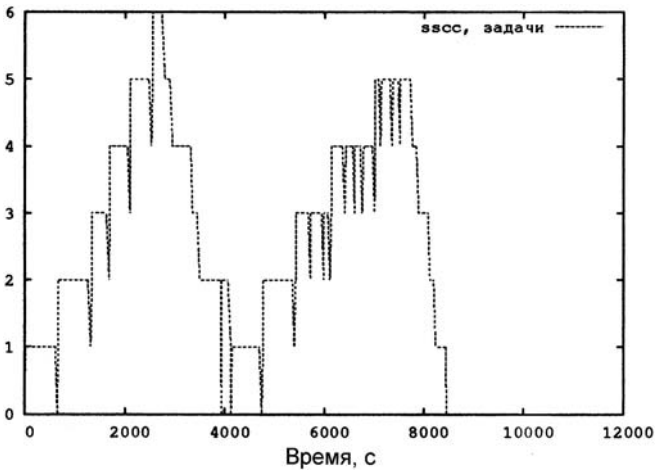


Рис. 2. Зависимость числа запущенных задач на sscc.grid.pp.ru от времени

Как некоторое теоретическое обоснование указанной тенденции было проведено имитационное моделирование алгоритма с описанными выше условиями.

В работе [9] была построена дискретно-событийная модель процесса диспетчеризации, используемого в GridWay, — система массового обслуживания $GI/G/D/M(SI, DI, MI, PI, DC)$. Рассмотрим ее модификацию, которая имеет следующие свойства.

1. В любой момент времени общее число ресурсов системы постоянно, что соответствует $N_i = D = \text{const}$, $DI = 0$.

В случае с кластером sscc.grid.pp.ru $D = 166$.

2. За одну итерацию планирования *SI* (в момент времени *iSI*) из очереди извлекается $cwnd_i$ задач, где $cwnd_i$ определяется в соответствии с алгоритмом планирования с контролем насыщения.

3. В любой момент времени число обслуживаемых задач не превышает *C*.

В случае с кластером sscc.grid.pp.ru $C = 5$.

Определение 2.1. *Система с перечисленными выше свойствами 1—3 называется системой массового обслуживания GI/G/M(SI, MI, PI, DC, D, C).*

Заметим, что система массового обслуживания $GI/G/M(SI, MI, PI, DC, D, C)$ является дискретно-

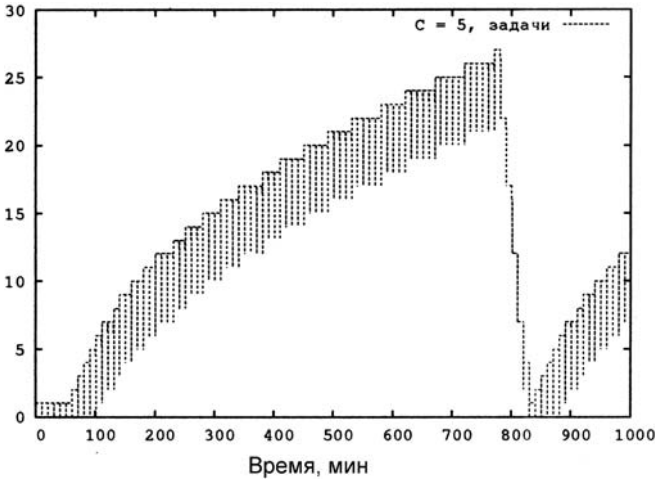


Рис. 3. Имитационная модель алгоритма планирования с контролем насыщения

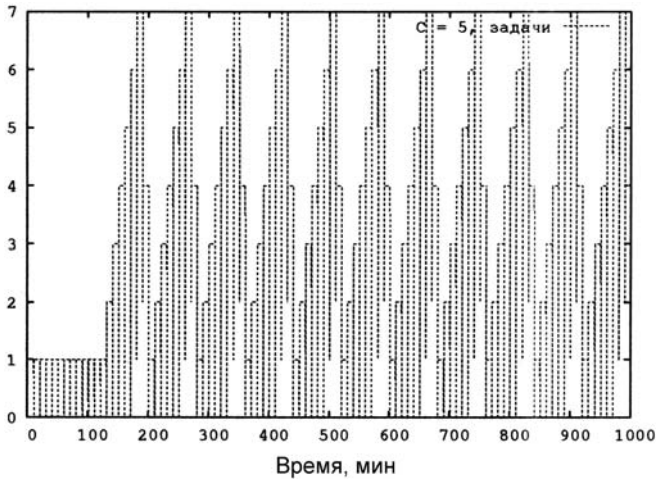


Рис. 4. Имитационная модель алгоритма планирования с контролем насыщения с параметрами *ALPHA* = 0,9 и *BETA* = 1,3

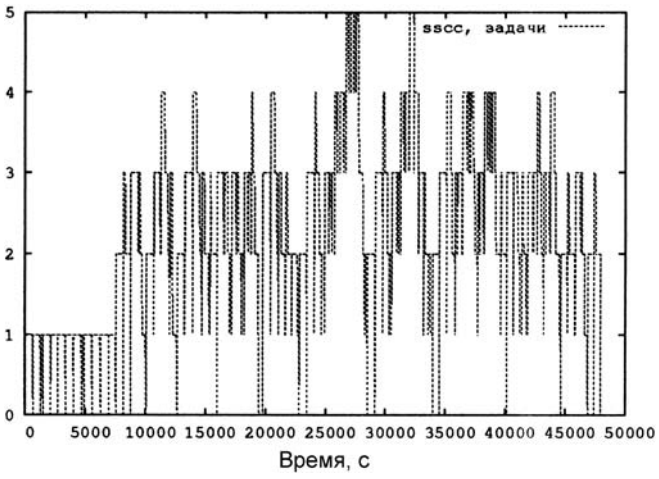


Рис. 5. Экспериментальная проверка алгоритма планирования с контролем насыщения с параметрами *ALPHA* = 0,9 и *BETA* = 1,3

событийной моделью процесса диспетчеризации задач на основе дисциплины планирования с контролем насыщения.

Поскольку алгоритм с контролем насыщения в качестве входных данных не использует число ресурсов, занятых локальными задачами, то без ограничения общности можно считать, что оно постоянно, т. е. $M_i = M = \text{const}$ и $MI = 0$.

Однако в результате компьютерного моделирования описанного выше процесса оказалось, что алгоритм не адаптируется под состояние узла (рис. 3). Причиной тому служит слишком "быстрая" адаптация тайм-аута *RTO* под время выполнение задачи. В результате работы алгоритма тайм-аут *RTO*, начиная с некоторого момента времени, покрывает время обслуживания задачи в независимости от того, как долго задача ждала освобождения одного из *C* ресурсов (напомним, что в описанной модели число обслуживаемых задач не превышает константу *C*). Таким образом, параметр *cwnd* оказывался значительно больше, чем *C*.

В то же время с помощью компьютерного моделирования удалось определить оптимальные значения констант *ALPHA* и *BETA*, входящих в определение тайм-аута *RTO*, для того чтобы алгоритм начал адаптироваться под состояние узла. Как видно из рис. 4, при *ALPHA* = 0,9 и *BETA* = 1,3 число задач, запущенных на узле, не превышает 7. Этот факт говорит о том, что при указанных значениях параметров алгоритм адаптируется под состояние узла.

Экспериментальная проверка алгоритма с контролем насыщения с параметрами *ALPHA* = 0,9 и *BETA* = 1,3 подтвердила результаты имитационного моделирования. Как следует из рис. 5, число задач, запущенных на узел, не превышает 5.

Результаты имитационного и экспериментального моделирования, описанные выше, свидетельствуют о появлении некоторого вопроса, который возникает в связи с медленной "инициализацией" алгоритма при указанных параметрах. Его суть заключается в том, что для первого увеличения параметра *cwnd* должно пройти довольно продолжительное время (см. рис. 4, 5). Причиной служит тот факт, что начальное значение параметра *SRTT*, входящего в определение тайм-аута *RTO*, равно константной величине *LBOUND*, которая гипотетически может быть на порядок меньше среднего времени выполнения задачи. Кроме того, чем меньше отношение *LBOUND* к среднему времени выполнения задачи, тем медленнее тайм-аут *RTO* адаптируется под среднее время выполнения задачи.

В этой связи в качестве начального значения *SRTT* можно использовать RTT_1, т. е. время выполнения первой задачи. Экспериментальная проверка алгоритма с контролем насыщения и начальным значением $SRTT = RTT_1$ подтвердила факт умень-

шения времени инициализации алгоритма. Таким образом, данное изменение алгоритма позволяет сократить время его адаптации в начальной стадии.

Заключение

Работа посвящена применению механизмов контроля насыщения (*Congestion Control*) для планирования ресурсов в Grid-среде. Приведены результаты тестирования алгоритмов планирования с контроля насыщения в сравнении со стандартным алгоритмом, используемом в диспетчере GridWay. Построена имитационная модель алгоритма с контролем насыщения, в результате компьютерной апробации которой найдены оптимальные параметры имитационной модели. Основной итог работы — алгоритм с контролем насыщения хорошо подходит для среды, где информация о ресурсах не является полной для принятия эффективного решения по планированию. Используя накопленные статистические данные, алгоритм справляется с планирование более эффективно, чем стандартный алгоритм, который используется в составе механизмов диспетчера GridWay.

Список литературы

1. **TOP500** Supercomputing Sites. URL:http://www.top500.org
2. **Human** Genome Project. URL: http://www.ornl.gov/sci/techresources/Human_Genome/home.shtml
3. **SETI**@home. URL: http://setiathome.berkeley.edu/
4. **distributed.nei:** Node Zero. URL: nttp: //distributed. net
5. **Foster I.** What is the Grid? A Three Point Checklist. Argonne National Laboratory & University of Chicago. 2002. July 20.
6. **RFC 793** — Transmission Control Protocol. URL: http://tools.ietf.org/html/rfc793
7. **GridWay** Metascheduler: Metascheduling Technologies for the Grid. URL: http://gridway.org
8. **GRID** МГУ. URL: http://grid.pp.ru
9. **Сериков Д. А.** К математическому моделированию процесса диспетчеризации задач в распределенной вычислительной среде // Информационные технологии. 2009. № 12. С. 17—24.

ИНФОРМАЦИЯ

22—23 апреля 2011 г. в г. Казани пройдет

Девятая Международная конференция специалистов в области обеспечения качества ПО — Software Quality Assurance Days (SQA Days-9).

SQA Days является одним из главных мероприятий в Восточной Европе, посвященных тематике тестирования и обеспечения качества ПО.

Тематика конференции:

- Функциональное тестирование
- Интеграционное тестирование
- Тестирование производительности
- Автоматизация тестирования и инструментальные средства
- Конфигурационное тестирование
- Тестирование удобства использования (usability)
- Тестирование защищенности (security)
- Статические методы обеспечения качества
- Внедрение процессов тестирования на предприятии
- Управление процессами обеспечения качества ПО
- Менеджмент команд тестировщиков и инженеров качества ПО
- Аутсорсинг тестирования
- Тестирование системных приложений (не Web), а также тестирование игр и приложений для мобильных устройств
- Мотивация проектной команды и сертификация специалистов в области обеспечения качества ПО

Подробности на сайте конференции: http://it-conf.ru/ru/content/339.htm

УДК 519.874

М. Х. Прилуцкий, д-р техн. наук, проф.,
e-mail: pril@iani.unn.ru,
В. С. Власов, канд. техн. наук, ст. науч. сотр.,
e-mail: vlasov_nn@mail.ru
Нижегородский государственный университет
им. Н. И. Лобачевского

Построение оптимальных по быстродействию расписаний в канонических системах "конвейер—сеть"

Рассматривается задача построения оптимального по быстродействию расписания в системах типа "конвейер—сеть". Для решения предлагаются вычислительные процедуры метода ветвей и границ с использованием эвристических схем нахождения верхних оценок.

Ключевые слова: *канонические системы типа "конвейер—сеть", комбинирование алгоритмов, оптимальное расписание, стохастические и детерминированные алгоритмы*

Введение

Модели управления изготовлением сложных изделий могут включать в себя несколько стадий, каждую из которых можно отнести либо к классу последовательного выполнения работ (конвейерные технологии), либо к классу распределения ресурсов в сетевых структурах. В работе предложена модель системы типа "конвейер—сеть". Рассматривается задача построения оптимальных по быстродействию расписаний в канонических системах "конвейер—сеть". Каноничность системы означает, что никакая деятельность не может быть активизирована до тех пор, пока не завершится выполнение всех деятельностей, непосредственно ей предшествующих.

Выделение класса систем "конвейер—сеть" позволило не только более естественно описывать в рамках поставленной модели многие инженерные и технические задачи, но и существенно сократило время их решения, используемые аппаратные ресурсы, а также оптимизировало поиск расписаний.

Для решения задач, относящихся к системам типа "конвейер—сеть", выделен метод, соединяющий в себе подход, связанный как с упрощением исходной задачи, а следовательно, со снижением ее математической сложности, так и с применением различных комбинаций конфигурируемых эвристических алгоритмов.

Существует широкий класс прикладных задач, формализация которых приводит к классу задач упорядочения работ и распределения ресурсов в канонических системах "конвейер—сеть". Типичными примерами таких задач являются задачи оптимального планирования и управления процессом производства изделий микроэлектроники, задачи инструментального производства, задачи планирования производства изделий машиностроения опытного производства.

К основным особенностям систем "конвейер—сеть" относятся:

- каноничность систем;
- одновременность поступления изделий на обработку;
- многостадийность систем — чередование стадий конвейерных и сетевых технологий;
- наличие групповых операций, которые должны выполняться последовательно и без перерывов;
- наличие времени пролеживания — минимально возможного интервала времени до начала выполнения следующей технологической операции;
- наличие межоперационного времени выполнения операций — максимально возможного интервала времени до начала следующей технологической операции.

Для совокупности выполняемых работ ставится оптимизационная задача минимизации времени завершения изготовления изделий.

1. Общая математическая модель

Исходные параметры математической модели

Пусть $T = \{0, 1, ..., T_0\}$ — множество тактов планирования. Обозначим J — множество всех работ, а $K(j)$ — множество работ, непосредственно предшествующих работе с номером j, $K(j) \subset J$, $j \in J$.

Пусть I — множество различных ресурсов. Обозначим n_i — срок годности ресурса i, $i \in I$, $n_i \in N$. Тогда $I^{\text{н}} = \{i | n_i = 1, i \in I\}$ — множество нескладируемых ресурсов, $I^{\text{с}} = \{i | n_i > T_0, i \in I\}$ — множество складируемых ресурсов, а $I^{\text{чс}} = \{i | 2 \leqslant n_i \leqslant T_0, i \in I\}$ — множество частично складируемых ресурсов. Обозначим $\varphi(j)$ функцию,

определяющую номер соответствующего j-й работе ресурса, $\varphi(j) \in I$, $j \in J$.

Пусть $V = \|v_{it}\|$ — матрица поступлений ресурсов в систему, где v_{it} обозначает количество ресурса с номером i, которое поступит в систему в такте t, $i \in I$, $t \in T$; $R = \|r_{ij}\|$ — матрица ресурсоемкостей, где r_{ij} обозначает количество ресурса с номером i, которое требуется для выполнения работы с номером j, $i \in I$, $j \in J$; W_{it} — количество i-го ресурса, которое может быть использовано в такте t для изготовления изделий, $i \in I$, $t \in T$.

Обозначим m_{ij}, M_{ij} — минимальную и максимальную интенсивности потребления работой с номером j ресурса с номером i, $0 \leqslant m_{ij} \leqslant M < \infty$, $i \in I$; $j \in J$, а через t_j^-, t_j^+ — минимальную и максимальную длительности выполнения работ, $j \in J$.

Пусть $G(j_s)$ — множество групповых работ, начинающихся с работы j_s, $G(j_s) = \{j_s, j_k, ..., j_t\}$, $j_s \in J$. Введем множество работ, являющихся начальными для соответствующих им групп, $G = \{\lambda_1, ..., \lambda_k\}$, $\lambda_1, ..., \lambda_k \in J$. Обозначим $t_j^{\min}$ — время пролеживания j-й работы, $j \in J^{\min}$, где $J^{\min}$ — множество работ, для которых определено время пролеживания; $t_j^{\max}$ — межоперационное время j-й работы, $j \in J^{\max}$, где $J^{\max}$ — множество работ, для которых определено межоперационное время.

Обозначим J^D множество работ, имеющих директивные сроки окончания, $J^D \subseteq J$, d_j — директивный срок окончания выполнения работы с номером j, $j \in J^D$.

Варьируемые параметры математической модели

$X = \{x_1, ..., x_{|J|}\}$ — вектор времен начала выполнения работ;

$Y = (y_1, ..., y_{|J|})$ — вектор времен окончания выполнения работ;

$Z = \|z_{ijt}\|$ — матрица интенсивностей, где z_{ijt} — интенсивность потребления ресурса с номером i работой с номером j в такт времени t, $i \in I$, $j \in J$, $t \in T$.

Ограничения математической модели

Естественные ограничения на переменные:

$$x_j \in T,\ y_j \in T,\ z_{ijt} \geqslant 0,\ i \in I,\ j \in J,\ t \in T.$$

Ограничения каноничности модели:

$$x_j \geqslant y_l,\ l \in K(j),\ j \in J.$$

Ограничения на интенсивность потребления ресурсов:

$$\begin{cases} m_{ij} \leqslant z_{ijt} \leqslant M_{ij}, \text{ если } t \in [x_j, y_j]; \\ z_{ijt} = 0, \text{ если } t \notin [x_j, y_j]; \\ i \in I,\ j \in J. \end{cases}$$

Ограничения на длительности выполнения работ:

$$t_j^- \leqslant y_j - x_j \leqslant t_j^+,\ j \in J.$$

Полное использование необходимых ресурсов означает выполнение работы:

$$\sum_{t \in T} z_{ijt} = r_{ij},\ i \in I,\ j \in J.$$

Ограничения для групповых работ:

$$x_j = y_k,\ k \in K(j),\ k, j \in G(\lambda_s),\ \lambda_s \in G.$$

Ограничения по времени пролеживания:

$$x_j \geqslant y_k + t_k^{\min},\ k \in (K(j) \cap J^{\min}),\ j \in J.$$

Ограничения по межоперационному времени:

$$y_k \leqslant x_j \leqslant y_k + t_k^{\max},\ k \in (K(j) \cap J^{\max}),\ j \in J.$$

Ограничения для директивных операций:

$$y_j \leqslant d_j,\ j \in J^D.$$

Ресурсные ограничения:

$$\sum_{j \in J} z_{ijt} \leqslant W_{it},\ i \in I,\ t \in T.$$

В рамках общей математической модели ставятся математические модели задачи упорядочения работ для систем с конвейерными технологиями и распределения ресурсов для систем с сетевыми технологиями.

2. Постановка оптимизационной задачи для систем "конвейер—сеть"

Поставленная задача является задачей построения оптимального по быстродействию расписания, и построение таких расписаний напрямую связано со временем простоя оборудования. Определим для каждого оборудования функцию простоя, связанную с неиспользованием оборудования на некотором временном участке. Время занятости оборудования на произвольном интервале $[0, t]$ при $t \in T$ находится как сумма времен выполнения операций, времена начала и окончания которых принадлежат рассматриваемому интервалу, а также времен выполнения операций, начавшихся на интервале $[0, t]$, но не завершенных к моменту времени t. Тогда время занятости i-го оборудования на интервале $[0, t]$ можно рассчитать следующим образом:

$$f_i^{\text{занят}}(x, y) = f_i^{\text{полн}}(y) + f_i^{\text{част}}(x, y),$$

где $f_i^{\text{полн}}(y) = \sum_{j: y_j \leqslant t} t_\varphi(j),\ j \in J,\ \varphi(j) = i,\ i \in I$:

$$f_i^{\text{част}}(x, y) = t - x_k,\ x_k < t < y_k,\ k \in J,\ \varphi(k) = i,\ i \in I.$$

Здесь функция $f_i^{\text{полн}}(y)$ определяет время занятости i-го оборудования для операции, время начала и завершения которых принадлежит рассматриваемому интервалу $[0, t]$, а функция $f_i^{\text{част}}(x, y)$ определяет время занятости оборудования операциями, начавшими свою обработку на интервале $[0, t]$, но не завершенными к моменту времени t, $i \in I$, $t \in T$.

Тогда функция, определяющая время простоя i-го оборудования на интервале $[0, t]$, будет определяться, как $If_i(x, y) = t - f_i^{\text{занят}}(x, y)$, $i \in I$.

Определим частные критерии оптимальности, связанные с минимизацией времени простоя конкретного оборудования, как $f_i(x, y) \to \min$, $i \in I$.

В качестве обобщенного критерия оптимальности будем рассматривать аддитивную свертку частных критериев по каждому виду оборудования при условии, что $t = \max\limits_{j \in J} y_j$:

$$F(x, y) = \sum_{i \in I} f_i(x, y) \to \min.$$

Функционал $F(x, y)$ определяет число тактов суммарного простоя оборудования на интервале от такта начала планирования до такта, соответствующего окончанию выполнения всех запланированных операций. Минимизация функционала отображает стремление максимизировать загрузку оборудования и оптимизировать быстродействие расписания.

3. Использование основных вычислительных процедур метода ветвей и границ для решения систем типа "конвейер—сеть"

Различают четыре основные вычислительные процедуры метода ветвей и границ, которые делятся на индивидуальные, зависящие от специфики задачи, и универсальные, которые являются общими для любых решаемых задач. К индивидуальным процедурам относятся процедура оценок (в общем случае — нахождение верхней и нижней оценок) и процедура ветвления. К универсальным процедурам относятся процедура отсева (отбрасывания неперспективных направлений) и процедура останова (определение оптимальности найденного решения).

Индивидуальные процедуры метода ветвей и границ.

Процедура оценок включает в себя определение верхней (достижимой) оценки V и нижней оценки H. В качестве верхней оценки выбирается минимальное значение критерия для перестановок, сгенерированных различными стохастическими и детерминированными алгоритмами. Значения нижних оценок определяются с учетом как длительностей выполнения операций, так и каноничности сетевой модели.

Процедура ветвления рассматривает допустимые варианты построения перестановок — на каждом шаге ветвления выбор осуществляется только из тех операций, для которых непосредственно предшествующие операции уже выполнены.

Универсальные процедуры метода ветвей и границ.

Процедура отсева предполагает, что если значение верхней (достижимой) оценки в одной из вершин дерева ветвлений не больше значения нижней оценки в другой вершине, то вторая вершина исключается из рассмотрения не в ущерб оптимальности.

Процедура останова определяет окончание процесса вычислений. Если осталась неотброшенной лишь одна вершина, в которой значения оценок совпадают, то найдено оптимальное решение задачи, которое определяется перестановкой, соответствующей верхней (достижимой) оценке.

Основным достоинством метода ветвей и границ является то, что остановив вычисления в любой момент времени, лучшее значение верхней оценки (рекорд) может быть принято за эвристическое решение задачи. Основным недостатком метода является необходимость определения оценок в каждой вершине дерева ветвления, а при большом числе исходных параметров число вершин становится очень большим, что не позволяет провести рассмотрение всех вершин дерева ветвлений.

4. Получение совокупности верхних оценок

4.1. Стохастические алгоритмы

Генетический алгоритм. В основу эволюционно-генетических алгоритмов заложена идея наследственности в биологических популяциях. Структура данных алгоритма состоит из набора хромосом. Работа алгоритма продолжается до тех пор, пока не будет выполнен критерий останова. В рассмотренной постановке расписание определяется допустимой перестановкой из n элементов. Представление допустимой перестановки формально является генетическим и задает хромосому в виде упорядоченной последовательности из n генов (n-мерного вектора). Ген, соответствующий j-й компоненте, определяет номер операции, которая будет выполняться j-й по порядку, $j = \overline{1, n}$. Работа генетического алгоритма основана на использовании ряда операторов кроссовера и мутации, которые, используя допустимые генотипы "родительских" особей, сохраняют допустимость генотипов потомков.

Алгоритм simulated annealing. Используется аналогия между процессом нахождения решения задачи и моделью охлаждения термодинамической системы. Считается, что процесс протекает при постоянно понижающейся температуре. На каждой итерации при температуре T система с некоторой вероятностью может перейти из состояния с энер-

гией E_1 в состояние с энергией E_2. Эта вероятность рассчитывается как

$$P(\Delta E) = \mathbf{e}^{-\Delta E/kT},$$

где $\Delta E = E_2 - E_1$ и k — постоянная Больцмана.

Работа алгоритма начинается с выбора случайного состояния π_1, удовлетворяющего ограничениям системы, для которого находится значение критерия E_1. Применяется оператор перехода из состояния π_1 в состояние π_2 и находится значение E_2 для этого состояния. Оператор перехода в алгоритме идентичен оператору мутации в генетическом алгоритме. Если $\Delta E \leqslant 0$ или $P(\Delta E) > \xi$, где ξ — случайное число, равномерно распределенное в диапазоне [0, 1], то π_2 принимается за текущее состояние системы. После этого происходит охлаждение системы. Условием остановки алгоритма является охлаждение системы до заданной температуры.

Алгоритм ant colonies. Идея алгоритма основана на моделировании поведения муравьев, связанного с их способностью быстро находить кратчайший путь от муравейника к источнику пищи и адаптироваться к изменяющимся условиям, находя новый кратчайший путь. При своем движении муравей метит свой путь феромоном, и эта информация используется другими муравьями для выбора пути. Дойдя до преграды, муравьи с равной вероятностью будут обходить ее справа и слева. Поскольку движение муравьев определяется концентрацией феромона, то следующие муравьи будут предпочитать более насыщенный феромоном путь, продолжая обогащать его феромоном. Моделирование испарения феромона гарантирует, что найденное локально оптимальное решение не будет единственным — муравьи будут искать и другие пути. В основе предлагаемого алгоритма лежит n-мерная квадратная матрица $P(t) = \|p_{ij}(t)\|$, определяющая состояние системы на такт работы алгоритма t. Элемент матрицы $p_{ij}(t)$ в такт t определяет вероятность того, что операция с номером i будет входить в искомую перестановку j-й по порядку. При этом в силу канонической особенности задачи вероятность выбора операции, для которой не выполнены все предыдущие, приравнивается нулю. В основу стратегии построения перестановки заложены как элементы стохастики, так и "предыстория" жизни системы, которая отображается матрицей $P(t)$. Каждая допустимая перестановка однозначно определяет расписание, а тем самым и значение критерия задачи, соответствующее этому расписанию.

4.2. Детерминированные алгоритмы

Фронтальный алгоритм. Предлагаемый алгоритм основан на идеологии "жадных алгоритмов": включенная в строящееся расписание на каком-то шаге работы алгоритма операция, в дальнейшем из расписания не исключается, причем для ее выполнения используются все доступные к данному моменту времени ресурсы в максимально возможном объеме.

На подготовительном этапе для каждой операции j проводится расчет временных характеристик: находятся моменты раннего начала выполнения $t_j^{\text{рн}}$, раннего окончания выполнения $t_j^{\text{рк}}$, позднего начала $t_j^{\text{пн}}$, позднего окончания $t_j^{\text{пк}}$ выполнения операций и резервы времени операций $r_j = t_j^{\text{пн}} - t_j^{\text{рн}} = t_j^{\text{пк}} - t_j^{\text{рк}}$, $j \in J$. Пусть t — произвольный такт планирования, $t \in T$. Назовем "фронтом операций" множество $F(t)$ — множество операций, любая из которых может начать выполняться с такта t, $t \in T$. Установим соответствие между фронтом операций и допустимой перестановкой, для чего перейдем от множества $F(t)$ к вектору $\rho(t)$, компоненты которого и будут определять порядок выполнения операций. Определим вектор $\rho(t)$ следующим образом. Если $F(t) = \{j_1, j_2, ..., j_k\}$, то $\rho(t) = (\rho_1, \rho_2, ..., \rho_k)$, $\rho_l \neq \rho_q$, $\rho_l, \rho_q \in F(t)$, где ρ_q — номер операции, выполняемой q-й по порядку. Упорядочим операции из множества $F(t)$ по неубыванию резервов времени. Далее последовательно просматриваем операции построенного вектора и включаем в строящееся расписание те из них, для которых свободно соответствующее оборудование. После этого изменяем значение такта планирования и заново формируем фронт операций, соответствующий этому такту.

Алгоритм, основанный на решении задачи о назначениях. Тот факт, что перестановки из n натуральных чисел можно представить в виде квадратных бистохастических булевых матриц, позволил для определения верхних оценок использовать аппарат решения задач о назначениях. Пусть P — множество различных решений системы "конвейер—сеть", а B — множество различных n-мерных бистохастических булевых матриц. Рассмотрим соответствие $\Gamma = (Q, P, B)$, где $Q = P \times B$. Соответствие Γ является функциональным, инъективным, всюду определенным и сюръективным, т. е. взаимно однозначным (биективным). Отсюда следует, что генерировать различные перестановки можно посредством решения задач о назначениях, для чего необходимо связать значения критерия задачи о назначениях, зависящие от коэффициентов функционала, со значениями критерия системы "конвейер—сеть". Это можно сделать, например, используя в качестве матрицы, определяющей коэффициенты критерия задачи о назначениях, матрицу $P(t) = \|p_{ij}(t)\|_{n \times n}$, найденную в результате работы муравьиного алгоритма. Кроме того, так как у задачи о назначениях в общем случае существует множество оптимальных решений, то имеет смысл

рассматривать все оптимальные решения задачи о назначениях, для каждого из которых находить соответствующую перестановку, строить по ней допустимое расписание и выбирать из построенных расписаний наилучшее с точки зрения критерия.

4.3. Метод комбинирования алгоритмов

Так как предложенные алгоритмы используют перестановку (или набор перестановок) в качестве исходных данных, а результатом работы алгоритмов также являются перестановки, то для наиболее эффективного решения задачи предлагается последовательное использование алгоритмов. Перестановка, соответствующая лучшему значению критерия рассматриваемой задачи, полученная одним из алгоритмов, используется как начальная для другого алгоритма. С помощью предложенных алгоритмов можно генерировать различные схемы построения допустимых расписаний, а тем самым определять верхнюю (достижимую) оценку в методе ветвей и границ.

4.4. Общая схема реализации алгоритма декомпозиции

Схема алгоритма разбиения системы "конвейер—сеть" представлена на рисунке. Как видно из схемы, алгоритм состоит из трех этапов:

- декомпозиция общей задачи на классы "конвейерных" технологий и сетевые модели. Основные преимущества данного алгоритма заключаются в получении моделей значительно меньшей размерности и конкретно определенной специфики;
- раздельное решение задач построения расписания "конвейерных" технологий и сетевых задач. Значительное сокращение размерности после первого этапа позволяет более точно находить решение каждой конкретной задачи и резко уменьшает объемы вычислений и потребление памяти;
- построение общего решения исходной задачи объединением значений частных подсистем.

Алгоритм разбиения системы "конвейер—сеть"

5. Вычислительный эксперимент

Для решения рассмотренного класса задач построена и реализована диалоговая программная система (среда Microsoft Windows XP, среда программирования Visual Studio.Net 2005, язык программирования C#). В программном комплексе реализованы описанные алгоритмы. Программное средство обладает интуитивно понятным интерфейсом, в ходе своей работы выводит пользователю всю необходимую информацию и время работы используемого алгоритма. Для проведения вычислительного эксперимента был программно реализован генератор задач, реализующий исходные данные для систем типа "конвейер—сеть".

Эксперименты проводились на машине со следующей конфигурацией: процессор AMD Athlon(tm) 64 X2 Dual Core processor 6000+ 3.01 ГГц, 2 Гбайт оперативной памяти.

Так как сетевая модель является более общей, чем конвейерная, то решение исходной задачи проводилось двумя способами:

- решение исходной задачи как единой сетевой канонической структуры;
- декомпозиция задачи на классы сетевых и конвейерных технологий.

В таблице приведены результаты эксперимента. Показано решение, полученное после декомпози-

Число работ	Число станков	$F_{к\text{-}с}$	$F_{кан}$	$w_{ср}$
20	10	837	837	
20	10	938	938	
20	10	954	954	0
20	10	911	911	
20	10	894	894	
50	20	1916	1935	
50	20	2012	2039	
50	20	1954	1978	≈0,012
50	20	1987	1987	
50	20	1953	1989	
100	20	4816	4893	
100	20	4765	4802	
100	20	4824	4916	≈0,02
100	20	4911	5023	
100	20	4832	4954	
500	30	21349	22064	
500	30	20931	21853	
500	30	21096	21985	≈0,042
500	30	21716	22971	
500	30	21533	22847	
1000	40	43528	45832	
1000	40	43719	45374	
1000	40	43101	45630	≈0,049
1000	40	44235	46012	
1000	40	43562	45961	

ции задачи на классы "конвейер—сеть" ($F_{к—с}$), решение задачи как единой канонической структуры ($F_{кан}$) и среднее отклонение полученных решений ($w_{ср}$).

Учитывая полученные данные, приведенные в таблице, можно сделать вывод о качестве предложенного алгоритма для решения задач в канонических системах "конвейер—сеть" большой размерности.

Заключение

Несмотря на то, что рассматриваемые задачи относятся к классу NP-трудных, и то, что реальные производственные задачи имеют большие размерности, применение алгоритма декомпозиции и метода ветвей и границ с использованием эвристических алгоритмов для нахождения верхних оценок позволяет в реальном времени находить допустимые решения задач планирования, формализованных в рамках канонических систем типа "конвейер—сеть". Это связано с тем, что даже рассмотрев не все вершины дерева ветвления, в процессе решения задачи можно многократно применять различные алгоритмы нахождения верхних оценок. А так как верхние оценки для рассматриваемого метода ветвей и границ являются достижимыми, то они определяют допустимые решения исходной задачи, лучшее из которых (с точки зрения критерия) может быть принято за эвристическое решение поставленной задачи.

Разработанная программная система внедрена в эксплуатацию при планировании производства изделий микроэлектроники (большие интегральные схемы и гибридные интегральные схемы) и инструментального производства, а также при планировании процесса изготовления изделий машиностроения.

Список литературы

1. **Батищев Д. И., Гудман Э. Д., Норенков И. П., Прилуцкий М. Х.** Метод комбинирования эвристик для решения комбинаторных задач упорядочения и распределения ресурсов // Информационные технологии. 1997. № 2. С. 29—32.
2. **Прилуцкий М. Х., Власов В. С.** Оптимизационные задачи распределения ресурсов при планировании производства микроэлектронных изделий // Системы управления и информационные технологии. 2009. № 1 (35). С. 38—43.
3. **Прилуцкий М. Х., Власов В. С.** Метод ветвей и границ с эвристическими оценками для конвейерной задачи теории расписаний // Вестник Нижегородского государственного университета. Математическое моделирование и оптимальное управление. Нижний Новгород: Изд-во ИНГУ. 2008. Вып. 3. С. 147—153.

УДК 519.876.5

И. В. Рудаков, канд. техн. наук, доц.,
e-mail: irudakov@yandex.ru,
А. В. Ребриков, студент,
e-mail: rebrikov_@mail.ru,
МГТУ им. Н. Э. Баумана

Неполная верификация сложных дискретных систем

Рассмотрен вопрос реализации метода неполной верификации моделей, реализующего статистически достоверную проверку и обеспечивающего максимальный показатель структурного покрытия элементов модели. Приводится описание метода формализации моделей в виде вероятностных автоматов, а также разработанных методов генерации наборов входных данных, обеспечивающих выполнение предъявляемых к системе верификации требований, а также проводится сравнение разработанных и существующих методов.

Ключевые слова: *неполная проверка моделей, верификация, автоматическая генерация входных данных, вероятностный автомат*

Сложные дискретные системы [1] используются для моделирования и описания объектов и систем широкого класса в различных областях человеческой деятельности и решения таких задач, как распознавание речи, криптоанализ, машинный перевод, моделирование работы вычислительных систем, систем автоматического управления и т. п.

При анализе функционирования сложных систем полная, или формальная, верификация модели [2, 3] таких систем может потребовать больших временных затрат [3], поэтому в целях сокращения затрат необходимо иметь возможность проведения неполной верификации [11] системы с заданной степенью доверия. Помимо сокращения временны́х затрат неполная верификация позволяет проводить проверку систем с нецелочисленной логикой, что невозможно сделать при полной верификации.

Необходимость разработки программного комплекса для неполной верификации моделей сложных дискретных систем обусловлена тем, что существующие аналоги являются узкоспециализированными и не обеспечивают требуемой степени

покрытия состояний модели или требуемого уровня автоматизации верификации [13].

Введение вероятностных процессов в детерминированные системы позволяет упростить их верификацию за счет уменьшения числа проводимых экспериментов, достаточное число которых определяется по формуле [4]

$$n = \frac{zp(1-p)}{\varepsilon^2}, \quad (1)$$

где n — минимальное число необходимых экспериментов; z — выбранный уровень доверия, на котором принимается решение о корректности работы системы; ε — относительная ошибка; p — относительная вариация выхода системы.

Одним из многочисленных методов формализации дискретных систем [9] являются вероятностные автоматы [5], в частном случае представляющие собой марковские цепи или скрытые и псевдомарковские модели. Кроме того, вероятностный автомат может быть сведен к байесовской сети доверия, обладающей важным свойством частичной обратимости, т. е. возможностью утверждать, что некоторый факт с определенной степенью вероятности является следствием другого факта. Данное свойство используется для генерации нечетких классов эквивалентности и последующей неполной верификации с равномерным распределением тестовых данных по этим классам.

Системы, представленные в виде вероятностных автоматов, можно использовать также и для моделирования взаимодействия двух независимых систем в условиях неполноты знаний систем друг о друге [10]. К примеру, одна из систем может быть отключена в момент обращения другой системы к ней либо же процесс взаимодействия не начнется совсем в силу недоступности одной из систем. С точки зрения другой системы данные явления могут быть рассмотрены как стохастические.

Для отражения структуры модели используется вероятностный автомат, дополненный нечеткой функцией вероятностей перехода. Пусть:

$S = \{s_i, i = 1, ..., n\}$ — множество состояний автомата;

$V = \{v_i, i = 1, ..., m\}$ — набор переменных системы;

$I \in S$ — начальное состояние автомата;

$F \subset S$ — множество завершающих состояний автомата;

$P: U \subset S \times S \to [0; 1]$ — четкая функция вероятностей перехода;

$PF: U \subset S \times S \to Fz$ — нечеткая функция вероятностей перехода, размечающая переход в виде набора правил типа *if* V_k *is* L_i *then P is* L_{pj}, определяющая вероятность перехода на основании нечеткого вывода;

$A = \{a_i, i = 1, ..., m\}$ — множество действий автомата, модифицирующих кортеж V;

$Action: S \to A$ — функция, сопоставляющая состоянию действие, выполняемое при входе в данное состояние.

Отметим, что $Action(I)$ есть не что иное, как инициализация кортежа V. Тогда автомат формализуется кортежем

$$M = (S, V, I, F, Action, P, PF). \quad (2)$$

Нечеткая функция перехода необходима для отражения неопределенности влияния исходных данных на вероятность перехода. Адекватность использования нечеткой функции переходов несложно показать, убедившись, что в общем случае кортеж (2) может являться универсальным аппроксиматором для любой математической системы. Для этого рассмотрим систему нечетких правил

$$R = R_i \text{ if } x_{ij} \text{ is } v_{jk} \text{ then } y \text{ is } v_{l0}, \quad (3)$$

где v_{jk} — терм одной из входных нечетких переменных; v_{l0} — терм выходной нечеткой переменной; x_{ij} — входные переменные; y — выходная переменная.

Алгоритм приведения нечеткой системы (3) к вероятностному автомату (2) сводится к следующим этапам:

1. Дефаззификация исходной нечеткой системы.

2. Расчет вероятностей перехода. Для каждого правила из (3) в (2) добавляется переход с вероятностью

$$\alpha_{i0} = \frac{\alpha_i}{\sum_{i=1}^{n} \alpha_i}, \quad (4)$$

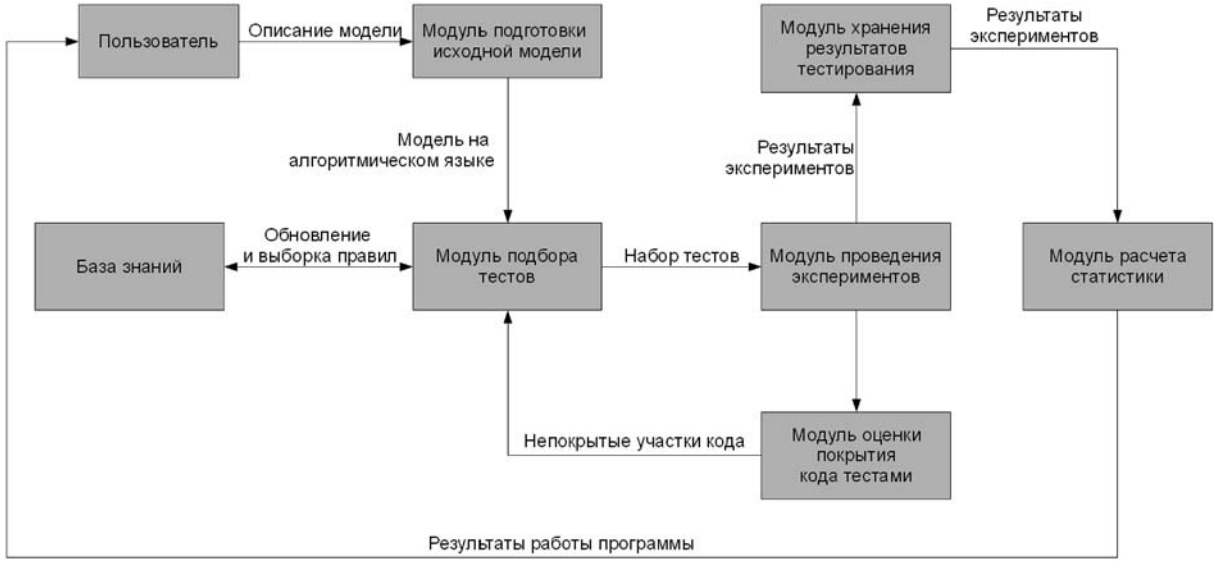
где α_i — уровень истинности [6] правила R_i.

3. Для каждого перехода добавляется состояние, в котором выходная переменная принимает вычисленное четкое значение y_i.

Тогда математическое ожидание выхода системы $M(f_0(x))$ определяется формулой

$$M(f_0(x)) = \sum_{i=1}^{n} \alpha_{i0} y_i = \frac{\sum_{i=1}^{n} \alpha_i y_i}{\sum_{i=1}^{n} \alpha_i} = f(x), \quad (5)$$

где $f(x)$ — передаточная функция исходной нечеткой системы; $f_0(x)$ — передаточная функция вероятностного автомата с нечеткой функцией переходов. Поскольку $f(x)$ является универсальным аппроксиматором [6], то и выражение $M(f_0(x))$ является универсальным аппроксиматором.



Структура программного комплекса для неполной верификации моделей

Модель системы, формализованная в виде кортежа (2), может быть задана с помощью различных языков описания данных, таких как JSON, XML или YAML [14], а действия автомата представляются на алгоритмическом языке программирования. Программный комплекс для неполной верификации моделей получает на вход файл с описанием автомата и дополнительные параметры верификации, в том числе уровень доверия и режим использования.

Структура программного комплекса для неполной верификации моделей (см. рисунок) состоит из следующих модулей:

- модуль подготовки исходной модели, осуществляющий преобразование исходных данных во внутренние структуры программного комплекса;
- модуль оценки покрытия подготовленного кода выбранными тестами;
- модуль подбора тестов, состоящий из простейшей базы знаний о структурах классов эквивалентности входных данных (проверка инициализации сложных структур, таких как ассоциативные массивы, ссылки на другие типы данных, проверки операций деления и элементарных функций), а также интерактивного модуля уточнения информации о составе входных и выходных данных;
- модуль генерации тестов для обеспечения максимального покрытия кода модели (модуль проведения экспериментов) [11, 12];
- модуль хранения накопленной в результате мониторинга информации о системе;
- модуль проведения эксперимента и расчета статистики, включающий, в том числе и систему нечеткого вывода.

Набор подобранных тестов будет считаться оптимальным, если либо покрытие кода признано допустимым (параметр допустимости задается пользователем), либо же состав тестов не может быть улучшен ни автоматически, ни интерактивно.

Использование программы возможно в следующих режимах:

- тестирование на заданном уровне доверия;
- проверка системы на основании данных, полученных при мониторинге работы аналогичной реальной системы.

Тестирование системы на заданном уровне доверия (1) состоит из стохастического тестирования и тестирования методом белого ящика. В последнем случае обеспечивается максимальное покрытие участков программного кода с помощью соответствующего модуля программного комплекса, реализующего алгоритмы автоматической генерации тестов [11, 12].

Если в модели не используются нечеткие переменные, то для автомата строится байесовская сеть доверия, в результате чего выявляется нечеткая зависимость между классами эквивалентности выходных и входных данных. Используя средства оценки степени покрытия кода тестами и выявленную зависимость между входными и выходными данными, можно скорректировать параметры тестирования. В случае же использования в модели нечетких переходов, которые в свою очередь зависят от переменных модели, осуществляется попытка построения зависимости с использованием модифицированного алгоритма обратного распространения ошибки для нейронечетких сетей [7].

Результатом работы программы являются:

- верификация системы на заданном уровне доверия;

- статистические характеристики верификации (вероятности ошибок первого и второго рода);
- описание тестов, при которых система повела себя некорректно.

Если число тестов, которые система не прошла, мало и укладывается в число допустимых ошибок на заданном уровне доверия, то верификация системы признается успешной.

Разработанный программный комплекс использовался для верификации системы автоматической регистрации услуг web-хостинга. Уровень доверия был выбран равным 0,95, что обусловлено требуемым уровнем автоматизации процессов. Заявка в системе регистрации услуг может находиться в одном из следующих состояний: неактивна, активна, приостановлена, удалена. В процессе перехода из одного состояния в другое система регистрации услуг взаимодействует с внешними системами, в частности, с панелью управления сервером ISPManager, используя протокол SOAP [15]. На запросы от системы регистрации услуг панель управления может возвращать успешный ответ или ответ, содержащий код ошибки. Также система регистрации может не получить ответ от системы в случае некорректной работы программного обеспечения на сервере.

Нечеткими переменными в данном случае выступали следующие параметры:

- средний уровень загрузки процессора сервера;
- число виртуальных хостов на сервере;
- среднее число запросов к серверу за секунду.

Неполная верификация системы заказа, приведенной к виду вероятностного автомата с нечеткой функцией переходов, позволила выделить ошибки в системе, возникающие в результате некорректной обработки ответов, содержащих код ошибки, а также в случае отсутствия ответа от панели управления сервером. После исправления ошибок в системе повторная верификация признана успешной на уровне доверия 0,95.

Список литературы

1. **Емельянов В. В., Ясиновский С. И.** Имитационное моделирование систем: Учеб. пособие. М.: Изд-во МГТУ им. Н. Э. Баумана, 2009. 584 с.

2. **Ben-Ari. M.** Principles of Spin. Vol. 3639 of Lecture Notes in Computer Science, San Francisco, August 2005. Springer-Verlag. 2008. 216 p.

3. **Burch J., Clarke E., McMillan K., Dill P. and Hwang L.** Symbolic model checking: 10^20 states and beyond // Information and Computation. 1992. Vol. 98. N 2. P. 142—170.

4. **Коваленко И. Н., Филиппова А. А.** Теория вероятностей и математическая статистика. М.: Высшая школа, 1983. 368 с.

5. **Бухарев Р. Г.** Вероятностные автоматы и процессы. М.: Знание, 1986. (Сер. Математика, кибернетика; Т. 6).

6. **Diskerson J. A., Kosko B.** Fuzzy Function approximation with supervised ellipsoidal learning // IEEE Transactions on systems, man and cybernetics. Part B: Cybernetics. Hanover, IEEE Systems, Man, and Cybernetics Society. 1996. Vol. 26. N 4.

7. **Тарков М. С.** Нейрокомпьютерные системы: Учеб. пособие. М.: Интернет-Университет информационных технологий; БИНОМ, 2006. 142 с.

8. **Хикс Ч. Р.** Основные принципы планирования эксперимента / Пер. с англ. М.: Мир, 1967.

9. **Финаев В. И., Павленко Е. Н., Заргарян Е. В.** Аналитические и имитационные модели: Учеб. пособие. Таганрог: Изд-во Технологического института ЮФУ. 2007.

10. **Кузьмин Е. В., Соколов В. А.** О некоторых подходах к верификации автоматных программ // Сборник докладов семинара Go4IT — шаг к новым технологиям Интернета. М.: Институт системного программирования, 2007. С. 43—48.

11. **Godefiroid P., Klarlund N. and Sen K.** DART: Directed Automated Random Testing // Proceedings of PLDI'2005 (ACM SIGPLAN 2005 Conference on Programming Language Design and Implementation). Chicago, June 2005. P. 213—223.

12. **Gupta N., Mathur A. P. and Soffa M. L.** Generating Test Data for Branch Coverage // Proceedings of the 15th IEEE International Conference on Automated Software. Grenoble, France. 2000.

13. **Кулямин В. В.** Методы верификации программного обеспечения // Всероссийский конкурсный отбор обзорно-аналитических статей по приоритетному направлению "Информационно-телекоммуникационные системы". М.: Ин-т системного программирования. 2008. 117 с.

14. **YAML:** YAML Ain't Markup Language [Электрон, ресурс]. URL: http://www.yaml.org/, свободный

15. **SOAP** Version 1.2 [Электрон. ресурс]. URL: http://www.w3.org/TR/soapl2-part1/, свободный

УДК 004.056

Я. Н. Имамвердиев, канд. техн. наук, зав. отд.,
e-mail: yadigar@lan.ab.az,
С. А. Деракшанде, соискатель,
e-mail: smdk364@yahoo.com,
Институт информационных технологий
национальной академии наук Азербайджана,
Азербайджан, Баку

Сервис-ориентированная эталонная модель для управления рисками информационной безопасности

Предлагается подход к формализации задачи управления рисками информационной безопасности (ИБ) на основе процессной и сервис-ориентированной моделей систем управления бизнес-процессами и информационными технологиями. Предложена эталонная модель рисков ИБ, состоящая из слоев бизнес-процессов, ИТ-сервисов, сервисов ИБ и угроз. Требования бизнес-процессов к ИБ отражаются в соглашениях об уровне ИТ-сервиса. В эталонной модели последствия рисков ИБ оцениваются с точки зрения достижения бизнес-целей, это позволяет повысить достоверность оценок рисков, экономически обосновать инвестиции в ИБ, а также сделать прозрачным процесс управления рисками ИБ.

Ключевые слова: *информационная безопасность, бизнес-процессы, эталонная модель, управление рисками, ИТ-сервис*

Введение

В современных условиях риски различного вида являются неотъемлемыми факторами деятельности любой организации, поэтому процессы управления рисками имеют решающее значение для успешного функционирования организаций. В связи с широким использованием информационных технологий в различных сферах жизнедеятельности организации управление рисками информационной безопасности (ИБ) становится более актуальным, так как информационные технологии (ИТ) позволяют не только повысить эффективность бизнес-процессов, но и могут стать источником огромного ущерба [1].

Основными задачами управления рисками ИБ являются идентификация и оценка наиболее значимых для бизнес-процессов рисков ИБ, оценка адекватности используемых средств контроля рисков, разработка мер по защите от возможных потерь, обеспечение необходимой информацией лиц, принимающих решения об инвестициях в ИБ [1, 2].

В настоящее время для управления рисками ИБ существует несколько стандартизованных международных и национальных методологий: ISO/IEC 27005:2007, NIST 800-30, OCTAVE, CRAMM и т. д. [1—5]. На основе некоторых из этих методологий разработаны различные инструментальные средства, например, программные продукты "Гриф" компании *Digital Security* (Санкт-Петербург) и "Авангард" Института системного анализа РАН [1].

Анализ показывает, что существующие методологии управления рисками имеют многочисленные недостатки [6—11]: значительная субъективность оценок рисков; высокие требования к профессионализму специалистов, проводящих оценку; трудность интерпретации результатов оценки; невозможность объективного сравнительного анализа различных вариантов защиты; плохая корреляция оценок рисков, полученных с помощью разных методик; отсутствие механизмов управления остаточными рисками; невозможность проведения оценки качества процесса реагирования на инциденты ИБ.

Методологии управления рисками ИБ нацелены в конечном итоге на лиц, принимающих решения (ЛПР), и служат тем инструментом, который позволяет проводить количественные и качественные оценки, обеспечивающие научную поддержку в процессе принятия решения. Несмотря на то, что существующие методологии управления рисками ИБ в основном предназначены для стратегического этапа управления, у них отсутствуют какие-либо привязки к бизнес-целям и бизнес-процессам организации, инвестиции в ИБ часто изолированы от разработки основных бизнес-процессов, они в большей степени ориентированы на поддержку ИТ-инфраструктуры. В результате, оценки рисков ИБ непрозрачны для ЛПР и в силу перечисленных выше недостатков ЛПР относятся с недоверием к этим оценкам.

Все, что влияет на бизнес-результаты, должно быть неразрывно связано с системой управления

организацией. Поэтому управление рисками ИБ не является самоцелью и изолированной задачей, оно должно интегрироваться с другими системами управления и стать составной частью общей системы управления организацией. В последнее время основным подходом к управлению организациями является процессный подход. Вся деятельность организации представляется в виде набора бизнес-процессов, а управление организацией — как управление бизнес-процессами [12, 13]. Для устранения некоторых из вышеуказанных недостатков в этой работе предлагается эталонная модель управления рисками ИБ, объединяющая современные парадигмы управления бизнес-процессами и ИТ-сервисами [14]. Такое слияние положительно влияет как на понимание рисков ИБ представителями бизнеса (концепция ИТ-сервиса более наглядна, чем термин "удаленный вызов процедуры"), так и на понимание бизнес-процессов разработчиками ИТ-систем, в результате процесс управления рисками ИБ становится более прозрачным, риски ИБ оцениваются по влиянию на конечный результат организации, и оценка рисков ИБ осуществляется при непосредственном участии потребителя — клиента бизнес-процессов.

1. Обзор последних работ

Идеи управления рисками во многом восходят к модели безопасности с полным перекрытием, разработанной еще в 1970-х годах. Эта модель исходит из постулата, что система безопасности должна иметь, по крайней мере, одно средство для обеспечения безопасности на каждом возможном пути воздействия нарушителя на ИТ-инфраструктуры [15]. В ней не рассматривается вопрос стоимости внедряемых средств защиты и соотношения затрат на защиту и получаемого эффекта.

В последние годы разными авторами ведутся работы по интеграции технологических подходов к управлению рисками ИБ и экономических подходов к управлению бизнес-процессами и наблюдается некоторая активность в этой интегративной области [16—26]. Работа [16] посвящена обзору исследований и идентификации актуальных проблем в этом направлении.

В работе [17] предлагается использовать функциональную декомпозицию для рисков в ИТ-инфраструктуре организации. Функциональная модель бизнес-операций используется в качестве критерия оценки критичности отдельных активов. Такой традиционный подход для организации не поддерживает процессно-ориентированный взгляд на бизнес-функции и не учитывает функциональные компоненты, которые могут поддерживать несколько бизнес-функций.

В работе [18] предлагается модель для оценивания стоимость—эффективность разных уровней безопасности с бизнес-процессами. Модель предназначена для планирования уровней безопасности при разработке программного обеспечения.

Авторы работы [19] предлагают метод анализа рисков ИБ на основе бизнес-модели. В методе используется количественный подход к непрерывности операций: сначала определяется важность различных бизнес-функций и уровень необходимости различных активов. На основе этих двух величин определяется цена каждого актива.

В работе [20] разработана методология, позволяющая обрабатывать требования безопасности к бизнес-процессам от их спецификации до их реализации. В частности, методология обеспечивает графическую концепцию для определения требований безопасности, репозиторию различных механизмов для реализации требований безопасности и набор эталонных моделей, позволяющих модифицировать бизнес-процессы. Кроме того, на основе объектно-ориентированных моделей процессов вводится инструмент для анализа безопасности бизнес-процессов и использования механизмов безопасности.

В работе [21] предлагается эталонная модель для управления рисками ИБ на основе бизнес-процессов. Модель состоит из слоев бизнес-процессов, ИТ-инфраструктуры, уязвимостей и угроз. В этой модели отсутствует сервис-ориентированный подход.

Таким образом, анализ работ показывает, что в контексте управления бизнес-процессами риск рассматривается в основном с точки зрения управления проектами [22, 23]. Некоторые работы концентрируются на ИБ и не интегрируют бизнес-процессы и анализ рисков. Какие-то работы сосредоточиваются на разработке безопасных бизнес-процессов на основе требований безопасности [24]. Другие работы предлагают анализировать требования безопасности к бизнес-процессам или верификации бизнес-процессов относительно политики безопасности [25].

2. Современные парадигмы управления

Как уже отмечалось, в настоящее время для большинства сфер управления основным подходом к управлению является так называемый процессный подход. Идея, представляющая всю деятельность организации в виде набора бизнес-процессов, а управление ее деятельностью — как управление бизнес-процессами, стала распространяться в конце 1980-х годов. Лучшие компании мира на практике доказали важность, эффективность, экономичность и прогрессивность перехода на клиентно-ориентированное производство товаров и услуг и про-

цессно-ориентированную структуру управления производством [12, 13].

Отметим, что сегодня система управления большинства организаций в постсоветском пространстве имеет функциональную направленность. Попытки внедрения автоматизации в функционально ориентированных организациях (программа внедрения АСУ 1970-х годов) привели к увеличению накладных расходов на обеспечение деятельности без повышения эффективности, а в ряде случаев при снижении эффективности в несколько раз [12]. При современных тенденциях ориентации на клиентов функциональный подход управления оказывается неэффективным. В таких структурах чрезмерно усложнен обмен информацией между различными подразделениями, что приводит к неоправданно длительным срокам выработки управленческих решений, и как следствие, к потере клиентов.

Большинство бизнес-процессов современной организации невозможно без поддержки ИТ, бизнес-процессы настолько тесно связаны с ИТ-инфраструктурой, что эффективность работы ИТ-подразделения оказывается одним из основополагающих факторов деятельности организации в целом. Вместе с тем, несмотря на огромный опыт использования ИТ в различных организациях, часто наблюдается несоответствие деятельности ИТ-подразделений ключевым бизнес-задачам организаций, это становится серьезным препятствием на пути повышения эффективности деятельности организаций [26].

Одним из путей преодоления упомянутого несоответствия является переход на относительно новую концепцию управления ИТ-сервисами. Ее сущность заключается в необходимости перехода от традиционной модели, где главная цель — поддержка ИТ-инфраструктуры, к модели, ориентированной на обслуживание основных бизнес-процессов организации. В этом случае ИТ-подразделение выступает в роли поставщика услуг для других подразделений. Требования качества предоставляемых услуг закрепляются в контрактах об уровне сервиса (*Service Level Agreement*, *SLA*), заключаемых между ИТ-подразделением и другими структурными единицами. В результате изменяется роль ИТ-подразделения в организации: из вспомогательной структурной единицы оно превращается в полноправного участника бизнес-процессов.

Становление концепции управления ИТ-сервисами связано с появлением библиотеки ITIL (*IT Infrastructure Library*) [27]. Закрепленные в ITIL концепции получили практическое воплощение в методиках *ITSM Reference Model* (Hewlett-Packard), *Microsoft Operations Framework* (Microsoft), *IT Process Model* (IBM), а также в упомянутых программных продуктах и ряда других компаний. Логическим результатом этих разработок стало принятие в 2005 г. международного стандарта ISO/IEC 20000 "Управление ИТ-сервисом" [28]. В этом стандарте формируются требования к 13 сервисам, объединенным в пять групп. Сегодня принципы и положения, изложенные в ISO 20000, признаны во всем мире и широко используются [29].

Под термином ИТ-сервис (ИТ-услуга) обычно понимается предоставление потребителям некоторой совокупности технических и организационных решений, которые обеспечивают поддержку одной или нескольких бизнес-функций (бизнес-процессов) потребителей и воспринимаются ими как единое целое. Например, "классический" ИТ-сервис — это предоставление доступа в Интернет.

Под ИТ-сервисами часто подразумевают услуги, которые ориентированы на потребителя вне организации, но стандарт ISO 20000 может быть также применен и к ИТ-сервисам, поставщики и потребители которых находятся внутри одной организации.

В общем случае ИТ-сервис характеризуется рядом параметров [26, 27]: функциональность, время обслуживания, доступность, надежность, производительность, конфиденциальность, масштаб, затраты. Как видно из приведенного списка, эти параметры охватывают также вопросы ИБ, так как ИБ является важнейшим показателем качества ИТ-сервиса. Требования бизнеса к ИБ должны быть отражены в соглашениях SLA. Задачей процесса управления ИБ в данном контексте является постоянное обеспечение безопасности ИТ-сервисов на согласованном с партнером уровне.

3. Эталонная модель рисков ИБ

В традиционных подходах к управлению рисками ИБ риск ИБ оценивается по двум (угроза, потенциальное воздействие) или по трем (угроза, уязвимость, потенциальное воздействие) факторам [1]. В предлагаемой ниже модели риски ИБ идентифицируются по двум факторам — угрозе и потенциальному воздействию.

Моделирование отношений между факторами рисков ИБ и их влияние на бизнес-процессы является очень сложной задачей. Нахождение подходящего абстрагирования от реальности и обеспечение интерпретации является задачей каждого подхода к моделированию. В этом разделе для моделирования причинно-следственных отношений рисков ИБ предлагается эталонная модель рисков ИБ. Эталонная модель — это абстрактное представление понятий и отношений между ними в некоторой проблемной области, этот подход используется в информатике (компьютерных науках) для уменьшения сложности [30]. Предложенная модель структурирована в соответствии с иерархической

Эталонная модель для управления рисками ИБ

моделью слоев абстрагирования. На основе этой эталонной модели можно построить более конкретные и детально описанные модели.

Предложенная модель состоит из четырех взаимодействующих слоев (см. рисунок). Ниже приводится краткое описание каждого из слоев.

Слой 4. Бизнес-процессы (BP)

Слой BP содержит идентифицированные бизнес-процессы организации. Бизнес-процесс — это совокупность действий (операций), направленных на достижение какой-либо определенной цели, например, бизнес-процесс "обслуживание клиента". Как правило, не существует стандартного списка бизнес-процессов, для каждой организации должен быть разработан собственный перечень основных и вспомогательных бизнес-процессов. Любые операции на предприятии происходят в рамках какого-либо бизнес-процесса.

Ориентируясь на количественную оценку рисков ИБ, каждый бизнес-процесс определяют таким образом, чтобы его денежный вклад в результат организации можно было вычислить. В общем случае бизнес-процессы можно моделировать независимо от ИТ-сервисов, например, используя инструмент ARIS (Архитектура интегрированных информационных систем), который предназначен для моделирования, реализации и оптимизации бизнес-процессов с помощью унифицированного языка моделирования (UML) [31].

Слой 3: ИТ-сервисы (IT-S)

Слой IT-S включает все ИТ-сервисы, которые используются компонентами из слоя BP и таким образом поддерживают бизнес-процессы. Идентификация ИТ-сервисов как компонентов слоя IT-S в значительной степени зависит от особенностей организации, существующей ИТ-инфраструктуры, квалификации персонала, стратегии развития и т. п. Упрощение этой задачи ожидается для организаций, использующих подход SOA.

В последнее десятилетие архитектура SOA стала основной при проектировании информационных систем организаций. По определению работы [32] "SOA — это каркас для интеграции бизнес-процессов и поддерживающей их ИТ-инфраструктуры в форме безопасных, стандартизованных компонентов — служб, которые могут использоваться многократно и комбинироваться для адаптации к изменению приоритетов в бизнесе". Реализация процедур бизнес-процессов как сервисов делает идентификацию ИТ-сервисов легкой.

Одновременно ИТ-сервисы являются точками нарушения функционирования бизнес-процессов. Неправильная работа ИТ-сервисов оказывает нежелательное влияние на бизнес-процессы, и идентифицированные ИТ-сервисы должны удовлетворять специфичным требованиям по ИБ бизнес-процессов. Зафиксированные в SLA требования по ИБ осуществляются специальными видами ИТ-сервисов — сервисами ИБ. Сервисы ИБ позволяют объединить экономический подход к обработке рисков ИБ с технологическим подходом на основе угроз. Поэтому они отнесены в отдельный слой и сервисы ИБ составляют ядро следующего слоя.

Слой 2. Сервисы ИБ (IS-S)

Слой IS-S включает все сервисы по обеспечению ИБ. В эталонной модели обеспечение ИБ рассматривается как сервис с определенным уровнем качества, предоставление которого обеспечивается определенными финансовыми, техническими и трудовыми ресурсами. Перечень сервисов ИБ, требования к ним, их функциональность, возможные методы реализации определяются исходя из требований бизнес-процессов к ИТ-сервисам. В литературе под сервисами ИБ обычно понимаются программно-технические меры [33]. В этой работе под сервисами ИБ подразумеваются наряду с традиционными сервисами ИБ (идентификация и аутентификация, управление доступом, протоколирование и аудит, шифрование, экранирование, туннелирование и т. д.) также административные и организационные меры. Сервисы ИБ обеспечивают заданную поддержку бизнес-процессов на протяжении их жизненного цикла.

Интеграция систем управления ИБ в систему процессов управления ИТ-сервисами и применение сервисно-ресурсного подхода при построении дают целый ряд преимуществ. В частности, появляется возможность правильной расстановки приоритетов для решаемых задач ИБ, повышения эффективности расходования ресурсов и средств, выделяемых на управление безопасностью, и как следствие — повышение управляемости системы ИБ в целом.

Слой 1. Угрозы (T)

Слой T содержит все известные угрозы, которые нацелены на ИТ-сервисы и могут стать причиной рисков ИБ — нарушения бизнес-процессов, в идеале они могут быть описаны вместе с вероятностями их реализации. Угроза — это совокупность условий и факторов, которые могут стать причиной нарушения целостности, доступности, конфиденциальности информации. Угрозы существуют всегда независимо от того, они реализуются как атаки или нет.

Стартовой точкой для идентификации и категорирования угроз могут стать различные каталоги угроз [3, 4, 34]. Например, в каталоге угроз BSI IT Baseline Protection Manual [34] приводится список из около 370 угроз, сгруппированных по пяти категориям (угрозы в связи с форс-мажорными обстоятельствами; угрозы на организационном уровне; угрозы, связанные с ошибками людей; угрозы, связанные с техникой; преднамеренные угрозы).

В рамках этих четырех слоев отношения между причинами и следствиями могут быть смоделированы исходя из нужд управления рисками ИБ, ориентированного на бизнес-процессы и ИТ-сервисы. Конечно, эти четыре слоя являются только "минимальным" разделением, их можно расширить при применении. Например, если оценка рисков выполняется по трем факторам (угроза, уязвимость, потенциальное воздействие), то в эталонную модель между слоями сервисов ИБ и угроз можно добавить слой уязвимостей.

4. Моделирование причинно-следственных отношений для рисков ИБ

Предложенная эталонная модель для рисков ИБ может служить основой для формального моделирования отношений между причинами рисков ИБ и их следствиями в бизнес-процессах или результатах корпорации. Для моделирования отношений предлагается использовать матричное представление. Различные слои эталонной модели рисков ИБ и отношения между этими слоями формально могут быть представлены следующими тремя матрицами: $BS = (b_{ij})_{m \times n}$, $IS = (s_{ij})_{n \times e}$, $ST = (t_{ij})_{e \times p}$, где m — число компонентов в слое BP; n — число компонентов в слое IT-S; e — число компонентов в слое IS-S; p — число компонентов в слое T.

Матричное представление служит структурой для формального описания причинно-следственных отношений для рисков ИБ. Применение эталонной модели рисков ИБ к бизнес-процессам и ИТ-инфраструктуре организаций требует отдельной концепции для определения формального моделирования отношений b_{ij}, v_{ij} и t_{ij} между компонентами. Простейшим подходом является моделирование отношений как бинарное отношение, т. е. определение отношения b_{ij} между бизнес-процессом BP_i и ИТ-сервисом S_j как

$$b_{ij} \in \{0, 1\} \; \forall \; i \in \{1, \ldots, m\} \wedge j \in \{1, \ldots, n\}, \quad (1)$$

где 0 означает, что отношение не существует, бизнес-процесс BP_i не использует ИТ-сервис S_j и 1 означает, что отношение существует, бизнес-процесс BP_i использует ИТ-сервис S_j. Отношения между другими слоями можно моделировать аналогичным образом.

Отношения между слоями BP и IT—S представляют технические поддержки бизнес-процессов ИТ-сервисами, а отношения между слоями IT—S, IS—S и T характеризуются существующим управлением ИБ организации.

Отношения между слоями могут быть смоделированы более точными методами, например, в форме вероятностей, распределений вероятностей или условных вероятностей, с помощью сетей Байеса. Однако определение таких вероятностей является проблематичным.

Более того, основным преимуществом эталонной модели является возможность более детального моделирования каждого слоя.

5. Пересмотр процессов управления рисками

Процессно- и сервис-ориентированное управление рисками ИБ требует некоторого пересмотра (расширения) процессов "традиционного" управления рисками, которое включает этапы идентификации, измерения, обработки и мониторинга (контроля) [5]. В этом разделе рассматриваются эти расширения, а также тема направлений дальнейших исследований.

Идентификация рисков. Стартовой точкой любой методологии управления рисками является идентификация угроз, которые могут быть причинами негативного влияния на конечные результаты организации. Для слоя 1 описанной эталонной модели существуют подробные списки [34], которые обеспечивают продуманную основу для категоризации рисков ИБ. Однако в таких списках лучших мировых практик перечислены известные изолированные угрозы, но нет никакой информации, как учесть причинно-следственные отношения или взаимозависимости между рисками. Эталонная модель для рисков ИБ предоставляет возможность определения и категоризации угроз в соответствии с формально описанным путем из угроз к бизнес-процессам.

Конечно, по крайней мере теоретически, такое расширение значительно увеличивает пространство возможных категорий рисков ИБ. Однако на практике, так как существующие категории можно сохранить и использовать, то это обеспечивает более подробную основу для процессно-ориентированного управления рисками ИБ.

Оценивание рисков. Существуют многочисленные методики управления рисками ИБ. В этих методиках риск ИБ оценивается по двум (угроза, потенциальное воздействие) или по трем (угроза, уязвимость, потенциальное воздействие) факторам. Эти факторы можно измерить качественно или количественно, поэтому большинство методов оценки рисков ИБ основаны на качественном и количественном подходах.

Одной из основных проблем методов оценки рисков ИБ является отсутствие репрезентативной статистики. Единственный путь получения объективных значений параметров факторов риска — накопление статистики по инцидентам ИБ. Предложенная модель позволяет привлечь к сбору статистических данных представителей разных подразделений организации. Работу по накоплению статистики целесообразно вести в рамках процедуры обработки инцидентов.

Накопление статистики по инцидентам помимо получения объективных данных, необходимых для обоснования вложений в ИБ, позволяет оценить эффективность функционирования ИТ-сервисов. Накопленная за определенный временной интервал статистика позволяет отследить общую тенденцию в сторону уменьшения или увеличения числа инцидентов.

Обработка рисков. Обработка рисков — процесс выбора и осуществления мер по модификации рисков. Улучшенное и более точное измерение рисков дает основание для нахождения более подходящих (оптимальных) решений о мерах избежания, смягчения, переноса и/или диверсификации существующих рисков. Ожидается вычисление остаточного риска более точно, с учетом влияний мероприятий на причинно-следственные отношения.

Мониторинг риска. Цель этапа мониторинга рисков заключается в непрерывном сравнении текущего и эталонного состояний, своевременном выявлении рисков, изменении приоритетов и планов преодоления рисков при изменении их вероятности и последствий, активации других этапов при необходимости. Мониторинг осуществляется благодаря информационным отчетам структурных подразделений и отдельных должностных лиц, аналитической деятельности специализированных служб. Отчетность в рамках мониторинга обеспечивает обратную связь.

Для эффективного осуществления мониторинга можно классифицировать риски ИБ и внести в систему мониторинга лишь ключевые для бизнес-процессов риски. При таком подходе средства будут направлены именно на выявление и мониторинг важных для бизнес-процессов рисков.

Результативность системы управления рисками в целом существенным образом зависит от эффективности системы мониторинга. Для автоматизированного обнаружения событий риска и непрерывной оценки влияний изменений на риски ИБ, существующие модели бизнес-процессов должны быть расширены моделями мониторинга. Для этих задач не существуют готовые решения, и они требуют дальнейших исследований.

Заключение

Реализация бизнес-процессов с помощью ИТ-сервисов повышает их эффективность и одновременно увеличивает риски ИБ. Предложенная эталонная модель позволяет формально моделировать отношения между угрозами к ИТ-сервисам и их влияние на бизнес-процессы, классифицировать риски ИБ по степени влияния на бизнес-процессы. Модель объединяет экономическую точку зрения на риски ИБ с технологической с помощью сервисов ИБ. Кроме того, становится очевидным необходимость управления, направленного на конечный результат, который оценивается потребителем — клиентом бизнес-процесса. Использование существующих бизнес-процессов позволяет использовать операционные данные, полученные во время выполнения бизнес-процессов, которые легко доступны. Поэтому сбор и анализ данных выполняются быстро, и это позволяет осуществлять непрерывный мониторинг и оценку рисков ИБ.

Список литературы

1. **Петренко С. А., Симонов С. В.** Управление информационными рисками. Экономически оправданная безопасность. М.: ДМК Пресс, 2004. 384 с.

2. **Астахов А. М.** Искусство управления информационными рисками. М.: ДМК Пресс, 2010. 312 с.

3. **Alberts C. J., Behrens S. G.** et al. Operationally Critical Threat, Asset and Vulnerability Evaluation (OCTAVE) Framework. Pittsburg, Carnegie Mellon. 1999. P. 1—69.

4. **Barber B., Davey J.** The use of the CCTA Risk Analysis and Management Methodology CRAMM // Proc. MEDINFO92, North Holland. 1992. P. 1589—1593.

5. **ISO/IEC** 27005:2007. Information Technology — Security Techniques // Information Security Risk Management. November 2007.

6. **Buyens K., De Win B., Joosen W.** Empirical and Statistical Analysis of Risk Analysis-driven Techniques for Threat Management // The Second International Conference on Availability, Reliability and Security (ARES'07). 2007. P. 1034—1041.

7. **Лопарев С., Шелупанов А.** Анализ инструментальных средств оценки рисков утечки информации в компьютерной сети предприятия // Вопросы защиты информации. 2003. № 4.

8. **Hamdi M., Boudriga N.** Computer and Network Security Risk Management: Theory, Challenges, and Countermeasures // International Journal of Communication Systems. 2005. Vol. 18, N 8. P. 763—793.

9. **Марков А. С., Цирлов В. Л.** Управление рисками — нормативный вакуум // Открытые системы. — 2007. № 7. URL: http://www.osp.ru/os/2007/08/4492873/ (дата обращения 15.01.10).

10. **McHugh J.** Quality of Protection: Measuring the Unmeasurable? // Proc. of the 2nd ACM Workshop on Quality of Protection (QoP'06), 2006. P. 1—2.

11. **Jaquith A.** Security Metrics: Replacing Fear, Uncertainty, and Doubt. NJ: Addison-Wesley Pearson Education, 2007. 336 p.

12. **Шеер А. В.** Бизнес-процессы Основные понятия, теория, методы. М.: Весть-МетаТехнология, 2005. — 173 с.
13. **Андерсен Б.** Бизнес-процессы. Инструменты совершенствования. М.: Стандарты и качество, 2007. 272 с.
14. **Giaglis G. M.** A Taxonomy of Business Process Modeling and Information Systems Modeling Techniques // International Journal of Flexible Manufacturing Systems. 2001. 13 (2). P. 209—228.
15. **Корт С. С.** Теоретические основы защиты информации: учеб. пособ. М.: Гелиос АРВ, 2004. 240 с.
16. **Jakoubi S., Tjoa S., Goluch G., Quirchmayr G.** A Survey of Scientific Approaches Considering the Integration of Security and Risk Aspects into Business Process Management // Proc. 20th International Workshop on Database and Expert Systems Application (DEXA'09). 2009. P. 127—132.
17. **Suh B., Han I.** The IS Risk Analysis Based on a Business Model // Information & Management. 2003. Vol. 41. P. 149—158.
18. **Neubauer T., Klemen M. D., Biffl S.** Business Process-based Valuation of IT-Security // ACM SIGSOFT Software Engineering Notes. 2005. Vol. 30, N 4. P. 1—5.
19. **Tjoa S., Jakoubi S., Quirchmayr G.** Enhancing Business Impact Analysis and Risk Assessment applying a Risk-Aware Business Process Modeling and Simulation Methodology // Third International Conference on Availability, Reliability and Security (ARES 08). 2008. P. 179—186.
20. **Herrmann P., Herrmann G.** Security Requirement Analysis of Business Processes // Electron Commerce Research. 2006. Vol. 6, N 3—4. P. 305—335.
21. **Sackmann S.** A Reference Model for Process-Oriented IT Risk Management // Proceedings of the 16th European Conference on Information Systems (ECIS 2008), 2008. URL: http://is2.lse.ac.uk/asp/aspecis/20080114.pdf (дата обращения 15.01.10).
22. **Sackmann S.** Assessing the Effects of IT Changes on IT Risk — A Business Process-Oriented View // Multikonferenz Wirtschaftsinformatik MKWI'08, Berlin; GITO-Verlag, 2008. P. 1137—1148.
23. **Halliday S., Badenhorst K., von Solms R.** A Business Approach to Effective Information Technology Risk Analysis and Management // Information Management & Computer Security. 1996. Vol. 4, N 1. P. 19—31.
24. **Roehrig S., Knorr K.** Security Analysis of Electronic Business Processes // Electronic Commerce Research. 2004. Vol. 4. P. 59—81.
25. **Rodriguez A., Fernandez-Medina E., Piattini M.** A BPMN Extension for the Modeling of Security Requirements in Business Processes // IEICE — Transactions on Information and Systems. 2007. Vol. E90-D, N 4. P. 745—752.
26. **Экономическая** информатика: введение в экономический анализ информационных систем: Учебник / М. И. Лугачев и др. М.: ИНФРА-М, 2005. 958 с.
27. **Грекул В. И., Денищенко Г. Н., Коровкина Н. Л.** Проектирование информационных систем. М.: Интернет-университет информационных технологий — ИНТУИТ.ру, 2008. 304 с.
28. **ISO/IEC** 20000-1:2005. Information Technology — Service Management. Part 1: Specification.
29. **Galup S., Quan J. J., Dattero R., Conger S.** Information Technology Service Management: an Emerging Area for Academic Research and Pedagogical Development // Proc. of the 2007 ACM SIGMIS Conference on Computer Personnel Research — SIGMIS-CPR '07. 2007. P. 46—52.
30. **Таненбаум Э.** Компьютерные сети. СПб.: Питер, 2003. 992 с.
31. **Дэвис Р., Брабендер Э.** BPM для начинающих. Моделирование бизнеса с ARIS Design Platform: пер с англ. М.: Серебряные нити, 2008. 436 с.
32. **Компас** в мире сервис-ориентированной архитектуры (SOA): ценность для бизнеса, планирование и план развития предприятия / Н. Биберштейн и др. — М.: Кудиц-пресс, 2007. 256 с.
33. **Тарасюк М. В.** Механизмы и сервисы безопасности информационных технологий: учеб. пособ. — СПб.: СПбГИТМО (ТУ), 2002. 88 с.
34. **BSI IT** Baseline Protection Manual. Bundesamt für Sicherheit in der Informationstechnik. 2000. URL: http://www.iwar.org.uk/comsec/resources/standards/germany/itbpm.pdf (дата обращения 15.01.10).

УДК 621.395

М. А. Дрюченко, аспирант,
e-mail: aldram@box.vsi.ru,
А. А. Сирота, д-р техн. наук, проф.,
e-mail: sir@cs.vsu.ru,
Воронежский государственный университет

Нейросетевые модели и алгоритмы стеганографического скрытия информации

Рассматривается возможность использования аппарата искусственных нейронных сетей в интересах создания стойкой стеганографической системы. Описывается статистическая модель стеганографического скрытия и извлечения информации на основе использования нейронных сетей прямого распространения. Исследуется стеганографическая стойкость предлагаемого метода и приводятся примеры его реализации.

Ключевые слова: *стеганографическое скрытие, контейнер, заполненный контейнер, нейронная сеть*

Введение и постановка задачи

Современные методы компьютерной стеганографии позволяют эффективно решать задачи защиты конфиденциальной информации, авторских прав на некоторые виды интеллектуальной собственности (*copyright*), а также скрытного хранения данных и их передачи в системах телекоммуникации. Принято выделять два основных метода компьютерной стеганографии [1]. К ним относятся методы, основанные на использовании специальных свойств компьютерных форматов данных, и методы, основанные на избыточности аудио- и видеоинформации. Одним из возможных подходов к дальнейшему развитию технологий скрытия данных является использование аппарата искусственных нейронных сетей (ИНС). Так, в работах [2, 3] описывается возможность использования нейронных сетей в интересах стеганографического скрытия информации (ССИ). В частности, в статье [2] с помощью специально обученной нейронной сети из битов контейнера (это файл или поток данных, структура и размер которого позволяют

"спрятать" необходимые данные) и битов встраиваемых данных проводится формирование новой битовой последовательности, которая в дальнейшем записывается в наименее значимые биты файла-контейнера. В то же время необходимо отметить, что большинство известных алгоритмов ССИ имеет один общий недостаток: используемые процедуры встраивания реализуют строго определенные последовательности операций преобразования данных, образующих конечное множество вариантов.

Основное отличие предлагаемого в данной работе подхода заключается в том, что специально обученные нейронные сети используются не для генерации стеганограмм из исходных данных (и дальнейшему их встраиванию по одному из известных алгоритмов), а для реализации самого скрывающего преобразования. В рамках этого подхода реализуется метод ССИ на основе нейросетевых функциональных моделей преобразования данных, при использовании которых процесс встраивания данных в файл-контейнер носит существенно менее прозрачный характер. Целью работы является разработка и исследование алгоритмов ССИ, реализуемых в рамках предлагаемого метода, анализ их стеганографической стойкости, оценка возможности извлечения скрытых данных при известном нарушителю алгоритме встраивания, а также обоснование и выработка правил по выбору.

Сформулируем в общем виде постановку задачи ССИ. Пусть Z, D, K есть соответственно множество возможных контейнеров, множество скрываемых сообщений и множество ключей, тогда процедура встраивания сообщений может быть представлена в виде отображения

$$F: Z \times D \times K \to \tilde{Z},$$
$$\tilde{z} = F(z, d, k),\ z \in Z,\ d \in D,\ k \in K,$$

где $\tilde{Z}$ — множество заполненных контейнеров. При этом требуется обеспечить $\|z - \tilde{z}\| \to \min$ и $F(z, d, k) \approx F(z + \varepsilon, d, k)$, т. е. свойства контейнера должны модифицироваться так, чтобы изменение контейнера, внесенное при встраивании данных стегосообщения, практически невозможно было бы выявить при визуальном и статистическом контроле, а само стегосообщение должно быть максимально устойчиво к различного рода искажениям.

Соответственно, постановка задачи ССИ с использованием аппарата ИНС может быть сформулирована следующим образом. Требуется с использованием функциональных возможностей нейронных сетей для любого вектора-контейнера $z \in R^n$ и вектора встраиваемых данных $d \in R^m$, $m \ll n$ построить отображения

$$\tilde{z} = F_1(z, d),\ \tilde{z} \in \tilde{Z},\ \|\tilde{z} - z\| \to \min,$$
$$\tilde{d} = F_2(\tilde{z}),\ \tilde{d} \in D,\ \|d - \tilde{d}\| \to \min,$$

где первый оператор реализует встраивание, а второй — восстановление информации.

Нейросетевые модели ССИ

В общем случае для решения данной задачи целесообразно использовать нейронные сети различных типов и архитектур, которые обеспечивают принципиальные возможности воспроизведения функциональных моделей преобразования данных в соответствии с приведенными общими соотношениями. Будем в простейшем варианте искать указанные отображения на основе *линейных нейронных сетей прямого распространения*.

Нейронную сеть (НС), реализующую оператор F_1, естественно искать в виде автоассоциативной нейронной сети, структура которой в общем виде приведена на рис. 1, *а*. Число нейронов в скрытом слое здесь $q \leqslant n + m$. Входной сигнал может быть представлен как составной вектор $y = (z^{\mathrm{T}}, d^{\mathrm{T}})^{\mathrm{T}}$ или $y = y_1 + y_2$, где $y_1 = (z_1, z_2, \ldots, z_n, 0, \ldots 0)^{\mathrm{T}}$, $y_2 = (0, \ldots 0, d_1, \ldots, d_m)^{\mathrm{T}}$, где $d = (d_1, \ldots, d_m)^{\mathrm{T}}$ — вектор, содержащий элемент встраиваемых данных.

В целях эффективного встраивания и последующего восстановления данных в ходе анализа заполненного контейнера $\tilde{z}$ необходимо осуществить "замешивание" z и d при выполнении преобразования F_1 с использованием ИНС, изображенной на рис. 1, *а*. При встраивании сообщения, образующего последовательность $d^{(p)}$, $p = \overline{1, P}$, для каждого ее элемента используется фрагмент контейнера, описываемый вектором $z^{(p)}$, $p = \overline{1, P}$. Соответственно, на выходе соответствующим образом обученной сети получается последовательность заполненных фрагментов контейнера $\tilde{z}^{(p)}$, $p = \overline{1, P}$. Далее без ограничения общности будем считать,

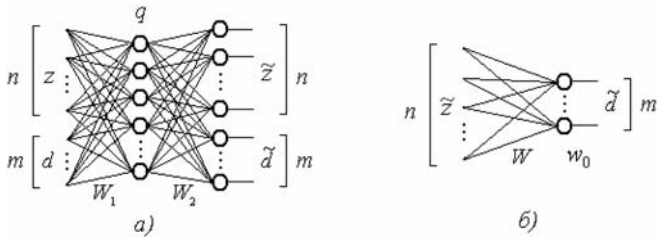


Рис. 1. Нейронная сеть, реализующая встраивание информации (*а*) и реализующая восстановление скрытой информации (*б*)

что размерность вектора сообщения $m = 1$, при этом $d \in \{-1, +1\}$ — скалярная величина, которая несет в себе один бит информации. Для восстановления скрываемых данных может быть использована вторая НС, архитектура которой показана на рис. 1, *б*. Данная сеть при подаче на нее последовательности входных сигналов $\tilde{z}^{(p)}$, $p = \overline{1, P}$, должна решать задачу классификации данных в целях выделения последовательности $\tilde{d}^{(p)}$, $p = \overline{1, P}$.

Статистический анализ процесса ССИ

Для анализа закономерностей процесса ССИ целесообразно рассмотреть следующую статистическую модель.

Пусть z — случайный вектор с математическим ожиданием $M[z] = 0$ и матрицей ковариации $M[zz^{\mathrm{T}}] = R_z$. Пусть d является случайной величиной, не зависящей от z и принимающей свои значения с одинаковыми априорными вероятностями $P(d = 1) = 0{,}5$ и $P(d = -1) = 0{,}5$. Тогда $M[d] = 0$ и дисперсия $M[d^2] = \sigma_d^2 = 1$. Обучение сети, приведенной на рис. 1, *а*, проводится по совокупности реализаций входного вектора $y^{(p)} = (z^{(p),\mathrm{T}}, d^{(p)})^{\mathrm{T}}$, $p = \overline{1, P}$, так, чтобы минимизировать величину

$$E = \frac{1}{2} \sum_{p=1}^{P} (y^{(p)} - W_2 W_1 y^{(p)})^{\mathrm{T}} (y^{(p)} - W_2 W_1 y^{(p)}). \quad (1)$$

Если число нейронов в скрытом слое совпадает с числом нейронов на выходе НС $q = n + m$, тогда эквивалентное преобразование, выполняемое НС, сходится к матрице $W = W_2 W_1 = I$ [4]. Это означает, что при $P \to \infty$ статистическая структура данных на выходе НС по отношению ко входу не изменяется. При последующем (после обучения) встраивании она будет определяться матрицей ковариации вектора y

$$R_y = \left(\begin{array}{c|c} R_z & 0 \\ \hline 0 & y_d^2 \end{array} \right),$$

т. е. компоненты $\tilde{z}$ и $\tilde{d}$ выходного вектора $\tilde{Y}$ будут некоррелированы. Это означает, что в данном случае практического встраивания данных в контейнер не происходит.

Пусть теперь $q \leqslant n$, т. е. в скрытом слое на один или более нейронов меньше, чем на выходе НС. В этом случае можно показать, что результирующее преобразование, выполняемое линейной двухслойной автоассоциативной нейронной сетью, приведенной на рис. 1, *а*, обучаемой для минимизации (1), эквивалентно применению линейного оператора вида

$$W = \hat{R}_{y11} \hat{R}_{y11}^{+} = W_2 W_1,$$

где $\hat{R}_{y11}$ — вырожденная матрица, полученная из выборочной по совокупности $y^{(p)}$, $p = \overline{1, P}$, $(P > m + n)$ матрицы ковариации R_{y11} при выполнении диагонализирующего преобразования и приравнивания к нулю $m + n - q$ ее последних собственных чисел; $\hat{R}_{y11}^{+}$ является псевдообратной матрицей для $\hat{R}_{y11}$:

$$\hat{R}_{y11} = \sum_{i=1}^{q} \lambda_q \tilde{\varphi}_i \tilde{\varphi}_i^{\mathrm{T}}, \quad \hat{R}_{y11}^{+} = \sum_{i=1}^{q} \frac{1}{\lambda_q} \tilde{\varphi}_i \tilde{\varphi}_i^{\mathrm{T}},$$

$$\tilde{\varphi}_i \tilde{\varphi}_j^{\mathrm{T}} = \begin{cases} 1, & i = j; \\ 0, & i \neq j. \end{cases}$$

Здесь изначально $\tilde{R}_{y11} = H \Lambda H^{\mathrm{T}}$, где H — матрица, столбцами которой являются собственные векторы $\tilde{R}_{y11}$: $H = (\tilde{\varphi}_1, ..., \tilde{\varphi}_n)$; $\Lambda = \mathrm{diag}\{\lambda_1, ..., \lambda_{n+m}\}$ — диагональная матрица собственных чисел.

Действительно, как показано в работе [4], при подаче на обученную подобным образом НС входного вектора y_1 на выходе будет получен сигнал в виде вектора

$$\tilde{y}_1 = \sum_{i=1}^{q} \alpha_i \tilde{\varphi}_i,$$

где α_i, $i = \overline{1, q}$, — коэффициенты разложения по первым $q \leqslant n$ собственным векторам $\tilde{R}_{y11}$.

При использовании указанной формы представления выполняемого НС преобразования вектор данных, формируемый в скрытом слое, имеет вид

$$\gamma = \hat{R}_{y11}^{+} y_1 = \sum_{i=1}^{q} \frac{1}{\lambda_i} \tilde{\varphi}_i \tilde{\varphi}_i^{\mathrm{T}} \left(\sum_{j=1}^{n} \alpha_j \tilde{\varphi}_j \right) = \sum_{j=1}^{q} \alpha_i \frac{1}{\lambda_i} \tilde{\varphi}_i.$$

Соответственно, вектор, получаемый на выходе НС, имеет вид

$$\tilde{y}_1 = \hat{R}_{y11} \gamma = \hat{R}_{y11} \hat{R}_{y11}^{+} y_1 = \sum_{j=1}^{q} \lambda_j \tilde{\varphi}_i \tilde{\varphi}_i^{\mathrm{T}} \sum_{i=1}^{q} \alpha_i \frac{1}{\lambda_i} \tilde{\varphi}_i =$$

$$= \sum_{j=1}^{q} \alpha_j \lambda_j \frac{1}{\lambda_j} \tilde{\varphi}_j = \sum_{j=1}^{q} \alpha_j \tilde{\varphi}_j.$$

Таким образом, при выполнении НС рассматриваемой архитектуры преобразования "вход—выход" осуществляется сжатие данных [4] с изменением их статистической структуры на выходе по отношению ко входу. В частности, при $q = n$ это преобразование сопровождается незначительным искажением вектора-контейнера и возможным "замешиванием" вектора d в вектор $\tilde{z}$, являющийся составной частью вектора, получаемого на выходе НС.

Теперь, чтобы проанализировать и оценить возможность восстановления данных в процессе ССИ, выполним анализ статистических характеристик вектора $\tilde{z}$ на выходе обученной после встраивания сети. Для этого предлагается следующая методика.

1. На вход сети подаются тестовые сигналы $y^+ = (0, 0, \ldots 0, 1)^{\mathrm{T}}$, $y^- = (0, 0, \ldots 0, -1)^{\mathrm{T}}$. При этом на выходе получаются векторы $\tilde{y}^+ = (m^+, \tilde{d})^+ = W_2 W_1 y^+$ и $\tilde{y}^- = (m^-, \tilde{d})^- = W_2 W_1 y^-$. Компоненты m^+ и m^- рассматриваются как математические ожидания полезного сигнала, соответствующие двум различным гипотезам при встраивании данных в контейнер $\tilde{z}$.

2. Оценивается матрица ковариации выхода сети в первых n компонентах, т. е. компонентах вектора $\tilde{z}$, при подаче случайного вектора $y_1 = (z_1, z_2, \ldots, z_n, 0)^{\mathrm{T}}$. Для этого вычисляется матрица

$$R^0_{\tilde{y}} = W_2 W_1 R^0_y W_1^{\mathrm{T}} W_2^{\mathrm{T}}, \; R^0_y = \begin{pmatrix} R_z & 0 \\ 0 & 0 \end{pmatrix}$$

и выделяется матрица $R_{\tilde{z}}$, являющаяся блочной в матрице $R^0_{\tilde{y}}$.

3. В итоге сигнал на выходе НС представляется в виде

$$\tilde{z} = W_2^y W_1 (Y_1 + Y_2), \; \tilde{z} = am^+ + (1 - a)m^- + n,$$

где $W_2^y = \| W_{ij}^{(2)} \|$, $i = \overline{1, n}$, $j = \overline{1, n}$; $W_{ij}^{(2)}$ — элементы матрицы W_2, $i = \overline{1, n+1}$, $j = \overline{1, n}$, где $a = 1$, если $d = 1$, $a = 0$, если $d = -1$; n — вектор флуктуаций (шума) с известной матрицей ковариации $R_n = R_{\tilde{z}}$.

Таким образом, для эффективного восстановления скрытых данных необходимо решить задачу классификации наблюдаемого вектора $\tilde{z}$ по его принадлежности к одному из классов H_1 и H_2, характеризующихся различными математическими ожиданиями m^+ и m^-, в присутствии шума n с известной матрицей ковариации R_n. Для этого используется вторая НС, архитектура которой показана на рис. 1, *б*. Заметим, что оптимальное по критерию максимума правдоподобия решающее правило в случае гауссовского распределения вектора n имеет вид

$$l(\tilde{z}) = \tilde{z}^{\mathrm{T}} R_n^{-1} (m^+ - m^-) - \\ - 0{,}5(m^+ + m^-)^{\mathrm{T}} R_n^{-1} (m^+ - m^-) \underset{<}{\overset{>}{}} 0. \qquad (2)$$

Вероятность суммарной ошибки при этом определяется соотношением

$$P_{OC} = \frac{1}{2} P_{12} + \frac{1}{2} P_{21} = P_{12} = 1 - \Phi(\alpha); \qquad (3)$$

$$\Phi^{(x)} = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} \mathbf{e}^{-\frac{t^2}{2}} dt,$$

$$\alpha = 0{,}5 \sqrt{(m^+ - m^-)^{\mathrm{T}} R_n^{-1} (m^+ - m^-)}.$$

Можно показать, что в рассматриваемом случае указанное решающее правило получается путем обучения простейшей однослойной линейной НС, структура которой приведена на рис. 1, *б*. При этом может быть сформулировано следующее утверждение: в результате обучения однослойной линейной НС (рис. 1, *б*) по совокупности $\{\tilde{z}^{(p)}, d^{(p)}, p = \overline{1, P}\}$ при $P \to \infty$ формируется преобразование входных данных в выходные, реализующее структуру оптимального решающего правила, сходную с формулой (2); вероятность ошибки при восстановлении ССИ на выходе этой НС стремится к величине, определяемой приведенным выше соотношением (3).

Данный результат получается путем нахождения необходимого и достаточного условия минимума целевой функции

$$E = \frac{1}{2} \sum_{p=1}^{P} (\tilde{d}^{(p)} - W\tilde{z}^{(p)} - w_0)^{\mathrm{T}} (\tilde{d}^{(p)} - W\tilde{z}^{(p)} - w_0)$$

при решении уравнений для частных производных по элементам вектора смещения и матрицы весов нейронной сети w_0, W

$$\frac{\partial E}{\partial w_0} = 0, \; \frac{\partial E}{\partial W} = 0,$$

а также доказательства положительной определенности матрицы вторых частных производных E по элементам w_0, W. В итоге после громоздких, но несложных преобразований можно показать, что

при подаче на вход НС вектора $\tilde{z}$ сигнал на выходе определяется следующим соотношением:

$$d = W\tilde{z} + w_0 = -a'\tilde{z}^{\mathrm{T}} S^{-1}(\tilde{m}_z^+ - \tilde{m}_z^-)^{\mathrm{T}} -$$
$$- a'\left(\frac{P_1}{P}\tilde{m}_z^+ + \frac{P_2}{P}\tilde{m}_z^-\right) S^{-1}(\tilde{m}_z^+ - \tilde{m}_z^-)^{\mathrm{T}} + m_d;$$

$$\tilde{m}_z^+ = \frac{1}{P}\sum_{\tilde{z}^{(p)} \in Z^+} \tilde{z}^{(p)};\ \tilde{z}^{(p)} = n^{(p)} + m^+ \in Z^+;$$

$$\tilde{m}_z^- = \frac{1}{P_1}\sum_{\tilde{z}^{(p)} \in Z^-} \tilde{z}^{(p)};\ \tilde{z}^{(p)} = n^{(p)} + m^- \in Z^-;$$

$$m_d = \frac{P_1}{P} - \frac{P_2}{P};\ S = \frac{P_1}{P}R_{zz}^+ + \frac{P_2}{P}R_{zz}^-;$$

$$R_{zz}^+ = \frac{1}{P_1}\sum_{\tilde{z}^{(p)} \in Z^+} (\tilde{z}^{(p)} - \tilde{m}_z^+)(\tilde{z}^{(p)} - \tilde{m}_z^+)^{\mathrm{T}};$$

$$R_{zz}^- = \frac{1}{P_2}\sum_{\tilde{z}^{(p)} \in Z^-} (\tilde{z}^{(p)} - \tilde{m}_z^-)(\tilde{z}^{(p)} - \tilde{m}_z^-)^{\mathrm{T}},$$

где a' — константа; P_1, P_2 — число обучающих патернов для первого и второго классов ($P_1 + P_2 = P$). Из представленного выражения следует, что матрица весов НС пропорциональна обратной матрице выборочной ковариации шума S. Если $P_1 = P_2$, то выражение для решающего правила записывается в виде

$$d(\tilde{z}) = \tilde{z}^{\mathrm{T}} S^{-1}(\tilde{m}_z^+ - \tilde{m}_z^-)^{\mathrm{T}} -$$
$$- \frac{1}{2}(\tilde{m}_z^+ - \tilde{m}_z^-)^{\mathrm{T}} S^{-1}(\tilde{m}_z^+ - \tilde{m}_z^-) \gtrless 0.$$

Необходимо отметить, что полученная структура решающего правила аналогична приведенной в работе [5]. Выполняемое нейронной сетью, изображенной на рис. 1, *б*, преобразование по своей структуре имеет вид линейной разделяющей функции и при $P \to \infty$ воспроизводит структуру оптимального решающего правила для гауссовских векторов (2).

Обобщенная схема встраивания и извлечения информации с использованием НС, изображенных на рис. 1, приведена на рис. 2.

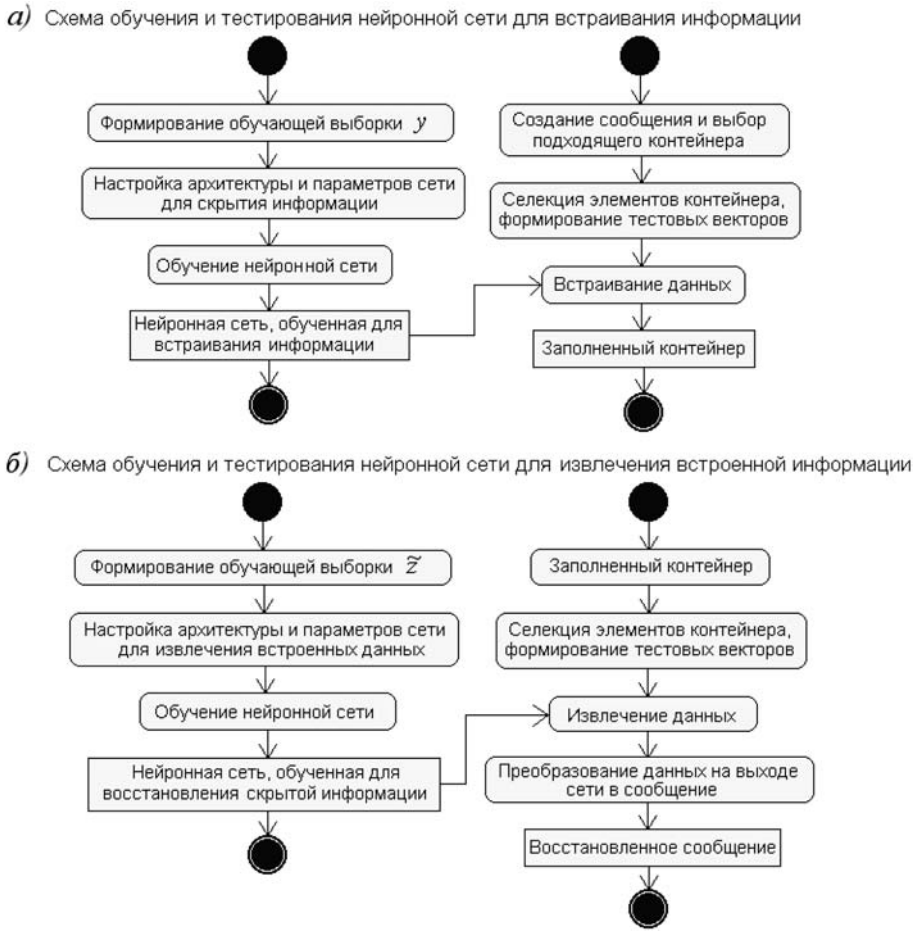


Рис. 2. Обобщенная схема встраивания (*а*) и извлечения (*б*) информации с использованием нейронных сетей

Схема встраивания информации включает два основных этапа. На первом этапе выполняется формирование обучающих множеств, настраиваются параметры НС (см. рис. 1, *а*) и происходит процесс ее обучения. Одновременно по сформированным векторам входных и целевых воздействий осуществляется обучение НС, реализующей процедуру восстановления (см. рис. 1, *б*). На втором этапе выбирается подходящий контейнер, формируются векторы входных воздействий и реализуется нейросетевой алгоритм ССИ. Извлечение данных реализуется путем подачи элементов контейнера, содержащих встроенное сообщение, на вторую НС.

Исследование предлагаемого метода ССИ

Рассмотренная статистическая модель позволяет исследовать закономерности процесса ССИ, например, влияние размерности и спектральных характеристик контейнера на достоверность и объем встраиваемой информации при заданном уровне искажений контейнера. Моделирование про-

цесса ССИ осуществлялось с использованием пакета MatLab 7.0. При проведении модельного эксперимента выполнялась генерация совокупностей встраиваемой последовательности $d^{(p)}$ и вектора-контейнера $z^{(p)}$, рассматриваемых как реализации случайного процесса с заданным коэффициентом корреляции ρ и дисперсией σ_n^2. Важной особенностью предлагаемого алгоритма ССИ является возможность значительного уменьшения амплитуды встраиваемого сигнала по сравнению с сигналом, использованным для обучения сети. При этом существенно уменьшается искажение контейнера, а НС, обученная для извлечения информации, способна успешно восстанавливать исходное сообщение. На рис. 3, *а*, *б* приведены зависимости вероятности ошибки при восстановлении одного бита встраиваемой информации от отношения амплитуды встраиваемой последовательности к σ_n, на рис. 3, *в* приведена зависимость вероятности ошибки восстановления скрытой информации от размерности входного вектора z, на рис. 3, *г* представлена зависимость вероятности ошибки восстановления скрытой информации от коэффициента корреляции ρ случайного процесса, генерирующего реализации вектора z. С увеличением размерности вектора-контейнера z уменьшается среднеквадратическая ошибка (СКО) искажения контейнера и ошибка восстановления скрытых данных. Одновременно уменьшается пропускная способность стегосистемы. Приемлемые значения пропускной способности должны определяться в рамках конкретной задачи ССИ с учетом обеспечения низкой вероятности вскрытия стегосистемы.

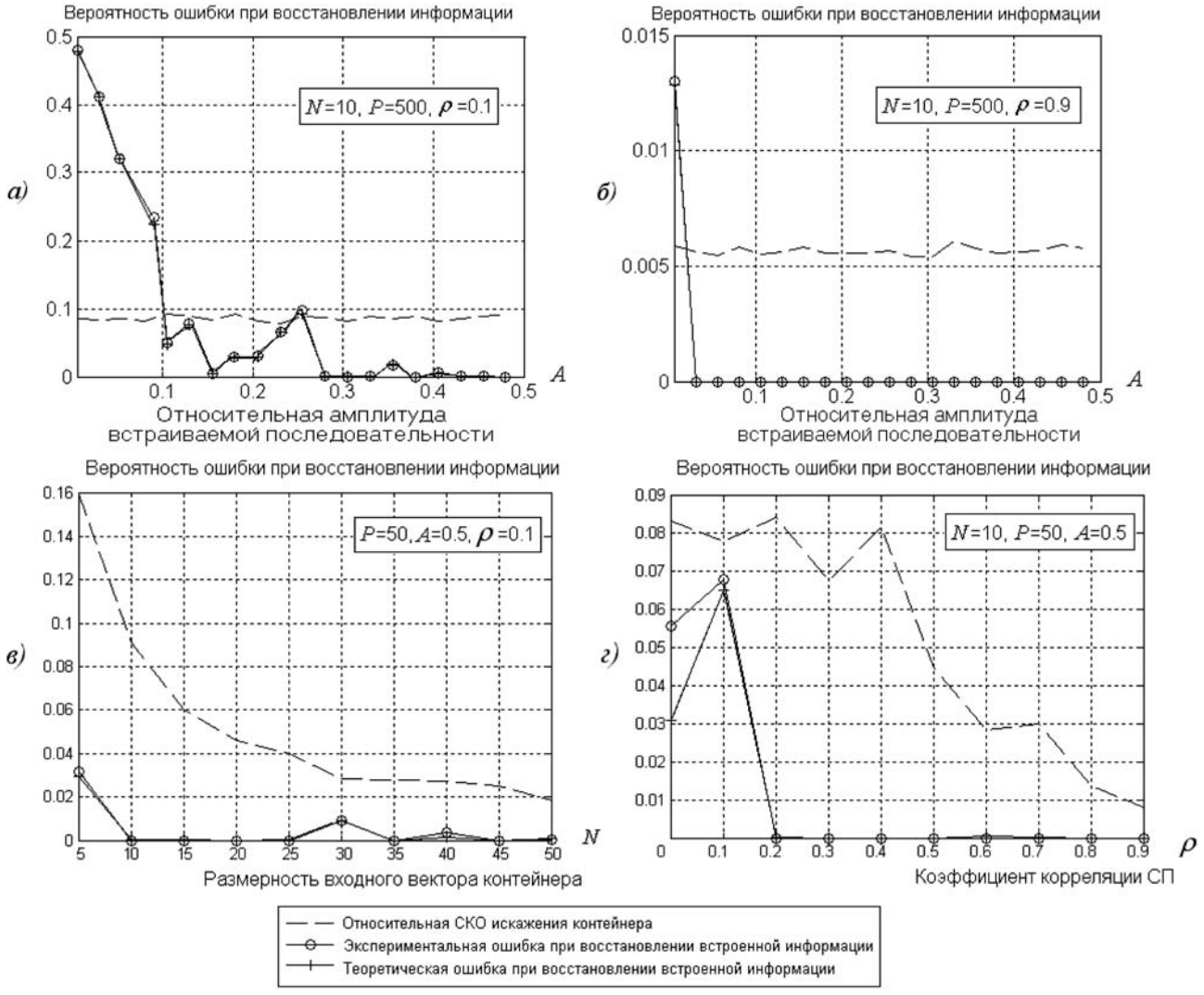


Рис. 3. Зависимости вероятности ошибки при восстановлении одного бита встраиваемой информации от отношения амплитуды встраиваемой последовательности к σ_n (*а*, *б*); зависимость вероятности ошибки восстановления скрытой информации от размерности входного вектора z (*в*); зависимость вероятности ошибки восстановления скрытой информации от коэффициента корреляции ρ случайного процесса, генерирующего реализации вектора z (*г*)

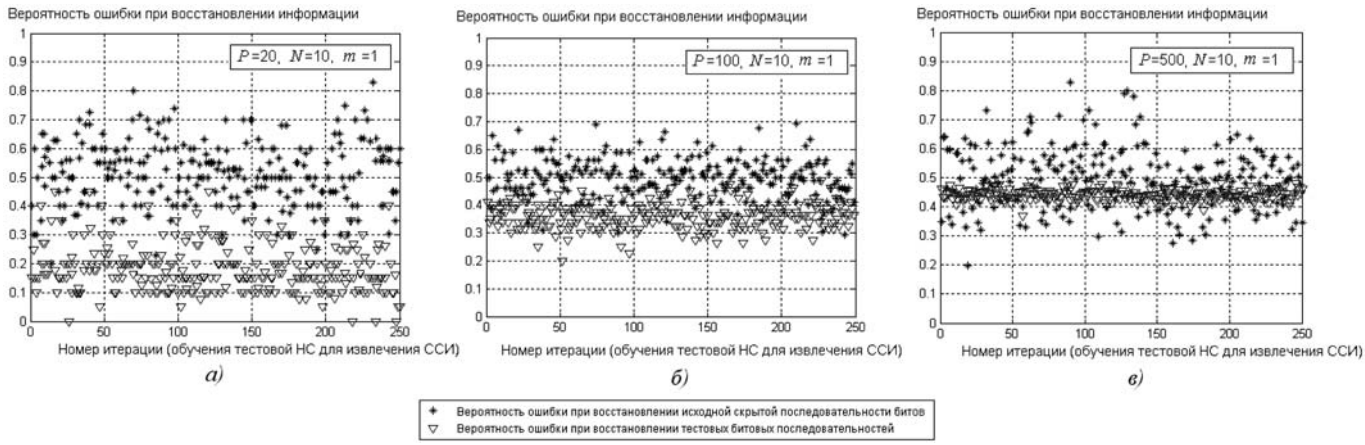


Рис. 4. Вероятности ошибок при восстановлении битов встроенных данных при обучении "ложных" нейронных сетей

В предлагаемом методе нейросетевого встраивания информации в качестве стеганографического ключа $K = \{\theta, T\}$ выступают параметры обученной для скрытия нейронной сети θ (ее архитектура, число нейронов по слоям и значения весовых коэффициентов), а также информация T о выбранных для встраивания элементах контейнера. Например, для скрытия может использоваться не весь контейнер целиком, а определенная его область. Если в качестве контейнера используется изображение, то для скрытия могут выбираться отдельные точки на изображении в последовательности, определенной с помощью криптографически безопасного генератора ПСП с известным обеим сторонам начальным значением.

Известно, что безопасность стегосистемы может полностью определяться секретностью ключа [1]. В этом случае нарушителю могут быть известны все алгоритмы работы стегосистемы и статистические характеристики множеств сообщений и контейнеров, но это не даст ему никакой дополнительной информации о наличии или отсутствии сообщения в данном контейнере. В нашем случае безопасность системы определяется как ключом, так и самим нейросетевым алгоритмом встраивания.

Оценка возможности извлечения скрытых данных при известных нарушителю принципах скрытия. По аналогии с криптоанализом в стегоанализе выделяют несколько типов атак, основанных на полноте информации о стегосистеме, которой обладает атакующая сторона, и возможности атакующего напрямую или опосредованно вмешиваться в работу стегосистемы.

Рассмотрим атаку на основе *известного заполненного контейнера*. Она предполагает, что у нарушителя есть один или несколько контейнеров, заполненных по одному стегоалгоритму. Нарушитель пытается обнаружить стегоканал и определить ключ. Определив ключ, нарушитель получит возможность анализа и извлечения встроенной информации из других контейнеров. В случае нейросетевого алгоритма ССИ определить ключ — значит подобрать необходимые параметры и обучить "ложную" НС, способную с минимальной ошибкой восстанавливать сообщение.

Предположим, что нарушителю известен набор элементов контейнера, содержащих встроенную информацию, или известен алгоритм, по которому выбираются элементы контейнера при формировании тестовой выборки для НС, реализующей стеганографическое скрытие. Пусть, кроме того, атакующей стороне известна размерность наборов данных, т. е. число входов n, число нейронов на выходе НС m, и возможно, число нейронов в скрытых слоях. Обладая информацией о $\tilde{z}$, нарушитель может попытаться сконструировать и обучить "ложную" НС и использовать ее для извлечения встроенных данных. В качестве целевых данных, предъявляемых на выходе НС, представленных на рис. 1, *б*, он может сгенерировать набор реализаций случайных векторов $g^{(i)}$, $i = \overline{1, K}$, где K — число таких реализаций. На рис. 4 представлены вероятности ошибок при восстановлении битов встроенных данных. Треугольными маркерами отмечены вероятности ошибок при восстановлении сгенерированных случайных векторов $g^{(i)}$, участвовавших в обучении "ложных" НС. Звездочками обозначены вероятности ошибок восстановления искомой скрытой последовательности d. Во всех трех случаях для различных значений P (длины обучающей выборки) вероятности ошибок восстановления искомой последовательности колеблются около 0,5, т. е. при известных нарушителю заполненном контейнере и принципах встраивания и извлечения данных обучаемая "ложная" НС не способна восстановить скрытую битовую последовательность. Кроме того, на рис. 4 видно, что с увеличением длины обучающей выборки тестовые нейронные сети обучаются хуже и вероятности

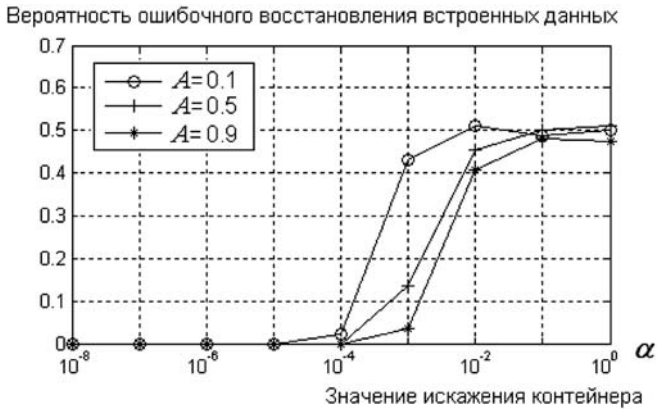


Рис. 5. Зависимости вероятности ошибочного восстановления встроенных данных при заданном уровне шума α, вносимого в контейнер для различных значений амплитуды встраиваемой последовательности *A*

ошибок восстановления данных возрастают и стремятся к 0,5.

Оценка возможности извлечения скрытых данных при намеренном искажении контейнера. Если у нарушителя имеется возможность внесения некоторого искажения в контейнер, передаваемый по стегоканалу, то это может привести к потере скрытого сообщения, если не использовать специальные методы (например, помехоустойчивое кодирование, расширение спектра сигналов и др.). На рис. 5 приведены зависимости вероятности ошибочного восстановления встроенных данных при заданном уровне шума, вносимого в контейнер для различных значений амплитуды встраиваемой последовательности A. Искаженный контейнер можно представить в виде

$$\tilde{z}' = \tilde{z} + \alpha v,$$

где $v = (v_1, ..., v_n)^T$ — случайный вектор возмущения; α — константа (амплитуда шума). Очевидно, что с увеличением амплитуды шумовых составляющих, добавляемых к контейнеру, вероятность ошибочного восстановления скрытой информации будет возрастать. Кроме того, на вероятность ошибочного восстановления в значительной степени влияет амплитуда встраиваемого сигнала — чем она меньше, тем сложнее восстановить исходный сигнал в присутствии шума.

Дальнейшее обсуждение вопроса робастности нейросетевого алгоритма ССИ по отношению к намеренным искажениям целесообразно вести исходя из выбора конкретного типа контейнера.

Пример и рекомендации по выбору контейнера

В предложенной функциональной модели ССИ для эффективного обучения НС входной сигнал $y = (z^T, d^T)^T$, как правило, подвергается нормировке, поэтому, чтобы не проводить дополнительных преобразований значений векторов контейнера из целочисленной в вещественную форму и обратно, в качестве контейнеров целесообразно выбирать файлы, содержащие большие массивы вещественных значений. Например, можно использовать форматы файлов для описания геометрии 3D-объектов. В этом случае скрытие информации происходит не в отдельные байты контейнера, а в само содержимое файла. Одним из простых и распространенных форматов для описания геометрии является формат файлов OBJ. Он содержит только 3D-геометрию, а именно: позицию каждой вершины, связь координат текстуры с вершиной, нормаль для каждой вершины, а также параметры, которые создают полигоны. В левой части таблицы приведен фрагмент исходного OBJ-файла, содержащий координаты вертексов (точек в трехмерном пространстве), в правой части — фрагмент этого же файла после работы нейросетевого алгоритма скрытия.

Фрагменты OBJ-файла описания геометрии до и после стеганографического скрытия

Исходный OBJ-файл				OBJ-файл, полученный в результате нейросетевого ССИ			
v	−0,700112	0,597674	−2,918275	v	−0,70107	0,59957	−2,91675
v	−0,699406	0,589033	−2,866541	v	−0,69903	0,59169	−2,86246
v	−0,704698	0,653815	−2,90036	v	−0,70328	0,65730	−2,89601
v	−0,665931	0,597674	−2,918275	v	−0,66136	0,59813	−2,91592
v	−0,666608	0,589033	−2,866541	v	−0,66604	0,59032	−2,86218
v	−0,66221	0,645174	−2,848626	v	−0,66121	0,64589	−2,84596
v	−0,661533	0,653815	−2,90036	v	−0,66005	0,65577	−2,89644
v	−0,699469	0,587007	−2,445402	v	−0,69637	0,58813	−2,44231
v	−0,704289	0,645208	−2,463317	v	−0,70497	0,64826	−2,46160
v	−0,70503	0,654166	−2,411583	v	−0,70324	0,65501	−2,40768
v	−0,666303	0,595964	−2,393667	v	−0,66381	0,59772	−2,38984
v	−0,666995	0,587007	−2,445402	v	−0,66589	0,59129	−2,44132
v	−0,662501	0,645208	−2,463317	v	−0,66830	0,64717	−2,46147
v	−0,661809	0,654166	−2,411583	v	−0,66755	0,65652	−2,40873

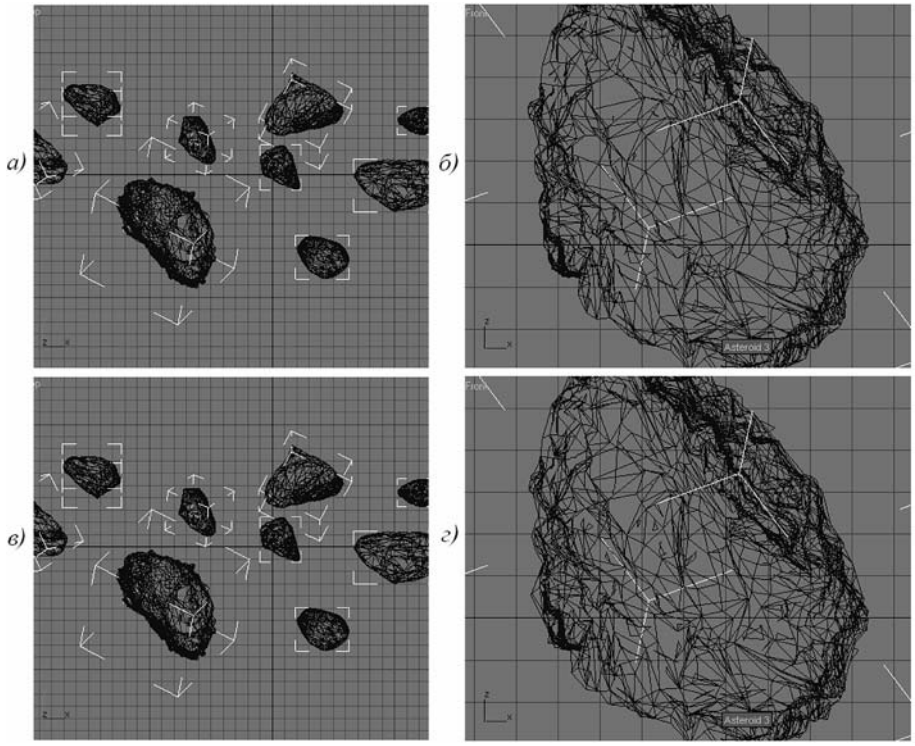


Рис. 6. Исходная 3D-модель "астероиды" (*а*); модель, содержащая ~3 Кбайт скрытновстроенной информации (*в*); увеличенные фрагменты исходного и заполненного контейнеров (*б*, *г*)

На рис. 6 представлен пример работы предлагаемого алгоритма ССИ для контейнера "астероиды" в формате OBJ. Здесь *а*, *б* — изображение исходной модели и ее увеличенного фрагмента; *в*, *г* — изображение модели, содержащей ~3 Кбайт скрытновстроенной информации и ее увеличенного фрагмента.

Заключение

Таким образом, в статье приводится теоретическое обоснование возможности использования ИНС для стеганографического скрытия. Полученные результаты наглядно демонстрируют, что нейронные сети могут успешно применяться в задачах ССИ. Получены зависимости вероятности ошибочного восстановления встроенной информации для различных параметров алгоритма ССИ, приведены общие рекомендации по выбору стеганографических контейнеров для повышения эффективности работы предлагаемого алгоритма.

Список литературы

1. **Грибунин В. Г.** Цифровая стеганография. СПб.: СОЛОН-Пресс, 2002. 280 с.

2. **Kavithal V., Easwarakumar K. S.** Neural Based Steganography PRICAI 2004: Trends in Artificial Intelligence. 2004. P. 429—435. URL: http://resources.metapress.com/pdf-preview.axd?code=q0bh4d8w9fjumdrj&size=largest

3. **Chang C. Y., Shen W.-C.** Using counter-propagation neural network for digital audio watermarking. URL: http://dspace.1ib.fcu.edu.tw/bitstream/2377/1060/1/ce07ncs002006000074.pdf

4. **Сирота А. А., Попов В. Г.** Свойства сходимости весов автоассоциативной двуслойной линейной нейронной сети при построении сжимающих отображений случайных векторов // Нейрокомпьютеры: разработка и применение. 2009. № 5. С. 3—11.

5. **Андерсон Т. В.** Введение в многомерный статистический анализ // М.: Физматлит, 1963. 500 с.

ИНФОРМАЦИЯ

УДК 004.652.4

Б. Г. Ильясов, д-р техн. наук, проф.,
Уфимский государственный авиационный технический университет,
А. А. Левков, канд. техн. наук, доц.,
Уфимский государственный авиационный технический университет,
e-mail: projekctor@gmail.com

Структурная оптимизация реляционных моделей сложных иерархических систем

Предлагается метод построения иерархических реляционных моделей сложных систем, позволяющий проводить полную нормализацию моделей и реализующий обобщения элементов модели в рамках реляционной парадигмы, а также снижающий число атрибутов в них.

Ключевые слова: *база данных, реляционная модель, иерархическая реляционная модель, нормализация, обобщение элементов реляционной модели, структурная оптимизация*

Введение

В настоящее время разработчики баз данных (БД) все чаще сталкиваются с необходимостью построения реляционных моделей (РМ) сложных иерархических систем (СИС) (таких как АСУТП, АСУП, ERP, CRM) как единого целого. Если раньше такие системы строили в виде небольших независимых модулей, изолированных друг от друга, то требования современных систем управления требуют сквозной интеграции разрозненных моделей в единое целое [1, 2]. Число сущностей в этих системах может превышать тысячи (в перспективе построение БД из десятков тысяч сущностей), что приводит к качественному росту сложности проектирования таких моделей [3, 4].

В настоящее время наиболее широко распространенным подходом к проектированию реляционной модели является построение многосвязного графа реляционных сущностей и их нормализация [5]. К недостаткам данного подхода можно отнести отсутствие средств по организации обобщений в РМ, что приводит к необходимости проектирования РМ в виде неразрывного "полотна" и использования исключительно экспертных знаний при нормализации сущностей [6].

Использование такого подхода оправдывает себя при моделировании систем, состоящих из десятков сущностей, но при моделировании систем из сотен и тысяч сущностей, охваченных тысячами связей, приходится прибегать к разбиению единого "полотна" проектирования на отдельные блоки, не имеющие прямого отражения в реализуемой реляционной схеме. По сути, определенные фрагменты РМ (сущности и связи) "вырывают" из контекста проектирования, заменяя на условные логические блоки для облегчения восприятия модели человеком, так как это показано ниже (рис. 1).

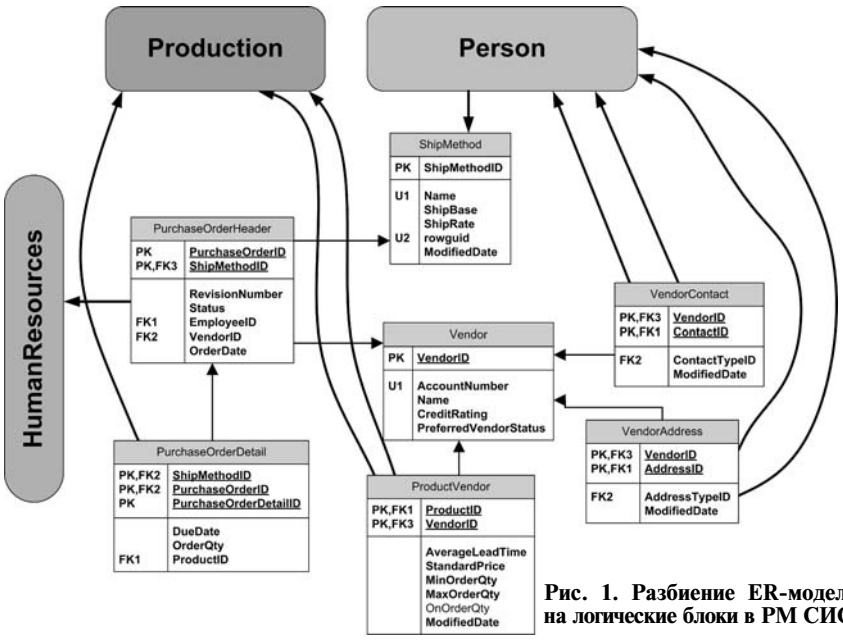


Рис. 1. Разбиение ER-модели на логические блоки в РМ СИС

Такие абстрактные логические блоки (*Person*, *HumanResources*, *Production*) не могут быть наделены никакими общими чертами, интегрированы в итоговую РМ как некий абстрактный элемент, не могут явно участвовать в ограничениях целостности (связях), запросах и т. д. Данные блоки вообще не являются элементами реляционной теории, а потому такие обобщения не могут быть отражены в РМ, т. е. выражения типа (*Person*) $\frac{ContactId}{\theta}$ (*VendorContact*) не только лишены смысла, но и невозможны. Обобщенные блоки не обладают эмерджентностью, отра-

зимой в РМ. Все это приводит к тому, что такая модель не адекватна реляционной модели данных. В реальной РМ приходится создавать большое число отдельных связей вместо связей между обобщениями.

Другой проблемой построения РМ СИС является нормализация реляционных отношений. Исходя из теории и практики нормализации, она осуществляется на выявленных функциональных зависимостях (ФЗ) между атрибутами в отношениях [7]. При выявлении ФЗ нельзя опираться на механический анализ уже существующих данных (если $r(R)$, $A \subseteq R$, $B \subseteq R$; $(A \rightarrow B)$, если $((\forall t_1, t_2 \in r : t_1(A) = t_2(A)) \Rightarrow (t_1(B) = t_2(B)))$, так как сама природа СИС такова, что существенная часть работы системы заключается в сохранении *новой* информации, поступающей на вход системы и являющейся отражением состояния ОУ. То есть существующая информация всегда неполна, и выявленные для нее функциональные зависимости могут оказаться неверными при поступлении новой информации.

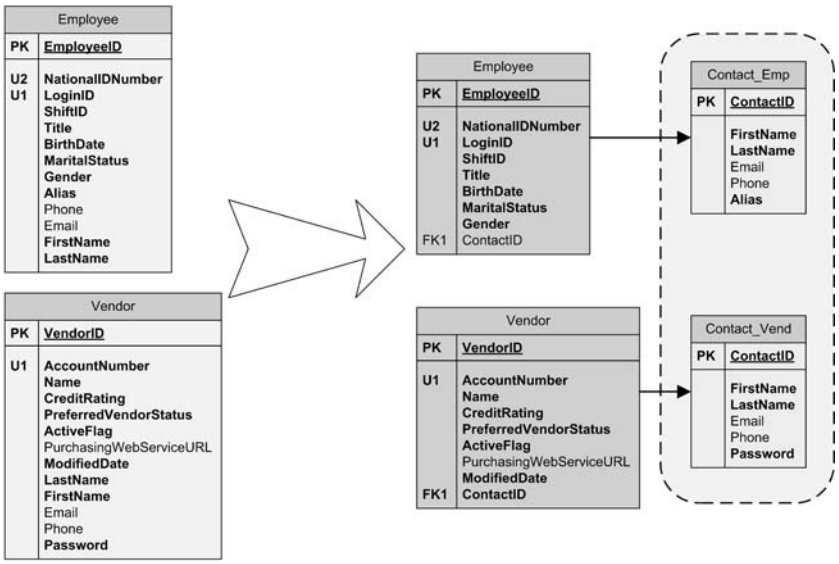


Рис. 2. "Распыленные" сущности при неполной нормализации

Для проведения полной нормализации необходимо построение единого ненормализованного отношения с последующим проведением операций анализа данных $O_{full} = 2^{|A|}$ на предмет определения функциональных зависимостей, где $|A|$ — число атрибутов в ненормализованном отношении. Очевидно, что такие операции транскомпьютабельны, так как число атрибутов в едином ненормализованном отношении может превышать десятки тысяч (при $|A| = 1000 O_{full} \approx 10^{301}$) — т. е. необходимо провести $O(\mathbf{e}^N)$ операций анализа.

Это вынуждает разработчиков осуществлять разбиение модели на сущности интуитивно и потом нормализовать эти, уже сравнительно небольшие, структуры. В таком случае необходимо выполнение $O_{part} = N \cdot 2^{|A|/N}$ операций анализа (если считать число атрибутов в сущности одинаковым, то $O(CN)$), однако модель в таком случае нормализуется лишь частично и возможно возникновение "распыленных" сущностей, т. е. ситуаций, когда производные сущности, полученные при нормализации из разных первоначальных сущностей, являются отражениями одной реальной сущности (рис. 2).

Все это привело к тому, что многие современные модели хранения СИС используют реляционные БД только в качестве хранилища информации, основываясь на объектно-ориентированном (ОО) представлении информации. Для построения РМ используются автоматические методы трансляции ОО модели.

Минусы данного подхода заключаются в том, что организованные на основе данных методик РМ СИС нарушают нормальные формы и теряют соответствие уже относительно самой реляционной модели данных [8], т. е. в сущности, перестают быть реляционными, а превращаются в "электронно-табличные". Все это приводит к падению производительности систем, возможному рассогласованию данных в них (собственно отказ использования от ограничения целостности по вторичным ключам и позволяет говорить о потере реляционности). Одной из типичных черт таких объектно-реляционных систем является отказ от массовой обработки данных в пользу построчной в связи с активным использованием инкапсуляции в объектной части.

Таким образом, современные методы построения РМ не позволяют провести полную нормализацию РМ сложных иерархических систем и не обеспечивают возможностей по реализации обобщений, что приводит к росту числа связей в модели и ее усложнению.

В данной работе авторами предлагается иерархическая структура организации реляционных сущностей, применение которой позволяет упростить нормализацию РМ СИС, а также реализовать обобщения посредством реляционных механизмов.

1. Структура организации РМ СИС

Наиболее эффективным решением по структуризации и нормализации РМ СИС, на взгляд авторов, является построение иерархии сущностей системы на основе отношений общее—частное (иерархия классов), где обобщенные сущности находятся на более высоких уровнях иерархии, а уточненные — на более низких [9]. Обобщенные сущно-

сти включают в себя только те атрибуты, которые входят во все потомки. Потомки включают в себя только те атрибуты, которые отсутствуют в родительских сущностях. Иерархия строится таким образом, чтобы сущности не содержали повторяющихся атрибутов — все они должны быть вынесены в родительские сущности [10]. Связи между сущностями, реализующие иерархию, осуществляются с помощью вторичных ключей, направленных от потомков к предкам и связывающих первичные ключи в сущностях (связи один-к-одному) (рис. 3, см. четвертую сторону обложки).

Такая структура позволяет более точно и менее затратно определить поведение элементов системы, легко изменять и дорабатывать модель без нарушения принципов нормализации.

С точки зрения процесса проектирования сложной РМ построение иерархии классов имеет не только высокую практическую ценность, но и важное методологическое значение: данный процесс позволяет последовательно разворачивать структуру даже самой сложной системы, уточняя функциональность ее элементов "шаг за шагом", без необходимости охватить все "поле" проектирования сразу, как требует того классический подход. Более того, последовательное уточнение иерархии классов позволяет более естественным образом отделять одни сущности от других в процессе их уточнения. В целом, построение иерархии классов основано на более полном отражении семантики системы, что позволяет более четко различать сущности.

Таким образом, разворачивание иерархии классов выступает в качестве альтернативы нормализации: позволяет выявить состав элементов системы и разложить их на элементарные, в рамках заданной функциональности, составляющие.

При построении данной структуры каждая последующая по иерархии сущность является уточняющей, а не самостоятельной полноценной сущностью. Для получения целостной сущности в данной структуре необходимо провести соединение всей ветви наследования сущности (порядок соединения значения не имеет):

$$E^{full} = \begin{cases} \text{если } \nexists\ E_{\text{пред}}, \text{ то } E \\ E \underset{\theta}{\overset{pk}{}} E^{full}_{\text{пред}}; \end{cases}$$

$$A_{E^{full}} = \begin{cases} \text{если } \nexists\ E_{\text{пред}}, \text{ то } A_E \\ A_E \cup A_{E^{full}_{\text{пред}}}, \end{cases}$$

т. е. $Customer = (Entry) \underset{\theta}{\overset{ID}{}} (Object) \underset{\theta}{\overset{ID}{}} (Customer1)$,

$A_{customer} = A_{entry} \cup A_{object} \cup A_{customer1}$.

2. Нормализация РМ СИС

В случае организации иерархии в процессе нормализации необходимо проводить анализ лишь только дочерних узлов друг с другом и родительским элементом (рис. 4, см. четвертую сторону обложки).

При иерархической организации РМ не могут возникнуть "распыленные" сущности, так как по требованию иерархизации все одинаковые элементы в потомках должны быть обобщены на уровне предка. Это дает гарантию того, что независимая нормализация каждой сущности на каждом уровне иерархии будет полной.

Так как каждый элемент в иерархической РМ является реляционной сущностью, то снимается проблема обобщений — все они могут участвовать в реляционных операциях: на них можно определять ограничения целостности, использовать их в реляционных операциях и т. д. На рис. 5, представлен общий вид иерархической РМ, использующей ссылки к обобщенным сущностям (вторичные ссылки, не участвующие в иерархии, показаны жирными стрелками):

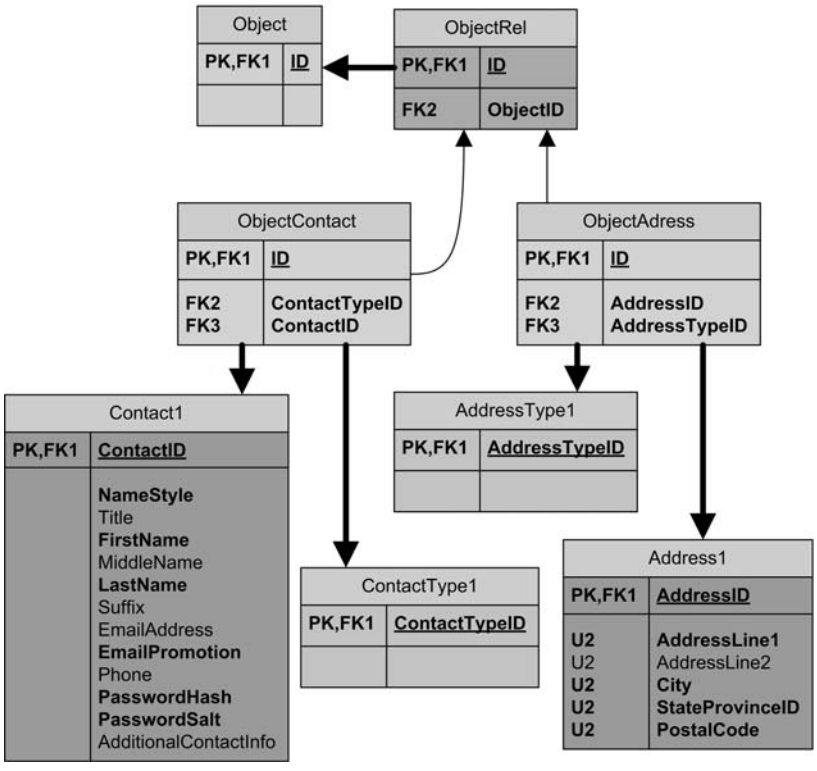


Рис. 5. Вторичные ключи в иерархической РМ СИС

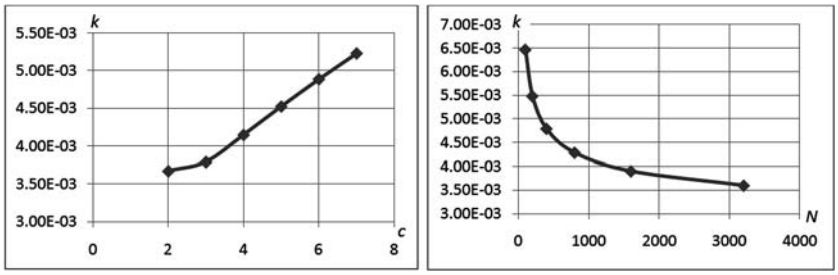
Как можно видеть из рис. 5, отношение (*Object*) $\frac{ObjectID}{\theta}$ (*ObjectRel*) возможно и определено как ограничение целостности в модели.

3. Параметры РМ СИС

Если считать дерево иерархии сбалансированным, а степень узла обозначить через c, то основными его характеристиками будут следующие:

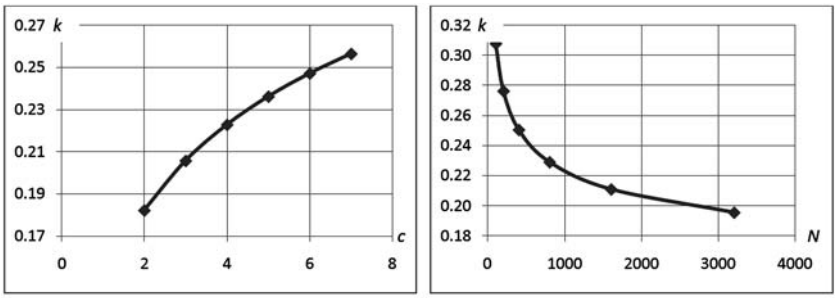
- число уровней в иерархии $U = \frac{\ln(N)}{\ln(c)} + 1;$ (1)
- общее число элементов в иерархии $N' = \frac{Nc-1}{c-1};$ (2)
- число нелистовых элементов $N_{\text{доп}} = \frac{N-1}{c-1};$ (3)
- число атрибутов в сущности $|A'_E| = \frac{|A|}{NU};$ (4)
- общее число атрибутов $|A'| = \frac{|A|}{NU}\frac{Nc-1}{c-1} \approx \frac{|A|}{U}.$ (5)

Для иерархической РМ можно провести полный анализ модели за $O_{hier} = N' \cdot 2^{|A'_E|}$ операций (при $c = 4$, $N = 100$ и $|A| = 1000$ $O_{hier} \approx 662$).



а) *б*)

Рис. 6. Зависимости относительной эффективности иерархической РМ: *а* —от степени узла иерархии (($|A| = 10\,000$, $N = 1000$); *б* — от числа сущностей ($c = 4$)



а) *б*)

Рис. 7. Зависимости относительного коэффициента оптимизации атрибутов сущностей: *а* — от степени узла иерархии ($|A| = 10\,000$, $N = 1000$); *б* — от числа сущностей ($c = 4$)

В случае иерархической РМ общее число операций анализа ограничено величиной $O(C^{\frac{\ln(c)}{\ln(Nc)}}N)$, т. е. эффективность иерархических РМ снижается с уменьшением числа уровней в иерархии и растет с увеличением числа сущностей. На рис. 6 представлена зависимость относительной эффективности иерархической структуры $k = \frac{O_{hier}}{O_{full}}$ от степени узла иерархии с ($|A| = 10\,000$, $N = 1000$).

Как можно видеть из данных зависимостей, чем меньше степень узла иерархии и чем больше число сущностей, тем иерархическая РМ эффективнее классической РМ. Это связано с уменьшением числа атрибутов в вершине при большей глубине дерева. Так как ссылки в РМ реализованы на основе атрибутов, то подобным же образом уменьшается число ссылок иерархической РМ. Все это позволяет без потери целостности строить более простые реляционные схемы за счет объединения подобных атрибутов и ссылок на родительском уровне иерархии. На рис. 7 изображены зависимости относительного коэффициента оптимизации атрибутов сущностей иерархических РМ $k = \frac{|A|}{|A_{hier}|}$.

Заключение

Оптимизация проектирования РМ СИС является важной научно-технической задачей: сложность таких моделей высока, что не позволяет проводить их полную нормализацию ($O(e^N)$), а отсутствие средств обобщений элементов модели приводит к необходимости многократного дублирования атрибутов в сущностях.

В данной статье предложена иерархическая организация РМ СИС по критерию общее—частное с декомпозицией сущностей по принципу пересечения множеств атрибутов. Данный подход позволяет проводить нормализацию РМ СИС за число шагов ($O(C^{\frac{\ln(c)}{\ln(Nc)}}N)$), меньшее полиномиального. Создание промежуточных сущностей-предков, агрегирующих подобные атрибуты сущностей-потомков, позволяет значительно уменьшить общее число атрибутов и связей в РМ СИС и использовать сущности-предки в качестве элементов-обобщений, что упрощает организацию ограничений целостности и позволяет строить более простые производные отношения в РМ СИС.

Список литературы

1. **Intersoft Lab.** Интеграция корпоративных приложений: основные понятия. URL: http://citcity.ru/11132/
2. **Гурьянов Л. В.** Интеграция АСУТП в АСУ предприятия. URL: http://www.krug2000.ru/reports/08-ent_acs_integr.pdf
3. **Khalilov A. I.** Data base organization in complex management information systems // Cybernetics and Systems Analysis. 02.02.2005.
4. **Cong Yu, Jagadish H. V.** Querying complex structured databases. VLDB. 2007.
5. **Heath I.** Unacceptable File Operations in a Relational Database // Proc. ACM SIGFIDET Workshop on Data Description. Access and Control. San Diego, Calif. 1971.
6. **Chen P.** The Entity-Relationship Model — Toward a Unified View of Data // ACM Transactions on Database Systems (TODS), 1976.
7. **Codd E. F.** A Relational Model of Data for Large Shared Data Banks // Communications of the ACM 13. June 1970.
8. **Abiteboul S., Patrick C. F., Schek H.** Nested relations and complex objects in databases. 1987.
9. **Смит Д. М., Смит Д. К.** Абстракции баз данных: агрегация и обобщение // СУБД. 1996. № 2.
10. **Буч Г.** Объектно-ориентированный анализ и проектирование. 2- е изд. М.: Бином, 2007. 560 с.

УДК 004.652.3(075)

Р. Ф. Халабия, канд. техн. наук, доц.,
Московский государственный университет
приборостроения и информатики,
e-mail: rustam-capitan@mail.ru

Организация и структура динамических распределенных баз данных

Рассмотрены проблемы организации баз данных. Предложена организация динамической распределенной базы данных, построенная на основе гибридных сетей. Описаны метаданные системы хранилищ данных как основы структуры динамической распределенной базы данных. Определены основные направления развития этой области.

Ключевые слова: *динамическая база данных, распределенная база данных, метаданные*

Введение

На современном этапе развитие компьютерных методов и средств реализации распределенных баз данных занимает центральное место в науке, промышленности и образовании. Информационные технологии, особенно за последние 10 лет, окончательно преодолели границы локального применения, а всемирную компьютерную сеть Internet по праву называют "седьмым континентом". В основу программных продуктов закладываются принципы интеграции не только объектно-ориентированного, но и документно-ориентированного подхода [1]. В настоящее время число компьютерных устройств разного типа увеличивается. Отмечается резкий рост мобильных планшетных компьютеров и ноутбуков, карманных компьютеров-смартофонов и мобильных телефонов-коммуникаторов, которые представляют собой мощные платформы, способные поддерживать новые приложения и сервисы, и проникают в любые процессы деятельности человека. Возникают проблемы хранения, обработки и передачи текстовых, графических и мультимедийных данных, а также их производства и поиска в базах данных, в которых, как правило, выбирается реляционная модель управления базами данных.

Однако эта модель становится малоэффективной при более глубоком анализе возникающих потребностей эксплуатации сложных информационных систем, так как она существенно ограничивает определение неоднородных данных [2]. В настоящее время многие системы управления базами данных построены на языке запросов SQL. Разработчики адаптируют системы текущим потребностям своей деятельности для обеспечения новых возможностей при реализации традиционных систем управления данными.

При проектировании приложений управления данными возникает класс проблем, для которых применение традиционных подходов разработки не являются оптимальными. Эти проблемы характеризуются:

- наличием огромных таблиц (сотни терабайт) и запросов к ним, обращающихся только к нескольким из многочисленных столбцов таблицы, а также потребность в просмотре таблиц в различных порядках сортировки;
- обращением к данным в основном по чтению, при этом запросы сводятся либо к выборке одной строки, либо к поиску на основе значений многозначных атрибутов, распространенность которых приводит к неэффективности использования реляционного представления;
- слабым информационным поиском, так как поисковые машины оперируют над полуструктурированными данными, а обрабатываемые ими запросы в большинстве случаев сводятся к поиску по ключевым словам, когда требуемым результатом является отсортированный список возможных ответов;

- возможностью кэширования части крупных наборов данных в небольших мобильных устройствах, но требуется дополнительная поддержка коммуникационных каналов к таким устройствам, что приводит к потерям информации;
- преобразованием XML-документов к канонической реляционной организации, сохранением их в базе данных, и обратном преобразованием при потребности их повторного использования; по мере возрастания числа создаваемых, передаваемых и обрабатываемых XML-документов эти преобразования становятся излишними, неэффективными и трудоемкими;
- лингвистической обработкой потоков данных, где их фильтрация выглядит подобно SQL-запросам, однако данный язык предназначается для работы с постоянно хранимыми таблицами, а запросы выполняются над потоком значений данных, поступающих в реальном времени.

В работе [3] показано, что стандартные системы БД некорректно решают эти проблемы, так как язык SQL является "правильным" языком запросов и разработчики используют модель реляционных баз данных для приложений, в которых отсутствует постоянное хранение данных.

Потоковая обработка данных представляет собой класс приложений, которые могут выиграть от использования подобного языка запросов поверх системы управления данными со свойствами, радикально отличающимися от свойств реляционных БД. Поскольку потоковые запросы обычно выполняются над данными, наблюдаемыми в течение определенного временного окна, требуется некоторое временное локальное хранилище данных, но это хранилище не обязано обладать свойствами персистентности, транзакционности, а также поддерживать выполнение сложных запросов. Реляционные базы данных хорошо приспособлены для обработки динамических запросов над сравнительно статическими или медленно изменяемыми данными, а этот класс приложений характеризуется статическим набором запросов над динамическими данными.

Для решения этих задач необходимы гибкие решения. Имеется несколько способов обеспечения гибкости в сегодняшней изменяющейся среде данных. При использовании подхода, предполагающего возврат к исходным принципам, для каждого отдельного приложения создается собственная служба хранения данных. Этот подход практически пригоден только для простейших приложений. Однако некоторые функционирующие сегодня приложения, требующие интенсивной обработки данных, не могут полагаться только на простые решения.

Для управления такими данными предлагается структура организации динамической базы данных, которая включает в себя не только множество классов документальных объектов, но их свойства, методы и события.

Организация н структура динамических распределенных баз данных

Система, в которой объем хранимой информации $V(t)$ меняется во времени t, является динамической распределенной базой данных, которую можно описать уравнением

$$V(t) = f(t)x(N(t),\ D(t)), \qquad (1)$$

где $f(t)$ — возможная функция распределения; x — зависимость состояния БД от N и D; N — общее число узлов БД в данный момент времени; D — число данных на узлах в данный момент времени.

Для организации такой системы можно использовать одноранговую компьютерную сеть. В таких сетях отсутствуют выделенные серверы, а каждый узел является как клиентом, так и сервером. В отличие от архитектуры клиент—сервер такая организация позволяет сохранять работоспособность сети при любом числе и сочетании доступных узлов. Основными недостатками одноранговой сети являются:

- низкая безопасность, т. е. возможность несанкционированного использования информации (открытость протокола доступа, его использования и подмены, предоставление ложной информации, открытость передачи данных, перехват данных);
- недостаточная надежность, при наличии внешних факторов, умышленных и случайных действий, которые создают последствия в работе сети (отключение провайдером сети либо несанкционированные действия с коммуникациями, отсутствие электричества в местных сетях и некорректная работа оборудования). Для устранения указанных недостатков целесообразно использовать гибридную сеть с использованием соединения "точка к точке" (Peer-To-Peer) (рис. 1).

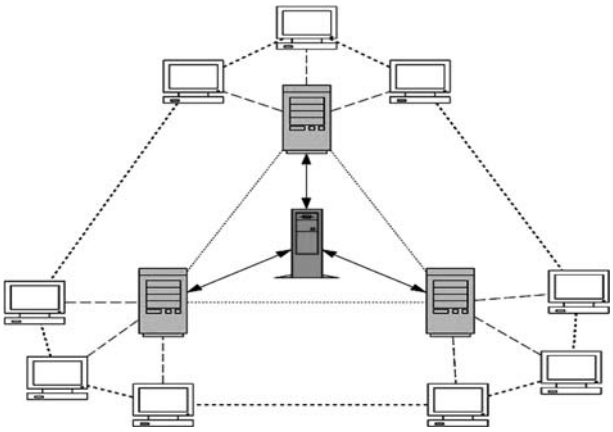


Рис. 1. Схема гибридной сети

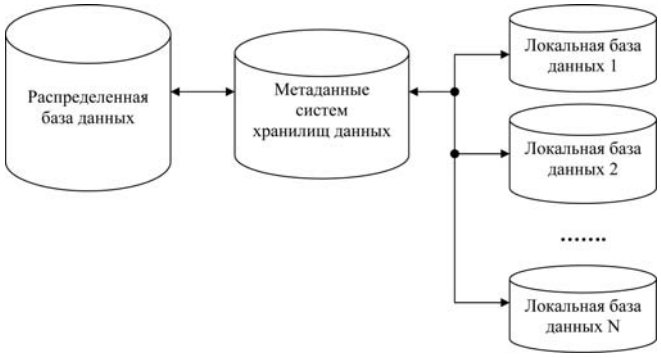


Рис. 2. Структура распределенной базы данных ДРБД

В гибридной сети все множество клиентов подключено друг к другу, в ней присутствуют серверы, выполняющие дополнительные функции, которые разделяются по рангам, функциональности и т. п.

Каждый узел является базой данных, все узлы объединяются в единое целое. Полученная таким образом архитектура позволяет определить ключевые понятия и структуру динамической распределенной базы данных (ДРБД) (рис. 2).

ДРБД является разделяемым информационным ресурсом (совокупностью локальных информационных ресурсов) в виде связанных распределенных данных через хранилище метаданных. Метаданные системы хранилищ данных обычно разделяют на два типа:

- **служебные метаданные,** используемые для функций извлечения, преобразования и загрузки, для переноса информации из транзакционных систем в хранилище;
- **интерфейсные метаданные,** используемые для описания экранов и создания отчетов.

Однако кроме служебных и интерфейсных типов метаданных в структуру ДРБД включаются следующие типы [4, 5]:

- **метаданные исходной системы** спецификации источников данных, таких как репозитории; описательная информация, например, частота обновления, юридические ограничения и методы доступа; информация о процессах, таких как график заданий и коды извлечения;
- **метаданные преобразования данных** (информация о получении данных, например о планировании передачи данных и их результатов, а также сведения об использовании файлов, управление таблицами измерений, преобразование и агрегирование, определения агрегатов данных, документирование проверок, работ и журналов);
- **метаданные СУБД** (содержание системных таблиц СУБД, рекомендации по обработке).

В ДРБД метаданные играют важную роль в реализации всей системы в целом, они используются следующим образом:

- **пассивно,** обеспечивая четкую документацию о структуре, процессе разработки и использовании системы хранилища данных, такая документация необходима всем участникам (т. е. конечным пользователям, системным администраторам, а также разработчикам приложений);
- **активно,** путем хранения конкретных семантических аспектов (например правил преобразования) в виде метаданных, которые можно интерпретировать и использовать во время исполнения. В этом случае процессы хранилища данных управляются метаданными. Следовательно, активные метаданные и дополнительная документация согласованно и унифицированно управляются в одном репозитории, при этом актуальность документации возрастает;
- **полуактивно,** за счет хранения статической информации (например, определений структур, спецификаций конфигураций), которую будет считывать другой программный компонент во время выполнения. Например, обработчикам запросов необходимы метаданные для проверки существования атрибутов. В отличие от активного использования здесь метаданные только читаются, но не исполняются.

Заключение

На данном этапе развития информационных технологий все чаще приходится сталкиваться с проблемами разработки и применения распределенных баз данных, построения аппаратного и программного обеспечения, организации процессов взаимодействия между всеми компонентами узлов сетевых архитектур. В данной работе выделены основные проблемы построения динамических распределенных баз данных, предложена и описана гибридная модель сети дли организации динамических распределенных баз данных. Приведено описание структуры ДРБД, в которой основную роль играет хранилище метаданных, разделенное на типы и функционирующее по предложенным схемам их использования.

Список литературы

1. **Буч Г.** Объектно-ориентированный анализ и проектирование. М.: Бином, 1998. 560 с.

2. **Таненбаум Э., вал Стеен М.** Распределенные системы: принципы и парадигмы. СПб.: Питер, 2003. 876 с.

3. **Stonebraker M.** Introduction to Chapter 1 ("The Roots"). Readings in Database Systems. 2nd ed. San Mateo, Calif.: Morgan Kaufmann, 1994.

4. **Ross M.** The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling. 2nd ed. — New York: Wiley, 2002.

5. **Kimball R.** et al. The Data Warehouse Lifecycle Toolkit. 2nd ed. — New York: Wiley, 2008.

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ЭКОНОМИКЕ И УПРАВЛЕНИИ

УДК 519.72:621.372

В. В. Савченко, д-р техн. наук, проф., зав. каф.,
Нижегородский государственный
лингвистический университет,
e-mail: svv@lunn.ru

Информационная теория колебаний биржевых котировок в динамике

Отталкиваясь от базовых понятий информационной теории колебаний рыночной конъюнктуры и общесистемного принципа минимума информационного рассогласования в метрике Кульбака—Лейблера, ставится и решается задача автоматической диагностики текущего состояния рынка ценных бумаг. Предложен новый алгоритм кластерного анализа. Даны оценки его эффективности, рассмотрены примеры практического применения.

Ключевые слова: *временной ряд, линейная авторегрессионная модель, динамика рыночной конъюнктуры, прогноз, типология рынка, критерий минимума информационного рассогласования*

Введение

Современный уровень достижений в области прогнозирования рыночной конъюнктуры представляют методы статистического анализа, объединенные общей идеей адаптивной авторегрессионной модели (АР-модели) колебаний биржевых котировок в динамике [1, 2]. Однако возможности данного класса методов объективно ограничиваются быстродействием современных алгоритмов адаптации АР-моделей под результаты наблюдений [3].

Качественно новый подход в том же направлении исследований открывает информационная теория колебаний рыночной конъюнктуры в динамике, которой и посвящена настоящая статья. В представленном исследовании она рассматривается под углом зрения классической задачи экономической диагностики, при этом используются модели и методы из предыдущих работ автора по типологии рынка ценных бумаг [3, 4].

Задача диагностики

Принцип действия большинства современных систем диагностики рынка в динамике основывается на сопоставлении его текущего состояния x с конечным набором альтернативных вариантов x_r^*, $r = \overline{1, R}$, из достижимой ретроспективы. Основным препятствием на этом пути является проблема вариативности динамики рынка в пределах одного и того же состояния [3]. Закономерным выходом из данной ситуации может служить статистический подход [4, 5] или сравнение тестируемого сигнала x одновременно с несколькими ($J_r \geqslant 1$) образцами колебаний $x_{r,j}$, $j = \overline{1, J_r}$, в пределах каждого возможного (r-го) состояния рынка.

Для принятия обоснованного решения в пользу состояния x_r^* исследователю будет достаточно в таком случае установить ρ_0 — степень близости сигнала x к любому из образцов в некоторой метрике $\rho(x/x_{r,j})$ Этим существенно ослабляется проблема вариативности рынка, а вслед за ней и проблема малых выборок наблюдений [1, 2]. Одновременно становится понятной и реализация критерия "достаточной степени близости" тестируемого сигнала x к эталону x_r^*: он должен войти в границы J_r-множества одноименных образцов-эталонов r-го состояния как полноправный ($J_r + 1$)-й его элемент. Задача в таком случае переходит в область изучения типологии рынка.

На первом, подготовительном, этапе ее решения для каждой из R рассматриваемых альтернатив x_r^*, $r = \overline{1, R}$, текущего состояния рынка x сначала требуется сформировать множество образцов колебаний его цен в динамике X_r в пределах соответствующей типологической единицы. Диагностика рынка сводится после этого к классической задаче статистической классификации сигнала x на ($R \times J_r$)-множестве его динамических образцов $\{x_{r,j}\}$. Наиболее близкий из них — это и есть искомое решение задачи диагностики, или ближайший аналог текущего состояния рынка в ретроспективе. Отталкиваясь от установленной типологии, исследователь получает возможность предвидеть будущее развитие рынка по содержанию его текущего состояния. Еще одна исключительно интересная для практики перспектива — "контекстный" анализ текущего и прогнозирование будущего состояния

рынка по принципу устойчивости в его динамике связанных типологических структур.

Ключевой вопрос по диагностике — о выборе критерия близости для однотипных колебаний рынка $\rho(x/x_{r,j})$. В рамках теоретико-информационного подхода роль указанного критерия выполняет величина информационного рассогласования (ВИР) в смысле Кульбака—Лейблера [6]:

$$\rho(x/x_{r,j}) \stackrel{\Delta}{=} \int\int \ln \frac{dP(x)}{dP_{r,j}(x)}\, P(dx)$$

между распределениями тестируемого сигнала $P(x)$ и (r, j)-го эталона $P_{r,j}(x)$. Здесь $\stackrel{\Delta}{=}$ — символ равенства по определению. Подробное обоснование ВИР в задачах распознавания образов приводится, например, в работах [7—9]. В дальнейшем мы еще вернемся к этому вопросу.

Теоретико-информационный подход

Несмотря на существующие различия в характере колебаний (образцах) некоторого r-го состояния рынка x_r^* все они воспринимаются его участниками (биржевыми игроками), как нечто общее, иначе типология рынка утратила бы для них свою информативность. Можно поэтому утверждать, что одноименные образцы-колебания $x_{r,j}$, $j = \overline{1, J_r}$, $J_r \gg 1$, для каждого отдельного состояния рынка в массовом сознании игроков группируются в соответствующие образы, или кластеры $X_r = \{x_{r,j}\}$, $r = \overline{1, R}$, вокруг определенного центра — эталонной метки данного образа. В информационной теории указанные эталоны определяются в строгом теоретико-информационном смысле [10]:

$$x_r^* = x_{r,v}: J_r^{-1} \sum_{j=1}^{J_r} \rho(x_{r,j}/x_{r,v}) =$$

$$= \min_{i \leqslant J_r} J_r^{-1} \sum_{j=1}^{J_r} \rho(x_{r,j}/x_{r,i}) \stackrel{\Delta}{=} \rho_r^*. \qquad (1)$$

Нетрудно увидеть, что именно в понятии информационного центра-эталона (ИЦ-эталона) r-го кластера X_r дается математически строгое описание свойств соответствующего состояния рынка в динамике. В результате будем иметь типологическую (кластерную) базу данных конкретного рынка, составленную из множества $\{x_r^*\}$ ИЦ-эталонов всех его типологических единиц X_r, $r = \overline{1, R}$. Одновременно становится очевидным и механизм диагностики текущего состояния рынка: для принятия решения в пользу r-го состояния X_r наблюдаемый сигнал x должен войти в состав его множества образцов $\{x_{r,j}\}$ по критерию

$$\rho_r(x) = \min_{k \leqslant R} \rho_k(x) \leqslant \rho_0 = (1...2)\rho_r^*, \qquad (2)$$

где $\rho_r(x) \stackrel{\Delta}{=} \rho(x/x_r^*)$, а ρ_0 — пороговый уровень. Отметим важную отличительную особенность данного правила: решение в каждый момент времени может быть принято либо в пользу определенной альтернативы из множества гипотетических состояний $\{X_r\}$, либо вообще не принято для сигналов x нечеткой (маргинальной) структуры. И в этом факте нет противоречия: при переходе рынка из одного состояния в другое возможны неустойчивые (пограничные) состояния. При этом требования в критерии (2) к степени близости тестового сигнала для разных состояний рынка x_r^*, строго говоря, разные.

Сделанный вывод можно наглядно проиллюстрировать геометрически (рис. 1). Здесь некоторый кластер X_r отображен в виде набора его точек-реализаций на плоскости (на рис. 1 отмечены кружками). Точка с минимальной суммой расстояний относительно всех других точек (затемнена) выступает в роли своеобразного "центра массы" данного кластера. В этом и состоит физический смысл понятия "ИЦ-эталон" из выражения (1). Тогда множество допустимых решений (2) в задаче диагностики — это все его (эталона) допустимые вариации в пределах кластера. А пороговый уровень ρ_r^* из выражения (1) — среднее значение ВИР относительно ИЦ-эталона на множестве элементов данного кластера. Нетрудно убедиться, что в этом механизме реализуется общесистемный принцип минимума информационного рассогласования (МИР) [6—10]. Указанный принцип подразумевает множество вариантов своей практической реализации.

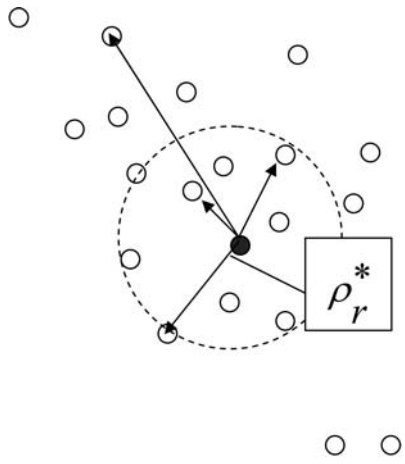


Рис. 1. Кластер одноименных типологических единиц

Пример реализации

В работе [7] показано, что для дискретных распределений вероятностей критерий МИР эквивалентен общесистемному критерию максимального правдоподобия, т. е. является оптимальным в байесовском смысле. Для непрерывных сигналов свойство оптимальности критерия МИР сохраняется, как известно [11], в гауссовском семействе распределений. Во многом именно поэтому в информационной теории рыночных колебаний [1—4] для описания каждого эталона x используется гауссовский (нормальный) закон с нулевым математическим ожиданием и автокорреляционной матрицей (АКМ) K_r размера $n \times n$, $n \geqslant 1$, где $r = \overline{1, R}$, В указанном случае решающая статистика МИР из выражения (2) раскрывается следующим образом [9]:

$$\rho_r(x) = 0{,}5[\mathrm{tr}(K_X \cdot K_r^{-1}) - \ln|K_X \cdot K_r^{-1}| - n].$$

Здесь K_X — выборочная оценка АКМ тестируемого сигнала x; $\mathrm{tr}(\cdot)$, $|\cdot|$ — соответственно след и определитель квадратной $(n \times n)$-матрицы. При дополнительном актуальном [10] условии нормировки сигнала x по энтропии, когда определители всех АКМ равны некоторой константе, можно записать

$$\rho_r(x) = 0{,}5[\mathrm{tr}(K_X \cdot K_r^{-1}) - n]. \qquad (3)$$

А следуя линейной АР-модели колебаний биржевых котировок в динамике [1—3] и переходя в частотную область обработки сигналов [11], окончательно будем иметь следующее выражение для оптимальной решающей статистики МИР [4]:

$$\rho_r(x) = $$

$$= (F+1)^{-1} \sum_{f=0}^{F} \frac{\left|1 + \sum_{m=1}^{p} a_r(m)\exp(-j\pi mf/F)\right|^2}{\left|1 + \sum_{m=1}^{p} a_x(m)\exp(-j\pi mf/F)\right|^2} - 1. \qquad (4)$$

Здесь $\{a_x(m)\}$, $\{a_r(m)\}$ — векторы АР-коэффициентов тестируемого сигнала x и r-го ИЦ-эталона x_r^* соответственно, оба одного порядка $p > 1$. Выражение в числителе (4) определяет квадрат амплитудно-частотной характеристики r-го обеляющего фильтра [11], настроенного на r-й эталон колебаний рынка x_r^*, $r = \overline{1, R}$. Это стандартная формулировка метода обеляющего фильтра (МОФ) в частотной области, где $f = 0, 1, ..., F$ — дискретная частота. Преимуществом данной интерпретации критерия МИР является прежде всего возможность его практической реализации в адаптивном варианте на основе быстрых вычислительных процедур АР-анализа [12].

Анализ эффективности

Эффективность МОФ (1)—(4) в задаче диагностики рынка может быть охарактеризована набором, или $(R \times R)$-матрицей $\|\alpha_{rv}\|$, вероятностей перепутывания состояний рынка, которые при довольно общих предположениях [11] определяются следующим выражением:

$$\alpha_{rv} \triangleq P\{\rho_r(x) < \rho_v(x)|_{x \in X_v}\} = [1 - \Phi_{M,M}(1 + \rho_{rv})],$$
$$v \neq r \leqslant R.$$

Здесь $P\{\cdot\}$ — символ вероятности случайного события; $\Phi_{M,M}(\cdot)$ — интегральная функция F-распределения Фишера с (M, M) степенями свободы; $M = L - p$; L — объем выборки наблюдений. Чем больше ВИР $\rho_{rv} \triangleq \rho_r(x_v)$ между двумя рассматриваемыми состояниями рынка, тем меньше вероятность ошибок при их различении. Например, зафиксировав вероятность ошибки на приемлемом уровне $\alpha_{rv} = 0{,}05$ для $M = 50$, с помощью таблиц F-распределения [13] придем к пороговому значению для ВИР, примерно равному 0,6. Он определяет требования к минимальной различимости состояний рынка по критерию МИР в критерии (2). Применяя их к разным рынкам и разным текущим состояниям рынков, исследователь получает возможность сравнивать рынки между собой по степени обусловленности колебаний рыночной конъюнктуры, их механизмам и перспективам. Иными словами, мы получили инструмент для проведения статистического анализа рыночной конъюнктуры в динамике.

Таким образом, матрица вероятностей перепутывания состояний рынка $\|\alpha_{rv}\|$, а вслед за ней и матрица их информационных рассогласований $\|\rho_{rv}\|$ — это способ математического описания типологии рынка в задачах экономической диагностики на основе информационной теории колебаний рыночной конъюнктуры. И первым шагом на пути к его осуществлению является подготовка данных, или создание типологической базы данных (ТБД) в виде конечного набора ИЦ-эталонов $\{x_r^*\}$ всех существующих типологических единиц колебаний рынка в динамике.

Этап подготовки данных

Указанная ТБД по каждому конкретному рынку, как правило, наблюдателю априори не известна. Здесь в общем случае требуется алгоритм с самообучением, или адаптивный алгоритм. Аналогичная задача рассматривалась в работе [14], в которой был предложен информационный $(R + 1)$-элемент. Это условный термин, обозначающий устройство, или алгоритм, для автоматической классификации

(распознавания) сигнала x в пределах заданного множества распределений P_r, $r = \overline{1, R}$. В основе его функционирования применяется все тот же принцип МИР, о котором много было сказано выше. Но в отличие от большинства своих известных аналогов $(R + 1)$-элемент имеет дополнительный $(R + 1)$-й выход, который при нарушении неравенства в критерии (2) сигнализирует об отказе одновременно от всех R альтернатив $\{X_r\}$. Задача в нашем случае сводится к последовательному применению критерия статистической классификации (2) при переменном (нарастающем) числе альтернатив $R = 1, 2, \ldots$.

Выделим в анализируемом сигнале $x = x(t)$, $t = 1, 2, \ldots$ в функции дискретного времени t первые L его отсчетов из соображений сохранения в них свойства приблизительной стационарности или однородности анализируемого распределения P. Полученный сегмент данных $\boldsymbol{x}_1 = \{x_1, \ldots, x_L\}$ образует минимальную динамическую единицу (МДЕ) рассматриваемого временного ряда. Например, на стандартном рынке ценных бумаг применительно к последовательности приращений $x(t) = c(t) - c(t-1)$ ежедневных цен закрытия биржи $c(t)$, $t = 1, 2, \ldots$ можно говорить [4] о равенстве $L = 60...100$ рабочим дням, что соответствует МДЕ длиной 3—5 месяцев. В подтверждение на рис. 2 представлена временна́я диаграмма ежедневных приращений $x(t)$ американского фондового индекса S&P-500 за период с 1970 г. по настоящее время. Здесь явно выделяются области стационарности временного ряда в широком смысле (по первым двум моментам распределения) продолжительностью несколько месяцев.

Будем использовать сформированную МДЕ в качестве выборки X_1 для оценивания АКМ сигнала x на первом шаге анализа. Соответствующий закон распределения $P_1 = N(K_1)$ — это первый элемент нашей будущей ТБД. После этого приравниваем $R = 1$ и возьмем последовательно второй сегмент временного ряда $x(t)$, $t = 1, 2, \ldots$ для анализа:

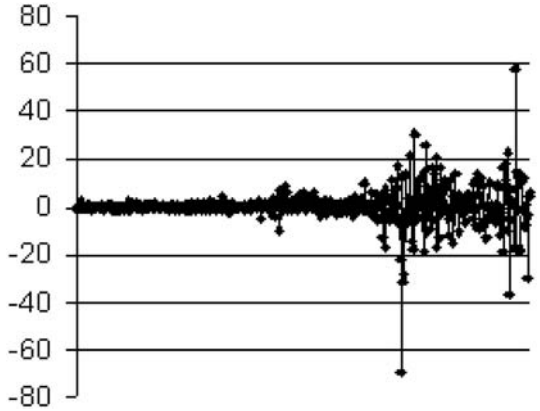


Рис. 2. Временная диаграмма ежедневных приращений индекса S&P-500

$\boldsymbol{x}_2 = \{x_{L+1}, \ldots, x_{2L}\}$. Следуя выражению для решающей статистики МИР (3), определим значение ВИР:

$$\rho(X_2, X_r) = \rho_r(x)\big|_{x = \boldsymbol{x}_2} \qquad (5)$$

сигнала $\boldsymbol{x}_2$ относительно первого элемента ТБД $\boldsymbol{x}_1^*$ (т. е. при $r = 1$). Полученный результат сопоставим согласно (2) с пороговым уровнем:

$$\min_{r \leqslant R} \rho(X_2, X_r) \leqslant \rho_0. \qquad (6)$$

При нарушении данного неравенства в нашем начальном варианте ТБД появится второй элемент $\boldsymbol{x}_2^*$, и вслед за этим приравняем число выявленных фонем $R = 2$. В противном случае принимается решение об объединении выборок X_1 и X_2 в один класс P_1. При этом сохраняется прежнее значение $R = 1$.

Вычисления по схеме (5), (6) повторяются циклически для всех последующих сегментов данных из наблюдаемого сигнала $x(t)$. Причем повторятся "нарастающим итогом" для переменного значения $R = 3, 4, \ldots$ Каждый очередной сегмент сопоставляется по правилу (6) одновременно со всеми R сформированными на данный момент эталонами. При этом не исключается возможность объединения одного и того же сегмента данных с элементами одновременно нескольких разных кластеров, что точно отвечает логике решаемой задачи. Это типичная формулировка информационного $(R + 1)$-элемента.

В результате обработки всего сигнала $x(t)$ будем иметь в конечном итоге искомую ТБД данного рынка $\{X_r\}$ с некоторым фиксированным числом элементов R^*. Напрашивается вывод: чем больше R^*, тем более многообразна с фундаментальной, типологической точки зрения его динамика в сравнении с динамикой других рынков. Однако здесь необходимо учитывать важное обстоятельство: объем ТБД — это производная величина от параметров алгоритма обработки сигналов (1)...(6) и, в частности, от основного его параметра — порогового уровня ρ_0 в выражениях (2) и (6). Так, при его увеличении расширяются границы и ослабляются требования к составу каждого кластера. За счет этого их суммарное число R^* пропорционально уменьшается. И наоборот, при понижении порога по ВИР объем результирующей ТБД R^* монотонно возрастает. Это известный [4] эффект кластеризации рынка в динамике по критерию МИР. Поэтому можно утверждать, что сравнительный анализ и сравнительную оценку типологии разных рынков следует проводить по ТБД, предварительно приведенным в сопоставимый вид по их параметрам и характеристикам. Пример такого рода анализа рассматривается далее.

Пример применения

Для практической апробации предложенной информационной теории были выбраны два фондовых рынка: самый развитый в мире рынок США и развивающийся рынок России. Первый из них был представлен двумя основными своими индексами: S&P-500 и Nasdaq, а российский рынок — индексом Российской торговой системы (РТС) и акциями нефтяной компании НК "Лукойл". По каждому из четырех финансовых инструментов с помощью электронной базы данных РБК с интернет-сайта http://quote.rbc.ru/exchanges/ были получены последовательности ежедневных цен закрытия торгов $c(t)$, $t = 1, 2, ...$ на биржах NYSE в США, ММВБ и РТС в России за период с 1970 г. и с 1995 г. соответственно, по 2009 г. В результате были получены четыре временных ряда $x(t)$, $t = 1, 2, ..., N$, разного объема N: от 10 095 по индексу широкого рынка США S&P-500 и до 2950 по акциям НК "Лукойл". Затем первый из них был предварительно разбит на короткие сегменты данных длиной $L = 80$ отсчетов, или один календарный квартал. И с их использованием согласно рекуррентной процедуре (5), (6) при переменном пороговом уровне $\rho_0 \leqslant 1{,}0$ была сформирована соответствующая ТБД. В аналогичном порядке вычислений для сравнения были сформированы и ТБД по всем другим финансовым инструментам из числа перечисленных выше. При этом все вычисления выполнялись в среде Matlab-7,2 с применением современного ПК. Расчет ВИР выполнялся в частотной области (4) методом Берга—Левинсона [12], имеющим предельно высокую скорость сходимости. Порядок АР-модели согласно рекомендациям работ [2, 3] был установлен во всех случаях постоянным и равным $p = 30$. Полученные результаты иллюстрируются двумя таблицами и четырьмя рисунками.

В табл. 1 и 2 для двух вариантов порога $\rho_0 = 1{,}0$; $\rho_0 = 0{,}8$ представлены (фрагментарно) две квад-

Таблица 1

Матрица ВИР при пороге 1,0

r/*v*	1	2	3	4	5	6	7	8	9	10	...	22	23	24
1	0,00	0,89	0,96	1,01	1,03	1,61	1,3	1,14	1,13	1,14	...	1,03	0,69	0,82
2	0,57	0,00	1,54	1,36	1,84	2,4	2,55	2,01	1,99	1,55	...	1,46	1,12	1,33
3	0,96	2,67	0,00	5,46	3,34	4,81	3,28	2,17	1,4	2,13	...	3,05	2,14	1,62
4	0,55	0,91	1,56	0,00	2,95	2,97	2,03	1,88	1,13	1,12	...	2,46	1,55	1,67
5	0,94	1,58	2,73	3,47	0,00	6,88	3,92	3,1	1,3	2,41	...	1,64	1,87	1,5
6	1	4,54	4,46	3,04	3,52	0,00	4,08	2,79	5,13	12,82	...	1,64	2,75	1,51
7	1,81	2,93	3,04	2,29	2,95	3,41	0,00	2,91	4,29	2,55	...	7,05	3,58	3
8	1,07	1,17	2,46	1,72	2,39	2,54	1,98	0,00	2,9	1,16	...	2,44	2,35	2,55
9	0,83	2,63	2,84	1,47	2,04	2,52	2,59	2,51	0,00	2,13	...	1,54	2,56	2,03
10	0,62	1,36	2,31	1,61	1,54	3,52	1,86	1,25	2,09	0,00	...	2,47	1,74	1,36
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
22	1,05	2,14	2,29	2,32	2,37	6,16	5,5	3,78	2,98	3,1	...	0,00	2,6	2,45
23	0,71	1,99	1,52	2,14	3,68	5,21	3,97	1,47	2,6	1,74	...	2,78	0,00	2,06
24	1,48	5,08	4,68	4,54	1,71	4,26	3,08	7,43	4,3	4,8	...	1,9	2,66	0,00

Таблица 2

Матрица ВИР при пороге 0,8

r/*v*	1	2	3	4	5	6	7	8	9	10	...	33	34	35
1	0,00	0,77	0,78	0,57	1,27	0,53	0,63	1,28	0,98	0,91	...	1,01	1,26	1,51
2	0,89	0,00	0,71	0,82	1,32	0,84	1,35	1,00	1,25	1,71	...	1,67	1,64	1,00
3	0,53	0,57	0,00	0,83	1,61	0,87	0,94	1,23	1,21	1,60	...	1,30	2,08	1,46
4	0,52	0,80	0,92	0,00	1,26	0,74	0,73	1,03	1,09	1,50	...	1,17	1,84	1,33
5	0,58	1,51	1,09	1,18	0,00	0,93	1,05	1,41	2,11	1,35	...	1,92	1,91	1,31
6	0,53	1,11	0,98	1,00	1,85	0,00	1,32	1,66	1,89	2,03	...	2,02	1,18	2,19
7	0,59	1,64	0,97	0,56	1,27	0,77	0,00	1,55	0,77	1,28	...	1,11	1,84	1,83
8	1,14	1,59	1,28	1,37	1,16	1,21	1,20	0,00	2,23	1,90	...	2,83	1,83	2,00
9	0,92	2,08	1,81	0,82	2,56	1,91	1,02	2,64	0,00	1,68	...	2,32	1,43	2,48
10	0,78	1,12	1,03	1,43	1,62	1,34	1,16	1,62	1,85	0,00	...	1,19	1,85	1,21
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
33	3,94	3,52	7,60	1,86	3,29	2,29	3,05	4,44	1,94	2,27	...	0,00	3,58	6,10
34	1,04	1,31	1,22	1,24	1,95	1,22	1,74	1,27	1,95	1,69	...	2,48	0,00	1,49
35	3,18	1,58	1,99	3,67	4,20	2,08	3,60	2,44	7,17	1,14	...	3,50	3,44	0,00

ратные матрицы ВИР $\|\rho_{rv}\|$ для двух ТБД широкого рынка США — порядков 24 и 35 соответственно. Из сравнения этих матриц видно, что в зависимости от предъявляемых в выражении (6) требований к однородности типологических единиц в пределах каждого отдельного кластера X_r меняется не только их суммарное число, но и состав эталонов динамических единиц (1), а также их информационные свойства. В таком случае возникает естественный вопрос: имеется ли на каждом рынке строгий оптимум в отношении порога ρ_0? В работе [4] на этот вопрос дается положительный ответ и приводится интервальная оценка указанного оптимума для фондовых рынков США и России. В соответствии с ней во всех дальнейших наблюдениях над каждым из изучаемых рынков порог по ВИР в выражениях (2) и (6) был зафиксирован на одинаковом уровне $\rho_0 = 0{,}8$ (см. табл. 2). А диагностика их состояний осуществлялась по критерию МИР в формулировке (2), (4) в ее адаптивном варианте реализации [12]. В результате был сделан первый вывод проведенного исследования: типология динамики фондового рынка США на данный момент включает в себя $R^* = 30...35$ элементов, и это почти в 3 раза больше объема ТБД развивающегося фондового рынка России. Вместе с тем, установлено, что указанная типология для всех рынков весьма устойчива во времени. Например, на том же рынке США за последние 10 лет сохранились (задействованы) в текущей динамике более 20 из 35 типологических единиц тридцатилетней давности. Ценность этого вывода для практики не требует комментариев, поэтому сразу перейдем к его иллюстрациям и развитию.

На рис. 3, *а—в* представлены три временные диаграммы: индекса высокотехнологичных компаний мира Nasdaq, индекса РТС (оба — в безразмерных единицах) и котировок ММВБ по акциям НК "Лукойл" (в рублях) на интервале 2006—2009 гг. — все в логарифмическом масштабе уров-

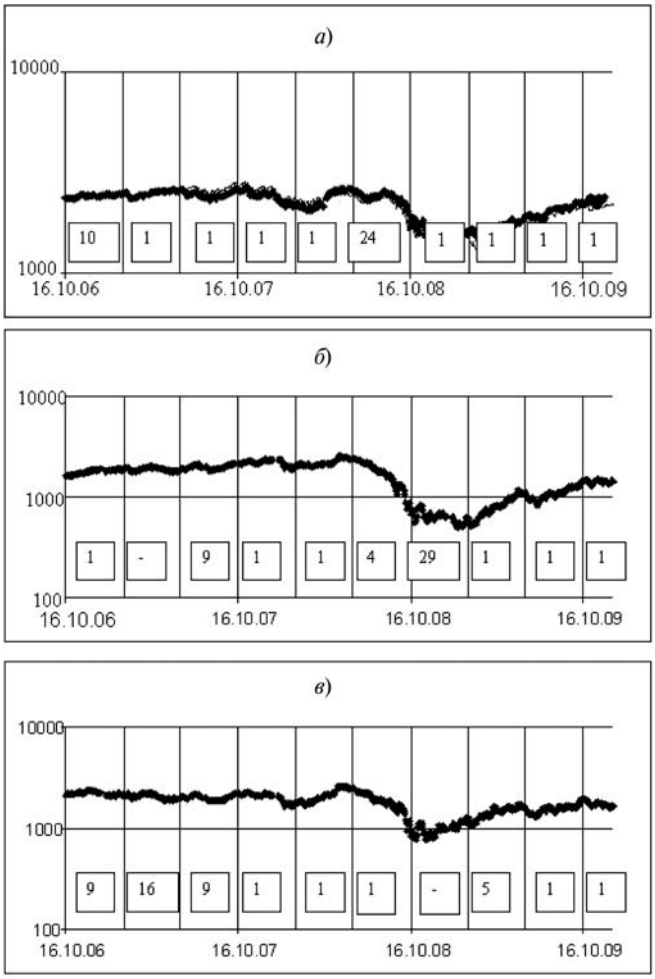


Рис. 3. Временные диаграммы:
а — индекса Nasdaq; *б* — индекса РТС; *в* — биржевых котировок акций НК "Лукойл"

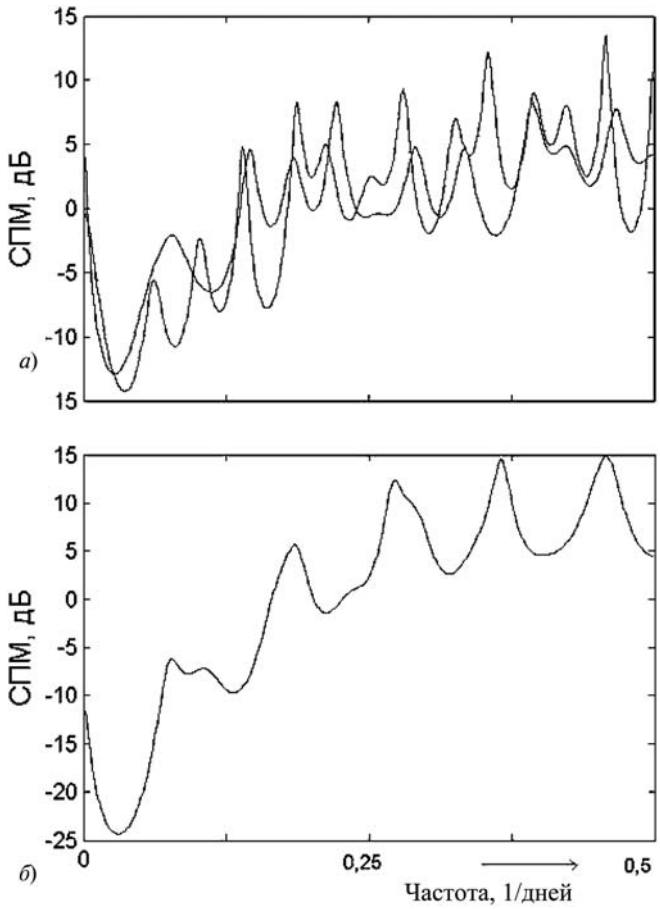


Рис. 4. АР-оценки СИМ динамики приращений котировок акций "Лукойл":
а — в состоянии 1 (две реализации); *б* — в маргинальном состоянии

ней. Здесь же над каждой осью абсцисс в границах ее промежуточных делений с интервалом примерно в один квартал (80 торговых дней) отмечены своими порядковыми номерами из ТБД по индексу S&P-500 согласно табл. 2 все выявленные в автоматическом режиме вычислений типологические единицы динамики рынка. Здесь прочерками обозначены маргинальные состояния рынка, которые не относятся ни к одной типологии.

Даже простой визуальный анализ представленных данных подтверждает их объективность и закономерность в каждом отдельном случае. А в результате их сопоставления друг с другом мгновенно следует вывод об общности ТБД разных рынков и по составу, и по свойствам отдельных элементов. При этом развивающийся характер российского фондового рынка проявляется в более частой (по сравнению с американским рынком) и вместе с тем в немотивированной иногда смене настроений среди его участников-игроков. Для подтверждения сказанного на рис. 4 представлены для сопоставления графики спектральной плотности мощности (СПМ) колебаний курса акций НК "Лукойл" в состоянии рынка 1 (рис. 4, *а*, две реализации) и в маргинальном состоянии (рис. 4, *б*). Отсюда, в частности, можно сделать главный вывод проведенного исследования: о возможности и, более того, целесообразности управления портфельными инвестициями на российском рынке ценных бумаг на основании данных прогнозирования фондового рынка США как определяющего будущую динамику их обоих.

Обсуждение полученных результатов

В основе предложенной теории колебаний рыночной конъюнктуры в динамике используется критерий минимума информационного рассогласования и два базовых понятия: кластера подобных (однотипных) колебаний и его информационного центра-эталона в метрике Кульбака—Лейблера (1). Условно говоря, общественное массовое сознание воспринимает и запоминает на будущее как нечто целое (в виде абстрактного образа — кластера) разные образцы (реализации) того или иного состояния рынка в динамике в соответствующей "сфере" своей памяти вокруг абстрактного (информационного) "центра" с заданным "радиусом". В результате получаем типологическую базу данных, различную для разных рынков и инструментов во всех странах мира. Практическая ценность такой базы данных представляется очевидной: это анализ текущего состояния рынка по принципу аналогии с одним из прошлых его же состояний и, как результат, оценивание наиболее вероятных состояний рынка в будущем. Поэтому проблема формирования и непрерывного обновления ТБД — одна из наиболее актуальных в теории и практике современного экономического анализа. В предложенном подходе она впервые преодолевается путем автоматической процедуры рекуррентного дополнения базы данных только теми из новых образцов колебаний, которые выходят за рамки первоначальных "границ" соответствующего кластера. После этого в памяти ПК автоматически смещается "центр" и корректируются границы обновленного образа-кластера. По убеждению автора статьи именно такого рода механизм реализуется в процессе восприятия рынка большинством (массой) его участников — это главный мотив проведенного исследования.

Список литературы

1. **Савченко В. В.** Прогнозирование социально-экономических процессов на основе адаптивных методов спектрального оценивания // Автометрия. 1999. 35. № 3. С. 99—108.

2. **Савченко В. В.** Теоретико-информационное обоснование линейных оценок прогнозирования // Автометрия. 2001. 37. № 5. С. 68—79.

3. **Савченко В. В.** Использование линейной авторегрессионной модели для прогнозирования динамики биржевых котировок // Автометрия. 2004. 40. № 4. С. 117—128.

4. **Савченко В. В., Пономарев Д. А.** Автоматическая периодизация случайных временных рядов с использованием метода обеляющего фильтра // Автометрия. 2009. 45. № 1. С. 56—64.

5. **Костерин А. Г.** Практика сегментирования рынка. СПб.: Питер, 2002.

6. **Кульбак С.** Теория информации и статистика. М.: Наука, 1967.

7. **Савченко В. В., Савченко А. В.** Принцип минимального информационного рассогласования в задаче распознавания дискретных объектов // Известия вузов России. Радиоэлектроника. 2005. Вып. 3. С. 10—18.

8. **Савченко А. В.** Метод направленного перебора альтернатив в задаче автоматического распознавания полутоновых изображений // Автометрия. 2009. 45. № 3. С. 90—98.

9. **Акатьев Д. Ю., Савченко В. В.** Обнаружение разладки случайного процесса по выборке на основе принципа минимума информационного рассогласования // Автометрия. 2005. 41. № 2. С. 68—74.

10. **Савченко В. В., Пономарев Д. А.** Оптимизация фонетической базы данных по группе дикторов на основе информационной теории восприятия речи // Информационные технологии. 2009. № 12. С. 7—12.

11. **Савченко В. В.** Различение случайных сигналов в частотной области // Радиотехника и электроника. 1997. 42. № 4. С. 426—431.

12. **Марпл С. Л.** Цифровой спектральный анализ и его приложения. М.: Мир, 1990.

13. **Мюллер П., Нойман П., Шторм Р.** Таблицы по математической статистике / Пер. с нем. под ред. В. М. Ивановой. — М.: Финансы и статистика, 1982.

14. **Савченко В. В.** Автоматическое распознавание речи методом дерева на основе информационного (R + 1)-элемента // Изв. вузов России. Радиоэлектроника. 2006. Вып. 4. С. 13—22.

ЖУРНАЛ В ЖУРНАЛЕ

НЕЙРОСЕТЕВЫЕ ТЕХНОЛОГИИ

№ 3
МАРТ
2011

УДК 004.932.2:004.048

П. В. Скрибцов, канд. техн. наук, ген. директор,
e-mail: skribtsov@pawlin.ru,
П. А. Казанцев,
канд. техн. наук, программист-математик,
А. В. Долгополов, программист-математик,
ООО "ПАВЛИН Техно"

Особенности реализации алгоритмов распознавания объектов на фото и видео с применением современных многоядерных процессоров[1]

Приводится описание универсального нейросетевого алгоритма для параллельных вычислений в системах с многоядерными процессорами. Алгоритм эффективен для использования в задачах реального времени: обнаружение различных объектов (лица, ключевые точки лица, автомобили, самолеты и др.) на фото высокого разрешения и видео. Алгоритм хорошо распараллеливается на современных графических процессорах и показал свою эффективность в задаче детекции лиц.

Ключевые слова: *распознавание объектов, алгоритмы распознавания объектов, анализ видео и фото, нейросетевые алгоритмы, распознавание лиц*

Введение

Распознавание объектов на фото и видео — сложная задача, так как система распознавания должна уметь идентифицировать объекты, несмотря на изменения освещенности, положения по отношению к видеокамере и формы объекта. Для решения этой задачи предлагается универсальный нейросетевой алгоритм, который хорошо распараллеливается, что позволяет эффективно ускорить его выполнение с помощью многоядерных процессоров. В связи с этим актуально применение алгоритма в задачах реального времени: обнаружение различных объектов (лица, ключевые точки лица, автомобили, самолеты и др.) на фото высокого разрешения и видео.

Общее описание алгоритма обнаружения объектов

Преобразование исходного изображения. Исходное изображение приводится к стандартному разрешению. Далее изображение преобразуется в черно-белое в градациях серого от 0 до 255. Затем возможны варианты использования устройств дополнительных быстрых преобразований-фильтров для устранения сильных перепадов яркости, бликов, усиления резкости границ и т. д.

Построение интегрального изображения. Следующим важным шагом является построение интегрального изображения [1], которое строится на основании исходного черно-белого отфильтрованного изображения со стандартным разрешением. Интегральное изображение применяется для быстрого расчета суммы яркостей пикселей в произвольных прямоугольных областях изображения (всего за четыре арифметические операции). Подобные суммы впоследствии используются как признаки изображения, являющиеся аналогами полей-рецепторов сечатки в ранних алгоритмах Розенблата [2]. Принцип построения интегрального изображения состоит в следующем. Каждый пиксель нового (интегрального) изображения с координатами (x, y) является суммой всех пикселей в прямоугольнике, левым верхним углом которого является точка с координатами (0, 0), а правым нижним — точка (x, y). Для изображения $I(x, y)$ интегральное изображение $II(x, y)$ определяется согласно формуле

$$II(x, y) = \sum_{x' \leqslant x,\, y' \leqslant y} I(x', y').$$

Расчет интегрального изображения на CPU проводится быстрым способом по рекуррентной формуле

$$II(x, y) = I(x, y) + II(x - 1, y) + \\ + II(x, y - 1) - II(x - 1, y - 1).$$

Затем изображение сэмплируется (выбирается) по ширине и высоте фрагментами — прямоугольниками с варьируемыми размерами. Изображение сэмплируется таким образом, чтобы весь прямоугольник попадал внутрь изображения — объект, частично попавший в кадр, не подлежит распознаванию.

Расчет и выбор информативных признаков. Каждый прямоугольник разбивается сеткой с $s_x = 20$ делениями по ширине и $s_y = 24$ делениями по высоте. Узлы данной сетки являются вершинами прямоугольников (рис. 1), яркости которых можно быстро вычислить по формуле

$$\sum_{A(x) < x' \leqslant C(x),\, A(y) < y' \leqslant C(y)} i(x', y') = \\ = I(A) + I(C) - I(B) - I(D) \qquad (1)$$

и использовать их в качестве входных значений для ступеней каскадов нейросетевых классификаторов.

Для более точного вычисления координат узлов сетки используется билинейная интерполяция.

[1] Работа выполнена в рамках работ по государственному контракту ГК-02.514.11.4127_24.11.09.

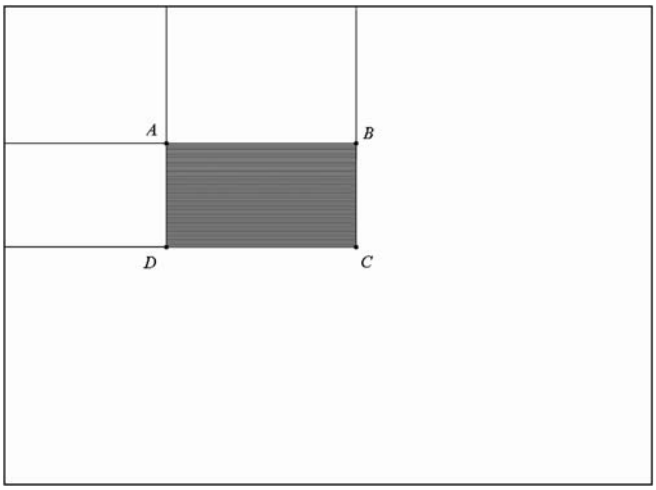


Рис. 1. Вычисление средней яркости внутри прямоугольника с использованием интегрального изображения

Значение интегрального изображения в точке (x, y) вычисляется по формуле

$$f(x, y) \approx f(x_1, y_2)(1 - x)(1 - y) + f(x_2, y_2)(1 - y)x + \\ + f(x_1, y_1)(1 - x)y + f(x_2, y_1)xy,$$

где $f(x_1, y_2), f(x_2, y_2), f(x_1, y_1), f(x_2, y_1)$ — известные значения интегрального изображения в координатах, образующих квадрат, внутрь которого попадает узел сетки (рис. 2, см. третью сторону обложки).

Далее каждое значение нормализуется путем умножения на коэффициент

$$\beta = \frac{\sigma_0}{A(\sigma + 1)},$$

где A — площадь просэмплированного прямоугольника; σ — стандартное отклонение (квадратный корень из дисперсии) внутри просэмплированного прямоугольника; $\sigma_0 = 42{,}5$ — величина, отображающая яркости 99,9 % пикселей в диапазон [0...255] при предположении, что яркости внутри просэмплированного прямоугольника распределены по нормальному закону.

Далее, на основе полученных значений интегрального изображения в узлах сетки, по формуле (1) можно вычислить значения яркости внутри требуемого прямоугольника-признака.

Также в каждую комбинацию добавляется дисперсия внутри просэмплированного прямоугольника.

Стандартное отклонение описывается выражением

$$\sigma = \sqrt{m^2 - \frac{1}{N}\sum_{i=1}^{N} x_i^2},$$

где m — среднее значение фрагмента; N — число пикселей фрагмента; x_i — значение i-го пикселя фрагмента. Для фрагмента изображения среднее значение вычисляется с помощью интегрального изображения, а $\sum_{i=1}^{N} x_i^2$ — с помощью квадратного интегрального изображения.

Квадратное интегральное изображение $ISI(x, y)$ рассчитывается из исходного изображения:

$$ISI(x, y) = \sum_{x' \leqslant x, y' \leqslant y} I(x', y')^2.$$

Информативный набор признаков формируется исходя из их информативности. Оценка информативности осуществляется по формуле (2), являющейся следствием формулы Байеса:

$$P_{error} = \int \{p(x|C_1)P(C_1)p(x|C_2)P(C_2)\}^{1/2} dx, \qquad (2)$$

где P_{error} — ошибка разделения классов; $P(C_1)$ и $P(C_2)$ — априорные вероятности появления классов C_1 и C_2 (объект и фон); $p(x|C_1)$ и $p(x|C_2)$ — значения функций плотности условных вероятностей. Величины в правой части формулы (2) рассчитываются на основе обучающей выборки. Далее, пары признаков ранжируются по величине P_{error} в порядке возрастания, в результате в верхней части списка оказываются пары признаков, для которых ошибка P_{error} — наименьшая.

Каскад нейросетевых классификаторов. Для ускорения алгоритма распознавания используют каскад нейросетевых классификаторов такой, что младшие ступени каскада осуществляют классификацию по меньшему набору признаков и являются нейронными сетями типа MLP меньших размеров. Младшие ступени каскада являются более "слабыми" классификаторами, а старшие — более "сильными". Суть данного метода состоит в том, что для распознавания некоторых фрагментов достаточно малых и быстрых классификаторов. На первую ступень каскада — наиболее слабый классификатор —подаются все возможные фрагменты изображения. На следующую, более сильную ступень каскада, подаются только те фрагменты, которые были пропущены первой ступенью, и т. д. (рис. 3).

Таким образом, к последнему каскаду доходят наиболее сложные для классификации фрагменты, которые классифицируются сильными и медленными классификаторами. Число фрагментов на старших каскадах достаточно мало, поэтому время работы старших каскадов незначительно. При текущей настройке каскада первая ступень отсеивает уже 75...78 % отрицательных примеров ("не объектов"), последняя — 99,999 %.

Более слабые каскады включают наиболее информативные комбинации признаков, а на каждой последующей ступени каскада к ним добавляются дополнительные признаки, усиливающие классификатор. При этом размеры нейронных сетей, осу-

ществляющих классификацию, увеличиваются так, чтобы они могли разделить примеры в увеличенных пространствах признаков.

Ступени каскадов настроены по обучающей выборке таким образом, чтобы было обеспечено распознавание всех объектов обучающей выборке, что дает уровень ложного отсева, равный нулю.

Каждая ступень каскада обучается на всех положительных примерах и некоторой части отрицательных примеров, которая была неправильно распознана всеми предыдущими обученными ступенями (рис. 4).

Для реализации этого алгоритма заводится массив весов негативных примеров. Изначально эти веса выставляются в значение 1. Если после обучения текущего каскада этот же каскад правильно распознает какой-либо отрицательный пример, то вес такого примера уменьшается по формулам алгоритма Adaboost [7], а если не распознает — увеличивается. Эти веса в дальнейшем играют роль вероятностей включения соответствующих отрицательных примеров в обучающую выборку. Таким образом, более вероятным становится добавление в обучающую выборку примера, который не был распознан предыдущими ступенями.

Рис. 3. Схема распознавания объекта

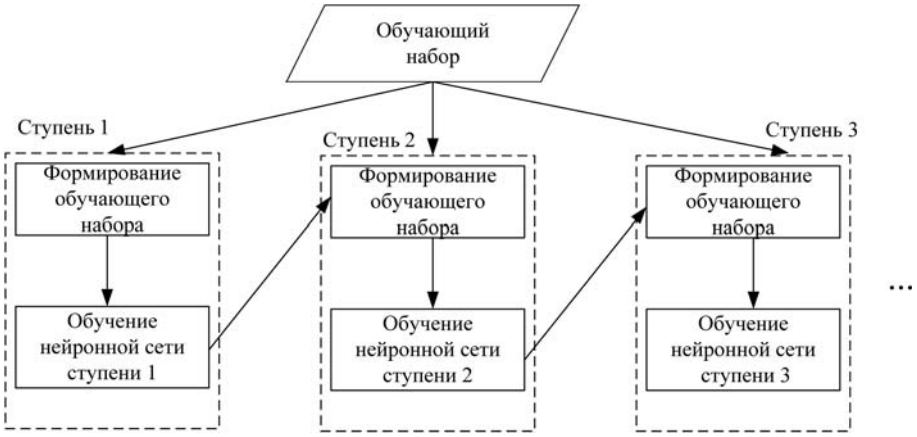


Рис. 4. Обучение многокаскадного классификатора объектов

Каждая ступень, кроме последней, представляет собой трехслойный персептрон с прямыми последовательными связями и нелинейными функциями активации. Размеры слоев увеличиваются в зависимости от размерности входного вектора.

Приращения числа нейронов в слоях выбраны экспериментальным способом. Обучение проводится по правилу обратного распространения ошибки.

Последняя ступень каскада обучается по алгоритму Adaboost, в котором базовыми классификаторами являются персептроны с двумя нейронами в скрытом слое и с одним нейроном в выходном слое. Использование алгоритма Adaboost для обучения только последней ступени обусловлено тем, что данный алгоритм позволяет получить более высокое качество распознавания на выборке малого размера за счет итеративного наращивания размера персептрона.

Формирование признаков и обучение сетей при таком подходе можно распараллелить с помощью графических процессоров. Обучение и расчет выходов нейронных сетей хорошо ускоряются на многоядерных процессорах [3].

Ускорение вычислений на современных многоядерных процессорах

В алгоритме ускорения вычислений наиболее вычислительно емкими являются: формирование интегральных изображений, формирование матриц признаков, расчет выходов нейронной сети, фильтрация фрагментов (*stream compaction*). Эти операции можно эффективно распараллеливать с помощью GPU (графические процессорные устройства) с помощью технологии NVIDIA CUDA [9].

Расчет выходов нейронных сетей ускоряется за счет ускорения умножения матриц. Расчет признаков хорошо распараллеливается с помощью CUDA, при этом одна нить рассчитывает один признак для одного фрагмента.

Диаграмма распределения операций детектирования на CPU представлена на рис. 5, *а*, на рис. 6 — диаграмма распределения операций после ускорения классификации с помощью GPU.

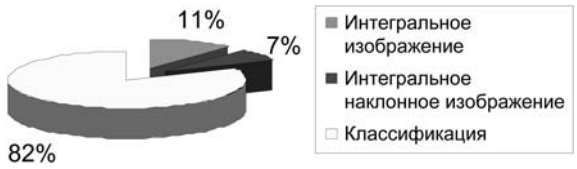


Рис. 5. Диаграмма распределения операций детектирования объекта на CPU

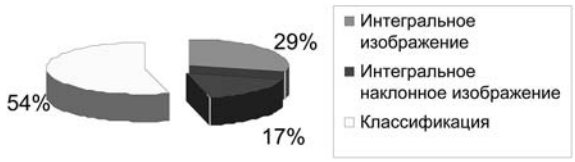


Рис. 6. Диаграмма распределения операций детектирования объекта после ускорения классификации с помощью GPU

По диаграммам видно, что если на CPU временем расчета интегральных изображений можно пренебречь, то после ускорения классификации время расчета интегральных изображений на CPU сопоставимо со временем классификации на GPU. Поэтому необходимо ускорение расчета интегральных изображений.

Построение интегрального изображения CUDA. На CUDA интегральное изображение рассчитывается в два этапа согласно формуле

$$II(x, y) = \sum_{x'=0}^{x} \sum_{y'=0}^{y} I(x', y').$$

Сначала рассчитывается интегральное изображение по столбцам, затем по строкам, при этом может использоваться алгоритм префиксного суммирования [1].

Алгоритм префиксного суммирования. Алгоритм префиксного суммирования формирует массив, в котором каждый элемент равен сумме предыдущих элементов исходного массива. Например, из массива

$$[a_0, a_1, a_2, ..., a_{n-1}]$$

алгоритм сформирует новый массив

$$[a_0, (a_0 + a_1), (a_0 + a_1 + a_2), ..., (a_0 + a_1 + a_2 + a_{n-1})].$$

Алгоритм префиксного суммирования состоит из двух частей: восходящей фазы (редукция) и нисходящей. Этот процесс легче всего описать с помощью двоичного дерева (рис. 7) [3].

Подробно о реализации на CUDA алгоритма префиксной суммы написано в [4].

На видеокарте NVIDIA 8800 GTX на небольшом числе элементов алгоритм префиксной суммы не дает ускорения, заметное ускорение ~6x достигается при числе элементов 16777216. Поэтому часто разработчики строят интегральное изображение на CPU, а затем копируют его на GPU [9].

Для построения интегрального изображения использовали следующий подход. Интегральное изображение по столбцам или по строкам рассчитывается следующим образом: одна нить рассчитывает один столбец или одну строку, при этом суммы хранятся в регистрах. Вместо транспонирования после расчета столбового изображения использовали текстурную память. Время построения и достигнутое ускорение на различных изображениях представлены на рис. 8 и 9.

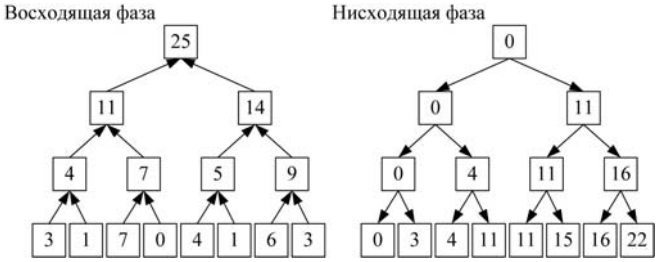


Рис. 7. Двоичное дерево, описывающее алгоритм префиксной суммы

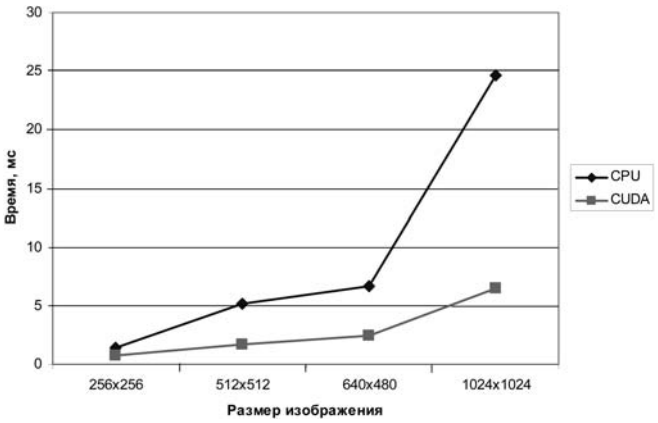


Рис. 8. Время построения интегрального изображения в зависимости от размера изображения (CPU:P4 3,2 ГГц, GPU: NVIDIA 8600GT)

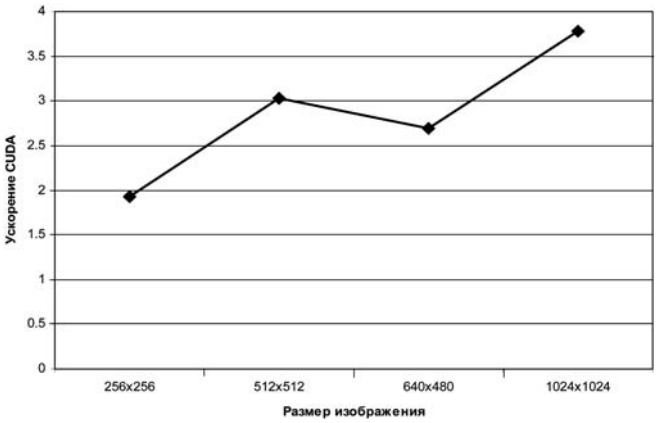


Рис. 9. Ускорение расчета интегрального изображения с помощью CUDA в зависимости от размера изображения (CPU:P4 3,2 ГГц, GPU: NVIDIA 8600GT)

Построение наклонного интегрального изображения. Для распознавания сложных наклоненных объектов используется наклонное интегральное изображение (наклон 45°) [4].

Наклонное интегральное изображение $Itilted(x, y)$ рассчитывается согласно формуле

$$Itilted(x, y) = \sum_{y' \leqslant y, y' \leqslant y - |x - x'|} I(x', y').$$

Наклонное интегральное изображение можно представить как сумму левого и правого наклонных интегральных изображений (рис. 10, см. третью сторону обложки):

$$Itilted(x, y) = Ileft(x, y) + Iright(x + 1, y - 1).$$

Для расчета левого и правого интегральных изображений сначала рассчитывается интегральное изображение по столбцам $C(x, y)$, как для обычного интегрального изображения.

Левое интегральное изображение $Ileft(x, y)$ вычисляется по реккурентной формуле

$$Ileft(x, y) = Ileft(x - 1, y - 1) + C(x, y).$$

Правое интегральное изображение вычисляется по аналогичной формуле

$$Iright(x, y) = Iright(x + 1, y - 1) + C(x, y).$$

Для ускорения расчета левого и правого интегральных изображений можно воспользоваться алгоритмом префиксной суммы. Суммы считаются по диагоналям столбового интегрального изображения в соответствующих направлениях (рис. 11, см. третью сторону обложки).

Фильтрация ложных гипотез с помощью CUDA. Для фильтрации гипотез, которые не прошли проверку на соответствующем каскаде, применяется алгоритм удаления элементов из массива (Stream Compaction) [8].

Алгоритм Stream Compaction состоит из двух этапов (рис. 12).

1. Префиксная сумма — на данном этапе формируется новый массив из "1" и "0". Элементам, которые необходимо включить в новый массив, соответствует "1", элементам, которые необходимо удалить, — "0". К этому массиву применяется алгоритм префиксной суммы.

2. Копирование выходов — в этой части алгоритма массив сумм, полученный на первом этапе, служит в качестве таблицы адресов, в которые необходимо записывать элементы нового массива.

Применение алгоритма распознавания объектов

Данный алгоритм был успешно применен к задаче детектирования лиц — одной из самых актуальных в системах компьютерного зрения (рис. 13, см. третью сторону обложки).

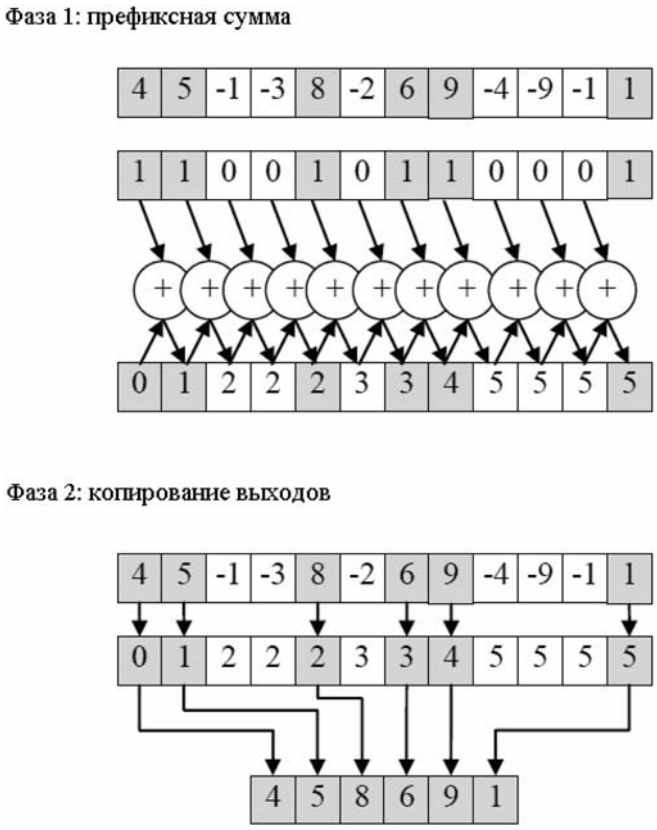


Рис. 12. Схема удаления элементов из массива

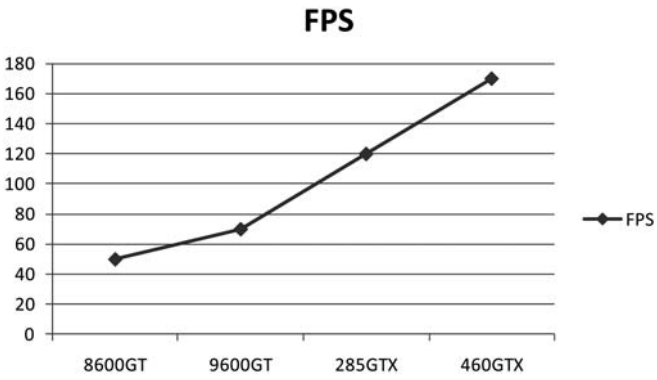


Рис. 14. Уровень FPS (кадров в секунду) при детектировании лиц на различных видеокартах (видеопоток 640 × 480)

На рис. 14 представлен уровень FPS при детектировании лиц в видеопотоке разрешением 640 × 480 на различных видеокартах. Тестирование проводилось на персональном компьютере следующей конфигурации:

- процессор Intel Core 2 Duo E8400 3,00 ГГц;
- память (RAM) 3,25 Гбайт, 3 ГГц.

Выводы

В статье был рассмотрен универсальный алгоритм распознавания объектов на фото и видео, который использует нейросетевые технологии. Дан-

ный алгоритм позволяет распознавать любые объекты — автомобили, самолеты, печатные символы, лица. Алгоритм хорошо распараллеливается на современных графических процессорах и показал свою эффективность в задаче детектирования лиц.

Список литературы

1. **Blelloch G. E.** Prefix Sums and Their Applications // School of Computer Science, Carnegie Mellon University, techreport, 1990.
2. **Rosenblatt F.** The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain // Psychological Review. 1958. Vol. 65, N 6. P. 386—408.
3. **Скрибцов П. В.** Аппаратное ускорение нейросетевых алгоритмов с применением графических процессоров // Труды III Междунар. конф. "Параллельные вычисления и проблемы управления". М.: ИПУ, 2006.
4. **Lienhart R., Kuranov E., Pisarevsky V.** Empirical Analysis of Detection Cascades of Boosted Classifiers for Rapid Object Detection // Pattern Recognition. 2003. Vol. 2781. P. 297—304.
5. **Harris M.** Parallel Prefix Sum (Scan) with CUDA. 2007, April.
6. **Хайкин С.** Нейронные сети полный курс. М.: Издат. дом "Вильямс", 2006.
7. **Viola P. Jones M. J.** Rapid Object Detection using a Boosted Cascade of Simple Features // IEEE CVPR. 2001.
8. **Billeter M., Olsson O., Assarsson U.** Efficient Stream Compaction on Wide SIMD Many-Core Architectures // In Proceedings of High Performance Graphics. August. 2009. P. 159—166.
9. **Harvey J. P.** GPU Acceleration of Object Classification Algorithms Using NVIDIA CUDA. September, 2009.

УДК 004.032.

А. И. Галушкин, д-р техн. наук, проф.,
ФГНУ "Центр информационных технологий и систем органов исполнительной власти",
г. Москва,
e-mail: neurocomputer@yandex.ru

Аналитические методы и нейросетевые технологии в решении задач по программе "Протеом человека"

Целью данной работы является определение путей создания аналитического ядра информационно-аналитической системы по программе "Протеом человека". Отмечены основные направления работ: применение нейросетевых суперкомпьютерных технологий в молекулярном моделировании в медицине и биоинженерии; использование нейросетевых технологий и методов семантического анализа для систематизации постгеномных данных; проведение исследований в области выявления структурно-функциональных взаимосвязей в белках методами смыслового дешифрования первичной структуры.

Ключевые слова: *нейросетевые технологии, геном, протеом, классификация, кластеризация, текстовая обработка*

Современные информационные технологии играют важную инструментальную роль в биологической проблеме исследования генома и протеома человека.

Целью данной работы является определение путей создания аналитического ядра информационно-аналитической системы по программе "Протеом человека".

Основными предпосылками роста числа необходимых для решения задач в современных и перспективных информационно-аналитических системах являются следующие:

- резкое увеличение объемов обрабатываемой информации;
- расширение множества типов обрабатываемой информации (параметрическая, текстовая, мультимедийная);
- появление сложных формализуемых и неформализуемых аналитических задач;
- повышение требований к качеству решения аналитических задач.

Основными направлениями работ в части создания информационно-аналитической системы — инструмента эффективной работы биологов-исследователей, будут следующие:

1. Применение нейросетевых суперкомпьютерных технологий в молекулярном моделировании в медицине и биоинженерии, в том числе в моделировании биомакромолекул, являющихся мишенями действия лекарственных препаратов или имеющих ценность для выявления механизмов развития социально значимых заболеваний.

2. Использование нейросетевых технологий и методов семантического анализа для систематизации постгеномных данных в целях повышения эффективности практического использования в медицине и здравоохранении.

3. Проведение исследований в области выявления структурно-функциональных взаимосвязей в белках методами смыслового дешифрования первичной структуры (дескриптомики).

Поскольку объем полезной информации в проекте "Протеом человека" превышает пределы возможностей персональных компьютеров по хранению и обработке данных, в пилотной фазе про-

граммы предусмотрена разработка информационно-аналитической системы распределенного (облачного) типа на платформе персональных суперЭВМ со специализированным программным обеспечением.

Создание подобной распределенной системы, проблемно-ориентированной на решение задач по программе "Протеом человека", позволит:

- резко повысить эффективность использования информации, необходимой для проведения исследований;
- качественно повысить вычислительную эффективность решения самих задач за счет организации "облачных" вычислений;
- повысить эффективность коллективной работы распределенных групп ученых-биологов.

Техническая основа информационно-аналитической системы для реализации программы "Протеом человека" должна включать в себя следующие компоненты:

- сетевой комплекс из 50 вычислительных узлов для предприятий биологической направленности;
- около 1500 персональных суперЭВМ, размещенных на данных предприятиях.

Работы по созданию информационно-аналитической системы должны проводиться в три этапа:

1. Создание опытного образца персональной суперЭВМ с системным программным обеспечением, инструментальными средствами и прикладным программным обеспечением первой очереди.

2. Реализация опытного района сетевого комплекса.

3. Организация серийного производства персональных суперЭВМ и реализация полномасштабного варианта сетевого комплекса.

Основными разделами работ по созданию информационно-аналитической системы в рамках программы "Протеом человека" являются следующие:

1. Разработка образцов персональных суперЭВМ для решения задач программы "Протеом человека".

2. Разработка общего системного программного обеспечения персональных суперЭВМ.

3. Средства обеспечения работы данных вычислительных средств в GRID-сети с использованием опыта ЦИТиС в работах по созданию информационной инфраструктуры отечественной наносети.

4. Разработка системы мониторинга GRID-сети.

5. Телекоммуникационная система.

6. Программные и технические средства обеспечения облачных вычислений.

7. Программные и технические средства работы с периферийным лабораторным оборудованием.

8. Отказоустойчивые системы хранилищ информации большого объема, включая варианты с высокой пропускной способностью.

9. Разработка базовых инструментальных систем программирования, ориентированных на решения задач по программе "Протеом человека".

10. Проведение работ по приобретению, изучению и освоению зарубежных пакетов программ, ориентированных на мультипроцессорные вычислительные системы с многоядерной архитектурой, адекватные задачам программы "Протеом человека", включая:

- пакеты программ решения задач биоинформатики, в том числе пакеты программ молекулярного моделирования;
- пакеты программ нейросетевого моделирования в задачах биоинформатики;
- библиотеки программ решения математических задач;
- системы управления базами данных и базами знаний;
- пакеты программ, реализующие модели общего вида, в том числе пространственные и пространственно-временные модели;
- пакеты программ для моделирования физических процессов, в том числе:
  — в масс-спектрометрии;
  — в микроскопии;
  — в моделировании квантовых эффектов.

11. Разработка отечественных пакетов программ для решения задач по программе "Протеом человека".

12. Разработка системы обеспечения информационной безопасности информационно-аналитической системы.

13. Разработка комплекса систем аналитической обработки информации в задачах программы "Протеом человека", включая:

- методы извлечения знаний, в том числе алгоритмы экстраполяции, классификации, кластеризации, заполнения неполных таблиц;
- обработку слабоструктурированной текстовой информации применительно к проблемам протеомики;
- реализацию экспертных систем;
- разработки современных методов и систем визуализации, в том числе 3D-стерео и интерактивные средства.

14. Разработка образцов и документации специализированной высокопроизводительной вычислительной системы решения задач биоинформатики.

15. Разработка систем обмена информацией с международными организациями и сетями по проблемам протеомики (с учетом опыта физиков-ядерщиков).

16. Разработка электронной библиотеки публикаций по проблеме протеомики.

17. Освоение в серийном производстве персональных суперЭВМ для решения задач программы "Протеом человека".

Рис. 1. Структура аналитического ядра информационно-аналитической системы по программе "Протеом человека"

Анализ существующих систем и возможных аналитических задач по программе "Протеом человека" позволил представить структуру аналитического ядра в виде, показанном на рис. 1.

Технология извлечения знаний (*data mining*) содержит следующие разделы:

- анализ и прогнозирование временных рядов;
- классификация или распознавание негативных объектов;
- кластеризация;
- заполнение неполных таблиц;
- задачи оптимизации;
- построение экспертных систем;
- поиск ассоциативных связей.

На рис. 2 представлен предлагаемый подход к созданию экспертных систем.

Информационным базисом биоинформатики являются результаты экспериментальной деятельности, зафиксированные в виде описательных документов, публикации — иными словами, представленные массивами слабоструктурированной информации (до 50 млн статей). В этой ситуации правомерна постановка задачи использования информационных технологий семантического анализа данных, обеспечивающих "глубокую" запросную фильтрацию текстовых массивов с высоким уровнем релевантности получаемых ответов и последующую автоматическую формализацию предметной области, например, в виде семантической сети, включающей ключевые понятия, их классификацию и сущностные связи между ними.

Основой работ в этой области будут технологии обработки больших массивов текстовой информации, в том числе:

- системы автоматического мониторинга Интернет-ресурсов и с привлечением экспертов-аналитиков;
- системы обработки массивов структурированной текстовой информации.

Система автоматического мониторинга информационных ресурсов сети Интернет и с привлечением экспертов-аналитиков включает в себя:

- технологию динамического эволюционного проектирования баз знаний;
- технологию метапоиска информации в сети Интернет на основе нейросетевой экспертной системы;
- технологию сбора текстовых данных из выделенных ресурсов информационного массива Интернет и анализа смысла текста на основе семантических сетей с применением нейросетевых методов.

В таблице представлены основные функциональные характеристики разработанных и используемых систем обработки больших массивов текстовой информации.

При необходимости обработки мультимедийной информации должен быть использован опыт решения конкретных практических задач:

- фильтрации изображений;
- сегментации изображений;
- сжатия изображений;
- выделения на изображениях элементов заданной формы;
- скелетонизации изображений;
- выделения на изображениях движущихся объектов;
- оценки качества изображений.

В аналитическом ядре возможно применение следующих технологий построения моделей:

- сценарные модели;
- модели, описываемые системой обыкновенных нелинейных дифференциальных уравнений;

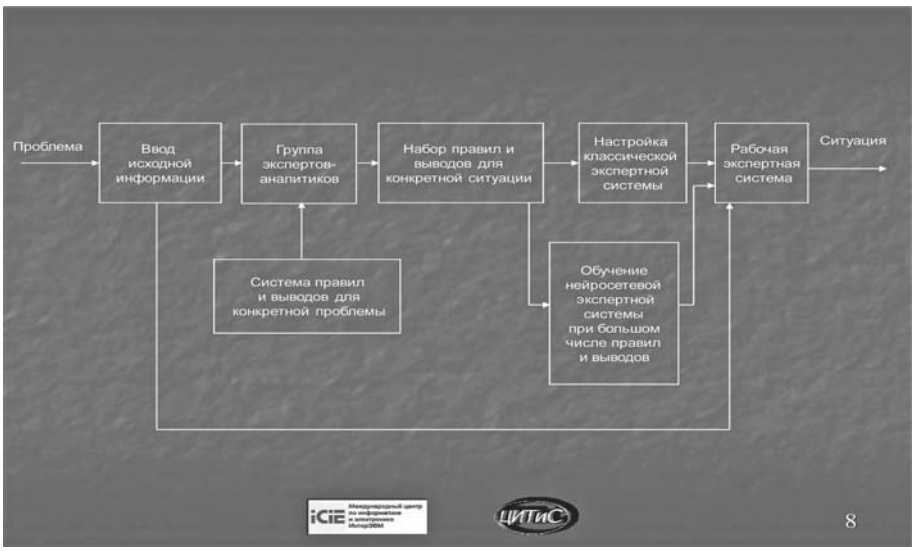


Рис. 2. Предлагаемый подход к созданию экспертной системы используемых систем обработки больших массивов текстовой информации

Основные функциональные характеристики разработанных и используемых систем обработки больших массивов текстовой информации

Алгоритм обработки	Входная информация	Выходная информация
Поиск ключевых слов	Комплект текстов	Ключевые слова
Поиск по ключевым словам	Ключевые слова	Проблемно-ориентированный текст
Классификация текстов	Комплект текстов	Тексты, классифицированные по тематике
Кластеризация текстов	Комплект текстов	Тексты, разбитые на группы без наличия предварительной информации о тематике
Поиск материалов по аннотации	Аннотация текстового материала	Комплект текстовых материалов, соответствующих аннотации
Реферирование	Текст	Реферат

- модели, описываемые системой дифференциальных уравнений в частных производных;
- нейросетевые модели сосредоточенных и распределенных динамических систем.

Средства визуализации результатов аналитических исследований по программе "Протеом человека" должны включать в себя:

- большие составные экраны;
- локальные и составные безочковые средства 3D- и стереовизуализации;
- персональные и коллективные средства визуализации с применением специальных очков.

В рамках программы "Протеом человека" должны быть использованы следующие перспективные технологии визуализации:

- графического 3D-моделирования;
- стерео 3D-визуализации;
- интерактивной 3D-визуализации (взаимодействие с данными в 3D и 3D Stereo);
- виртуальной реальности;
- дополненной (*Augmented Reality*) (синтезирующей элементы виртуальной реальности и элементы реального мира);
- геоинформационные (GIS) технологии.

Ниже приведена последовательность этапов при решении аналитических задач по программе "Протеом человека".

1. Формулировка прикладной аналитической задачи на физическом уровне.

2. Математическая формулировка прикладной аналитической задачи.

3. Количественная оценка сложности решаемой прикладной аналитической задачи.

4. Выбор логического базиса и метода решения прикладной аналитической задачи.

5. Моделирование алгоритма решения прикладной аналитической задачи на стандартной рабочей станции при реальных количественных показателях ее сложности.

6. Оценка времени решения прикладной аналитической задачи.

7. Выбор направления и технологии аппаратной реализации алгоритма решения прикладной аналитической задачи в случае, когда в соответствии с требованиями заказчика время решения нужно резко сокращать.

Необходимо отметить роль нейросетевых технологий в решении аналитических задач по программе "Протеом человека".

Нейросетевые технологии весьма активно используются в геномике и протеомике и синтезе лекарственных соединений уже в течение 20 лет [2—5].

Необходимо также отметить достаточно широкое применение нейросетевых технологий в разделах вычислительной химии [1].

Применение нейросетевых технологий в рассматриваемой области знаний и интеграция результатов привели в настоящее время к формированию нового направления — *Computational Neurogenetic* [8].

В программе "Протеом человека" активное применение нейросетевых технологий обязательно во всех направлениях аналитических исследований:

- извлечение знаний;
- обработка больших массивов текстовой информации;
- обработка мультимедийной информации;
- построение моделей.

Список литературы

1. **Баскин И. И., Палюлин В. А., Зефиров Н. С.** Нейроматематика — будущее вычислительной химии // Нейрокомпьютер. 1997. № 3, 4.

2. **Zhang W., Shmulevich I.** (ed.). Computational and Statistical Approaches to Genomics. Kluwer Academic Publishers. 1999.

3. **Wu C. H., McLarly J. W.** Neural Networks and Genom Informatics. Elsevier, 2000.

4. **Zupan J., Gasteiger J.** Neural Networks in Chemistry and Drug Design. Wiley-VCH. 1999.

5. **Галушкин А. И.** Нейрокомпьютеры в Китае на рубеже тысячелетий. М.: Горячая линия — ТЕЛЕКОМ. 2004.

6. **Galushkin A. I.** Neural Network Theory. Springer. 2007.

7. **Галушкин А. И.** Нейронные сети: основы теории. М.: Горячая линия — Телеком. 2010.

8. **Kasabov N.** Computational Neurogenetic. IJCNN. 2008.

УДК 004.032.26

И. М. Воронков[1], вед. инж.,
e-mail: voronkov@inevm.ru,
И. В. Кречетов[1], инженер,
e-mail: neurocomputer@yandex.ru,
А. А. Харламов[2], д-р техн. наук, зав. лаб.,
e-mail: kharlamov@analyst.ru
[1] Международный центр по информатике и электронике (Интер ЭВМ)
[2] Федеральный институт развития образования

Обработка больших массивов текстовой информации и перспективы ее развития для информационно-аналитических систем, программная и аппаратная реализация

Рассматриваются нейросетевой подход к решению задач текстовой обработки на основе построения семантической сети для документов, а также вопросы аппаратной реализации алгоритмов построения семантической сети для документов.

Ключевые слова: *семантическая сеть, нейросетевые алгоритмы, текстовая обработка, графические ускорители, ПЛИС*

Введение

Задачи обработки больших объемов текстовой информации представляют собой достаточно сложную систему работы на естественном языке, основное назначение которой повышение эффективности за счет интеллектуальной обработки документов, в том числе с помощью технологии извлечения информации, а также благодаря средствам визуализации и коррекции результатов. Кластеризация текстовой информации — это наиболее эффективный способ обработки больших объемов информации, суть которого разбиение множества обрабатываемых документов на классы (кластеры, рубрики), содержащие близкие по содержанию документы.

Семантическая сеть на основе нейронной сети

В основе алгоритма кластеризации лежит подход, связанный с построением для каждого документа (и для базы в целом) смыслового портрета (смысловой сети), включающей ключевые слова и устойчивые словосочетания.

В задачах обработки текстов хорошо известно использование персептронов для формирования так называемой модели языка. Упрощенно ее можно сформулировать следующим образом. Текст сканируется окном длины в n слов со сдвигом на одно слово за один такт. На вход персептрона подаются фрагменты текста длины n слов. Предварительно словарь должен быть проиндексирован, таким образом на вход персептрона подаются последовательности кодов слов. Происходит обучение персептрона на множестве текстов. Если выборка была представительной, в режиме распознавания предсказывается вероятность появления $(n + 1)$-го слова вслед за n-м словом. Эта модель хорошо работает на нефлективных языках, где число словоформ одного слова сравнительно невелико, так как требуется представительная выборка для формирования модели языка. Причем речь идет о модели, содержащей информацию о триграммах, т. е. при $n = 3$. Известно, что для русского языка эта модель нереализуема в лоб: слишком большого объема обучающая выборка необходима для ее формирования.

Несмотря на то, что тексты представляют собой синтагматические структуры, т. е. информационные элементы выстраиваются в последовательности, лингвистическая информация имеет многоуровневую иерархическую структуру [1]. На каждом уровне иерархии имеется словарь событий данного уровня, в котором элементы противопоставлены друг другу, т. е. формируют парадигмы (словари). Если иметь в виду, что на входе мы имеем буквы, то это, соответственно, уровни иерархии снизу вверх: морфологический, лексический, синтаксический. Сверху надстраиваются еще два экстралингвистических уровня: семантический и прагматический.

На морфологическом уровне формируются словари префиксов, постфиксов и флективных морфем, а также словарь корневых морфем, попросту — приставок, суффиксов, окончаний и корней слов. Отдельно может формироваться словарь слогов.

На лексическом уровне формируется словарь корневых основ (корень слова плюс суффикс), словарь лексем (слов и их словоформ), словарь устойчивых словосочетаний.

Синтаксический уровень представляется словарем синтаксем. Синтаксемы — это сочетания слов, в которых есть хотя бы два полнознаменательных слова, главное и второстепенное. Каждое такое сочетание характеризует отдельный грамматический факт. Такие синтаксемы разбиваются на кластеры: существительного, глагола, прилагательного и причастия.

Представления экстралингвистических уровней похожи друг на друга. Они характеризуют сочетаемость слов в предложениях (высказываниях) [2].

Если рассмотреть внимательно различные способы формирования семантических представлений [3], окажется, что все они сводятся к сетевому: к простой ассоциативной сети, или к неоднородной, в которой связи размечены метками отношений [4]. Ну а сеть — это пары ключевых понятий. Отличия семантики от прагматики трудно разделимы. Но в целом можно сказать, что семантическое представление опосредует модель мира, а прагматическое — модель отдельной ситуации. Таким образом, с одной стороны, семантическое представление, включающее в себя в качестве вершин сети потенциально все понятия, характеризующие сущности мира, в их взаимосвязях, является статичным представлением. С другой стороны, прагматическое представление, описывающее конкретную ситуацию, включающее в свой состав те же понятия, является динамическим представлением, развертывающимся как цепь на семантической сети, или несколько параллельных цепей.

Независимо от уровня представления языковой информации можно заметить, что она является динамической: текстовые события развертываются во времени. Это накладывает отпечаток на способы обработки текстовой информации: временная структура начинает играть главную роль.

Статические нейронные сети могут использоваться для представления последовательности статических событий без учета длительности события во времени. Например, с помощью персептрона можно представить последовательность букв в слове, последовательность слов в тексте [5]. Модель предсказывает следующее событие на основе предыдущих $n - 1$ событий.

Статические сети из нейроподобных элементов могут использоваться и для распознавания динамических образов, однако в этом случае для учета временной структуры информации прибегают к специальным приемам, например, заводят на дополнительные входы сети информацию с задержками [6]. Сети, имеющие в своем составе элементы задержки, называются динамическими.

Для автоматического формирования эталонов динамических образов в стационарную сеть должны быть введены обратные связи с элементами задержки с выходов сети на ее входы. Такие сети называются рекуррентными [7]. В наиболее общем виде это выражается в использовании трех одинаковых стационарных сетей, две из которых включены в прямой поток, а третья — в обратную связь [8]. Значение текущего входа поступает на сеть одновременно с текущим состоянием. Оба вектора проходят через стационарную прямую сеть для получения выходного вектора и следующего вектора состояния.

Такая структура позволяет эффективно автоматически формировать эталоны слов, содержащие контекстные знания (длиной в слово), но встречает определенные трудности в случае запоминания контекстной информации большей длины [9]. В случае длинных последовательностей сеть сваливается в локальные оптимумы на основе информации о более коротких контекстах: при наличии в длинной последовательности более коротких повторяющихся отрезков, система учится распознавать их, а не всю длинную последовательность.

Правила каждого уровня представления языковой информации имеют собственную природу, поэтому их эффективное представление и использование сталкивается с проблемами, характерными для интегрированного использования больших объемов существенно неоднородной информации [10]. Использование искусственных нейронных сетей одного типа — на основе нейроподобных элементов с временной суммацией сигналов [11], позволяет построить иерархические структуры, эффективно интегрирующие многоуровневое языковое представление [12].

Особенно важно при анализе больших массивов текстов на естественном языке учесть синонимические и иерархические отношения между понятиями. Для этого на основе терминов, содержащихся в словарях, автоматически строятся названия узких предметных рубрик, содержащих близкие по смыслу документы. Эти же термины являются узлами гипертекстовой структуры на множестве документов, которую необходимо сформировать при обработке больших массивов текстовых документов.

Автоматическое построение узких предметных рубрик с гипертекстовой структурой, отражающей актуальное состояние базы документов, в сочетании с использованием "жестких" рубрикаторов верхнего уровня позволяет существенно повысить точность и полноту поискового механизма по всему объему базы документов.

В основе механизма автоматической рубрикации лежит нейросетевая технология построения семантического портрета текста (множества текстов — рубрики) в виде ассоциативной сети, на основе которой, а также исходного текста, автоматически строится гипертекстовая структура, для которой упомянутая выше сеть является средством эффективной навигации по тексту (множеству текстов).

Семантическая сеть — это множество понятий текста, т. е. слов и словосочетаний, связанных между собой по смыслу. В семантическую сеть включены не все слова текста, а лишь наиболее значимые, несущие основную смысловую нагрузку. При этом в сеть не входят общеупотребительные слова, а также слова, очень редко встречающиеся в тексте (этот параметр — частоту встречаемости, можно настраивать по своему желанию). Поэтому, с одной стороны, семантическая сеть достаточно

точно представляет смысл текстов, а с другой стороны, позволяет отбросить несущественную информацию.

Каждое понятие семантической сети характеризуется числовой оценкой, так называемым смысловым весом. Связи между парами понятий, в свою очередь, также имеют характеристики — веса связей. Эти оценки позволят сравнить относительный вклад различных понятий и их связей в общий смысл текста, исследовать текстовый материал по пластам — смысловым срезам различной глубины.

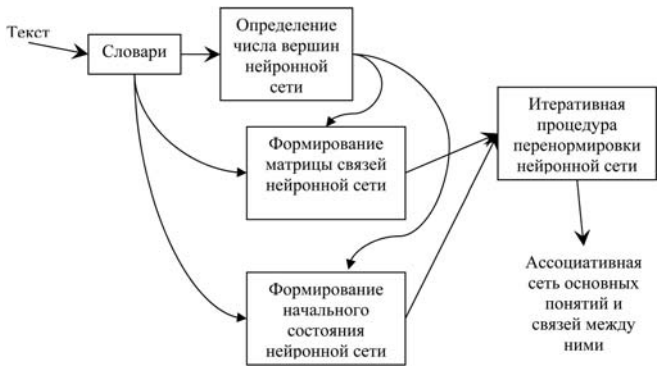
Наиболее подходящим типом нейронных сетей для создания семантических сетей являются сети прямого распространения, такие как сети Хопфилда [13]. В обработке текстового документа можно выделить в качестве специальных блоков обработки следующие блоки:

- "Словари";
- "Определение числа вершин";
- "Формирование матрицы связей";
- "Формирование начального состояния нейронной сети".

Схема алгоритма обработки текстовой информации на основе сети Хопфилда приведена на рисунке.

В блоке "Словари" происходит первоначальная фильтрация входного текста с использованием специальных словарей часто употребляемых слов и словосочетаний, которые не несут смысловой нагрузки, а выполняют связующую роль. Полученный в результате фильтрации текст поступает на вход блоков "Определение числа вершин", "Формирование матрицы связей" и "Формирование начального состояния".

Блок "Определение числа вершин" определяет число необходимых нейронов для обработки входной текстовой информации, а блоки "Формирование матрицы связей" и "Формирование начального состояния" создают первоначальное статистическое представление текстовой информации — сеть слов (понятий) с их связями.



Алгоритм обработки текстовой информации на основе сети Хопфилда

Для обработки текстовых документов используется нейронная сеть Хопфилда с обратными связями, которая представляет собой однослойную сеть нейронов, на вход которых подаются выходы всех остальных нейронов сети.

Обозначив вершинами сети понятия, извлеченные из текста на этапе частотной обработки, получаем, что степень связанности i-го понятия с j-м проецируется в вес связи от i-го нейрона. В то же время частота встречаемости понятия в тексте будет соответствовать изначальному уровню возбуждения соответствующего нейрона. С помощью итеративной процедуры перенормировки, аналогичной алгоритму сети Хопфилда, можно перейти от частотного портрета текстовой информации к семантической сети ключевых понятий анализируемого текста.

В результате такой перенормировки меняются первоначальные числовые характеристики слов. Слова, которые в сети связаны со словами с большим весом, в том числе через промежуточные слова, в результате такой процедуры повышают свой вес, вес остальных слов равномерно уменьшается. Полученная числовая характеристика слов — их смысловой вес — характеризует степень их важности в тексте.

В итерационном процессе для каждого нейрона определяется взвешенная сумма его входов по формуле

$$S_i = \sum_{j=1}^{N} w_{i,j} x_j^t, \qquad (1)$$

где $w_{i,j}$ — матрица весов связей нейронов, а x_j^t — значение выхода j-го нейрона на итерации t; N — число нейронов.

Затем вычисляется новое значение выхода нейрона как функция активации (2) от усредненной взвешенной суммы его входов (3):

$$\sigma(x) = \frac{A}{1 + \mathbf{e}^{B(1-x)}}; \qquad (2)$$

$$x_i^{t+1} = \sigma\left(\frac{S_i}{M}\right), \qquad (3)$$

где $M = \frac{1}{N}\sum_i S_i$, а $A = 100$ и $B \in (0...5,0)$ являются настраиваемыми параметрами.

Итерационный процесс останавливается по прошествии заданного числа итераций либо при достижении нейронной сетью слабо изменяющегося состояния, которое определяется на основании сравнения приращения энергии сети с заданным порогом в виде следующего неравенства:

$$|E^{t+1} - E^t| < \delta. \qquad (4)$$

Энергия нейронной сети на каждом шаге итераций рассчитывается по формуле

$$E^t = \frac{1}{N^2} \sum_{i=1}^{N} x_i \sum_{j=1}^{N} w_{i,j} x_j^t. \quad (5)$$

На этапе предварительной обработки текстового документа индексируются корневые основы слов и формируется матрица связей на основании частоты совместной встречаемости слов (понятий). Слова (понятия) считаются связанными, если они встречаются в одном предложении в анализируемом тексте. Вес матрицы связи вычисляется по формуле

$$w_{ij} = z_{ij}/z_j,. \quad (6)$$

где z_j — частота встречаемости слова в анализируемом тексте; z_{ij} — частота совместной встречаемости слов в анализируемом тексте.

Применение аппаратных ускорителей

В работе [14] представлены структурно-архитектурные решения нейрочипов и нейроплат на базе ПЛИС, реализующих указанную сеть, а также способы организации вычислений с использованием графических ускорителей, которые более подробно описаны в работе [15]. Эти разработки основаны на нейросетевой технологии TextAnalyst [16], которая в реализации на персональном компьютере имеет ограничения по объему обрабатываемой информации как вследствие ограничений объема оперативной памяти, так и вследствие ограничений по быстродействию. Эти ограничения возникают в процедуре итеративной перенормировки весовых коэффициентов понятий ассоциативной сети. Применение аппаратных ускорителей позволяет снять эти ограничения, в некоторых случаях значительно поднять потолок по объему обрабатываемой информации.

Тенденцией развития современных вычислительных платформ является увеличение степени параллелизма исполнения. Поэтому эффективность реализации любого алгоритма на параллельных архитектурах зависит от степени параллелизма алгоритма — способности разделения любой входной задачи на достаточно большое число подзадач, каждая из которых может быть решена независимо на отдельных вычислительных модулях, между которыми синхронизация и обмен данными затруднены (т. е. являются дорогостоящими операциями) или просто невозможны. К счастью, программные и аппаратные реализации обработки текстовой информации с применением семантических сетей на основе искусственных нейронных сетей имеют значительную вычислительную сложность и достаточно большую степень параллелизма, что позволяет эффективно использовать их на большинстве параллельных архитектур.

В условиях заметного отставания роста вычислительной мощности от роста пропускной способности памяти в современных вычислительных средствах многие общие рекомендации по оптимизации, действенные для предыдущих поколений вычислительной техники, зачастую бесполезны, если не сказать, что вредны для современных архитектур. Одним из основных методов оптимизации алгоритмов под современные архитектуры является экономия пропускной способности (т. е. сокращение числа обращений) памяти DRAM, часто действенным даже в тех случаях, когда это влечет за собой некоторое увеличение числа выполняемых математических операций. Кроме того, на графическом процессоре экономия пропускной способности памяти часто достигается за счет использования разделяемой памяти, имеющей гораздо большую пропускную способность и гораздо меньшие задержки доступа, чем DRAM, для обмена промежуточными результатами между потоками. Для аппаратной реализации на базе современных ПЛИС аналогичная экономия достигается за счет использования памяти, расположенной на плате с ПЛИС, доступ к которой происходит с существенно меньшими задержками, при этом результаты вычислений могут передаваться практически сразу после первоначальной задержки на прохождение сигнала по всей вычислительной цепочке. Поэтому при программировании, проектировании аппаратных реализаций можно выделить примерно следующую последовательность действий:

- загрузить данные из глобальной памяти в разделяемую память;
- синхронизовать потоки блока потоков, чтобы все они гарантированно записали результаты чтения глобальной памяти; инициализировать все параллельные вычислительные блоки в случае аппаратной реализации на ПЛИС;
- обработать данные в разделяемой памяти (при необходимости синхронизируя потоки блока потоков во избежание конфликтов доступа к общим участкам памяти);
- синхронизовать блок потоков повторно, чтобы завершить все вычисления над данными в разделяемой памяти, осуществить запись части результатов вычислений в глобальную память при аппаратной реализации на ПЛИС;
- записать результаты вычислений обратно в глобальную память; в случае аппаратной реализации на ПЛИС запись результатов совмещается с вычислениями и возможно дополнительной загрузкой данных из глобальной памяти.

Заключение

В заключении можно сказать, что вычислительный процесс на плате с ПЛИС управляется с помощью контроллера, интегрированного в ПЛИС. Управление процессом работы платы осуществляется посредством передачи управляющего слова в соответствующий регистр от управляющей программы. Таким образом, аппаратная реализация на базе ПЛИС позволяет обрабатывать в потоковом режиме большие объемы текстовой информации, скорость обработки в этом случае ограничивается только полосой пропускания интерфейса между платой с ПЛИС и оперативной памятью персонального компьютера или сервера. Однако программная реализация на современных графических процессорах позволяет практически любой персональный компьютер (в том числе и мобильный) превратить в мощный инструмент для анализа текстовой информации без затрат на дополнительное дорогостоящее оборудование; производительности современных графических процессоров зачастую более чем достаточно для большинства задач обработки текстовой информации.

Список литературы

1. **Киров Е. Ф.** Теоретические проблемы моделирования языка. Казань: Изд-во Казанского университета, 1989.
2. **Рахилина Е. В.** Когнитивный анализ предметных имен: семантика и сочетаемость. М.: Русские словари, 2000.
3. **Кузнецов И. П.** Семантические представления. М.: Наука, 1986.
4. **Осипов Г. С.** Приобретение знаний интеллектуальными системами: Основы теории и технологии. М.: Наука. Физматлит, 1997.
5. **Schwenk H.** Efficient Training of Large Neural Networks for Language Modeling // Proc. International Joint Conference on Neural Networks, 2004. P. 3059—3064.
6. **Lippman R. P., Gold B.** Neural-Net Classifiers Useful for Speech Recognition // IEEE 1-st International Conference on Neural Networks. Vol. 4. San Diego, Calif, 1984.
7. **Tsoi Ah Chung, Back A. D.** Locally recurrent globally feedforward networks: A critical review of architectures // IEEE Trans. on Neural Networks. 1994. Vol. 5, N 2. P. 229—239.
8. **Robinson A. J., Fallside F.** Static and dynamic error propagation networks with application to speech coding // Neural Information Processing System / Ed. D. Z. Anderson. N.-Y.: American Institute of Physics. 1988. P. 632—641.
9. **Bianchini M., Gori M.** On the Problem of Local Minima in Recurrent Neural Networks // IEEE Transactions on neural networks. 1994. Vol. 5, N 2. P. 167—176.
10. **Методы** автоматического распознавания речи / Под. ред. У. Ли. Т. 2. М.: Мир, 1983.
11. **Харламов А. А.** Нейроподобные элементы с временной суммацией входного сигнала и блоки ассоциативной памяти на основе этих элементов // Вопросы кибернетики. Устройства и системы / Под ред. Н. Н. Евтихиева. М.: МИРЭА, 1983. С. 57—68.
12. **Харламов А. А.** Нейросетевая технология представления и обработки информации (естественное представление знаний). М.: Радиотехника, 2006. 89 с.
13. **Hopfield J. J.** Neural networks and physical systems with emergent collective computational abilities // Proc. Natl. Acad. Sci. 1982. 79. P. 2554—2558.
14. **Воронков И. М., Лобов С. А., Кречетов И. В.** Применение аппаратных ускорителей нейросетевых вычислений для задач текстовой обработки // Материалы 4-й Международной научной молодежной школы "Нейроинформатика и системы ассоциативной памяти" в рамках 5-й Международной молодежной научно-технической конференции "Высокопроизводительные вычислительные системы". ВПВС-2008, Таганрог, сентябрь 2008. С. 131—135.
15. **Воронков И. М., Лобов С. А.** Использование графических ускорителей для задач текстовой обработки с применением нейросетевых технологий // Материалы 51-й научной конференции МФТИ — Всероссийская молодежная научная конференция с международным участием "Современные проблемы фундаментальных и прикладных наук". Долгопрудный. Ноябрь 2008.
16. **Харламов А. А., Ермаков А. Е., Кузнецов Д. М.** TextAnalyst — комплексный нейросетевой анализатор текстовой информации // Вестник МГТУ им. Н. Э. Баумана. 1998. № 1. С. 32—36.

ИНФОРМАЦИЯ

CONTENTS

Адрес редакции:

107076, Москва, Стромынский пер., 4

Телефон редакции журнала **(499) 269-5510**

E-mail: it@novtex.ru

Дизайнер *Т.Н. Погорелова*. Технический редактор *Е. В. Конова*.

Корректор *Т. В. Пчелкина*.

Сдано в набор 11.01.2011. Подписано в печать 18.02.2011. Формат 60×88 1/8. Бумага офсетная. Печать офсетная. Усл. печ. л. 9,8. Уч.-изд. л. 10,85. Заказ 133. Цена договорная.

Журнал зарегистрирован в Министерстве Российской Федерации по делам печати, телерадиовещания и средств массовых коммуникаций.

Свидетельство о регистрации ПИ № 77-15565 от 02 июня 2003 г.

Отпечатано в ООО "Подольская Периодика"

142110, Московская обл., г. Подольск, ул. Кирова, 15