

# ИНФОРМАТИКА, ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА И УПРАВЛЕНИЕ

УДК 519.2:004.421.5:004.7

**В. Н. ЗАДОРЖНЫЙ**Омский государственный  
технический университет,  
г. Омск

## РАСТУЩИЕ СЕТИ С ПОТЕРЯМИ УЗЛОВ

Разрабатываются методы анализа растущих сетей со случайными потерями узлов. На основе теории случайных графов предпочтительного связывания выводятся уравнения для распределения степеней вершин в переходном и стационарном режимах. Предлагаются численные методы решения полученных уравнений. Приводятся примеры расчетов. Оценивается устойчивость распределения степеней вершин к потерям.

**Ключевые слова:** теория сетей, случайные графы предпочтительного связывания, переходные и стационарные распределения степеней связности.

**1. Введение.** В науке о сетях информационные, транспортные, электрические, социальные и многие другие большие сети рассматриваются как явление природы, исследование которого приводит к познанию общих законов развития сетей и к разработке стратегий управления сетевыми процессами. В последнее время от разумного и своевременного решения этих важных вопросов все более зависит не только прогресс, но и само существование человеческой цивилизации.

Математической основой теории сетей является теория случайных графов, которая со времени создания классической теории простейших случайных графов [1] обогатилась развитым математическим аппаратом исследования множества более адекватных и сложных графовых моделей. Одними из наиболее перспективных моделей сетей являются графы предпочтительного связывания. Первая и наиболее

популярная до настоящего времени версия графов предпочтительного связывания — это графы Барабаши — Альберт (графы БА), предложенные в работах [2–4]. Появление графов БА привело к взрывному росту числа публикаций по теории сетей (см., например, [5–8]), в которых предлагаются и исследуются случайные графы, представляющие собой обобщения модели БА. В работах [9–11] предлагается и исследуется класс графов с нелинейным правилом предпочтительного связывания (НППС), а также разрабатывается математический аппарат для точного расчета стационарных характеристик графов с НППС.

Граф с НППС выращивается из небольшого графа без петель (затравки) путем добавления к нему в моменты  $t = t_1, t_2, \dots$  очередного *приращения графа* — новой вершины со случайным числом  $x$  исходящих из нее дуг. Концы дуг приращения связыва-

ются со случайно выбираемыми вершинами графа. Вероятность  $p_i$  того, что дуга выберет для связи вершину  $i$ , пропорциональна весу  $f$  этой вершины, определяемому через ее степень связности  $k_i$ :

$$p_i = \frac{f(k_i)}{\sum_j f(k_j)}. \quad (1)$$

При неограниченном добавлении приращений формируется бесконечный граф. Таким образом, случайный граф с НППС задается двумя параметрами — распределением  $\{r_k\}$  вероятностей случайной величины  $x$  — числа дуг у приращения — и весовой функцией  $f(k) \geq 0$ . Будем считать, что  $f(k) > 0$  тогда и только тогда, когда  $g \leq k \leq M$  (где  $g \geq 0$ ,  $M \leq \infty$ ) и обозначать  $f(k)$  в виде  $f_k$ .

**2. Постановка задачи.** В теории случайных графов с НППС [9–11] решаются задачи анализа (расчета характеристик) графов, заданных параметрами  $\{r_k\}$  и  $f(k)$ , и задачи синтеза (калибровки) графов, т.е. задачи определения таких  $\{r_k\}$  и  $f(k)$ , которые обеспечивают наличие у выращиваемых графов заданных характеристик.

Решения задач анализа охватывают в качестве частных случаев результаты, полученные в других работах. Если число  $x$  дуг приращения считать постоянным ( $x = m = \text{const}$ ) и положить  $f(k) = k$ , то граф с НППС превращается в граф БА, а соответствующие найденные для графов с НППС соотношения — в соотношения для графов БА (известные ранее или новые). Если положить  $f(k) = k^g$ , то при  $x = m = \text{const}$  из соотношений для графов с НППС получается ряд результатов, найденных в [6] для графов со степенной весовой функцией.

Вместе с тем целый класс решенных для графов с НППС задач — задач калибровки — не имеет аналогов в частных теориях графов предпочтительного связывания. Это обеспечивает графам с НППС серьезные преимущества при построении моделей реальных больших сетей.

Найденные в теории графов с НППС аналитические выражения и методы точного численного расчета таких характеристик, как распределение степеней связности (РСС) вершин или дуг графа обеспечивают этой теории принципиальные преимущества по скорости и точности вычислений в сравнении с методами имитационного моделирования (ИМ).

Преимущества теории случайных графов с НППС становятся еще более значимыми в результате вывода уравнений динамики, описывающих переходные процессы развития графов при  $t < \infty$ , и в результате разработки численных методов их решения [12]. Численные методы расчета особенно важны при вычислении хвостов таких распределений, которые, оказываясь степенными (фрактальными), на порядки усложняют применение ИМ [13].

В статье [14] получен результат, обеспечивающий теории графов с НППС еще одно качественное преимущество: выведены уравнения динамики для графов с НППС, случайно теряющих дуги в ходе выращивания графов. Эти результаты позволяют исследовать реальные сети с потерями связей.

В настоящей статье ставится и решается задача анализа графов с НППС со случайными потерями вершин. На каждом шаге выращивания таких графов наряду с присоединением очередного приращения может быть (с вероятностью  $p$ ) потеряна одна случайно выбранная вершина (с.в.в.). При потере вершины удаляются и все инцидентные ей дуги графа. Требуется вывести уравнения динамики, позволя-

ющие по заданным параметрам  $p$ ,  $\{r_k\}$  и  $f$  рассчитывать переходные процессы в растущих графах. На основе уравнений динамики требуется получить уравнения для стационарных РСС вершин и разработать численные методы их решения. Необходимо также на примерах расчета графов с потерями вершин показать возможность получения содержательных выводов о моделируемых реальных сетях с потерями узлов.

**3. Вывод уравнений динамики.** В процессе выращивания графа будем рассматривать его РСС вершин как вектор-функцию  $\mathbf{q}(t) = \{q_k(t)\}$  скалярного аргумента  $t$ , где  $q_k$  — вероятность того, что с.в.в. графа имеет степень связности  $k$ . Например, кольцо на  $N > 2$  вершинах имеет РСС  $\mathbf{q} = \{q_0, q_1, q_2\} = \{0, 0, 1\}$ , так как в кольце степень с.в.в. с вероятностью 1 равна двум. В общем случае длина (число координат) вектора  $\mathbf{q}$  равна  $K + 1$  (где  $K = \max\{k\}$ ) и меняется во времени  $t$ .

Поскольку на каждом шаге выращивания графа к нему добавляется одна вершина и теряется в среднем  $p$  вершин, то, пренебрегая малыми размерами заправки, среднее число  $N(t)$  вершин в графе можно считать равным  $(1 - p)t$ .

Для получения уравнений динамики графов с потерями естественно опереться на выведенные в [12] уравнения динамики графов без потерь:

$$N_k(t + 1) = N_k(t) + r_k + mP_{k-1}(t) - mP_k(t), \quad k \geq g, \quad (2)$$

где  $N_k$  — среднее число вершин графа, имеющих степень  $k$ ;

$r_k = P(x = k)$  — вероятность того, что приращение графа имеет  $k$  дуг;

$m = \sum kr_k$  — средняя степень приращения графа;

$P_k(t) = \frac{q_k(t)f_k}{\bar{f}(t)}$  — вероятность того, что по правилу (1)

дуга приращения выберет вершину графа, имеющую степень  $k$ . В этом последнем выражении  $\bar{f}(t) =$

$= \sum_{k=g}^M f_k q_k(t)$  — средний вес вершин графа на шаге  $t$ .

С учетом того, что  $N_k(t) = q_k(t)N(t)$  и что в графе без потерь  $N(t) = t$ , уравнения (2) приводятся в [12] к виду

$$q_k(t + 1) = \frac{tq_k(t) + r_k + mP_{k-1}(t) - mP_k(t)}{t + 1},$$

или, с учетом приведенного выше для  $P_k(t)$  выражения, к виду

$$q_k(t + 1) = \frac{tq_k(t) + r_k + \frac{m}{\bar{f}(t)} [q_{k-1}(t)f_{k-1} - q_k(t)f_k]}{t + 1}, \quad k \geq g. \quad (3)$$

Уравнения (3) позволяют рассчитать РСС  $\mathbf{q}(t)$  графа без потерь при заданном начальном РСС  $\mathbf{q}(t_0)$ , которое определяется начальным графом-заправкой, содержащим  $N_0 = t_0$  вершин. Эти уравнения, как и лежащее в их основе соотношение (2), являются асимптотически точными. С ростом  $t$  рассчитываемые по этим уравнениям РСС  $\mathbf{q}(t)$  быстро сходятся к точным РСС.

Теперь для вывода уравнений динамики, описывающих эволюцию графов с потерями, добавим к правой части соотношения (2) слагаемые, учитывающие влияние на величину  $N_k(t + 1)$  возможной

t	N(t)	<f>(t)	kcp(t)	q0(t)	q1(t)	и т.д.
3	3	2,0000	2,000	0	0	1
4	3,2	2,5000	2,500	0	0,125	0,25
5	4	2,7500	2,750	0,00625	0,0875	0,43125
6	4,8	2,8958	2,896	0,008594	0,088305	0,408878
7	5,6	2,9896	2,990	0,010213	0,087697	0,418237
8	6,4	3,0540	3,054	0,011358	0,088227	0,421808

Рис. 1. Таблица на Excel для расчета РСС  $q(t)=\{q_k(t)\}$  растущего графа с потерями вершин

(с вероятностью  $p$ ) потери с.в.в. графа. Обозначая через  $A_k$  подмножество (слой) вершин, имеющих степень  $k$ , и через  $B(l, k)$  — подмножество дуг (туннель), исходящих из вершин слоя  $A_l$  и заходящих в вершины слоя  $A_k$ , можно отметить следующее.

1. Поскольку с.в.в. с равной вероятностью может оказаться любой из  $N$  вершин графа, то в слое  $A_k$  она выбирается с вероятностью  $q_k$ . И первое корректирующее правую часть соотношения (2) слагаемое, равное  $[-pq_k(t)]$ , учитывает среднее число вершин, теряемых в слое  $A_k$  непосредственно за счет выбора удаляемой вершины в этом слое.

2. Если с.в.в. удаляется из некоторого (любого) слоя  $A_l$ , то вместе с нею удаляются все инцидентные ей дуги. В среднем одну вершину слоя  $A_l$  с вершинами

слоя  $A_k$  связывает  $\frac{|B(l, k)| + |B(k, l)|}{|A_l|}$  дуг. Удаление

этих дуг уменьшает степени соответствующих вершин слоя  $A_k$ . Поэтому такие вершины из слоя  $A_k$  уходят. Представим их среднее число в виде

$$\frac{|B(l, k)| + |B(k, l)|}{|A_l|} = \frac{1}{N} \frac{|B(l, k)| + |B(k, l)|}{|A_l|/N} = \frac{1}{N} \frac{|B(l, k)| + |B(k, l)|}{q_l}$$

и, усредняя по выборам с.в.в. во всех слоях, находим безусловное среднее число вершин, теряемых в слое  $A_k$  за счет снижения их степени:

$$\sum_l q_l \frac{1}{N} \frac{|B(l, k)| + |B(k, l)|}{q_l} = \frac{1}{N} \sum_l |B(l, k)| + |B(k, l)|$$

Последняя сумма представляет здесь общее (среднее) число дуг, инцидентных вершинам слоя  $A_k$ . Заменяя ее эквивалентным более простым выражением  $kN_k$ , находим, что среднее число вершин, уходящих из слоя  $A_k$  из-за снижения их степени, составляет  $kN_k/N = kq_k$ .

Таким образом, вторая поправка к правой части соотношения (2) равна  $[-pkq_k(t)]$ , где множитель  $p$  учитывает вероятность потери с.в.в.

3. Рассуждая аналогично, нетрудно определить, что из-за снижения степеней в слое  $A_{k+1}$  из него в слой  $A_k$  на шаге  $t$  приходит в среднем  $p(k+1)q_{k+1}(t)$  вершин.

Учитывая все три найденные поправки, получаем из (2) следующее уравнение эволюции графа с потерями вершин:

$$N_k(t+1) = N_k(t) + r_k + mP_{k-1}(t) - mP_k(t) - pq_k(t) + p(k+1)q_{k+1}(t) - pkq_k(t), \quad k \geq g, \quad (4)$$

где, принимая во внимание неизбежное появление вершин со степенью  $k=0$ , следует положить  $g=0$ .

Для удобства расчетов РСС  $q(t)=\{q_k(t)\}$  остается привести уравнение (4) к виду, аналогичному уравнению (3), описывающему граф без потерь. Для этого перепишем (4) с заменами  $N_k(t+1) = N(t+1)q_k(t+1) = (1-p)(t+1)q_k(t+1)$  и  $N_k(t) = N(t)q_k(t) = (1-p)(t)q_k(t)$  следующим образом:

$$(1-p)(t+1)q_k(t+1) = (1-p)(t)q_k(t) + r_k + mP_{k-1}(t) - mP_k(t) - pq_k(t) + p(k+1)q_{k+1}(t) - pkq_k(t), \quad k \geq 0. \quad (5)$$

Применяя обозначение  $q = (1-p)$  и подставляя в (5) выражения для  $P_{k-1}(t)$  и  $P_k(t)$ , получаем формулу для расчета РСС вершин на шаге  $t+1$  через РСС на шаге  $t$ :

$$q_k(t+1) = \frac{1}{t+1} \left[ tq_k(t) + \frac{r_k}{q} + m \frac{q_{k-1}(t)f_{k-1} - q_k(t)f_k}{q\bar{f}(t)} - p \frac{q_k(t) - (k+1)q_{k+1}(t) + kq_k(t)}{q} \right], \quad k \geq 0. \quad (6)$$

Естественно, что при вероятности потерь  $p=0$  (а тогда  $q=1$ ) эта формула превращается в формулу (3) эволюции графа без потерь.

**4. Расчет переходных процессов.** Формула (6) позволяет на одном листе Excel легко рассчитывать РСС  $q(t)=\{q_k(t)\}$  для любого шага  $t > t_0$ , начиная с РСС, заданного для  $t=t_0$ . Не зависящие от  $t$  последовательности  $\{k\}$ ,  $\{r_k\}$  и  $\{f_k\}$  фиксируются в строках верхней «шапки» листа Excel аналогично тому, как это делается при расчете графов без потерь вершин [12]. Скалярные показатели, зависящие от времени, такие как средняя степень вершин  $\bar{k}(t) =$

$= \sum_{k \geq 0} kq_k(t)$  и их средний вес  $\bar{f}(t) = \sum_{k \geq 0} f_k q_k(t)$  можно вычислять в столбцах левой части листа.

На рис. 1 показан фрагмент листа с расчетом РСС  $q(t)=\{q_k(t)\}$  растущего графа, определяемого параметрами  $p=0,2$ ,  $\{r_1; r_2\} = \{0; 1\}$ ,  $f_k = k$  при  $1 \leq k \leq 200$  (при прочих  $k$  вес  $f_k = 0$ ). Этот граф отличается от графа БА с параметром  $m=2$  лишь тем, что максимальная степень его вершин ограничена и на каждом шаге наращивания графа может с вероятностью  $p=0,2$  потеряться с.в.в. графа.

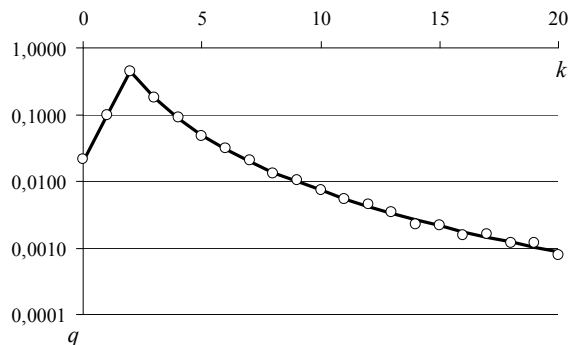


Рис. 2. Полученное в ИМ и расчетное РСС растущего графа с потерями вершин ( $t = 500$ )

Основная часть листа, в которой выполняется расчет вероятностей  $q_k(t)$ , начинается ячейкой **G7**, которая на рис. 1 выделена для того, чтобы записанная в ней в виде выражения  $=1/A8*(A7*G7+G3/B$3+D$2/B$3*(F7*F$4-G7*G$4)/C7-B$2/B$3*G7+B$2*N$2/B$3*N7-B$2*G$2/B$3*G7)$  формула (6) была видна в строке набора формул (в верхней части листа). Ячейка **G7** копируется в лежащие правее нее ячейки, а затем вся полученная строка ячеек копируется в нижележащие строки. Начальное РСС задается в ячейках **G6**, **H6** и т.д. В данном случае начальное РСС соответствует графу-кольцу на трех вершинах.

Колонка  $N(t)$  содержит среднее число вершин в графе ( $N \sim qt$ ) и в расчетах, согласно формуле (6), не участвует.

На рис. 2 построен график РСС  $q(t) = \{q_k(t)\}$  при  $t = 500$  (РСС рассчитано на листе, показанном на рис. 1). График результатов расчета РСС (сплошная линия) сопоставляется с графиком результатов ИМ (маркеры).

Согласованность результатов расчета с результатами ИМ подтверждает правильность асимптотически точной формулы (6) и ее пригодность для практических расчетов.

Заметим, что в ряде случаев вероятности «второго порядка малости», которыми мы пренебрегли при выводе формулы (6), например, вероятности того, что с одной вершиной свяжется несколько дуг приращеня, могут сказаться на точности расчета ранних стадий переходного процесса. Например, при большом среднем числе дуг приращеня, большой вероятности потерь и малом размере затравки (малом начальном времени  $t_0$ ) рассчитываемые малые вероятности  $q_k$  могут принимать отрицательные значе-

ния. Эта ситуация легко корректируется выбором подходящего достаточно большого значения  $t_0$ . Обычно при  $p < 0,9$  достаточно брать  $t_0$  в пределах от 10 до 60 единиц времени.

**5. Уравнения стационарного распределения степеней.** Обозначим через  $Q_k$  финальные (стационарные) вероятности степеней  $k$ , т.е.  $Q_k = \lim_{t \rightarrow \infty} q_k(t)$ , и решим задачу вычисления стационарного РСС  $Q = \{Q_k\}$ .

Умножив равенство (6) на  $(t + 1)$ , перепишем его следующим образом:

$$(t + 1)q_k(t + 1) - tq_k(t) = \frac{r_k}{q} + m \frac{q_{k-1}(t)f_{k-1} - q_k(t)f_k}{q\bar{f}(t)} - p \frac{q_k(t) - (k + 1)q_{k+1}(t) + kq_k(t)}{q}, \quad k \geq 0. \quad (7)$$

Здесь левая часть  $(t + 1)q_k(t + 1) - tq_k(t) = tq_k(t + 1) - tq_k(t) + q_k(t + 1)$  содержит дифференциал  $tq_k(t + 1) - tq_k(t)$ , который сходится с ростом  $t$  к нулю, если  $q_k(t)$  сходится к пределу  $Q_k$  не медленнее, чем со степенной скоростью  $ct^{-a}$  (где  $a > 1$ ). На практике, при решении конкретных задач, это условие легко проверяется и подтверждается на этапе расчета переходных процессов. Поэтому при  $t \rightarrow \infty$  положим  $tq_k(t + 1) - tq_k(t) \rightarrow 0$  и оставим в левой части равенства (7) лишь слагаемое  $q_k(t + 1)$ . Учитывая теперь, что при  $t \rightarrow \infty$  величина  $q_k(t + 1)$  сходится к  $Q_k$ , величины  $q_{k-1}(t)$ ,  $q_k(t)$ ,  $q_{k+1}(t)$ , сходятся, соответственно, к  $Q_{k-1}$ ,  $Q_k$ ,  $Q_{k+1}$ , а средний вес вершин  $\bar{f}(t)$  сходится к стационарному среднему весу  $\langle f \rangle$ , из равенства (7) получаем:

$$Q_k = \frac{r_k}{q} + m \frac{Q_{k-1}f_{k-1} - Q_k f_k}{q\langle f \rangle} - p \frac{Q_k - (k + 1)Q_{k+1} + kQ_k}{q}, \quad k \geq 0.$$

Выражая отсюда  $Q_k$  и добавляя формулу для вычисления  $\langle f \rangle$ , получаем систему уравнений

$$\begin{cases} Q_k = \frac{r_k + \frac{mf_{k-1}}{\langle f \rangle} Q_{k-1} + p(k + 1)Q_{k+1}}{1 + \frac{mf_k}{\langle f \rangle} + pk}, & k \geq 0, \\ \langle f \rangle = \sum_{k=0}^M f_k Q_k, \end{cases} \quad (8)$$

решая которую численными методами, можно рассчитывать искомые стационарные вероятности  $Q_k$ .

O7		Текущее		Новое	
K	L	M	N	O	P
1	fk=k				
2			p = 0,2		<f> текущее
3			m = 2,0000		<f> новое
4					3,3290380
5	<k> =	3,333334			sum
6	m =	2,0000			1,000000003
7	k	rk	fk	Новое Qk	Текущее
8	-1	0	0	0	6,4957E-06
9	0	0	0	0,019937	0,019937
10	1	0	1	0,099685	0,099685
11	2	1	2	0,448776	0,448776
12	3	0	3	0,179372	0,179372

Рис. 3. Фрагмент листа, предназначенный для расчета стационарного РСС  $\{Q_k\}$

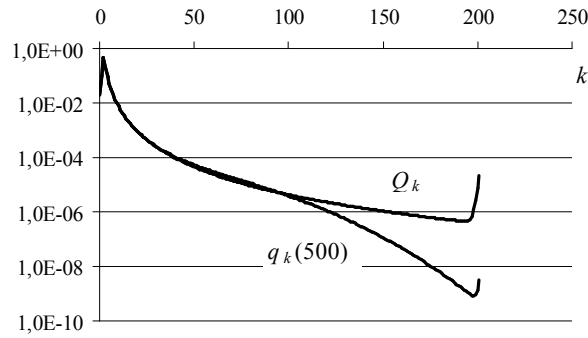


Рис. 4. Графики переходного при  $t = 500$  и финального РСС графа с потерями вершин

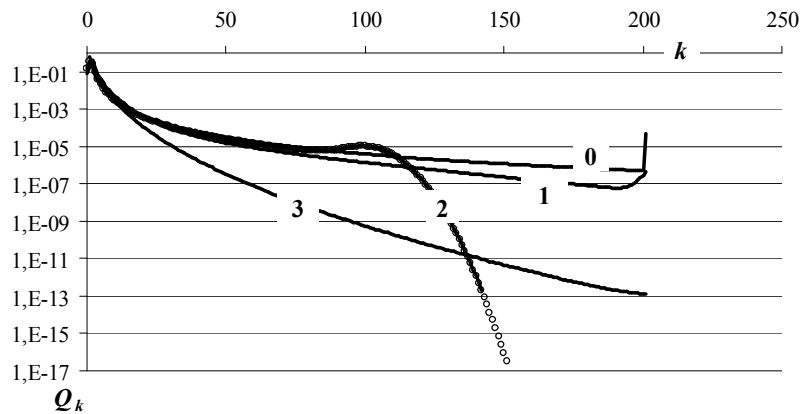


Рис. 5. Состояния большой сети при различных сценариях воздействия на нее

**6. Расчет стационарного распределения степеней.** На рис. 3 показан фрагмент листа Excel с решением системы уравнений (8) для графа, рассмотренного в предыдущем разделе статьи. Решение выполняется методом простых итераций. В колонке **О** в формате «значение» записано текущее приближение распределения  $\{Q_k\}$ . В качестве начального приближения  $\{Q_k\}$  можно взять, например, РСС  $\{q_k(t)\}$  при  $t = 500$ , рассчитываемое методами предыдущего раздела при некоторой произвольной загрузке графа. В ячейке **РЗ** в формате «значение» находится текущее приближение среднего веса  $\langle f \rangle$  (его начальное приближение также можно взять из расчета переходного процесса развития графа). Через текущие приближения РСС и  $\langle f \rangle$  по формулам (8) в колонке **Н** и ячейке **РЗ** рассчитываются новые приближения для  $\{Q_k\}$  и, соответственно, для  $\langle f \rangle$ . Далее эти новые приближения нужно скопировать и вставить на место текущих приближений (специальной вставкой, как значения). Для такого копирования и вставки удобно создать простой макрос, с помощью которого копирование и вставка выполняются одновременным нажатием выбранной комбинации клавиш. В ячейке **Р7** вычисляется показатель погрешности — корень квадратный из суммы квадратов разностей между текущими приближениями вероятностей  $Q_k$  и их новыми значениями. На рис. 3 этот показатель умножен на 1000. Комбинацию клавиш, вызывающую макрос, можно просто придавить, не отпуская, тогда все пересчеты будут происходить с высокой скоростью, за секунды снижая показатель погрешности до нужных малых значений.

На рис. 4 сравниваются графики стационарного РСС  $\{Q_k\}$  графа, рассмотренного в предыдущем разделе, и переходного РСС, достигаемого на шаге  $t = 500$ .

Из рисунка видно, что в ходе эволюции графа вероятности  $q_k$  невысоких степеней  $k$  связности мало отличаются от стационарных вероятностей  $Q_k$  уже при относительно небольших значениях  $t$  (в данном случае при  $t = 500$ ). Но в такое же время  $t$  вероятности больших степеней  $k$  могут отличаться от соответствующих стационарных вероятностей на несколько порядков.

**7. Заключение.** Полученные в статье уравнения (6) и (8) позволяют эффективно рассчитывать переходные и стационарные процессы эволюции графов с нелинейным правилом предпочтительного связывания, растущих в условиях случайной потери вершин.

Это обеспечивает возможность в рамках науки о сетях ставить и решать задачи анализа и синтеза широкого класса реальных стохастических сетей и разрабатывать эффективные стратегии управления сетевыми процессами. К таким задачам относятся, в частности, задачи динамической идентификации растущих сетей (включая информационные, транспортные, электрические, социальные, террористические и другие сети) и задачи выбора и оптимизации управленческих решений, актуальные для широкого класса таких сетей.

В качестве примера на рис. 5 сравниваются финальные РСС  $\{Q_k\}$  растущей сети, получающиеся при различных приблизительно эквивалентных по стоимости сценариях воздействия на сеть.

В данном примере сеть моделируется графом, описанным в разделе 4 статьи.

Цифрой 0 на рис. 5 помечено финальное РСС, достигаемое при базовом сценарии развития, когда связи и узлы в сети не теряются. Из графика можно видеть, что при размерах сети в несколько миллионов узлов она будет содержать сотни узлов, име-

ющих максимальную степень связности 200 (см. «хвостик» в правом конце графика).

График 1 соответствует сценарию роста сети при потере в среднем половины узлов: здесь вероятность  $p$  взята равной 0,5. Предположим, что целью воздействия на сеть является недопущение ее заметного роста. Тогда сценарий «устранения» в среднем каждого второго узла в ходе роста сети следует признать наилучшим. Ведь, несмотря на такое воздействие, сеть приходит в итоге практически в то же состояние, что и в базовом сценарии.

График 2 — это две практически совпадающие линии, одна представляет результат «финального просеивания случайно выбранной половины узлов», другая — результат «финального просеивания случайно выбранной половины связей», каковое производится однократно в отношении уже сформировавшейся сети. Этот сценарий лучше первого, но такая «война» с узлами/связями уже сформированной сети заканчивается не вполне удовлетворительно, так как в сети еще остается большое число узлов-лидеров с высокой степенью связности, близкой к 100.

Наилучшим оказывается сценарий, результат которого представлен графиком 3. Этот результат достигается при потере случайно выбранных связей. На каждом шаге выращивания сети с вероятностью 0,5 теряется одна случайно выбранная связь. В результате при размерах сети в несколько миллионов узлов в ней уже трудно найти лидера с хотя бы 50 связями. Для расчета этого сценария использовались уравнения, выведенные в статье [14].

Здесь следует пояснить, что в базовом сценарии сеть растет при  $m = 1$ , а в сценариях с ликвидацией половины узлов/связей — при  $m = 2$ , т.е. сеть имеет «ресурс сопротивления», обеспечивающий сохранение «средней скорости» ее роста. Несмотря на одинаковый «ресурс сопротивления», качество сети при разных сценариях воздействия на нее оказывается принципиально различным.

Рассмотренный пример показывает, что найденные новые теоретические результаты позволяют получать важные содержательные выводы о свойствах моделируемых растущих сетей даже при отсутствии информации о деталях их структуры.

#### Библиографический список

1. Erdos P., Renyi A. On the evolution of random graphs // Publications of the Mathematical Institute of the Hungarian Academy of Sciences 5. 1960. P. 17–61.

2. Barabasi A. L., Albert R. Emergence of scaling in random networks // Science. 1999. 286. P. 509–512.

3. Barabasi A. L., Albert R. Statistical mechanics of complex networks // Rev. Mod. Phys. 2002. 74. P. 47–97.

4. Barabasi A. L. Scale-free networks: A decade and beyond // Science. 2009. 325. P. 412–413.

5. Amaral L. A. N., Scala A., Barthélemy M., Stanley H. E. Classes of small-world networks // Proc. Natl. Acad. Sci. U.S.A. 2000. 97. P. 11149.

6. Krapivsky P. L., Redner S. Organization of growing random networks // Phys. Rev. E 63. 2001. 066123.

7. Newman M. The structure of scientific collaboration networks // Proc. Natl. Acad. Sci. USA. 2001. 98. P. 404–409.

8. Clauset, A., Shalizi, C.R., Newman, M. Power-law distributions in empirical data, Rev. Mod. Phys. 2009. 51. P. 661–703.

9. Задорожный В. Н. Случайные графы с нелинейным правилом предпочтительного связывания // Проблемы управления. 2011. № 6. С. 2–11.

10. Zadorozhnyi V., Yudin E. Growing Network: Nonlinear Extension of the Barabasi-Albert Model // Communications in Computer and Information Science. 2014. Vol. 487. P. 432–439.

11. Zadorozhnyi V. N., Yudin E. B. Growing network: models following nonlinear preferential attachment rule, Physica A: Statistical Mechanics and its Applications. 2015. Vol. 428. P. 111–132. DOI: 10.1016/j.physa.2015.01.052.

12. Задорожный В. Н., Бадрызов В. А. Переходные процессы в растущих сетях с нелинейным правилом предпочтительного связывания // Омский научный вестник. Сер. Приборы, машины и технологии. 2016. № 1 (145). С. 95–99.

13. Задорожный В. Н., Кутузов О. И. Проблемы генерации случайных величин с фрактальными распределениями // Омский научный вестник. Сер. Приборы, машины и технологии. 2012. № 3 (113). С. 20–24.

14. Задорожный В. Н., Юдин Е. Б. Уравнения динамики степеней узлов в растущих сетях с потерями связей // Динамика систем, механизмов и машин. 2016. № 1. Т. 3. С. 340–346.

**ЗАДОРЖНЫЙ Владимир Николаевич**, доктор технических наук, доцент (Россия), профессор кафедры автоматизированных систем обработки информации и управления.

Адрес для переписки: zwn2015@yandex.ru

Статья поступила в редакцию 11.01.2017 г.

© В. Н. Задорожный

## Книжная полка

**Батенькина, О. В. Технологии визуализации результатов профессиональной деятельности в области информационных систем : моногр. / О. В. Батенькина, О. Н. Ткаченко. – М. : Магистр, 2016. – 113 с.**

Монография посвящена исследованию методов визуализации на всех этапах профессиональной деятельности специалистов в области информационных систем и технологий. Рассмотрены сферы и направления использования визуализации в области информационных технологий, описаны методы визуализации, которые классифицированы по области применения, уровням визуализации, методам представления информации; представлены инструментальные средства визуализации информации. Исследованы основные принципы преобразования данных, визуального отображения и представления, методы проектирования визуального образа для систематизации и отображения их в удобной форме с учетом психологических особенностей восприятия для привлечения внимания к наиболее важным единицам информации. Авторами представлены рекомендации по использованию методов визуализации в процессе проектирования, разработки и внедрения информационных систем и технологий, которые могут быть полезны студентам высших учебных заведений, обучающимся по направлению подготовки 09.04.02 «Информационные системы и технологии».

## КАЛИБРОВКА СЛУЧАЙНЫХ ГРАФОВ ПРЕДПОЧТИТЕЛЬНОГО СВЯЗЫВАНИЯ ПО РАСПРЕДЕЛЕНИЯМ СТЕПЕНЕЙ ВЕРШИН И РЕБЕР

Разрабатываются методы калибровки случайных графов предпочтительного связывания при моделировании больших растущих сетей. Графы калибруются по распределениям степеней связности вершин и ребер. Математической основой методов является теория случайных графов с нелинейным правилом предпочтительного связывания. Полученные результаты расширяют возможности адекватного моделирования и исследования реальных растущих сетей (социальных, телекоммуникационных, транспортных, террористических, финансовых и т.д.).

**Ключевые слова:** случайные графы, распределения степеней связности вершин и ребер (дуг), нелинейное правило предпочтительного связывания.

*Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 16-31-60023 мол\_а\_дк.*

**1. Введение.** В науке о сетях (Network Science) установлено, что свойства широкого класса больших сетей определяются механизмом их роста, называемым правилом предпочтительного связывания. Такие сети моделируют случайными графами, выражаемыми по правилу предпочтительного связывания, первая версия которого [1] включает следующие два пункта.

1. Граф растет за счет добавления к нему на каждом шаге времени  $t \in \{1, 2, \dots\}$  новой вершины с  $x = m = \text{const}$  ребрами (называемой приращением графа [4]). Исходный небольшой граф (затравка) задается произвольно.

2. Каждое из  $x$  ребер новой вершины присоединяется свободным концом к случайно выбираемой вершине графа. Вероятность  $p_i$  присоединения ребра к вершине  $i$  пропорциональна локальной степени связности  $k_i$  этой вершины:

$$p_i = \frac{k_i}{\sum_{j=1}^N k_j}, \quad i = 1, \dots, N, \quad (1)$$

где  $N$  — число вершин графа.

Это правило обуславливает тем большую вероятность  $p_i$  связывания нового ребра с вершиной  $i$ , чем больше ребер с нею уже связано.

В других версиях правила предпочтительного связывания вероятность  $p_i$  может определяться индивидуальной предпочтительностью вершины [2], зависеть от времени [3] или от иных факторов.

Теория случайных графов с нелинейным правилом предпочтительного связывания (НППС) [4–6] развивает теорию графов предпочтительного связывания,

охватывая в качестве частных случаев ряд ее ответвлений (таких как [1, 7]) и решая новые задачи. К ним относится и задача калибровки графов, моделирующих реальные сети. В [4–6, 8, 9] предлагаются методы, алгоритмы и программы, позволяющие калибровать графы с НППС по распределениям степеней связности (РСС) вершин. В данной статье впервые решается задача калибровки графов с НППС одновременно и по РСС вершин, и по РСС ребер.

**2. Исходные положения.** *Распределения степеней вершин.* В теории случайных графов с НППС [4–6, 8, 9] приращения графа в общем случае являются стохастическими и ориентированными, т.е. представляют собой новые вершины со случайным числом  $x$  исходящих дуг;  $P(x=k) = r_k$ ,  $g \leq k \leq h$ ,  $r_g + \dots + r_h = 1$ . Здесь  $g \geq 1$  — наименьшая степень приращения.

Конец каждой дуги приращения связывается со случайно выбранной вершиной графа. Вероятность  $p_i$  связывания с вершиной  $i$  пропорциональна некоторой функции (весу)  $f$  от степени связности  $k_i$  этой вершины:

$$p_i = \frac{f(k_i)}{\sum_{j=1}^N f(k_j)}, \quad i, j = 1, \dots, N, \quad (2)$$

где вес  $f(k) > 0$ , если  $g \leq k \leq M$ , иначе  $f(k) = 0$  (здесь  $M \leq \infty$ ). Для функции  $f(k)$  целочисленного аргумента  $k$  иногда удобнее использовать обозначение  $f_k$ .

Обозначим через  $N_0$  число вершин в затравке графа. Отсчет времени будем начинать с момента  $t_0 = N_0$ . На шаге времени  $t = t_0 + 1$  к графу добавляется первое приращение, дуги которого связываются

с вершинами имеющегося графа по правилу (2), и число вершин в графе возрастает на единицу. Далее эта операция повторяется на каждом новом шаге времени  $t$ . Поэтому всегда сразу после добавления очередного приращения выполняется равенство  $N=t$ .

В статье [4] решены задачи анализа и синтеза (калибровки) графов с НППС в части финального РСС вершин, достигаемого при  $N \rightarrow \infty$ . Решения этих задач даются следующими основными соотношениями.

При заданном распределении  $\{r_k\}$  числа  $x$  дуг в приращении и заданной весовой функции  $f(k)$  стационарное РСС вершин графа определяется формулой:

$$Q_k = \frac{r_k \langle f \rangle + m f_{k-1} Q_{k-1}}{\langle f \rangle + m f_k}, \quad k = g, g+1, g+2, \dots \quad (f_{g-1} = 0), \quad (3)$$

где  $Q_k$  — вероятность того, что случайно выбранная вершина будет иметь степень связности  $k$ ;  $m = S_k(kr_k)$  — среднее число дуг в приращении, а значение  $\langle f \rangle$  среднего веса вершины рассчитывается численно путем решения системы уравнений (3) с добавленным к ней уравнением

$$\langle f \rangle = \sum_{k=g}^M f_k Q_k. \quad (4)$$

При этом вероятности  $Q_k$  и средний вес  $\langle f \rangle$  определяются одновременно, а в качестве проверочного равенства используется формула для определения средней степени  $\langle k \rangle$ , которая по построению графа должна быть равна  $2m$ :

$$\langle k \rangle = \sum_{k=g}^{M+1} k Q_k = 2m. \quad (5)$$

В [4] предложена простая процедура численного решения системы (3), (4).

Задача синтеза (калибровки) графа с НППС по РСС вершин является обратной по отношению к задаче анализа. Задано РСС  $\{Q_k\}$ , требуется найти распределение  $\{r_k\}$  числа дуг в приращении и функцию  $f(k)$  такие, чтобы выращивался граф с РСС вершин  $\{Q_k\}$ . Решение этой задачи, также найденное в [4], дается следующей формулой, вытекающей из формулы (3):

$$f_k = \frac{Q_{k-1}}{Q_k} f_{k-1} + \frac{r_k}{Q_k} - 1, \quad k = g, g+1, g+2, \dots, \quad f_{M+1} = 0, \quad \langle f \rangle = m, \quad (6)$$

где  $Q_{g-1} = 0$ . При этом вероятности  $r_{g'}, \dots, r_h$  следует задавать так, чтобы среднее число дуг в приращении  $m = \sum_k (kr_k)$  было равно  $\langle k \rangle / 2$ , а веса  $f_{g'}, \dots, f_h$  в (6) были неотрицательными. В статье [6] эти условия формализованы так:

$$0 \leq r_i \leq 1, \quad \sum_{i=g}^h r_i = 1, \quad m = \sum_{i=g}^h i r_i = \frac{\langle k \rangle}{2}, \quad \sum_{i=g}^k r_i \geq \sum_{i=g}^k Q_i \quad (\forall k = \overline{g, h}). \quad (7)$$

Обычно имеется бесконечно много таких наборов  $r_{g'}, \dots, r_h$ . Любой из них при весах (6) обеспечивает требуемое финальное РСС  $\{Q_k\}$  вершин графа.

**Распределения степеней дуг/ребер.** Задача анализа РСС дуг/ребер для графов с НППС впервые решена в общем виде в статье [10]. При заданных  $\{r_k\}$  и  $f(k)$  выращивается граф с финальным распределением вероятностей дуг, определяемым формулой

$$Q_{l,k} = \frac{f_{k-1}(l r_l Q_{k-1} + m^2 Q_{l,k-1}) + f_{l-1} m^2 Q_{l-1,k}}{m(\langle f \rangle + m f_k + m f_l)}, \quad l, k = g, g+1, g+2, \dots, \quad (8)$$

где  $Q_{lk}$  — вероятность того, что случайно выбранная дуга исходит из вершины со степенью  $l$  и заходит в вершину со степенью  $k$ . Матрица  $\mathbf{Q} = \|Q_{lk}\|$  рассчитывается по формуле (7) построчно. Вероятности  $Q_k$  определены выше.

При  $x=m=g$  имеем  $r_g=1$  (приращения графа с НППС становятся постоянными). Формула (8) при этом превращается в формулу для  $Q_{lk}$ , найденную в [6]. Если еще положить, что  $f(k)=k$ , то граф с НППС становится графом Барабаши–Альберт, и формула (8) превращается в формулу РСС дуг этого графа, найденную в [9].

Если граф с НППС используется для моделирования сетей с неориентированными связями, то дуги в нем заменяются ребрами. Обозначая через  $\Theta_{lk}$  финальную вероятность того, что случайно выбранное ребро, проходимое в случайном направлении, ведет из вершины со степенью  $l$  в вершину со степенью  $k$ , получаем  $\Theta_{lk} = (Q_{lk} + Q_{kl})/2$ . Следовательно, матрица  $\Theta = \|\Theta_{lk}\|$  РСС ребер графа может быть рассчитана по матрице  $\mathbf{Q} = \|Q_{lk}\|$  РСС дуг с помощью следующего преобразования:

$$\Theta = \frac{1}{2} (\mathbf{Q} + \mathbf{Q}^T). \quad (9)$$

Возможность быстрого точного расчета РСС ребер по формулам (9), (8) позволяет поставить комплексную задачу калибровки графов с НППС одновременно и по РСС вершин, и по РСС ребер.

**3. Постановка и решение комплексной задачи калибровки графов с НППС.** Постановка задачи: по заданным РСС  $\{Q_k\}$  и  $\tilde{\Theta} = \|\tilde{\Theta}_{lk}\|$  требуется найти распределение  $\{r_k\}$  и функцию  $f$ , которые обеспечивают РСС вершин  $\{Q_k\}$  и доставляют минимум отклонения РСС  $\Theta$  степеней ребер от заданного РСС  $\tilde{\Theta}$ :

$$R(\Theta, \tilde{\Theta}) = \max_{l,k} |\Theta_{lk} - \tilde{\Theta}_{lk}| \rightarrow \min. \quad (10)$$

Целевая функция  $R(\Theta, \tilde{\Theta})$  экстремальной задачи (10) представляет собой расстояние Колмогорова между распределением  $\Theta$  степеней ребер выращиваемого графа и заданным распределением  $\tilde{\Theta}$ . Варьируемыми переменными являются вероятности  $r_{g'}, \dots, r_h$ . Область допустимых решений (ОДР) определяется ограничениями (7). Как показано в [6], эта ОДР непустая, если  $h \geq h^*$ , где  $h^*$  является наименьшим  $h$ , удовлетворяющим неравенству

$$\sum_{i=0}^{h-1} [1 - F_Q(i)] \geq m, \quad (11)$$

где  $F_Q(i) = \sum_{k=0}^i Q_k$ . Далее будем полагать, что выбрано значение  $h \geq h^*$  и исключим из рассмотрения малоинтересный и маловероятный случай, когда ОДР (7) содержит единственную точку.

Таким образом, при заданном РСС вершин  $\{Q_k\}$ , определяющем параметр  $m = \langle k \rangle / 2$ , и заданном РСС



$Q_l$	$l$	$k$										
		0	1	2	3	4	5	6	7	8	9	10
0	0	0	0	0	0	0	0	0	0	0	0	0
0,34145	1	0	0	0,00142	0,00232	0,00231	0,00211	0,00185	0,00153	0,00171	0,00149	0,00148
0,42246	2	0	0,00142	0,0045	0,00415	0,00372	0,00372	0,00303	0,00273	0,00323	0,00246	0,00235
0,09664	3	0	0,00232	0,00415	0,00233	0,00161	0,00114	0,00092	0,00083	0,00093	0,00075	0,00092
0,03874	4	0	0,00231	0,00372	0,00161	0,0012	0,00055	0,00057	0,00032	0,00037	0,00041	0,00046
0,01985	5	0	0,00211	0,00372	0,00114	0,00055	0,00041	0,00032	0,00018	0,00026	0,00013	0,00017
0,01273	6	0	0,00185	0,00303	0,00092	0,00057	0,00032	0,00031	0,00019	0,0001	0,0002	9,3E-05
0,00941	7	0	0,00153	0,00273	0,00083	0,00032	0,00018	0,00019	0,0001	0,00012	0,00012	0,0001
0,00722	8	0	0,00171	0,00323	0,00093	0,00037	0,00026	0,0001	0,00012	6,2E-05	8,3E-05	8,3E-05
0,00571	9	0	0,00149	0,00246	0,00075	0,00041	0,00013	0,0002	0,00012	8,3E-05	4,1E-05	0,0001
0,00462	10	0	0,00148	0,00235	0,00092	0,00046	0,00017	9,3E-05	0,0001	8,3E-05	0,0001	0

Рис. 1. РСС  $\{Q_l\}$  узлов и РСС  $\Theta = \|\Theta_{lk}\|$  связей сети АС (фрагмент)

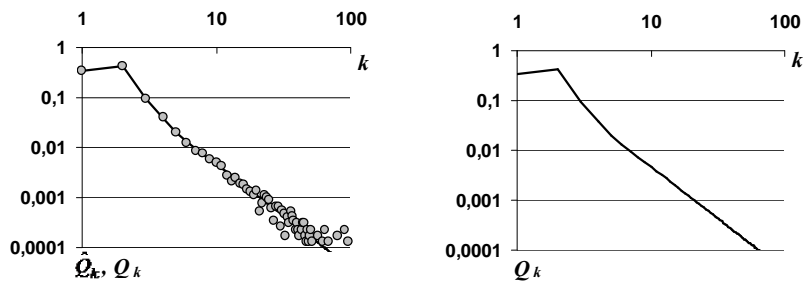


Рис. 2. Слева эмпирическое распределение  $\bar{Q}$  (маркеры) степеней узлов сети АС и сглаженное распределение  $Q_k$  (линия). Справа выделен график сглаженного распределения  $Q_k$

$\bar{\Theta} = \|\bar{\Theta}_{lk}\|$  имеем задачу выбора в области (7) решения  $r_{g'}, \dots, r_{h'}$ , определяющего посредством цепочки преобразований (6), (3), (8), (9) распределение  $\Theta = \|\Theta_{lk}\|$  степеней ребер выращиваемого графа, которое минимизирует целевую функцию  $R(\Theta, \bar{\Theta})$  (10). При этом заданное РСС вершин  $\{Q_k\}$  реализуется графом точно, а заданное РСС ребер  $\bar{\Theta} = \|\bar{\Theta}_{lk}\|$  реализуется с минимальной в смысле расстояния Колмогорова погрешностью.

Численные эксперименты показывают, что задача (10), (7) эффективно решается стандартными градиентными методами и что если в ее постановке заменить расстояние Колмогорова  $\max_{l,k} |\Theta_{lk} - \bar{\Theta}_{lk}|$

мерой погрешности  $\sum_{l,k} (\Theta_{lk} - \bar{\Theta}_{lk})^2$ , то решение задачи практически не изменяется.

**4. Пример решения комплексной задачи калибровки.** В качестве примера рассмотрим решение поставленной задачи калибровки при моделировании сети автономных систем (АС) Интернета по данным [11]. Сеть содержит 22 963 узла и 48 436 ребер, максимальная степень вершины  $k_{max} = 2390$ . Начальные фрагменты РСС  $\{Q_k\}$  узлов сети и РСС  $\bar{\Theta} = \|\bar{\Theta}_{lk}\|$  ее связей представлены на рис. 1.

Важно заметить, что РСС узлов моделируемой сети (первый столбец на рис. 1) представляет собой сглаженное распределение (график которого показан на рис. 2 справа) полученное из эмпирического распределения (рис. 2 слева) степеней узлов. Это сглаживание, основанное на тщательном теоретическом исследовании сети АС в статьях [4, 6], минимизирует влияние погрешностей эмпирических оценок вероятностей на результаты решения комплексной задачи калибровки.

Сглаженное распределение  $\{Q_k\}$  точно реализуется графом с НППС, калиброванным в [6] по РСС вершин. Этот граф определяется в [6] следующими параметрами:  $g = 1, h = 5; r_1, \dots, r_5 = 0,34145, 0,42246, 0,09664, 0,0645, 0,07495; m = 2,1093; \{f_1, \dots, f_5\} = \{0; 0; 0; 0,6329; 3,8769\}; f_k = 0,8949k$  при  $k > 5$ . Взяв это сглаженное РСС  $\{Q_k\}$  за основу, мы можем теперь точно воспроизвести его в графе с НППС при любом  $\{r_k\}$ , принадлежащем ОДР (7), если веса  $f$  определим по формулам (6). Такой расчет весов выполняется на компьютере «мгновенно» и при  $\langle f \rangle = m$  эти веса вместе с использованным  $\{r_k\}$  действительно определяют по рекуррентной формуле (3) в точности то же самое РСС  $\{Q_k\}$  вершин (сглаженное), которое требуется реализовать. Таким образом, первая часть рассматриваемой комплексной задачи калибровки графа решается без каких-либо вычислительных проблем. Остается решить вторую часть задачи — приблизить РСС ребер графа к заданному РСС  $\bar{\Theta} = \|\bar{\Theta}_{lk}\|$ .

На рис. 3 слева приведен график рассчитанного по формулам (6), (3), (8), (9) РСС  $\Theta = \|\Theta_{lk}\|$  ребер графа с НППС, калиброванного в [6] только по РСС вершин (параметры для генерации этого графа приведены выше).

Справа на рис. 3 приведен график РСС  $\bar{\Theta} = \|\bar{\Theta}_{lk}\|$  связей в моделируемой сети АС (фрагмент матрицы  $\bar{\Theta}$  см. на рис. 1). Сравнение графиков на рис. 3 показывает, что назвать их близкими можно лишь с большой натяжкой.

Поэтому имеет смысл попытаться сблизить реализуемое РСС  $\Theta$  ребер графа с известным РСС  $\bar{\Theta}$  связей в моделируемой сети, решая задачу (10), (7). Из (11) и заданного РСС вершин  $\{Q_k\}$  нетрудно определить, что мы можем использовать любое  $h \geq 5$ . Поскольку цепочка преобразований (6), (3), (8), (9),

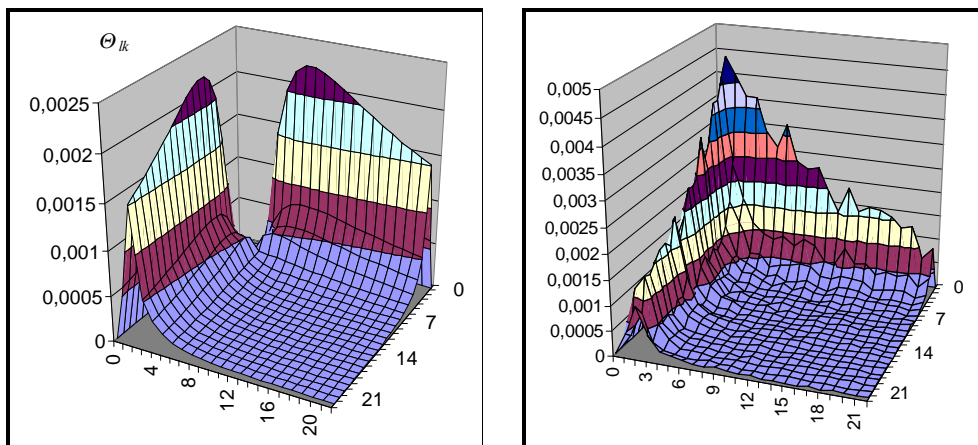


Рис. 3. Слева РСС ребер графа, калиброванного в [6] только по РСС вершин, справа РСС связей в моделируемой сети АС

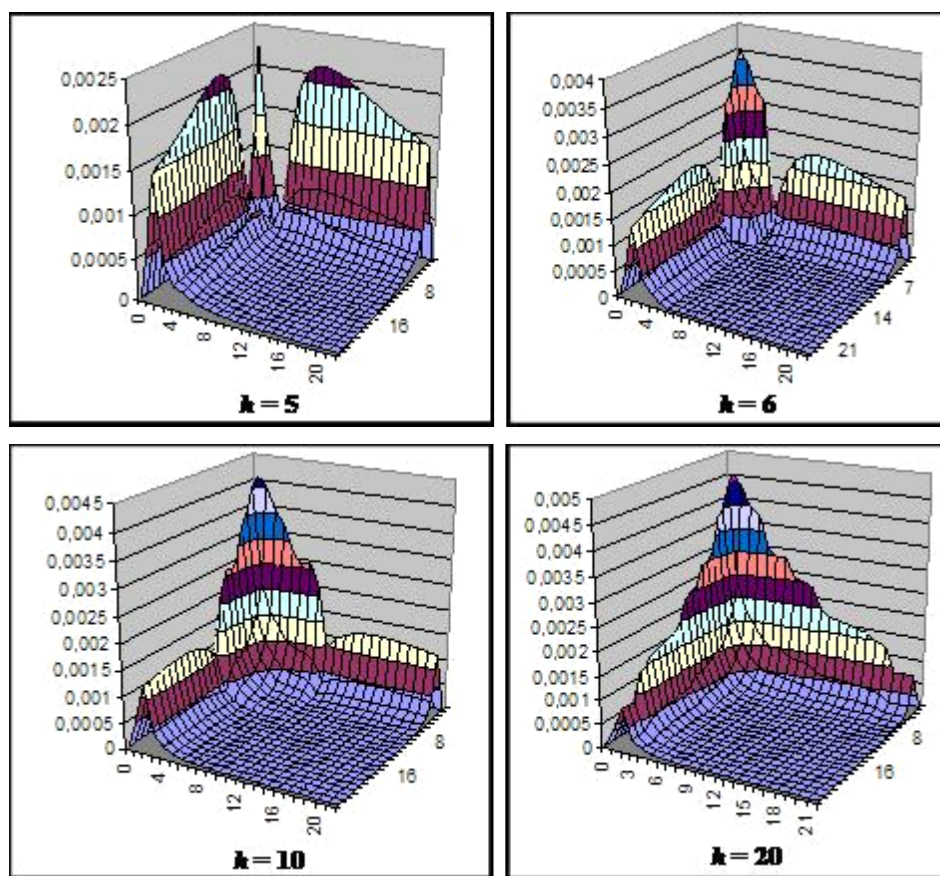


Рис. 4. РСС ребер графа, калиброванного по РСС ребер при  $h = 5, 6, 10$  и  $20$

определяющая  $\Theta$  через значения варьируемых параметров  $r_1, \dots, r_h$ , легко реализуется на одном листе Excel, мы можем при выбранном  $h$  быстро минимизировать целевую функцию (10) с помощью сервиса «поиск решения».

На рис. 4 представлены графики РСС ребер графа, реализуемого в результате такой оптимизации распределения  $\{r_k\}$  при различных значениях  $h$ .

Нетрудно видеть, что РСС ребер графа с ростом  $h$  приближаются к РСС связей в моделируемой сети (показанному на рис. 3 справа). При этом заданное слаженное РСС вершин  $\{Q_k\}$  при всех рассмотренных  $h$  реализуется точно. Параметры для выращивания графа при  $h = 20$  таковы:

1)  $\{r_1, \dots, r_{20}\} = \{0,37009; 0,45798; 0,106; 0,0331; 0,00214; 0,00819; 0,00384; 0; 0,00386; 0; 0; 0,0038; 0,00173; 0; 0; 0; 0,00093; 0,00047; 0,00788\}$ ;

2)  $\{f_1, \dots, f_{20}\} = \{0,08389; 0,15189; 0,76084; 1,75237; 2,52761; 3,5848; 4,25984; 4,5506; 5,43441; 5,71515; 5,92737; 7,25874; 8,181; 8,51582; 8,806; 9,0512; 9,25108; 10,0695; 10,6298; 17,8403\}$ ;

если  $k \geq 21$ , то  $f_k = 0,89202 \cdot k$ .

При таких  $\{r_k\}$  и  $f$  генерируется граф с  $m = 2,109055$  и  $\langle f \rangle = m$ , с заданным РСС вершин  $\{Q_k\}$  и с РСС ребер  $\Theta$ , близким к РСС связей в моделируемой сети АС.

**5. Заключение.** Постановка комплексной задачи калибровки и разработанный в статье метод ее решения существенно продвигают возможности использования графов с НППС для адекватного моделирования больших растущих сетей. Приведенный пример комплексной калибровки графа, моделирующего сеть АС, демонстрирует эффективность разработанного метода. Точно рассчитанное РСС вершин (см. рис. 2) и все точно рассчитанные

РСС ребер калиброванного при различных  $h$  графа (см. рис. 4) подтверждены моделированием, т.е. непосредственным выращиванием графов.

В приведенном примере решения комплексной задачи используется сглаженное РСС узлов моделируемой сети. Это позволяет существенно ускорить и упростить решение комплексной задачи в части реализации требуемого РСС вершин и тем самым упрощает и ускоряет ее решение в части реализации требуемого РСС ребер графа. При этом РСС *связей* в моделируемой сети не сглажено, т.е. содержит существенные стохастические погрешности. Это приводит к следующим двум выводам.

Во-первых, предложенный метод обладает значительной робастностью. Несмотря на погрешности в исходных данных (т.е. в заданном РСС связей сети), метод позволяет достаточно быстро найти параметры для выращивания графа, имеющего РСС ребер, весьма близкое к заданному.

Во-вторых, погрешности в заданном РСС связей очевидным образом сказываются на процессе и результатах комплексной калибровки графа. В процессе оптимизации распределений  $\{r_k\}$  градиентным методом (при фиксированных значениях  $h$ ) нет оснований полагать, что задача оптимизации выпуклая, и поэтому приходится несколько раз повторять оптимизацию при разных начальных условиях, чтобы убедиться, что процесс не остановился в каком-нибудь неудачном локальном минимуме целевой функции. На результатах калибровки это сказывается таким образом, что мы получаем не просто «хороший» граф, но и, одновременно, граф несколько «перекалиброванный». Так, сравнивая РСС ребер калиброванного графа (последний график на рис. 4) с заданным РСС связей сети АС (рис. 3 справа), нетрудно заметить, что РСС калиброванного графа стремится повторить «неровности» заданного РСС, порожденные стохастическими погрешностями.

Возможность «перекалибровки» графа «под погрешности» в исходных данных ставит задачу минимизации этих погрешностей. Такая минимизация погрешностей может быть достигнута за счет разработки методов сглаживания двумерных РСС связей сетей. Если моделируются сети большего размера (сеть АС содержит лишь 22 963 узла, а не миллионы узлов, как многие реальные сети), то исходные данные имеют меньшие стохастические погрешности, и проблема сглаживания становится не столь острой.

В целом полученные в статье результаты демонстрируют наличие у теории случайных графов с НППС широких возможностей для дальнейшего развития и значительных преимуществ перед частными моделями [1–3, 7], в которых заранее фиксируются вид функции предпочтения  $f$  и число дуг в приращении графа. Тонкая и точная комплексная калибровка графов с НППС, как показывает имитационное моделирование, приводит к лучшему соответствию структурных характеристик таких графов характеристикам моделируемых сетей, например, по частоте встречаемости типовых подграфов — мотивов [12]. Вместе с тем комплексная калибровка в об-

щем случае не гарантирует достаточное соответствие выращиваемых графов моделируемым сетям по частоте встречаемости мотивов. Это ставит перед теорией случайных графов с НППС задачу такого расширения числа параметров у приращений графа, которое позволило бы эффективно решать задачу калибровки графов и по встречаемости мотивов.

#### Библиографический список

1. Barabasi A. L., Albert R. Emergence of scaling in random networks // *Science*. 1999. 286. P. 509–512.
2. Bianconi G., Barabasi A. L. Competition and multiscaling in evolving networks // *Europhys. Lett.* 54. 2001. P. 436.
3. Dorogovtsev S. N., Mendes J. F. F. Effect of the accelerated growth of communications networks on their structure // *Phys. Rev. E*. 63. 2001. 025101.
4. Задорожный В. Н. Случайные графы с нелинейным правилом предпочтительного связывания // *Проблемы управления*. 2011. № 6. С. 2–11.
5. Zadorozhnyi V. N., Yudin E. B. Growing Network: Nonlinear Extension of the Barabasi-Albert Model, in *Communications in Computer and Information Science*. 2014. Vol. 487. P. 432–439.
6. Zadorozhnyi V. N., Yudin E. B. Growing network: models following nonlinear preferential attachment rule // *Physica A: Statistical Mechanics and its Applications*. 2015. Vol. 428. P. 111–132. DOI: 10.1016/j.physa.2015.01.052.
7. Krapivsky P. L., Redner S. Organization of growing random networks // *Phys. Rev. E* 63. 2001. 066123.
8. Задорожный В. Н., Юдин Е. Б. Система агентного моделирования «SIMBIGRAPH» // *Навигатор в мире науки и образования*. 2012. № 4-7 (20–23). — С. 536.
9. Zadorozhnyi V. N., Yudin E. B. Structural properties of the scale-free Barabasi-Albert graph // *Automation and Remote Control*. 2012. Vol. 73. №. 4. P. 702–716. DOI: 10.1134/S0005117912040091.
10. Задорожный В. Н. Уравнения динамики концевых степеней дуг в растущих графах // *Динамика систем, механизмов и машин*. 2016. № 1. Т. 3. С. 327–336.
11. Структура автономных систем сети Интернет, воссозданная на основе BGP таблиц, 2006. URL: <http://www-personal.umich.edu/~mejn/netdata/as-22july06.zip> (дата обращения: 01.09.2009).
12. Задорожный В. Н., Юдин Е. Б. Расчет числа сетевых мотивов методом случайной выборки каркасов // *Омский научный вестник. Сер. Приборы, машины и технологии*. 2015. № 2 (140). С. 208–211.

**ЗАДОРЖНЫЙ Владимир Николаевич**, доктор технических наук, доцент (Россия), профессор кафедры автоматизированных систем обработки информации и управления Омского государственного технического университета.

Адрес для переписки: zwn2015@yandex.ru

**ЮДИН Евгений Борисович**, кандидат технических наук, старший научный сотрудник Института математики им. С. Л. Соболева СО РАН, Омский филиал. Адрес для переписки: udinev@asoiu.com

Статья поступила в редакцию 20.12.2016 г.

© В. Н. Задорожный, Е. Б. Юдин

## ПРИМЕНЕНИЕ НЕПРЕРЫВНОГО АНАЛИЗА КАЧЕСТВА ПРИ СОЗДАНИИ СИСТЕМЫ УСОВЕРШЕНСТВОВАННОГО УПРАВЛЕНИЯ ТЕХНОЛОГИЧЕСКИМ ПРОЦЕССОМ УСТАНОВКИ СЕРНОКИСЛОТНОГО АЛКИЛИРОВАНИЯ

В данной статье рассматривается задача повышения оперативности данных о качестве продуктов для системы усовершенствованного управления технологическим процессом (СУУТП) установки серноокислотного алкилирования. Это необходимо для того, чтобы система усовершенствованного управления могла решать проблемы в темпе технологического режима. Была поставлена цель — разработать виртуальные анализаторы качества основных продуктов установки. На основе регрессионного анализа лабораторных и технологических данных работы установки было получено несколько моделей виртуальных анализаторов качества и была проведена оценка их адекватности, выбрана конечная модель.

**Ключевые слова:** виртуальный анализатор, СУУТП, серноокислотное алкилирование.

**Введение.** В условиях нестабильной ситуации в мировой экономике энергетических ресурсов, снижения спроса и стоимости продукции ведущие компании нефтегазовой отрасли ищут способы снижения издержек и получения максимальной прибыли. Традиционными способами повышения эффективности производства являются применение новых технологий, модернизация существующего оборудования, что требует существенных затрат и окупается через значительный промежуток времени. Другим менее затратным и быстро окупающимся методом является повышение качества управления за счет средств усовершенствованного управления процессом (АРС или СУУТП).

Существуют различные технологии построения СУУТП [1], такие как:

- А. Прогнозирующее управление по модели.
- В. Линейное программирование.
- С. Нелинейное управление.
- Д. Компенсация запаздывания.

Классическая архитектура СУУТП представлена на рис. 1. Установка серноокислотного алкилирования предназначена для производства алкилбензина. Сырьем для установки служит бутан-бутиленовая фракция, содержащая около 50 % олефинов, входящих в реакцию, и изобутановая фракция. Важность установки алкилирования для нефтеперерабатывающего завода состоит в том, что она производит высокооктановый бензин комбинированием легких углеводородов с изобутаном, чем повышается ценность

этих фракций. Получаемый алкилат имеет высокое октановое число (по исследовательскому методу =96), низкую упругость паров (по Рейду 0,43 кг/см<sup>2</sup> абс.) и не содержит ароматических углеводородов. Согласно данным фирмы Yokogawa [2], внедрение СУУТП на установки серноокислотного алкилирования может принести потенциальной экономической прибыли до 72 млн руб. в год (рис. 2), поэтому разработка и внедрение данной системы является одной из приоритетных задач для НПЗ.

Для успешной реализации СУУТП необходимым условием являются виртуальные анализаторы (ВА) качества продуктов достаточной степени достоверности. Принцип действия ВА основан на непрерывном определении показателя качества по математической модели, описывающей его взаимосвязь с текущими значениями измеряемых технологических переменных. ВА позволяют оценивать не измеряемые непосредственно, но необходимые показатели качества продукта по таким измеряемым параметрам технологического процесса, как температура, давление, расход, непрерывно контролируемым современными системами управления.

**Постановка задачи.** Рассмотрим принцип построения ВА на примере показателя качества «Температура конца кипения стабильного алкилбензина». Данный параметр качества строго регламентирован, и не должен превышать 205 °С. Поскольку для построения виртуального анализатора используются лабораторные данные, то для оценки его точности

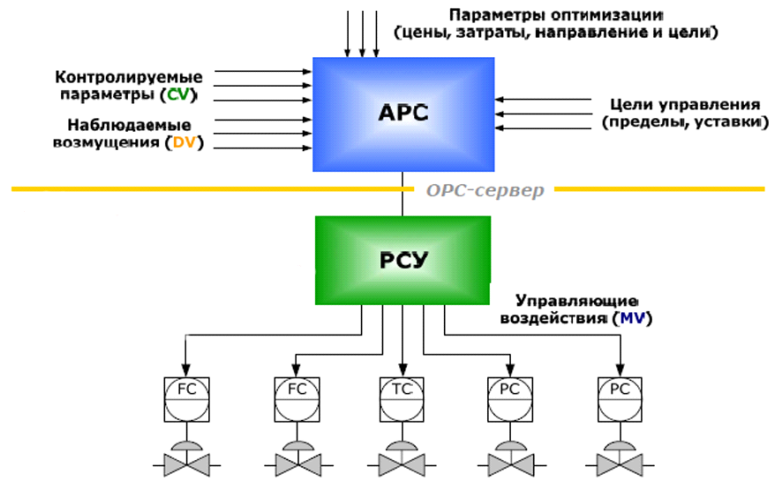


Рис. 1. Классическая архитектура СУУТП

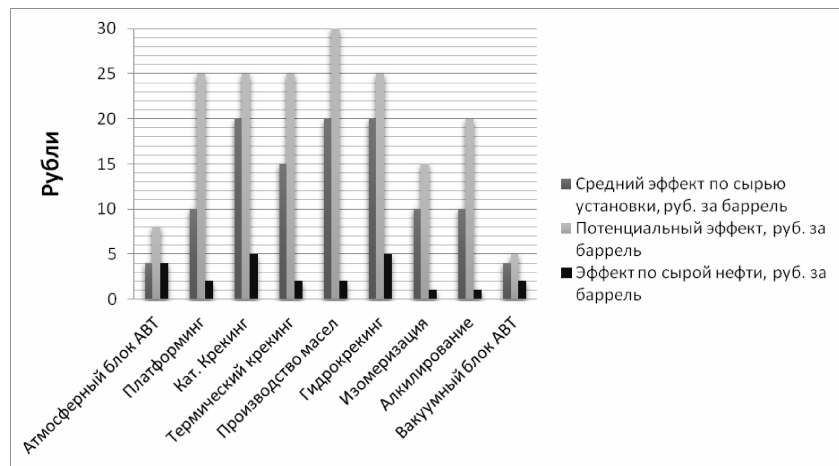


Рис. 2. Типичные эффекты внедрения СУУТП на НПЗ

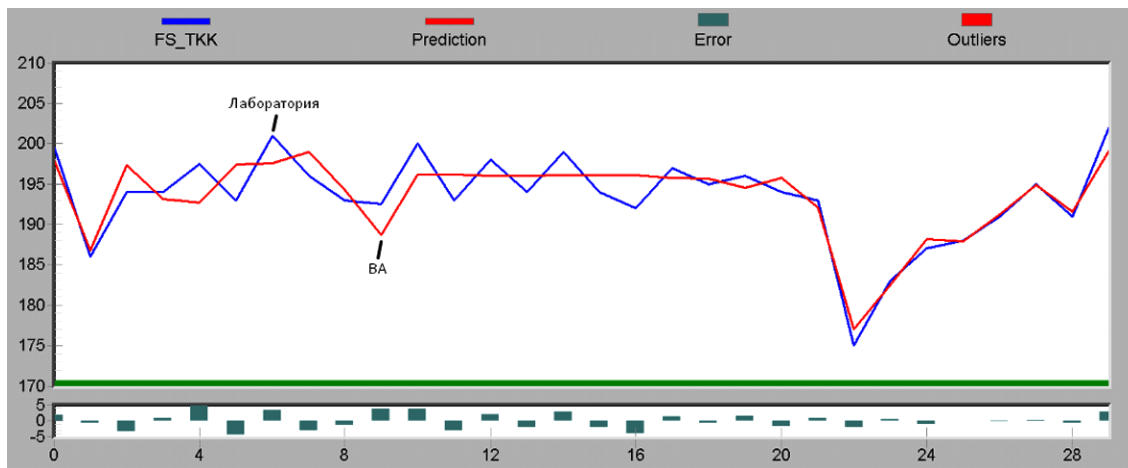


Рис. 3. График лабораторных данных и показания модели виртуального анализатора № 1

примем требования на точность лабораторного анализа согласно нормативной документации (воспроизводимость данного анализа составляет  $3,5\text{ }^{\circ}\text{C}$ ) [3].

Качество виртуальных анализаторов напрямую зависит от исходных ретроспективных данных. Для получения адекватной и работоспособной модели необходимо использовать выборку данных для стабильного технологического режима. Также по причине того, что технологический процесс является

динамической системой, поэтому данные имеют свойство быстро устаревать и не отражать действительное состояние технологического процесса. Опираясь на описанные выше условия, используем данные за период с 01.07.16 по 01.09.16, когда установка работала в одном режиме, а устаревание данных не превышает 6 месяцев.

**Теория.** Для реализации оптимизационных функций в современных ВА применяется широкий спектр

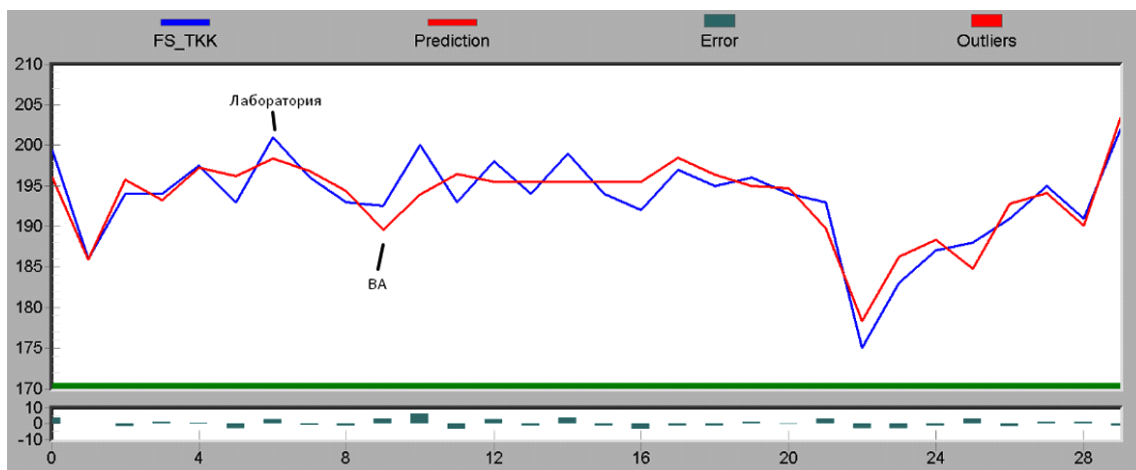


Рис. 4. График лабораторных данных и показания модели виртуального анализатора № 2

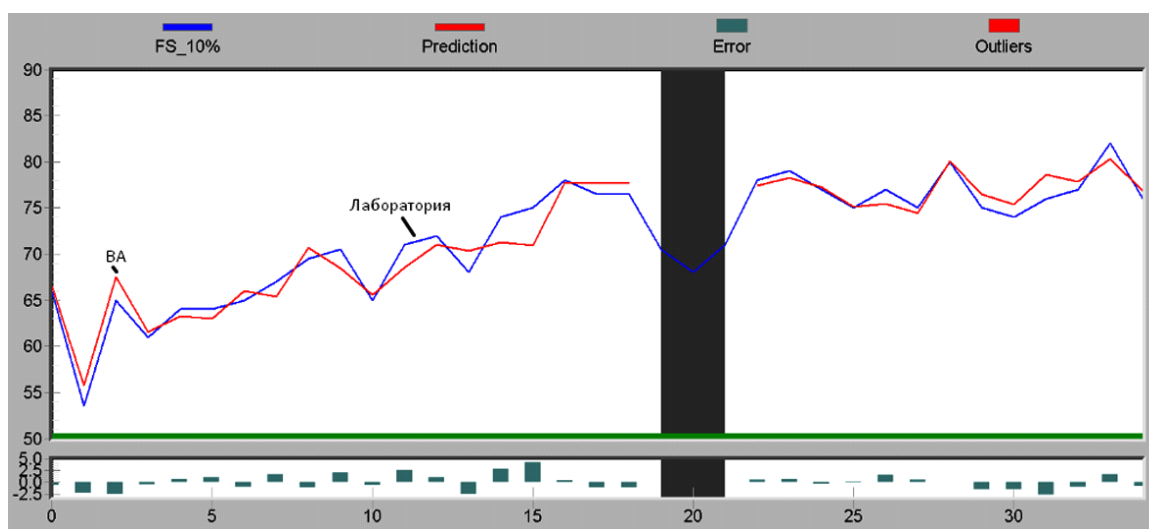


Рис. 5. График лабораторных данных и показания модели виртуального анализатора «Температура точки 10 % алкилбензина»

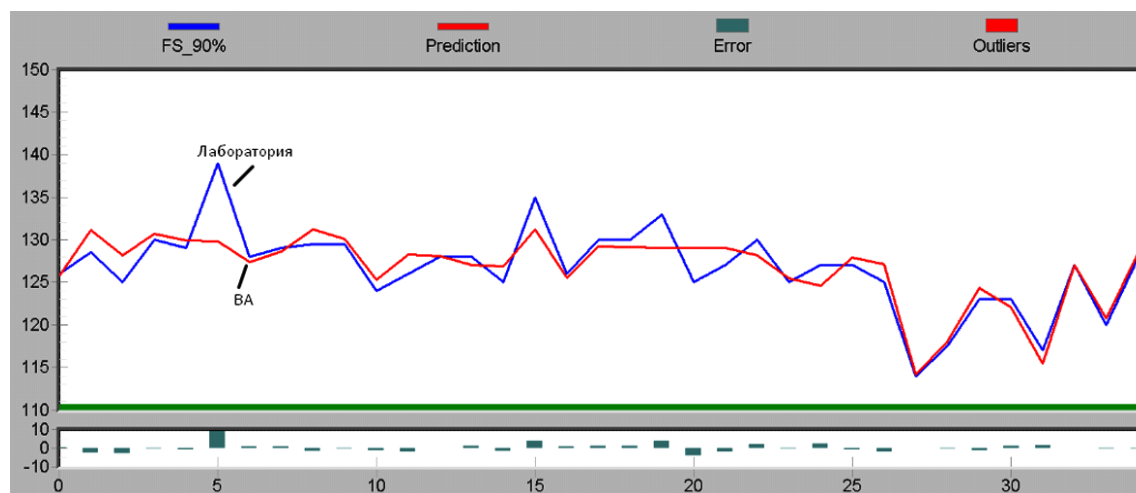


Рис. 6. График лабораторных данных и показания модели виртуального анализатора «Температура точки 90 % алкилбензина»

как традиционных алгоритмов и методов анализа данных: наименьших квадратов (НК), робастной регрессии (РР), проекции на латентные структуры (ПЛС) (случай линейных моделей), гребневой регрессии (ГР) и алгоритмы ACE (Alternating conditional expectation — чередующееся условное математическое ожидание) (нелинейные непараметрические

модели)), так и относительно авангардных: нейронные сети, генетические алгоритмы. Тенденции развития математического аппарата ВА связаны с методами «машинного обучения» [4].

Интеллектуальный анализ данных представляет собой процесс поддержки принятия решения, основанный на поиске в этих данных скрытых законо-

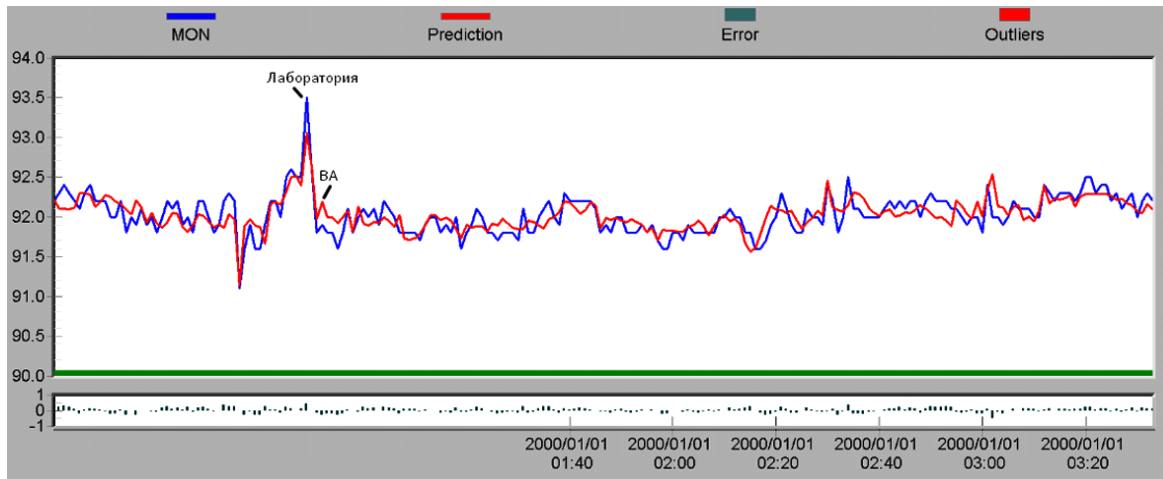


Рис. 7. График лабораторных данных и показаний модели виртуального анализатора «Октановое число по моторному методу алкилбензина»

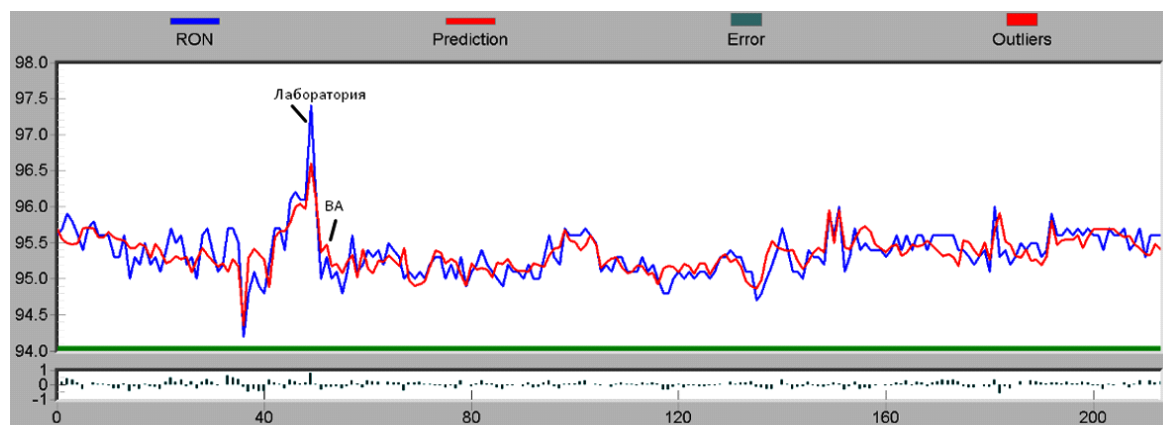


Рис. 8. График лабораторных данных и показаний модели виртуального анализатора «Октановое число по исследовательскому методу алкилбензина»

Таблица 1  
Статистические оценки моделей ВА

	Модель № 1	Модель № 2
RMSE	2,44	2,47
Sigma	2,48	2,52
Index	19,84	20,47

мерностей (шаблонов информации). При этом накопленные сведения обобщаются до информации, которая может быть охарактеризована как знания.

На текущий момент многие поставщики программных продуктов для систем усовершенствованного управления предоставляют решения для создания ВА в виде конфигурируемых систем, относительно простых в употреблении. Большинство из них снабжены графическими пакетами для представления результатов и направления действий операторов [5].

**Результаты экспериментов.** В результате разработки ВА были получены модели, графики которых представлены на рис. 3 и 4.

Для оценки адекватности модели виртуального анализатора необходимо [6]:

- провести визуальную оценку графиков;
- оценить модели, используя статистические характеристики;
- проверить физико-химическую природу влияния параметров, используемых в модели ВА.

При визуальной оценке необходимо определить повторяемость графика показаний данных ВА и лабо-

раторных данных. На рис. 4 модель виртуального анализатора № 2 повторяет лабораторные данные лучше, чем модель анализатора № 1 на рис. 3.

Для определения адекватности модели используем следующие статистические оценки:

А. RMSE — среднеквадратическая ошибка прогноза, которая фиксирует влияние смещения и среднеквадратическое отклонение.

В. Sigma — среднеквадратическое отклонение, которое фиксирует распределение ошибки. Данный параметр не должен превышать воспроизводимость лабораторного анализа.

С. Index — соотношение между среднеквадратической ошибкой прогноза и дисперсией выхода.

В табл. 1 представлены статистические оценки моделей ВА. Следует отметить, что Sigma всех моделей меньше, чем воспроизводимость лабораторного анализа, однако модель № 1 имеет статистические оценки лучше, чем модель № 2.

Из табл. 2 и 3 следует, что параметры уравнения модели № 1 адекватно описывают технологический процесс, в отличие от модели № 2 (коэффициент

Таблица 2  
Коэффициенты регрессии для переменных, используемых в модели 1

Позиция	Описание	Коэффициент регрессии
FIC203	Расход сырья, насыщенного изобутаном, в колонну С-331	1,4853
PIA303	Давление верха колонны С-332	0,76707
TI206	Температура верха колонны С-331	-2,6353
TIC301	Температура на 3-й тарелке колонны С-332	0,50210

Таблица 3  
Коэффициенты регрессии для переменных, используемых в модели 2

Позиция	Описание	Коэффициент регрессии
FIC302	Расход орошения от насосов Р-336А, В в колонну С-332	1,5757
PI307	Давление в сепараторе В-336	0,20695
PIA217	Давление в кубе колонны С-331	-183,78
TIC301	Температура на 3-й тарелке колонны С-33	0,82751

Таблица 4  
Модели ВА показателей качества продуктов установки сернокислотного алкилирования

Наименование ВА	Статическая оценка			Уравнение ВА	Ссылки на рисунки
	RMSE	Sigma	Index		
Температура точки 10 % алкилбензина	1,58	1,61	6,18	$0,19567 \cdot \text{FIC204} + 0,39524 \cdot \text{TI202} + 0,80927 \cdot \text{TI305} - 99,646$	Рис. 5
Температура точки 90 % алкилбензина	2,33	2,36	25	$6,0697 \cdot \text{TI205} - 6,8397 \cdot \text{TI206} + 0,42338 \cdot \text{TI209} - 1,7797 \cdot \text{TI304} + 1,4663 \cdot \text{TI305} - 0,67073 \cdot \text{TIC301} + 138,72$	Рис. 6
Октановое число по моторному методу алкилбензина	0,15	0,15	38	$0,11339 \cdot \text{FIC101} - 0,217 \cdot \text{FIC202} - 0,061053 \cdot \text{FIC306} - 0,012344 \cdot \text{FIC307} + 11,824 \cdot \text{PIC104} - 0,26380 \cdot \text{TI208} - 0,34194 \cdot \text{TI305} + 0,12244 \cdot \text{PIA304} + 92,245$	Рис. 7
Октановое число по исследовательскому методу алкилбензина	0,2	0,2	39	$0,10730 \cdot \text{FIC101} - 0,24919 \cdot \text{FIC202} - 0,041488 \cdot \text{FIC306} - 0,014129 \cdot \text{FIC307} - 1,3362 \cdot \text{PDIA302} + 20,747 \cdot \text{PIC104} - 0,028537 \cdot \text{TI101} - 0,57195 \cdot \text{TI208} + 93,379$	Рис. 8

перед параметром «Расход орошения от насосов Р-336А,В в колонну С-332» имеет знак «+», а должен быть «-» [7]).

По описанной выше методике, помимо ВА «Температура конца кипения алкилбензина», также были разработаны модели ВА, показателей качества нефтепродуктов установки сернокислотного алкилирования. Модели ВА и их статические оценки представлены в табл. 4.

**Обсуждение результатов.** Используя оценки, данные моделям виртуальных анализаторов выше, можно сделать вывод о том, что модель № 1 лучше, чем модель № 2 по статистическим оценкам и коэффициентам уравнения регрессии, используемых переменных. Поэтому в составе СУУТП будет использована модель № 1.

**Выводы и заключения.** В результате исследований были получены модели виртуальных анализаторов качества «Температура конца кипения алкилбензина»; данный параметр строго регламентирован и требует постоянного контроля. Проведен анализ разработанных моделей, в результате которого была выбрана модель № 1 со следующими характеристиками: среднеквадратическая ошибка прогноза — 2,44; среднеквадратическое отклонение — 2,48;

соотношение между среднеквадратической ошибкой прогноза и дисперсией выхода — 19,84. Разработанная модель виртуального анализатора качества позволяет анализировать качество продуктов установки сернокислотного алкилирования. Использование полученных моделей в составе СУУТП позволит решать задачи оптимизаций в темпе производственного процесса для получения дополнительной экономической прибыли.

#### Библиографический список

1. Wade H. L. Basic and advanced regulatory control: system design and application. 2nd ed. Research Triangle Park: ISA — The Instrumentation, Systems, and Automation Society. 2004. 387 p.
2. Рылов М. А. Обзор систем усовершенствованного управления технологическими процессами // Исследовано в России: эл. журн. 2013. 008. С. 120—130.
3. Бадикова А. Д., Цадкин М. А. Разработка химической технологии сернокислотного алкилирования изопарафинов олефинами с применением в качестве сырья бензинов термодеструктивного происхождения // Вестник Башкирского университета. 1997. № 2. С. 46.
4. Дуго Г. Б., Дуго Н. Б., Можаровский И. С., Торгашов А. Ю. Исследование моделей виртуальных анализаторов



массобменного технологического процесса ректификации // Информатика и системы управления. 2011. № 4 (30). С. 17–26.

5. Бахтадзе Н. Н. Виртуальные анализаторы (идентификационный подход) // Автоматизация и телемеханика. 2004. № 11. С. 3–23.

6. Дуго Г. Б., Дуго Н. Б., Можаровский И. С., Торгашов А. Ю. Разработка моделей показателей качества ректификационных колонн, функционирующих в предельных режимах // Идентификация систем и задач управления: тр. IX Междунар. конф. SICPRO'12, 30 января – 2 февраля 2012 г. М., 2012. С. 211–221.

7. Friedman Y. Z. Alkylation Product Separation Control // HYDROCARBON PROCESSING. 09. 2008. Vol. 87. no. 9. P. 178.

8. Бельков Ю. Н., Кнеллер Д. В., Торгашов А. Ю., Файрузов Д. Х. Система усовершенствованного управления установкой первичной переработки нефти: создание, внедрение, сопровождение // Автоматизация и телемеханика. 2013. № 8. С. 3–23.

**ИБАТУЛЛИН Альберт Амирович**, кандидат технических наук, доцент кафедры автоматизации и робототехники.

Адрес для переписки: Ibatullin77@yandex.ru

**ОГУДОВ Андрей Александрович**, магистрант гр. АТПМ-151 факультета элитного образования и магистратуры.

Адрес для переписки: Andreyogudov@gmail.com

**ХАКИМОВ Рустам Анварович**, магистрант гр. АТПМ-151 факультета элитного образования и магистратуры.

Адрес для переписки: Machinestandard@mail.ru

Статья поступила в редакцию 23.12.2016 г.

© А. А. Ибатуллин, А. А. Огудов, Р. А. Хакимов

УДК 681.5:658.5

**А. А. ИБАТУЛЛИН  
А. А. ОГУДОВ  
Р. А. ХАКИМОВ**

Омский государственный  
технический университет,  
г. Омск

## РАЗРАБОТКА МОДЕЛИ НЕЧЕТКОГО ПИД-РЕГУЛЯТОРА КОНТУРА УПРАВЛЕНИЯ ТЕМПЕРАТУРЫ В КОЛОННЕ ДЕИЗОБУТАНИЗАЦИИ

В данной статье представлена проблема управления температурного контура в колонне деизобутанизации на установке сернокислотного алкилирования. Проанализированы основные проблемы и характерные особенности настройки данного контура. Была поставлена цель — разработать ПО, которое позволит учитывать нелинейности и высокую инерционность технологического процесса. Выявлена и обоснована необходимость внедрения нечеткого управления. На основе проведенных исследований была разработана модель нечеткого ПИД-регулятора, а также создан пакет надстроек, реализующий принципы нечеткой логики, в АСУТП Centum VP фирмы Yokogawa.

**Ключевые слова:** нечеткое управление, алкилирование, Centum VP.

**Введение.** Анализ качества подстройки ПИД-регуляторов посвящено немало трудов [1, 2]. При плохих подстройках оператор будет вести процесс вручную, что повышает нагрузку на оператора, ухудшает качество продукции, повышает риск возникновения аварийной ситуации. ПИД-регуляторами на стандартной логике не учитывают нелинейные зависимости параметров процесса. Решить ряд названных проблем можно при помощи применения правил нечеткой логики для подстройки коэффициентов ПИД-регулятора.

При построении ПИД-регулятора на нечеткой логике необходимо решить следующие задачи:

— разработать математическую модель объекта регулирования;

— произвести идентификацию контура регулирования;

— разработать базу правил для нечеткой логики;

— провести моделирование работы контура с применением нечеткой логики;

— разработать ПО для реализации методов нечеткой логики в АСУТП.

**Постановка задачи.** Рассмотрим контур управления температуры в колонне С-331 на установке сернокислотного алкилирования (рис. 1). Перепад температуры между 46 и 16 тарелками (датчик ТУС207) регулируется за счет подогрева куба колонны. Куб подогревается в теплообменнике Е-335, в трубчатую часть которого подается пар (расход пара регистрируется прибором 33FRC201). Расход пара в теплообменник управляется при помощи регулирующего клапана 33LV201 на линии откачки конденсата и датчика уровня LIRCA201 в конденсаторе. Данный контур представляет собой слож-

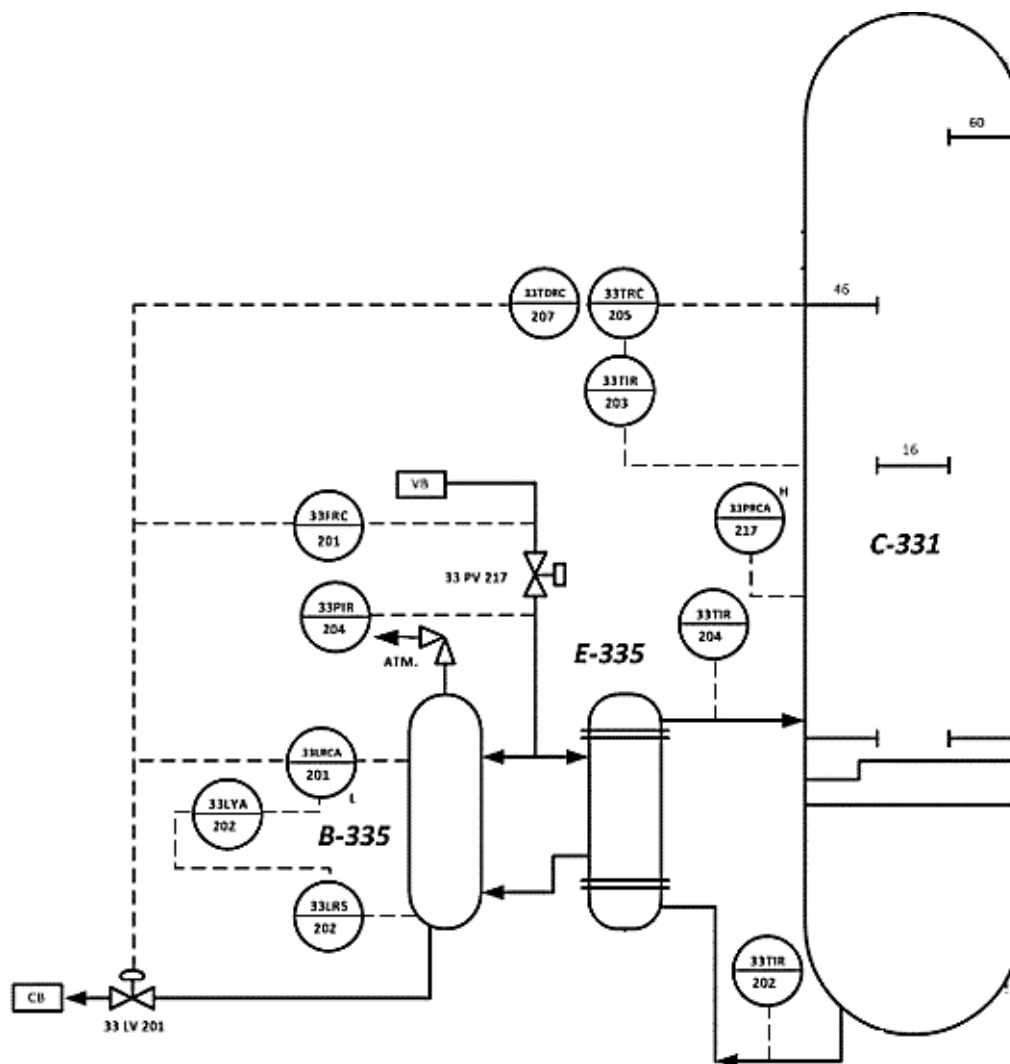


Рис. 1. Колонна деизобутанизации установки сернокислотного алкилирования

ный каскад и имеет существенные проблемы в конфигурировании, а именно вследствие того, что конденсатор находится ниже теплообменника, датчик уровня постоянно залит, а также присутствует высокая инерционность теплового процесса, поэтому для увеличения адаптивности системы управления предлагается использовать методы нечеткой логики для автоматической подстройки коэффициентов ПИД-регулятора.

**Теория.** Теория нечетких множеств была предложена американским математиком Лотфи Заде в 1965 г. [3]. Применение теории нечетких множеств позволяет описывать нечеткие понятия и знания, оперировать этими знаниями и делать нечеткие выводы. Нечеткое множество представляет собой совокупность элементов произвольной природы, относительно которых нельзя с полной определенностью утверждать, принадлежит ли тот или иной элемент рассматриваемой совокупности. Формально нечеткое множество  $A$  определяется как множество упорядоченных пар или кортежей вида  $\langle x, \mu_A(x) \rangle$ , где  $x$  является элементом некоторого универсального множества, или универсума,  $E$ , а  $\mu_A(x)$  — функция принадлежности, которая ставит каждому из элементов  $x$  универсального множества некоторое действительное число из интервала  $[0, 1]$ . При этом  $\mu_A(x) = 1$  означает, что элемент  $x$  из  $E$  определенно принадлежит нечеткому множеству  $A$ , а значение  $\mu_A(x) = 0$  означает, что элемент  $x$  из  $E$  определенно не принадле-

жит нечеткому множеству  $A$ . С помощью нечеткой логики обрабатывают неопределенные данные и ситуации, используя ассоциативные понятия человека — лингвистические термины.

Разработка и применение систем нечеткого вывода включают в себя ряд этапов, которые выполняются с помощью основных положений нечеткой логики [4]. Нечеткий вывод характеризуется следующими этапами:

1. Формирование базы правил (БП) систем нечеткого вывода, предназначенной для формального представления эмпирических знаний или знаний экспертов в проблемной области. В системах нечеткого вывода используются правила, в которых условия сформулированы в терминах нечетких лингвистических высказываний.

2. Фаззификация ( $\Phi$ ) входных переменных, представляющая собой процедуру нахождения значений функций принадлежности нечетких множеств (термов) на основе обычных (четко определенных) исходных данных. Целью этапа фаззификации является установление соответствия между конкретным значением отдельной входной переменной системы нечеткого вывода и значением функции принадлежности соответствующего ей значения входной лингвистической переменной.

3. Операции блока логического заключения ( $\Lambda$ ), выполняющего функции, аналогичные функциям вычислительного устройства в классической системе



Выборка данных из Pi Datalink

Дата	33FRC201	33LRCA201	33T203
02.01.2015 00:00:00	17,57973671	99,99977875	69,68743896
03.01.2015 00:00:00	17,06762505	99,99977875	69,70450592
04.01.2015 00:00:00	16,64416695	99,99977875	69,09999847
05.01.2015 00:00:00	17,00856781	99,99977875	70,05821991

регулирования, и определяющего степень истинности условий по каждому из правил, хранящихся в БП.

4. Дефазификация (ДФ) выходных переменных в системах нечеткого вывода, представляющая собой процедуру или процесс нахождения обычного (четко определенного) значения для каждой из выходных лингвистических переменных. Цель дефазификации заключается в том, чтобы, используя результаты логического заключения, получить значение каждой из выходных переменных, которое может быть использовано специальными устройствами, внешними по отношению к системе нечеткого вывода.

Для разработки базы правил и проверки работоспособности необходимо произвести идентификацию объекта регулирования. Процесс получения (синтеза) математического описания объекта на основе экспериментально полученных сигналов на его входе и выходе называется идентификацией объекта [5]. Существуют две причины, ограничивающие применение точных моделей. Первой из них является невозможность аналитического решения системы уравнений, описывающих ПИД-регулятор с моделью высокого порядка (а именно аналитические решения получили наибольшее распространение в ПИД-регуляторах с автоматической подстройкой). Вторая причина состоит в том, что при большом числе параметров и высоком уровне шума измерений количество информации, полученной в эксперименте, оказывается недостаточным для идентификации тонких особенностей поведения объекта. Выбор оптимальной модели обычно основан на критерии достаточности качества регулирования при минимальной сложности модели. Для нелинейных процессов и при повышенных требованиях к качеству регулирования разрабатывают модели с индивидуальной структурой, основываясь на физике процессов, протекающих в объекте управления. Если процесс любой сложности аппроксимировать моделью первого порядка с транспортной задержкой, то полученные таким способом постоянная времени  $T$  и задержка  $L$  называются соответственно эффективной постоянной времени и эффективной задержкой. Теория ПИД-регуляторов хорошо развита для линейных объектов управления. Однако практически все реальные объекты имеют нелинейность типа «ограничение управляющего воздействия». Ограничение может быть связано, например, с ограниченной мощностью нагревателя при регулировании тепловых процессов, с ограничением площади сечения перепускного клапана, с ограничением скорости потока жидкости и т.п. [6].

**Результаты экспериментов.** Для разработки математической модели объекта управления используем ПО Aspen Hysys. На рис. 2 представлена разработанная схема. Можно сказать, что невозможно настроить контур при данной конфигурации, используя стандартные методы настройки ПИД-регуляторов. Поэтому необходимо выполнить другую конфигурацию контура и использовать аппарат

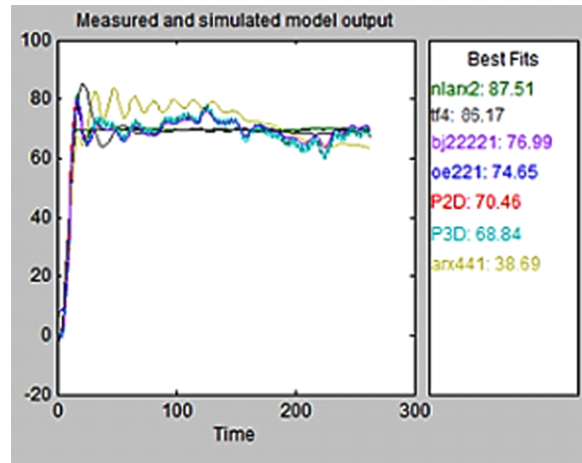


Рис. 3. Модели, полученные в ПО Matlab

нечеткой логики для автоматической подстройки ПИД-регулятора. Для разработки базы правил нечеткой логики, а также для последующего моделирования контура необходимо произвести идентификацию модели в ПО Matlab. Для идентификации объекта использовалась выборка данных из Pi Datalink. Часть выборки представлена в табл. 1.

Для получения математической модели используется пакет System Identification Tool ПО Matlab. В нашем случае наиболее оптимальной моделью будет считаться модель tf4, с достоверностью 86,17, т.к. сигнал модели tf4 меньше остальных выходит за границы «достоверности» (рис. 3).

В результате синтеза модели получили передаточную функцию теплообменника. Рассмотрим построение регулятора на базе нечеткой логики для подстройки ПИД-регулятора контура управления температуры в колонне деизоизобутанизации установки сернокислотного алкилирования. На рис. 4 изображена структурная схема контура управления температурой.

Структура нечеткого регулятора выглядит следующим образом:

- 1) база правил;
- 2) функция принадлежности входов, в нашем случае это функция для изменения скорости температуры;
- 3) функция принадлежности выходов, в нашем случае это функции для пропорционального и интегрального коэффициентов ПИД-регулятора, так как данный тип регулятора получил наибольшее распространение на практике.

Нечеткой базой правил [7] называется совокупность нечетких правил «Если — то», определяющих взаимосвязь между входами и выходами исследуемого объекта. Для данной системы были разработаны следующие правила, представленные в табл. 2, где  $bn$  — большое отрицательное значение,  $ml$  — малое отрицательное значение, 0 — нулевое значе-

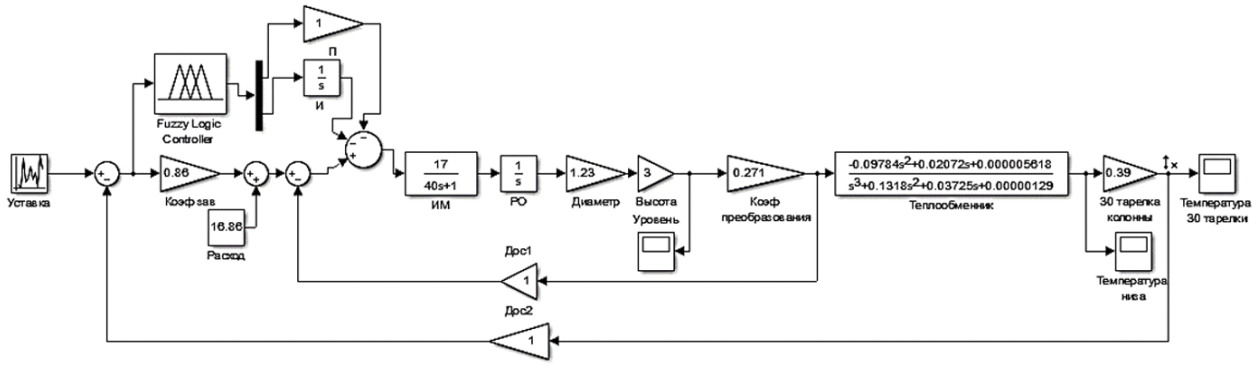
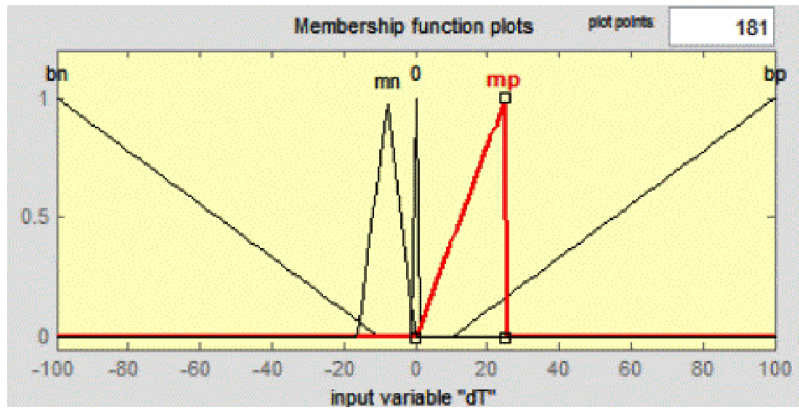
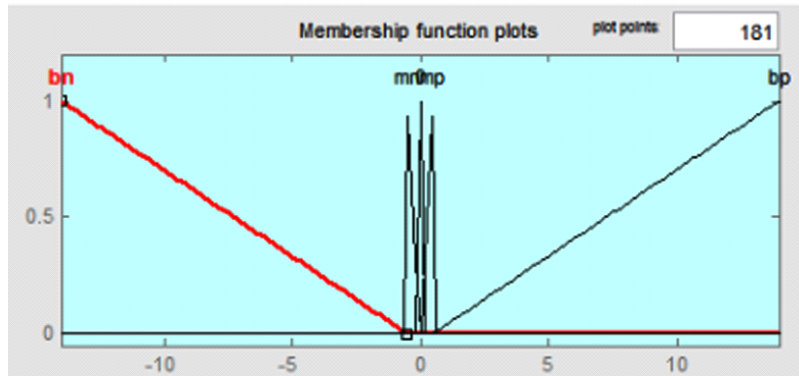


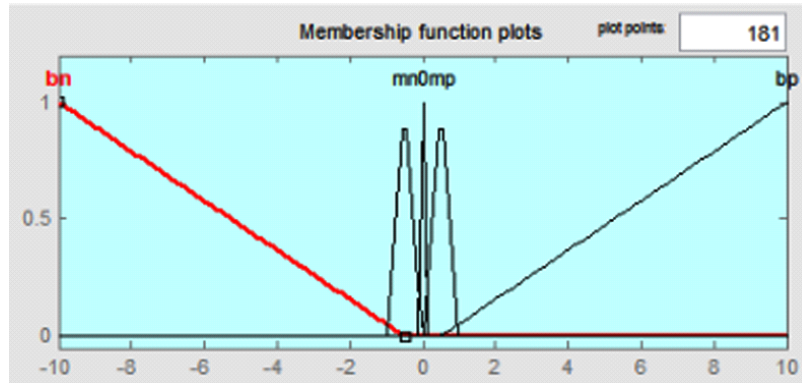
Рис. 4. Структурная схема контура управления температуры



а



б



в

Рис. 5. Структура нечеткого регулятора  
а — входной терм; б, в — выходные термы коэффициентов  $K_p$  и  $K_i$

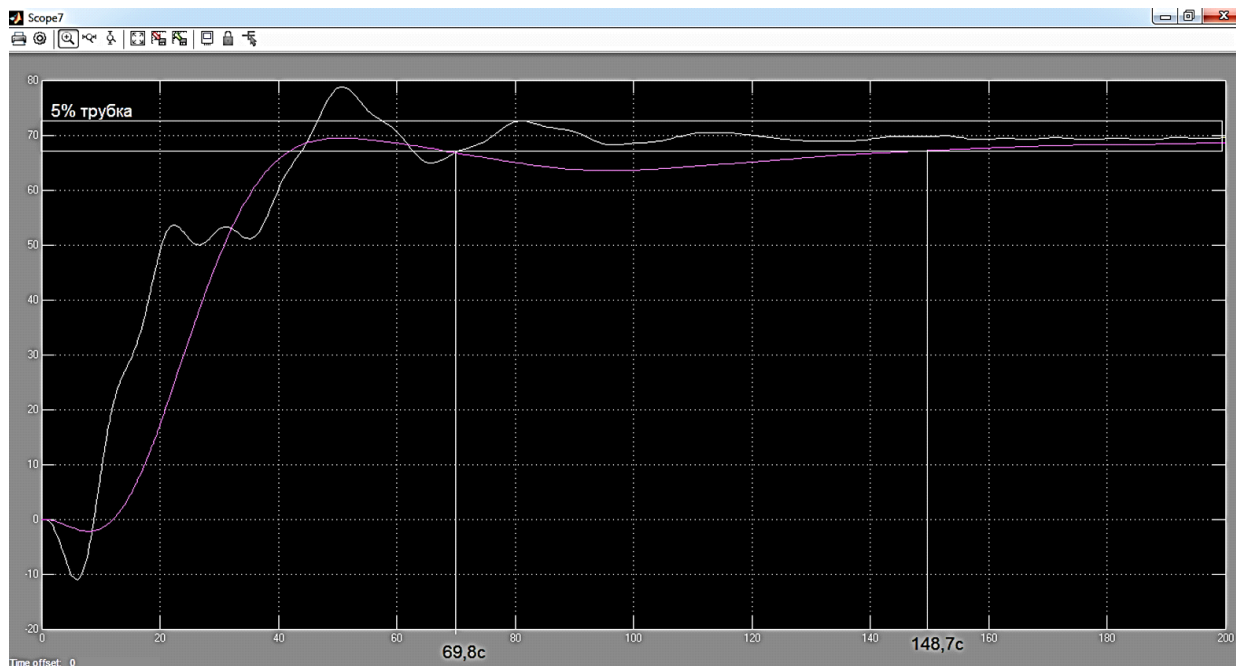


Рис. 6. Переходная характеристика САУ температуры

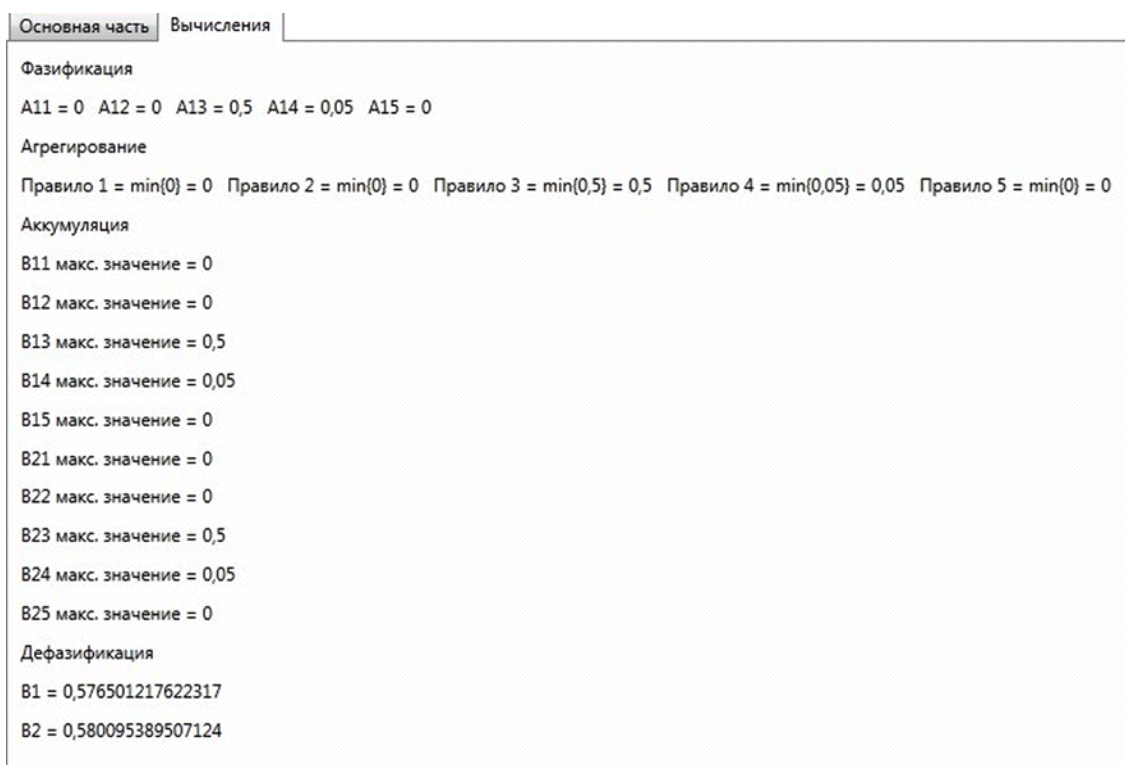


Рис. 7. Интерфейс программы «Нечеткая логика»

ние,  $tr$  — малое положительное значение,  $br$  — большое положительное значение.

Ниже описаны одна входная и две выходные переменные, для каждой выбраны пять треугольных термов (рис. 5).

Итоговую оценку качества настроенной системы производим по графику переходного процесса (рис. 6). Для полученного переходного процесса в САУ величина перерегулирования равна 11 %. Время регулирования процесса равно 69,8 с.

Для внедрения полученных правил необходимо разработать ПО, реализующее принципы нечеткой логики, в АСУТП Centum VP фирмы Yokogawa [8].

На рис. 7 представлена программа «Нечеткая логика», написанная на языке C++. Данную программу можно внедрить как модуль в АСУТП. В данный момент проводятся работы по связи ПО с математической моделью в HYSYS.

**Обсуждения результатов.** Внедрение рассмотренной системы позволит снизить стандартное отклонение ключевых производственных параметров за счет подстройки ПИД-регулятора, что позволит установке работать ближе к технологическим ограничениям, сохраняя при этом исходную приемлемую вероятность количества нарушений технологических параметров (либо их отсутствие), также

Таблица 2  
База правил  
нечетного регулятора

$\Delta T$	$Ki$	$Kp$
$Vn$	$bp$	$Vp$
$Mn$	$mp$	$Mp$
0	0	0
$Mp$	$mn$	$Mn$
$Vp$	$bn$	$bp$

позволит получить дополнительную экономическую прибыль за счет работы в предельном режиме.

**Выводы и заключения.** Результатом работы является ПО для решений проблем, связанных с управлением технологическим процессом в колонне деизобутанизации на установке сернокислотного алкилирования. ПО позволяет учитывать нелинейности и высокую инерционность процесса для подстройки системы управления, что позволяет увеличить точность управления и повысить ее адаптивность к возмущениям. Перевод в автоматический режим также снизит нагрузку на оператора. Однако перед внедрением прежде всего необходимо произвести испытание ПО непосредственно на математической модели объекта управления в Aspen Hysys. Работы в данном направлении ведутся. Однако результаты модели могут отличаться от реального процесса, поэтому перед внедрением необходимо произвести предварительные испытания на установке для корректировки базы правил.

#### Библиографический список

1. Новиков С. И., Шахнович В. Р., Сафронов А. В. Методы нечеткой логики в задачах автоматизации тепловых процессов

электростанций // Вестник Ивановского гос. энергет. ун-та имени В. И. Ленина. 2010. Вып. 4. С. 72–75.

2. Борисов В. В., Круглов В. В., Федулов А. С. Нечеткие модели и сети. М.: Горячая линия – Телеком, 2007. 284 с.

3. Чернецкая И. В., Чернецкий В. О. Нечеткие регуляторы в системах автоматического регулирования // Вестник Южно-Уральского гос. ун-та. 2006. Вып. 16. С. 156–159.

4. Shreiber R. Air Flow Control Using Fuzzy Logic // Microchip Technology Inc. Application Note AN600. 1997. 21 p.

5. Stotts L., Kleiner B. H. New developments in fuzzy logic computers // Industrial Management & Data Systems. 1995. Vol. 95. P. 22–27.

6. Tanaka K., Sano M. A robust stabilization problem of fuzzy controller systems and its applications to backing up control of a truck-trailer // IEEE Trans. Fuzzy Systems. 1994. Vol. 2. no 2. P. 119–134.

7. Захаров В. Н., Ульянов С. В. Нечеткие модели интеллектуальных промышленных регуляторов и систем управления. IV. Имитационное моделирование // Известия РАН. Техническая кибернетика. 1994. № 5. С. 169–211.

8. Ротац В. Я. Возможен ли синтез нечетких регуляторов с помощью теории нечетких множеств // Промышленные АСУ и контроллеры. 2004. № 1. С. 33–34.

**ИБАТУЛЛИН Альберт Амирович**, кандидат технических наук, доцент кафедры автоматизации и робототехники.

Адрес для переписки: Ibatullin77@yandex.ru

**ОГУДОВ Андрей Александрович**, магистрант гр. АТПм-151 факультета элитного образования и магистратуры.

Адрес для переписки: Andreyogudov@gmail.com

**ХАКИМОВ Рустам Анварович**, магистрант гр. АТПм-151 факультета элитного образования и магистратуры.

Адрес для переписки: Machinestandard@mail.ru

Статья поступила в редакцию 23.12.2016 г.

© А. А. Ибатуллин, А. А. Огудов, Р. А. Хакимов

## Книжная полка

**Мезенцев, К. Н. Мультиагентное моделирование в среде NetLogo : учеб. пособие / К. Н. Мезенцев. – СПб. : Лань, 2015. – 174 с.**

Учебное пособие может быть использовано для создания имитационных моделей с использованием методов агентного моделирования в таких предметных областях, как биология и социология. Рассмотренные в пособии модели и методы агентного моделирования могут быть использованы при дипломном проектировании и в процессе диссертационного исследования. Пособие предназначено для бакалавров, магистров и аспирантов, обучающихся по направлениям «Информатика и вычислительная техника», «Прикладная информатика», «Программная инженерия», изучающих дисциплины в предметной области «Моделирование систем».

**Олифер, В. Компьютерные сети. Принципы, технологии, протоколы : учеб. / В. Олифер, Н. Олифер. – СПб. : Питер, 2016. – 992 с. – ISBN 978-5-496-01967-5.**

Пятое издание одного из лучших российских учебников по сетевым технологиям, переведенного на английский, испанский, португальский и китайский языки, отражает те изменения, которые произошли в области компьютерных сетей за 6 лет, прошедших со времени подготовки предыдущего издания: преодоление локальными и глобальными сетями рубежа скорости в 100 Гбит/с и освоение терабитных скоростей; повышение эффективности и гибкости первичных оптических сетей за счет появления реконфигурируемых мультиплексоров ввода – вывода (ROADM) и применения суперканалов DWDM, работающих на основе гибкого частотного плана; развитие техники виртуализации сетевых функций и услуг, приведшее к распространению облачных сервисов; выход на первый план проблем безопасности. Рекомендовано Министерством образования и науки Российской Федерации в качестве учебного пособия для студентов высших учебных заведений, обучающихся по направлению «Информатика и вычислительная техника» и по специальностям «Вычислительные машины, комплексы, системы и сети», «Автоматизированные машины, комплексы, системы и сети», «Программное обеспечение вычислительной техники и автоматизированных систем».

## АСИНХРОННЫЕ ПРОЦЕССЫ ВЗАИМОДЕЙСТВИЯ КАК ОСНОВА ПОНЯТИЯ ИНФОРМАЦИИ

Формализованное понятие информации оказывается недостаточным для охвата фактически используемого содержания в технических науках с учетом не дискретных процессов управления. Предметом исследования является выработка определения, обобщающего дискретный и аналоговый подход и основанного на асинхронных воздействиях на управляемую систему, связывая локальные свойства информации и энтропийные оценки множественных взаимодействий. Вместо сообщения как конечной последовательности знаков, предложено рассматривать более широкое понятие прообраза кусочного отображение динамики воздействия. В качестве модели нестационарного потока сообщений используется асинхронные по характеру взаимодействия, изучавшиеся в теории последовательных взаимодействующих процессов.

**Ключевые слова:** информация, сообщение, управление, воздействие, взаимодействие, асинхронные процессы.

**1. Введение.** Несмотря на очень широкое использование термина *информация*, определение стоящего за ним понятия до сих пор вызывает серьезные проблемы. Существует гуманитарное представление об информации как о *сведениях, сообщениях, знаниях*, но все использованные для этого разъясняющие слова имеют очень широкое и, в общем случае, нечеткое толкование. К современному моменту определения информации включены в ряд мировых стандартов и законов. Правовое значение термина дает Российский закон № 149-ФЗ «Информация (от лат. Informatio — «разъяснение, изложение, осведомленность») — это сведения (сообщения, данные), независимо от формы их представления». Международный стандарт ISO/IEC 10746-2 от 1996 года определяет информацию как знания о предметах, фактах, идеях и т.п., которыми могут обмениваться люди в рамках конкретного контекста.

Интерес к систематическому использованию понятия информации в естественных науках с опорой на математические представления возник после работы К. Шеннона по математической теории связи [1]. Предложенное им определение количества информации как научной и технической меры возможности коммуникационных каналов по передаче сообщений стало опорной точкой этого направления.

Терминологическое связывание *информации* с *данными*, обрабатываемым компьютерами, и с любыми данными, передаваемыми, хранимыми и представляемыми для человеческого использования в современных устройствах практически смешало границы указанных понятий.

**2. Обзор различных подходов к информации.** Уточнения потребовались для сложившейся как науки информатики. В начале 70-х годов польский ученый М. Мазур сделал попытку систематического построения теории информации, названную им «качественная теория информации» [2]. Он исходил из своего понимания термина сообщения, определяя его как «физическое состояние, определенным образом отличающееся от других физических состояний в цепи управления» [3, с. 33]. При этом указывалось,

что «воздействие в цепи управления состоит из определенного числа сообщений». Далее он вводил промежуточные понятия информационной ассоциации и поперечного множества сообщений. Информационная ассоциация по Мазуру — это «пара сообщений из поперечного множества сообщений». Поперечное множество сообщений — множество сообщений в произвольном месте цепи управления. В качестве завершения определения он утверждал, что «информация — преобразование одного сообщения информационной ассоциации в другое сообщение той же ассоциации». Все это позволяло провести последовательный логический анализ информационных процессов как процессов преобразования сигналов в контурах управления с обратной связью и вести логическую классификацию различных видов информирования с учетом семантических аспектов передаваемой информации. К сожалению, за формальными границами построенной таким образом теории не были найдены приложения использованных подходов и их приложения для смежных областей применения. Более того, из предложенного М. Мазуром подхода логически следовало, что любое физическое состояние, сколько-нибудь отличное от другого соседнего во времени или пространстве состояния, порождает некоторую информацию как формальное преобразование между ними. Этот подход вел к утверждению, что все физического пространство заполнено информацией, вне зависимости от конкретных специализированных систем хранения и преобразования информации. Условие наличия цепи управления в исходных определениях несколько не мешало подобным неестественным заключениям, поскольку рассмотрение именно «физических состояний» приводит к тому, что действие того же закона всемирного тяготения ведет к неизбежным «управляющим» воздействиям любой материальной точки на соседние, и даже на более удаленные материальные точки.

Несколько позже Ф. Бауэр предложил определение информации, опираясь на более простое понятие *сообщения*, которое представляет собой цепочку



символов из некоторого алфавита [3]. Его подход состоит в указании связи между сообщениями и информацией, которая записывается в виде отображения  $N \rightarrow J$ , где  $N$  — сообщение, а  $J$  — информация, причем правило соответствия называют *интерпретацией* сообщения. Последнее определение международного стандарта использует именно эту точку зрения, декларируя, что хотя информация должна использовать некоторую форму *представления*, превращаясь при этом в *данные*, чтобы ею можно было технически обмениваться, но все же информация есть в первую очередь *интерпретация* такого представления (стандарт ISO/IEC/IEEE 24765: 2010 Systems and software engineering). Поэтому в строгом смысле информация отличается от данных, хотя в неформальном контексте эти два понятия чаще всего используются как синонимы.

Техническое понимание информации оказалось тесно связанным с основными определениями кибернетики. Это междисциплинарное направление науки было описано Н. Винером, как «наука об управлении и связи в машинах и живых организмах» (*Cybernetics: or, Control and Communication in the Animal and the Machine*) [4]. Внутри же текста книги было дано определение кибернетики через информацию, а именно она описывалась как наука «об общих закономерностях получения, хранения, передачи и преобразования информации в сложных управляющих системах, будь то машины, живые организмы или общество». В другом месте Винер также писал «*Информация — это обозначение содержания, полученное нами из внешнего мира в процессе приспособления к нему нас и наших чувств*».

Последнее замечание, несмотря на отсутствие в первоисточнике развертывания из него последующих выводов, особенно интересно тем, что привлекает внимание к существенной стороне информации, а именно ее глубинной генетической связи с окружающим миром.

**3. Анализ проблем в общетехническом понятии информации.** Чтобы показать недостаточность сложившегося к настоящему времени понятия «информационных сообщений», как некоторых последовательностей или комбинаций знаков или физических состояний, рассмотрим исторически первую управляющую систему, называемую *центробежным регулятором*. Это устройство было приспособлено Дж. Уаттом для регулирования устойчивой скорости вращения паровой машины. Он представлял собой достаточно примитивную механическую конструкцию с массивными шариками на раздвигаемом параллелограмме из реек, закрепленную и вращающуюся от вала паровой машины. К раздвижной части этой конструкции было приделан механический стержень, одним концом закрепленный в выемке муфты, а другим соединенный с тягой привода дроссельной заслонки в трубе, подающей пар к рабочему цилиндру машины.

Это устройство оказалось действительно эффективным регулятором, широко использовавшимся с тех пор. Его теория исчерпывающе была описана значительно позже русским механиком и математиком И. А. Вышеградским в связи с необходимостью его усовершенствования, связанной с выявленной неустойчивостью регулирования в более мощных паровых машинах второй половины XIX века [5].

С учетом современных возможностей механический стержень, передающий обратное управляющее воздействие на вход, может быть заменен информационным каналом, использующим электрическую

или оптическую реализацию (электрический или оптический кабель), по которому передаются кодированные данные о текущей скорости вращения вала. При этом в систему нужно встраивать механико-электронный преобразователь линейного смещения в цифровой код, обратный преобразователь цифрового кода в линейное смещение, использовать усилители для передачи сигналов и реализацию исполнительного механизма, который под воздействием поступающего на его вход цифрового кода выполняет перемещение заслонки.

Использование такой конструкции центробежного регулятора «со встроенной электроникой и цифровым кодированием» в составе паровой машины превращает рассматриваемую машину в кибернетическую систему в строгом определении «по Винеру». Но, с прагматической точки зрения и функциональности, замена сложным и менее надежным оборудованием простейшего механического стержня не дает ничего, точнее, обеспечивает ту же функциональность, но путем огромного усложнения конструкции.

Описание же поведения указанного механического стержня, как реального механизма обратной связи, не позволяет сколько-нибудь явно вводить дискретные сообщения, «передаваемые им». Таким образом, теоретическая привязка определения информации к явно выделяемым и дискретным по строению сообщениям оказывается не всегда применимой.

В работе [6] предлагалось расширенное понимание информации как системы специфических топологических инвариантов отображаемого мира, что позволяло охватить и существенные особенности аналоговых технических систем управления, но указанное расширение было направлено в первую очередь на информацию в естественных системах и ориентировано в большей степени на сенсорные образы внешних воздействий.

Перефразируя в качестве полумеры определение Винера, можно сказать, что информация — это *что-то* получаемое из внешнего мира в процессе приспособления к нему. Тогда можно предложить следующее определение. «Информация — это кусочное отображение динамики окружающего мира во внутреннее состояние управляющей системы». Понятие внешнего мира достаточно общее и несколько расплывчато для технических применений, поэтому желательно его несколько сузить, оттеняя самое главное для нашей темы. Термин *мир* в этом определении целесообразно сузить до совокупности *асинхронно взаимодействующих процессов* [7]. Причем отображение не просто во что-нибудь, а именно *внутрь управляющей системы*. Просто «система», не являющаяся управляющей, не может быть сколько-нибудь автономной частью динамического мира. Такая система либо формальная (продукт отображения или даже воображения некоторой управляющей системы — человека, социума, но не является отображением динамики, а только представляет собой формальную статическую систему), либо она подчиненная собственная часть управляющей системы и не является взаимодействующей с окружающим миром, непосредственным окружением для такой части служит согласованный — синхронный — мир внутреннего времени управляющей системы.

Окончательно уточненное определение будет гласить: «Информация — это кусочное отображение динамики воздействий *асинхронно взаимодействующих процессов* во внутренние состояния управляющей системы в совокупности с динамикой обратных воздействий от системы».

Чтобы яснее осознать это определение, нужно принимать во внимание, что кибернетическая управляющая система управляет объектом управления с учетом *посторонних воздействий* на этот объект. Жесткая функциональность воздействия внутренних механизмов для задания детерминированного поведения теоретически возможна, но тогда имеет место чисто функциональный автомат, поведение которого никак не изменяется под влиянием каких-то внешних воздействий, а определяется только внутренними закономерностями. Установление зависимости, что внешние воздействия на кибернетическую систему необходимо соотносятся с ее внутренним состоянием, были сделаны У. Р. Эшби [8].

**4. Существенность асинхронного характера взаимодействия.** Новая часть переопределения общего понятия информации заключается в учете именно *динамики* внешних воздействий и внимании к *асинхронному* характеру этих воздействий. Дело в том, что реальное поведение управляемой системы вынуждено хотя бы косвенно отражать временные последовательности воздействий, а не просто их формализованную статику, причем никаких оснований для безусловного оценивания интервалов между такими воздействиями и момента их появления в общем случае быть не может. Это не значит, что в частных случаях подобные проявления не могут иметь место. С общих позиций вполне допустимо синхронные взаимодействия рассматривать как частный случай асинхронных, нельзя только постулировать обратное утверждение, пытаясь рассматривать асинхронные взаимодействия с помощью средств и методов, предназначенных для учета и использования синхронных. Методологически ситуация здесь подобна изучавшейся Э. Дейкстрой в связи с проблемой мультипрограммирования. Им указано, что «многие проблемы мультипрограммирования, ставившие нас прежде в тупик, являются всего лишь следствием априорной тенденции придавать детерминированности слишком большое значение; в конце концов, я пришел к тому, что стал считать недетерминированность естественной ситуацией, при этом детерминированность свелась к довольно банальному частному случаю» [9, с. 11].

Асинхронность внешних воздействий относительно управляющей системы приводит к тому, что отображение динамики этих воздействий получается только кусочным и отражающим взаимодействия этого одного или нескольких одновременных внешних воздействий, которые в таком случае оказываются синхронными в этом рассматриваемом взаимодействии друг с другом, но все же асинхронными относительно управляемой системы. Эти кусочные отображения динамики принципиально не могут быть расширены до всего общего процесса взаимодействия управляющей системы с внешними источниками воздействий, так как в противном случае существование такого предельно расширенного отображения дает теоретическую возможность синхронизации всех этих воздействий на основе внешней динамики данной управляющей системы.

Вынужденная кусочность отмеченных отображений влечет дискретизацию формализации этих отдельных отображений, их принципиальную отделимость друг от друга и возможность использования конечных систем для представления образов таких отображений внутри управляющей системы. Вместо упрощенного понятия сообщения как конечной последовательности знаков из некоторого алфавита, следует рассматривать более широкое понятие «про-

образа кусочного отображения динамики воздействия».

Указанный в определении информации фрагмент «в совокупности с динамикой обратных воздействий от системы» представляет в нем то, что, иными словами, называют *интерпретацией* информации. Действительно, даже в обычном смысле интерпретация относится к явлениям и в общем случае процессам внешнего для управляющей системы мира, а не только к внутреннему миру самой системы. Интерпретация команд вычислительного устройств компьютера заключается в реальных действиях, реализуемых в динамике, над текущими данными, а не просто формальные коды этих команд, значимые только по своей статической сущности. Интерпретация формализмов всегда оказывается представлением этих формализмов в частной совокупности более конкретной системы, которая формально оказывается уже внешним миром для исходного формализма.

Вместо упрощенного понятия сообщения следует рассматривать более широкое понятие *прообраза кусочного отображения динамики воздействия*. Для систем и устройств с аналоговыми сигналами воздействия приходится применять более широкое представление о множестве, описывающем асинхронные воздействия на управляющую систему, чем дискретные сообщения. В общем случае нужно рассматривать указанный прообраз как конечную последовательность элементов, которые принадлежат многомерным сегментам непрерывных значений. В частном случае сегмента с совпадающими границами получаем дискретное значение.

Например, программы компьютера при воздействии на электронную кнопку в графической системе управления получают от аппаратуры мыши сигналы о перемещении, нажатии на кнопку. Далее обработчики прерываний формируют коды координат мыши и сравнивают получаемые при этом значения с действующими на текущий момент границами изображения электронной клавиши, изображение на клавише при этом имеет чисто декоративное значение.

**5. Связь информации с энтропией.** Построенное уточненное определение понятия информации дает возможность по-новому понять меру количества информации, введенную К. Шенноном. Для этого достаточно обратить внимание на аналогию в термодинамике. Термодинамическое определение энтропии оценивает степень хаотичности движений в системе механических частиц, но, по существу, анализируемых элементарных составляющих движений глубоко связано с суммарной оценкой динамического множества ударных взаимодействий между этими частицами. Оценивает она, в конечном счете, степень разнообразия этих взаимодействий во всей рассматриваемой системе, для которой вычисляется энтропия [10].

Информация, согласно выведенному определению, для отдельного элементарного взаимодействия между управляющей системой и окружающей ее средой, представляет и фиксирует в некоторых внутренних состояниях управляющей системы отдельный акт этого взаимодействия и реакции во внешнюю среду для него. Переход от этого кусочного представления и отображения к множеству всех взаимодействий для рассматриваемой системы должен учитывать разбиение этого множества на одинаковые элементы (одинаковые по элементарной информации взаимодействия), количество таких одинаковых взаимодействий и их вероятность, абст-

рактно представляющую частоту их проявлений в реальной картине взаимодействий. В результате глубокой аналогии между случайными отдельными взаимодействиями как в термодинамической системе, так в абстрактной совокупности управляющей системы и ее окружения, получаются одинаковые аналитические оценки суммарной численной характеристики, описывающей всю совокупность.

Получается, что энтропийный подход для количества информации от Шеннона изучает и оценивает не информацию, а исключительно некоторые последовательности аргументов обращения к информации. Энтропия устанавливает связь между макро- и микросостояниями термодинамической системы. Аналогичным образом, количество информации устанавливает связь между отдельными взаимодействиями управляющей системы с ее средой и совокупностью всех таких взаимодействий. Существо же и особенности конкретной информации как в отдельных элементах, так и всей совокупности, это количество никак не затрагивает, поскольку описывает только свойства последовательностей во времени поступающих воздействий, но не их результаты и действия отдельных компонентов управляющей системы.

**6. Заключение.** Предложено уточнение понятия информации, охватывающее не только дискретные по элементному составу сообщения, но и воздействия более сложного математического строения. Предложено рассматривать асинхронные воздействия произвольной природы, которые в общем случае могут описываться многомерными сегментами действующих континуальных значений. Показана внутренняя связь рассмотренных конструкций с энтропией в термодинамике. Из исследования вытекает, что информация в глобальных сетях и средах массовой информации не является пассивным передаваемым средством, в любом использовании информация оказывается либо фиксацией во внутренних состояниях управляющей системы самого

факта внешнего воздействия вместе с его возможными атрибутами для последующего использования в обратных воздействиях, либо последовательностью директив для последующих воздействий.

#### Библиографический список

1. Шеннон К. Математическая теория связи // Работы по теории информации и кибернетике. М.: Иностранная литература, 1963. С. 243–322.
2. Мазур М. Качественная теория информации. М.: Мир, 1974. С. 240.
3. Бауэр Ф., Гооз Г. Информатика / пер. с нем. В. К. Сабальфельд. М.: Мир, 1976. 484 с.
4. Винер Н. Кибернетика, или Управление и связь в животном и машине. М.: Советское радио, 1968. 326 с.
5. Максвелл Д. К., Вышнеградский И. А., Стодола А. Теория автоматического регулирования (линеаризованные задачи). М.: Изд-во АН СССР, 1949. 430 с.
6. Флоренсов А. Н. Динамические аспекты информационной компоненты сложных систем: моногр. Омск: изд-во ОмГТУ, 2007. 344 с.
7. Хоар Ч. Взаимодействующие последовательные процессы / пер. с англ. А. А. Бульонковой; под ред. А. П. Ершова. М.: Мир, 1989. 264 с.
8. Эшби У. Р. Введение в кибернетику / под ред. В. А. Успенского. М.: Изд-во иностранной литературы, 1959. 432 с.
9. Дейкстра Э. Дисциплина программирования: пер. с англ. / под ред. Э. З. Любимского. М.: Мир, 1978. 275 с.
10. Стратонович Р. А. Теория информации. М.: Советское радио, 1975. 424 с.

**ФЛОРЕНСОВ Александр Николаевич**, кандидат технических наук, доцент (Россия), доцент кафедры информатики и вычислительной техники.  
Адрес для переписки: lorensov@yandex.ru

Статья поступила в редакцию 09.12.2016 г.

© А. Н. Флоренсов

## Книжная полка

**Керниган, Брайан У. Язык программирования C / Брайан У. Керниган, Деннис М. Ритчи. – 2-е изд. – М.: Вильямс, 2016. – 288 с. – ISBN 978-5-8459-1874-1, 0-13-110362-8.**

Классическая книга по языку C, написанная самими разработчиками этого языка и выдержавшая в США уже 34 переиздания! Книга является как практически исчерпывающим справочником, так и учебным пособием по самому распространенному языку программирования. Предлагаемое второе издание книги было существенно переработано по сравнению с первым в связи с появлением стандарта ANSI C, для которого она частично послужила основой. Книга не рекомендуется для чтения новичкам; для своего изучения она требует знания основ программирования и вычислительной техники. Книга предназначена для широкого круга программистов и компьютерных специалистов. Может использоваться как учебное пособие для вузов.

**Новожилов, О. П. Архитектура ЭВМ и систем : учеб. пособие / О. П. Новожилов. – М.: Юрайт, 2016. – 528 с. – ISBN 978-5-9916-6882-8.**

Учебное пособие посвящено архитектуре современных компьютеров. Рассмотрен широкий круг вопросов, касающихся структурно-функциональной организации микропроцессоров, микропроцессорных систем, компьютеров и компьютерных систем, основной компьютерной памяти и дисковых накопителей, а также различных типов периферийных устройств. Соответствует актуальным требованиям Федерального государственного образовательного стандарта высшего образования. Материал излагается с учетом двухступенчатой подготовки специалистов (бакалавр — магистр) и может быть полезен аспирантам и преподавателям других специальностей.

# РАСЧЕТ ЧИСЛА МОТИВОВ НА ТРЕХ УЗЛАХ МЕТОДОМ СЛУЧАЙНОЙ ВЫБОРКИ КАРКАСОВ В СЕТЯХ С НАПРАВЛЕННЫМИ СВЯЗЯМИ

Задача разработки эффективных алгоритмов для расчета частот встречаемости неизоморфных связных подсетей (мотивов) на заданном количестве узлов является актуальной задачей теории сетей. Комбинаторно-логический характер этой задачи обуславливает большие затраты времени и / или оперативной памяти при расчете сетей, содержащих сотни тысяч узлов. В предлагаемой статье для решения данной задачи развивается основанный на статистическом подходе метод случайной выборки каркасов и разрабатывается алгоритм для расчета встречаемости 3-мотивов в сетях с направленными связями. Предлагается реализация этого алгоритма с использованием параллельных вычислений. Приводятся результаты численных экспериментов. При сравнении разработанного алгоритма с другими известными алгоритмами в ряде случаев выявляются его значительные преимущества по точности, быстродействию и затратам оперативной памяти.

**Ключевые слова:** сетевые мотивы, встречаемость мотивов, статистическое моделирование.

*Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 16-31-60023 мол\_а\_дк.*

**1. Введение.** Расчет частот встречаемости различных неизоморфных подсетей из заданного их набора — сетевых мотивов — является одной из актуальных задач науки о сетях (Network Science). При этом особый интерес представляют не все возможные мотивы [1], а только те из них, которые в реальной сети встречаются чаще, чем в ее рандомизированной версии, в которой ребра перераспределены случайно. После выявления сетевых мотивов (как правило, тех, которые появляются часто) задача исследователей-прикладников заключается в том, чтобы понять, какую функцию выполняют эти мотивы, какую роль играют в сети, почему они часто встречаются. В частности, в [2] при анализе геновых сетей мотив 11 на рис. 1 рассматривается как «петля прямой связи» (ген-мишень регулируется двумя другими генами-источниками, одним напрямую, а другим опосредованно). В большинстве случаев назначение этого мотива исследователи определяют как «блок И» (для активации мишени нужна активация обоих генов-источников), либо «блок ИЛИ» (для активации мишени достаточно активации одного из генов-источников). Возможна постановка другой задачи: на основе анализа частот встречаемости мотивов нужно сделать вывод о качестве графовой модели сети. В частности, даже при совпадении распределения степени связности дуг графовой модели и распределения степени связности связей исследуемой сети [3], графовая модель может отличаться от моделируемой сети по частотам встречаемости мотивов. Таким образом, задача расчета частот встречаемости мотивов является составной частью задачи структурной идентификации сетей. Важность расчета час-

тот встречаемости мотивов и сложность задачи подтверждается многочисленными публикациями, в которых предлагаются различные комбинаторные методы [4–6], сокращающие перебор, и методы распределения вычислений [7, 8].

Многие существующие подходы к расчету частот встречаемости мотивов ориентированы на работу с сетями, размер которых не может превышать десятков тысяч узлов и связей, что объясняется тем, что первоначально мотивы получили широкое распространение в биоинформатике, где размеры сетей относительно невелики. Однако все чаще концепция мотивов используется в исследовании социальных и других больших сетей, что выдвигает новые требования к применяемым подходам.

В табл. 1 представлены характеристики сетей, а в табл. 2 — время, которое затрачивается на расчет частот встречаемости мотивов программами MFinder, Fanmod и AccMotif. Программа MFinder — первая известная программа для расчета частот встречаемости мотивов, которая осуществляет полный перебор всех возможных подграфов. Популярная программа Fanmod реализует алгоритм, позволяющий ускорить перебор с помощью алгоритма ESU [4]. Программа AccMotif позволяет существенно ускорить расчет. В качестве тестового стенда в вычислениях данной статьи выступал моноблок HPZ1 G2 Workstation, имеющий 4-ядерный центральный процессор Intel Xeon E3-1245 3,3 ГГц с поддержкой технологии Hyper-Threading и 8 ГБ оперативной памяти стандарта DDR3-1333. Данные о сетях были получены из баз данных М. Ньюмена [9] и Ю. Лесковца [10].

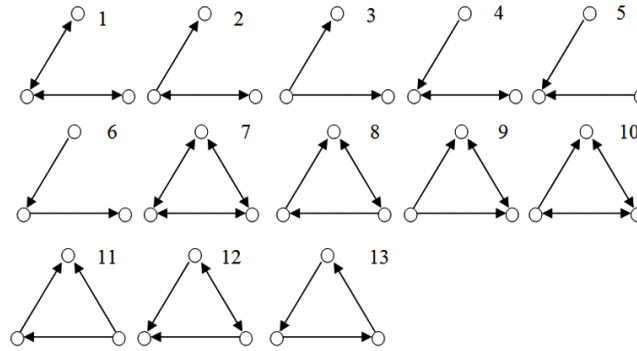


Рис. 1. Возможные мотивы на трех узлах

Характеристики исследуемых сетей

Таблица 1

Сеть	Тип сети	Число узлов	Число связей
Internet [9]	неориентированная	22963	48436
Email-Enron [10]	неориентированная	36692	183831
Political Blogs [9]	ориентированная	2224	18956
Twitter [10]	ориентированная	81306	1768149
Email-EuAll [10]	ориентированная	265214	420045
G+ [10]	ориентированная	107614	13673453

Время расчета частот встречаемости мотивов на трех и четырех узлах с использованием программ MFinder, AccMotif, FanMod

Таблица 2

Сеть	MFinder		AccMotif		FanMod	
	Число узлов в мотиве		Число узлов в мотиве		Число узлов в мотиве	
	3	4	3	4	3	4
Internet	670 с	>10 <sup>4</sup> с	0,0084 с	3,96 с	6,846 с	8134,78 с
Email-Enron	1015 с	>10 <sup>4</sup> с	0,31 с	33,1 с	17,951 с	10097 с
Political Blogs	10,0 с	5364 с	0,025 с	2,35 с	0,846 с	144 с
Twitter	>10 <sup>4</sup> с	>10 <sup>4</sup> с	невозможно		295 с	>10 <sup>4</sup> с
Email-EuAll	>10 <sup>4</sup> с	>10 <sup>4</sup> с	невозможно		102 с	>10 <sup>4</sup> с
G+	>10 <sup>4</sup> с	>10 <sup>4</sup> с	невозможно		>10 <sup>4</sup> с	

Как можно видеть из табл. 2, программа AccMotif имеет ощутимое преимущество по скорости расчета. Однако ускорение расчета достигается за счет использования больших объемов оперативной памяти, размеры которой в современных компьютерах ограничены. Так, при расчете частот встречаемости мотивов сети Email-Enron, содержащей 36692 узла, программа AccMotif использует 1,8 ГБ, а при расчете сети Gnutella, содержащей 62586 узла, — 4 ГБ. Не менее требовательным к размеру оперативной памяти является алгоритм G-Tree, предложенный в работе [11]. Практически это означает, что на используемом стенде, поддерживаемом всего 8 ГБ оперативной памяти, подобного рода алгоритмы не могут работать с сетями, содержащими уже сотни тысяч узлов.

Для больших сетей необходимо использовать большое количество оперативной памяти (что все равно имеет свои пределы, ведь использование оперативной памяти в алгоритмах AccMotif [12] приблизительно пропорционально квадрату числа узлов сети). Альтернативой этому является разработка ме-

тодов статистической оценки частот встречаемости мотивов.

В статистических подходах часто используют не абсолютные частоты мотивов, а относительные:

$$c_k^i \stackrel{\text{def}}{=} \frac{S_k^i}{\sum_j S_k^j},$$

где  $S_k^i$  — абсолютная частота встречаемости  $i$ -го мотива размера  $k$ ;

$c_k^i$  — относительная частота  $i$ -го мотива размера  $k$ .

На сегодняшний день известно 3 алгоритма, дающих статистическую оценку частот встречаемости мотивов.

1. Алгоритм случайного выбора ребра [13], реализованный в программе MFinder. Суть этого алгоритма заключается в том, что в сети равновероятно выбирается связь, далее случайно выбираются две инцидентные ее концевым узлам другие связи, и тем самым всего выбираются три или четыре узла. Эти узлы с учетом всех имеющихся связей между ними

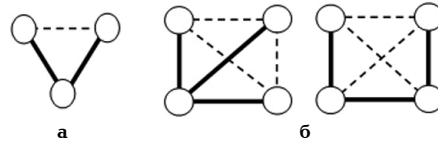


Рис. 2. Каркасы: а — для 3-мотивов, б — для 4 мотивов.  
Жирной линией обозначены связи, входящие в каркас;  
пунктирной — возможные связи в найденном на данном каркасе мотиве

```

Input: Graph (V, E), N
Output: motif3_1, motif3_2, motif3_3, motif3_4, motif3_5, motif3_6, motif3_7, motif3_8,
motif3_9, motif3_10, motif3_11, motif3_12, motif3_13

A1 ← ∅, A2 ← ∅, ...
count_S1 ← 0, ..., count_S13 ← 0, Nv ← 0
1. foreach v ∈ V do
    k ← |Adj(v)|
    Nv ← Nv + k × (k - 1) / 2
    Add v to Ak
end
2. foreach Ak do Pk ← |Ak| × k × (k - 1) / 2 / Nv end
3. for i = 1, N do parallel
    Generate r ~ {Pk}
    v ← get from Ar, random
    v1, v2 ← get from Adj(v) random (v2 ≠ v1)
    if {v, v1, v2} ∈ S4 count_S1 ++; if {v, v1, v2} ∈ S2 count_S2 ++;
    if {v, v1, v2} ∈ S5 count_S3 ++; if {v, v1, v2} ∈ S4 count_S4 ++;
    if {v, v1, v2} ∈ S6 count_S5 ++; if {v, v1, v2} ∈ S6 count_S6 ++;
    if {v, v1, v2} ∈ S7 count_S7 ++; if {v, v1, v2} ∈ S8 count_S8 ++;
    if {v, v1, v2} ∈ S9 count_S9 ++; if {v, v1, v2} ∈ S10 count_S10 ++;
    if {v, v1, v2} ∈ S11 count_S11 ++; if {v, v1, v2} ∈ S12 count_S12 ++;
    if {v, v1, v2} ∈ S13 count_S13 ++;
end do
4. motif3_1 ← count_S4 / N × Nv, motif3_2 ← count_S2 / N × Nv, motif3_3 ← count_S3 / N × Nv,
motif3_4 ← count_S4 / N × Nv, motif3_5 ← count_S5 / N × Nv, motif3_6 ← count_S6 / N × Nv,
motif3_7 ← count_S7 / N / 3 × Nv, motif3_8 ← count_S8 / N / 3 × Nv, motif3_9 ← count_S9 / N / 3 × Nv,
motif3_10 ← count_S10 / N / 3 × Nv, motif3_11 ← count_S11 / N / 3 × Nv,
motif3_12 ← count_S12 / N / 3 × Nv, motif3_13 ← count_S13 / N / 3 × Nv
5. return motif3_1, ..., motif3_13

```

Рис. 3. Алгоритм расчета частот встречаемости мотивов на трех узлах

задают найденный мотив. Счетчик частоты встречаемости этого мотива увеличивается на единицу. Далее описанная операция повторяется заданное количество раз. Однако данный алгоритм дает смещенную оценку частот встречаемости мотивов. Для компенсации смещения можно применять ресурсоемкую процедуру вычисления поправочных коэффициентов [13].

2. Алгоритм RandESU [4], реализованный в программе FanMod. Суть алгоритма, как и алгоритма ESU, заключается в направленном переборе заданного размера  $k$  подсетей в анализируемой сети, в результате этого перебора строится дерево, листьями которого являются имеющиеся в сети  $k$ -мотивы. В работе [4] доказывается, что, регулируя параметр процедуры расчета частот встречаемости мотивов — вектор  $\langle \alpha \rangle$ , задающий вероятности отсечения ветвей дерева, можно уменьшить число проверяемых подсетей, получая при этом несмещенные оценки для относительных частот мотивов. Но контролировать точность вычислений при использовании этого алгоритма можно только косвенно.

3. Метод случайной выборки каркасов [14] заключается в многократно реализуемом процессе равновероятного выбора каркаса — связной подсети на трех (четырех) узлах, которая входит во все или несколько возможных мотивов на трех (четырех) узлах. Для мотивов на трех узлах (3-мотивах) такой каркас один и представляет собой  $v$ -образный под-

граф, для 4-мотивов в методе случайной выборки каркасов используется 2 подграфа:  $u$ -образный и  $w$ -образный подграфы, изображенные на рис. 2. Для 3-мотивов выбор каркаса можно реализовать посредством выбора случайного узла, вероятность выбора которого пропорциональна числу возможных путей длины два ( $v$ -образного подграфа), центром которых является данный узел. После чего каркас выбирается путем равновероятного выбора двух инцидентных выбранному узлу связей. Среди недостатков можно отметить, что предложенный метод был реализован только для неориентированных сетей.

2. **Постановка задачи.** В данной работе развивается метод случайной выборки каркасов (МВК). Алгоритм, предложенный в статье, расширяет возможности использования МВК, позволяя рассчитывать число 3-мотивов в сетях с направленными связями, а также использует преимущества параллельного программирования. Результатом работы алгоритма являются оценки относительных (абсолютных) частот встречаемости мотивов на трех узлах (рис. 1).

3. **Расчет частот встречаемости 3-мотивов.** В предлагаемом алгоритме расчета частот встречаемости мотивов (см. рис. 3) узлы сети разбиваются на подмножества  $A_i$ , в каждом из которых содержатся узлы с одинаковой степенью связности  $i$  (и, соответственно, одинаковым числом путей длины два, центром которого является этот узел — каркасов для 3-мотивов). Общее число всех каркасов сохраняется

Результаты расчета относительной частоты встречаемости 3-мотивов с использованием MBK,  $N = 100\ 000$ 

Номер мотива	PolBlogs		Twitter		Email-EuAll		G +	
	точное	MBK	точное	MBK	точное	MBK	точное	MBK
1	0,03203	0,03147	0,05857	0,05787	0,02707	0,02739	0,01255	0,01229
2	0,08363	0,08292	0,09035	0,09197	0,02405	0,02382	0,07607	0,07636
3	0,14708	0,14585	0,12603	0,12483	0,01458	0,01490	0,22829	0,22878
4	0,10675	0,10637	0,11626	0,11768	0,12026	0,11883	0,05114	0,05107
5	0,42153	0,42393	0,45847	0,45716	0,75790	0,75835	0,45193	0,45124
6	0,12045	0,11967	0,10051	0,10077	0,05477	0,05542	0,12297	0,12357
7	0,00265	0,00271	0,00727	0,00723	0,00025	0,00023	0,00144	0,00150
8	0,00367	0,00375	0,00220	0,00216	0,00008	0,00004	0,00110	0,00107
9	0,01510	0,01520	0,00721	0,00728	0,00033	0,00033	0,00498	0,00477
10	0,00946	0,00986	0,00976	0,00987	0,00036	0,00034	0,00326	0,00332
11	0,04294	0,04355	0,01367	0,01357	0,00023	0,00023	0,03378	0,03353
12	0,01429	0,01430	0,00935	0,00927	0,00012	0,00010	0,01237	0,01238
13	0,00042	0,00042	0,00035	0,00034	0,00001	0,00001	0,00011	0,00011

Таблица 4

Время расчета частот встречаемости мотивов на трех узлах, полученное с помощью точного алгоритма FANMOD, RandESU и алгоритма на основе MBK,  $N = 100\ 000$ 

Сеть	FANMOD ESU, с	RandESU $\langle \alpha \rangle = \langle 0,5; 0,5; 0,5 \rangle$ , с	Время работы алгоритма, использующего MBK, с			
			1 поток	2 потока	4 потока	8 потоков
Political Blogs	0,846	0,118	2,246	1,393	1,044	0,881
Twitter	294,971	43,22	8,535	5,104	3,716	3,221
Email-EuAll	102,434	12,67	33,930	16,882	12,015	7,479
G +	2025698	108074	101,378	61,589	35,126	28,318

в переменной  $N_v$ . После этого, с учетом возможностей распараллеливания, выполняется  $N$  реализаций случайного выбора каркаса и устанавливается, какой из 13 возможных мотивов (см. рис. 1) найден на выбранных на каждой итерации трех узлах. После выполнения всех итераций относительная частота встречаемости  $i$ -го мотива  $C_k^i$  определяется отношением числа  $motif3_i$  успешных обнаружений  $i$ -го мотива к общему числу опытов  $N$ , что следует также разделить на поправочный коэффициент, основанный на том, сколько каркасов можно найти в мотиве. Для получения оценки абсолютной частоты встречаемости мотивов относительная частота встречаемости домножается на число каркасов в сети —  $N_v$ .

**4. Результаты экспериментов.** Реализация параллельного многопоточного программирования выполнена с использованием платформ Fork-Join Framework и JSR 335: Streams. Результаты расчетов относительной частоты встречаемости мотивов при реализации на трех узлах представлены в табл. 3.

В табл. 4 представлены результаты экспериментов по скорости работы разработанного алгоритма по сравнению с алгоритмом ESU и RandESU программы FANMOD. Для MBK выбрано число опытов  $N = 100\ 000$ . В этом случае достигается большая точность расчета частот встречаемости мотивов при использовании MBK относительно алгоритма RandESU с параметром  $\langle \alpha \rangle = \langle 0,5; 0,5; 0,5 \rangle$ .

Можно видеть, что коэффициент ускорения вычислений растет с ростом сети. Так, для алгоритма

RandESU при расчете частот встречаемости мотивов в сети Twitter коэффициент ускорения равен 13,4, для сети пользователей электронной почты Email-EuAll — 1,7, а для сети пользователей социальной сетью G+ коэффициент ускорения достигает 3800. При этом точность разработанного алгоритма не меньше точности алгоритма RandESU.

В табл. 5 представлены результаты экспериментов по скорости работы разработанного алгоритма MBK при  $N = 25\ 000$ . Точность результатов сопоставима с точностью алгоритма RandESU с параметром алгоритма  $\langle \alpha \rangle = \langle 0,5; 0,5; 0,5 \rangle$ . При этом для сети Email-EuAll алгоритм MBK относительно менее точного алгоритма RandESU работает в 4,6 раза быстрее.

Результаты предварительных экспериментов в табл. 6 показывают также, что MBK может ускорить вычисления при расчете частот встречаемости 4-мотивов. На данный момент реализована лишь версия программы для сетей с ненаправленными связями. Для обнаружения мотивов в сетях с направленными связями необходимо реализовать процедуру определения изоморфных подсетей. Проблема заключается в том, что для случая неориентированных сетей может быть 198 различных 4-мотивов, в то время как в неориентированном случае их всего шесть.

**5. Заключение.** В данной работе предлагается алгоритм расчета частот встречаемости мотивов на трех узлах в сетях с направленными связями с помощью метода случайной выборки каркасов (MBK).

Время расчета частот встречаемости мотивов на трех узлах, полученное с помощью точного алгоритма FANMOD, RandESU и алгоритма на основе MBK,  $N = 100\ 000$

Сеть	FANMOD ESU, с	RandESU <a> = =<0,5; 0,5; 0,5>, с	Время работы алгоритма, использующего MBK, $\bar{v} \leq 0,01$ , с			
			1 поток	2 потока	4 потока	8 потоков
Internet	6,846	0,696	2,70	1,51	1,00	0,81
Email-Enron	4,3	2,96	1,87	0,11	0,74	0,65

Таблица 6

Время расчета частот встречаемости мотивов на четырех узлах, полученное с помощью точного алгоритма FANMOD, RandESU и алгоритма на основе MBK,  $N = 100\ 000$

Сеть	FANMOD ESU, с	RandESU <a> = =<0,5; 0,5; 0,5; 0,5>, с	Время работы алгоритма, использующего MBK, с			
			1 поток	2 потока	4 потока	8 потоков
Internet	8134,78	200,844	18,026	37,456	10,898	8,786
Email-Enron	10097	608,592	14,167	28,572	9,012	7,818

Реализована программа для ускорения расчета с использованием параллельных вычислений. Результаты позволяют говорить о возможностях ускорения расчетов при лучшей точности по сравнению с известным алгоритмом RandESU. Для социальной сети G + + при расчете мотивов на трех узлах скорость работы программы при соизмеримой точности в 3800 раз быстрее, чем в алгоритме RandESU, реализованном в программе Fanmod.

Результаты численных экспериментов показывают, что чем больше исследуемая сеть и суммарное число мотивов в ней, тем большего ускорения удастся достичь при использовании MBK. Поскольку многие исследуемые в Network Science сети содержат десятки миллионов узлов и связей, дальнейшее развитие MBK является актуальным. Одним из направлений такого развития может стать использование распределенных вычислений. Например, можно, как в работе [15], разбить сеть на компоненты и обрабатывать каждый компонент с использованием распределенных вычислений. В этом плане возможно применение технологии MapReduce [15, 16]. Другим направлением развития MBK является разработка алгоритмов для расчета мотивов на четырех узлах в ориентированных сетях. Эта техническая проблема во многих других пакетах решается за счет использования ускоренных методов определения изоморфных подграфов [17, 18].

#### Библиографический список

- Milo R., Shen-Orr S., Itzkovitz S. [et al.]. Network motifs: simple building blocks of complex networks // *Science*. Oct. 2002. Vol. 298 (5594). P. 824–827.
- Mangan S., Alon U. Structure and function of the feed-forward loop network motif // *Proc. Natl. Acad. Sci. U.S.A.* Oct. 2003. 100 (21). P. 11980–5.
- Задорожный В. Н. Уравнения динамики конечных степеней дуг в растущих графах // *Динамика систем, механизмов и машин*. 2016. № 1. Т. 3. С. 327–335.
- Wernicke S., Rasche F. Fanmod: a tool for fast network motif detection. *Bioinformatics* // 2006. Vol. 22. № 9. P. 1152–1153.
- Marcus D., Shavitt Y. Efficient counting of network motifs // *Proceedings of the 2010 IEEE 30th International Conference on Distributed Computing Systems Workshops*, ser. ICDCSW'10. Washington, DC, USA: IEEE Computer Society. 2010. P. 92–98.
- Grochow J. A., Kellis M. Network Motif Discovery Using Sub-graph Enumeration and Symmetry-Breaking (PDF) // *RECOMB*. 2007. P. 92–106. DOI:10.1007/978-3-540-71681-5\_7.

7. Gonzalez J. E., Low Y., Gu H. [et al.]. PowerGraph : Distributed Graph-Parallel Computation on Natural Graphs // 10th USENIX Symposium on Operating Systems Design and Implementation (OSDI). 2012. P. 17–30.

8. Elenberg E. R., Shanmugam K., Borokhovich M., Dimakis A. G. Beyond Triangles: A Distributed Framework for Estimating 3-profiles of Large Graphs // *KDD*. 2015. P. 229–238.

9. Newman M. Network data. URL: <http://www-personal.umich.edu/~mejn/netdata/> (дата обращения: 12.12.2016).

10. Lescovec Yu. Stanford Large Network Dataset Collection. URL: <http://snap.stanford.edu/data/index.html> (дата обращения: 12.12.2016).

11. Ribeiro P, Silva F. G-Tries: an efficient data structure for discovering network motifs // *ACM 25th Symposium On Applied Computing* – Bioinformatics Track. Sierre, Switzerland. 2010. P. 1559–1566.

12. Luis A. Meira, Vinicius R. Mбximo, Blvaro L. Fazenda. Arlindo Flбvio da Conceizro: acc-Motif: Accelerated Network Motif Detection // *IEEE. ACM Trans. Comput. Biology Bioinform.* 2014. Vol. 11 (5). P. 853–862.

13. Kashtan N., Itzkovitz S., Milo R., Alon U. Efficient sampling algorithm for estimating subgraph concentrations and detecting network motifs // *Bioinformatics*. Jul. 2004. Vol. 20. № 11. P. 1746–1758.

14. Юдин Е. Б., Задорожный В. Н. Расчет числа сетевых мотивов методом случайной выборки каркасов // *Омский научный вестник. Сер. Приборы, машины и технологии*. 2015. № 2 (140). С. 208–211.

15. Gonzalez J. E., Low Y., Gu H., [et al.]. PowerGraph : Distributed Graph-Parallel Computation on Natural Graphs // 10th USENIX Symposium on Operating Systems Design and Implementation (OSDI). 2012. P. 17–30.

16. Pagh R., Tsourakakis C. E. Colorful triangle counting and a MapReduce implementation // *Information Processing Letters*. 2012 Mar. 112 (7). P. 277–281.

17. Cordella L. P., Foggia P., Sansone C., Vento M. An improved algorithm for matching large graphs // *Proc. of the 3rd IAPR TC-15 Workshop on Graphbased Representations in Pattern Recognition*. 2001. P. 149–159.

18. Solnon C. All Different-based Filtering for Subgraph Isomorphism // *Artificial Intelligence*. 2010. 174 (12-13). P. 850–864.

**ЮДИН Евгений Борисович**, кандидат технических наук, старший научный сотрудник.

Адрес для переписки: [udinev@asoiu.com](mailto:udinev@asoiu.com)

Статья поступила в редакцию 26.12.2016 г.

© Е. Б. Юдин



## ИНСТРУМЕНТАЛЬНАЯ СРЕДА ФОРМИРОВАНИЯ ВНЕШНИХ КЛЮЧЕЙ НА СХЕМЕ РЕЛЯЦИОННОЙ БАЗЫ ДАННЫХ

В работе рассматривается оригинальное программное обеспечение для автоматизации построения избыточного множества ссылочных ограничений на данные (внешних ключей). Эти ограничения позволяют регламентировать бизнес-правила в использовании информации на предприятии, которая хранится в реляционной базе данных и обслуживается системой управления базами данных. Разработанное программное обеспечение автоматически ищет возможные варианты ссылочных ограничений, оставляя за пользователем право принятия или отклонения этих ограничений. В автоматическом режиме определяются и удаляются избыточные ссылочные ограничения целостности.

**Ключевые слова:** базы данных, ссылочные ограничения целостности, внешние ключи.

*Работа выполнена при финансовой поддержке РФФИ, проект № 15-41-04436-р\_сибирь\_a*

Данные, хранимые в базе данных (БД), должны удовлетворять правилам, которые приняты за основу информационной политики компании [1]. Ограничения целостности на данные являются основой реализации такой политики. Кроме того, они оказывают существенное влияние на процесс интеграции схем нескольких БД при слиянии компаний или расширении сферы их деятельности [2].

Ссылочные ограничения целостности на данные (referential integrity) являются одним из основных видов ограничений в БД, которые позволяют сохранить структурную ее целостность. В большинстве существующих систем управления базами данных (СУБД) поддерживается такой вид ограничений и задаются эти ограничения в виде связей (relationship) на схеме БД. При проектировании схемы БД ссылочные ограничения задаются проектировщиком «вручную» по мере формирования очередных отношений (relation) БД. Однако в настоящее время стали широко использоваться средства автоматизации проектирования схем БД. При этом связи между автоматически сформированными отношениями на схеме БД по-прежнему необходимо формировать либо корректировать вручную. Следствием сказанного является то, что проблема автоматизации построения корректного и избыточного набора ссылочных ограничений целостности является актуальной.

Формирование схемы БД [3, 4] начинается с исследования зависимостей: функциональных, многозначных, соединений и включения. Первые три вида зависимостей используются для проектирования сущностей (отношений) БД, что позволяет реализовать принцип независимости данных и, как следствие, устойчивость проекта БД при последующей модернизации. Последний вид зависимостей (включения) являются теоретической основой для ссылочных ограничений целостности.

Путь  $U = \{A_1, A_2, \dots, A_n\}$  — множество атрибутов, определенных в БД,  $[R_i]$  — множество атрибутов на которых определено отношение  $R_i$ ,  $[R_i] \subseteq U$ ,  $R = (R_1,$

$R_2, \dots, R_k)$  — БД,  $S = \{[R_1], [R_2], \dots, [R_k]\}$  — схема БД, на которой пока еще не определены ссылочные ограничения целостности. Значения атрибутов, по которым связаны отношения, заимствуются из главных отношений (справочников) в подчиненные отношения. Между ними устанавливается связь **1:M** либо **1:1** по направлению от главного отношения к подчиненному. Рассмотрим формальное определение зависимостей включения (inclusion dependencies) [5]:

**Определение 1.** Пусть  $[R_i]$  и  $[R_j]$  — схемы отношений (не обязательно различные),  $V \subseteq [R_i]$  и  $W \subseteq [R_j]$ ,  $|V| = |W|$ , тогда соотношение  $R_i[V] \subseteq R_j[W]$  называется зависимостью включения.

В определении 1  $|V|$  — мощность множества  $V$ ,  $R_i[V] = \pi_V(R_i)$  — проекция отношения  $R_i$  по атрибутам  $V$ . Далее будем предполагать, что условие  $V = W$  является необходимым для установления связи. Такие зависимости включения называются типизированными (typed) [6, 7].

Обозначим:  $PK(R_i)$  или просто  $PK(i)$  — первичный ключ отношения  $R_i$ ;  $L(i, j, X)$  — связь **1:1** либо **1:M** от  $R_i$  к  $R_j$ , установленная по множеству атрибутов  $X$ , где  $R_i$  главное отношение, а  $R_j$  подчиненное отношение;  $L_1(i, j, X)$  — связь **1:1** от  $R_i$  к  $R_j$ ;  $L_M(i, j, X)$  — связь **1:M** от  $R_i$  к  $R_j$ . В отношении  $R_i$  может существовать множество альтернативных первичных ключей и соответствующих им связей, в которых  $R_i$  будет главным или подчиненным.

**Определение 2.** Между отношениями  $R_i$  и  $R_j$  допустима связь  $L_1(i, j, X)$ , если  $X = PK(R_i) = PK(R_j)$  и для любых реализаций  $R_i$  и  $R_j$  выполнено  $\pi_X(R_i) \subseteq \pi_X(R_j)$ .

**Определение 3.** Между отношениями  $R_i$  и  $R_j$  допустима связь  $L_M(i, j, X)$ , если  $PK(R_i) \neq PK(R_j)$  и  $PK(R_i) \subseteq [R_j]$ .

В определении 3 отношение  $R_j$  может содержать неопределенные значения для атрибутов, не принадлежащих первичному ключу. Заметим, что определения 2 и 3 соответствуют типизированным зависимостям включения. Ограничение целостности, задаваемое связью  $L_M(i, j, X)$ , не подразумевает выполнение

условия  $\pi_X(R_j) \subseteq \pi_X(R_i)$ , где  $X = [R_i] \cap [R_j]$ , поскольку атрибуты  $X$ , не принадлежащие  $PK(R_j)$ , могут принимать неопределенные значения, тогда как в  $R_i$  им соответствуют определенные значения. Смысл ограничения в том, что неопределенное значение какого-либо атрибута в  $R_j$  может быть заменено только тем определенным значением, которое есть в  $R_i$ . Поиск связей, соответствующих определениям 2 и 3, достаточно просто алгоритмуется, что позволит выявить большинство ссылочных ограничений целостности в автоматическом режиме.

В работе [8] для корректного решения рассматриваемой проблемы вводится и исследуется понятие ациклических схем БД. Введена интерпретация ациклических схем БД в виде ассоциированных гиперграфов, доказана теорема о цикличности такого гиперграфа. Представлен алгоритм автоматического построения множества всевозможных ссылочных ограничений и далее предложен алгоритм автоматического удаления избыточных ссылочных ограничений целостности. В данной работе рассмотрим практическую реализацию этих алгоритмов в среде MS Access.

Перед началом работы в текущую БД Access необходимо загрузить две формы «Создание\_связей» и «Удаление\_Связей». Форма «Удаление\_Связей» является вспомогательной и предназначена для удаления всех установленных связей между отношениями в текущей БД. Форма «Создание\_связей» является основной и предназначена для формирования неизбыточного и актуального набора связей между отношениями текущей БД.

Внешний вид формы «Создание\_связей» представлен на рис. 1. В настройках внешнего вида формы удалено стандартное поле перехода по записям, поскольку форма предназначена не для работы с конкретным отношением, а для работы со всеми отношениями БД одновременно, и конкретные записи отношений в форме не анализируются.

После загрузки формы «Создание\_связей» необходимо выбрать все входные отношения, на которых уже должны быть заданы первичные и альтернативные ключи. Для этого на форме присутствует элемент управления «Поле со списком». При открытии этого поля появляется список, состоящий из множества отношений исходной БД. Далее должны быть выбраны все отношения, между которыми необходимо построить ссылочные ограничения целостности (связи); каждый раз выбирается только одно отношение из списка.

После того как выбраны все необходимые отношения для составления связей, необходимо нажать элемент управления «Кнопка». На рис. 1 данная кнопка имеет подпись «Закончить выбор таблиц и сформировать связи».

При добавлении отношений в поле для дальнейшего формирования связей возможно возникновение неопределенностей. Эти неопределенности возникают в связи с тем, что подчиненное отношение не может существовать без главного, например, отношение «Расписание» не может существовать без определяющих его отношений «Предметы», «Преподаватели», «Список групп», «Неделя», «Начало занятий». В программе реализованы определения необходимости построения текущей связи, указывающие, какие отношения необходимо добавить в рассмотрение для ее корректного выполнения. Если очередная связь не соответствует семантике системы, то следует нажать кнопку «Отмена» (рис. 2), после чего связь между двумя указанными отношениями не

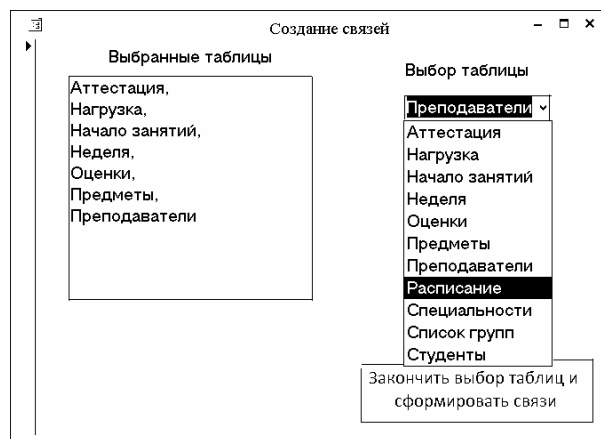


Рис. 1. Форма «Создание связей»

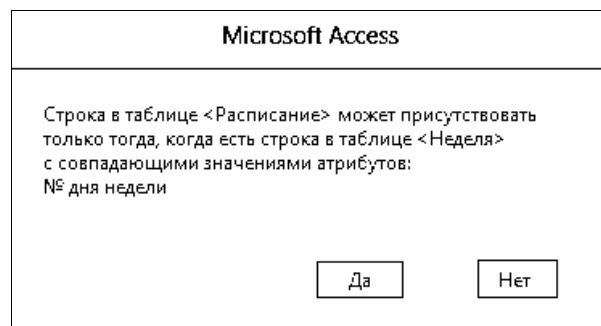


Рис. 2. Предупреждение о необходимости создания связей

будет устанавливаться. Эта функция необходима для создания ациклических схем БД, а также для недопущения создания связей, не предусмотренных семантикой в прикладной области. Если для всех неглавных отношений на вход программы поданы их главные отношения, то программа корректно выполнит операцию создания неизбыточного набора связей, в противном случае выходом программы будет БД без связей. На рис. 2 показан пример определения семантики связи между двумя отношениями, также это сообщение напоминает о невозможности существования одного отношения без другого.

Другая форма «Удаление\_связей» является вспомогательной и служит только для удаления построенных связей на схеме БД. Она становится необходима в тех случаях, когда приходится тестировать написанные алгоритмы формирования неизбыточного набора связей, а также перед применением разработанных алгоритмов над готовыми схемами БД, где связи были уже построены с ошибками, неудовлетворяющими ссылочным ограничениям целостности. Форма «Удаление\_связей» содержит только один элемент управления «Кнопка», при нажатии которой в текущей БД удаляются все связи. Аналогично форме «Создание\_связей» в текущей форме удалено стандартное поле перехода по записям.

Рассмотрим основные принципы работы программного обеспечения, что необходимо знать при организации взаимодействия с другими приложениями. При открытии формы «Создание\_связей» происходит инициализация глобальных переменных: *dbms*, *strtab*, *strrel*, *str\_tab*. Переменная *dbms* имеет тип *Database* и соответствует текущей открытой БД, над которой применяются алгоритмы создания связей, в ней хранятся все отношения с их первичными

ключами, а также все установленные связи. Остальные три переменные имеют одинаковый тип *String*, в них хранятся строки, содержащие служебную информацию. Переменная *strtab* служит для хранения названий отношений в элементе управления «Поле со списком», где присутствуют названия всех отношений для рассматриваемой БД. Значения переменной *strrel* соответствуют всем связям, построенным в данной БД, и имеют следующую структуру: имя подчиненного отношения, далее, после запятой, указывается имя главного отношения, и далее, через запятую, указываются имена атрибутов, участвующих в данной связи. Переменная *str\_tab* служит для хранения имен отношений, на которые будут накладываться связи, этот список можно будет увидеть в элементе управления «Поле».

Процедура *Form\_Open* соответствует событию открытия формы, в представленном случае это есть форма «Создание связей», в нее можно передавать значения целочисленных переменных для выявления кода ошибки при выполнении этой процедуры. Вызов *Form\_Open* главным образом служит для создания списка имен отношений в элементе управления «Поле со списком». В теле данной процедуры объявляется 2 переменные: *fld* и *tbf*. Переменная *fld* имеет тип *Control* и используется для хранения данных, находящихся в элементе управления «Поле со списком». Переменная *tbf* имеет тип *TableDef* и служит для хранения одного отношения и используется для перебора всех отношений в исходной БД. При выполнении рассматриваемой процедуры переменной *db* присваивается значение текущей БД. Затем в цикле формируется список имен отношений, который помещается в текстовую переменную *strtab*, далее задается значение *fld* как элемента управления поля со списком и задается его свойство *RowSource* как значение переменной *strtab*.

После выбора очередного отношения из списка отношений необходимо добавить его имя в элемент управления «Поле» и включить в рассмотрение для дальнейшего создания связей на отношениях. Для этого и выполняется процедура *SpisokTablic\_AfterUpdate*. Внутри данной процедуры объявляются две переменные *txt* и *tab1*. Переменная *txt* имеет тип *ComboBox* и используется для извлечения текущего имени отношения. Переменная *tab1* имеет тип *String* и используется для сохранения полученного имени отношения, которое необходимо добавить к существующему списку отношений.

В процедуре *SpisokTablic\_AfterUpdate* используется следующая последовательность операций: в переменную *txt* записывается значение, отображаемое на элементе управления «Поле со списком», после чего оно преобразуется в текстовую переменную, избавленную от пустых символов, и дописывается в глобально объявленную переменную *str\_tab*. В окончании процедуры осуществляется перенос значения переменной *str\_tab* в элемент управления «Поле» для дальнейшей работы со списком отношений.

Процедура *Связи\_Click* является основной в разрабатываемой программе. Из нее вызываются все вспомогательные функции и процедуры, служащие для создания связей. Внутри процедуры кроме служебных переменных объявляются переменные *pdj* и *pdj* текстового типа *String* и используются для хранения текущих *i*-го и *j*-го отношений, между которыми и проверяется необходимость установки одной из возможных связей. Кроме того, в данной процедуре используется текстовое значение *msgs*, которое служит для формирования текстового сообщения

о необходимости создания той или иной связи, в нее поступают значения тех отношений, которые не могут существовать по отдельности. Еще одна переменная, используемая в данной процедуре, — *fldp*, которая имеет тип *Field* и используется для определения атрибутов каждого отношения.

Как уже было ранее упомянуто, процедура *Связи\_Click* является основной при создании связей на схеме БД. В теле данной функции присутствует многомерный цикл, основная цель которого просмотреть зависимости включения между каждой парой отношений. В процессе выполнения цикла осуществляется запись в переменную *strrel* значения всех связей, присутствующих на входной схеме БД. Каждый раз при этом выдается сообщение о необходимости создания той или иной связи вместе с напоминанием о невозможности существования одного отношения без другого. После выполнения цикла из функции *Связи\_Click* вызываются две последующие процедуры *min\_rel* и *gen\_rel*, служащие для формирования избыточного набора связей и для непосредственной записи в БД.

После создания полного набора связей на схеме БД необходимо избавиться от избыточных связей, наличие которых необязательно, но возможно. Для реализации этой цели реализована функция *min\_rel*, которая вызывается после проверки всех пар отношений на выполнение условия включения. В теле процедуры *min\_rel* реализован алгоритм построения замыкания множества отношений для каждой пары отношений, связанных условием ссылочной целостности [8]. Для текущей пары отношений в переменной типа *String* формируется список отношений, попадающих в замыкание главного отношения рассматриваемой пары в соответствии с условием ссылочной целостности. При этом условие ссылочной целостности для текущей пары не учитывается и будет считаться избыточным, если подчиненное отношение текущей пары попадет в замыкание. Логическая переменная *pod*, имеющая тип *Boolean*, используется для определения необходимости дальнейшего построения замыкания при добавлении очередного отношения в замыкание.

При реализации построения связей на схемах БД так же используются вспомогательные функции. Они используются только с целью получения удобства при создании программного комплекса.

Из рассмотрения предыдущего материала следует, что в основе работы программного обеспечения лежит обработка символьных строк различной структуры и содержания. Стандартных функций VBA для этой цели недостаточно. Поэтому была разработана функция *str\_tek*, которая служит для последовательной выборки необходимых подстрок в текущей строке.

Еще одна вспомогательная функция — *sub\_set*, которая возвращает одно логическое значение. На вход функции *sub\_set* поступает два аргумента текстового типа. Функция выдает истинное значение в случае, когда второй аргумент является подстрокой первого аргумента, соответственно, функция возвращает ложное значение, если второй входной аргумент не присутствует в первом. В структуре функции *sub\_set* присутствует бесконечный цикл, из которого программа выходит при условии конца главной строки или при условии конца вспомогательной строки. Отличие рассмотренных процедур от стандартных функций VBA в учете структуры обрабатываемых строк.

Кроме перечисленных универсальных программ, которые без труда могут быть переписаны для ра-

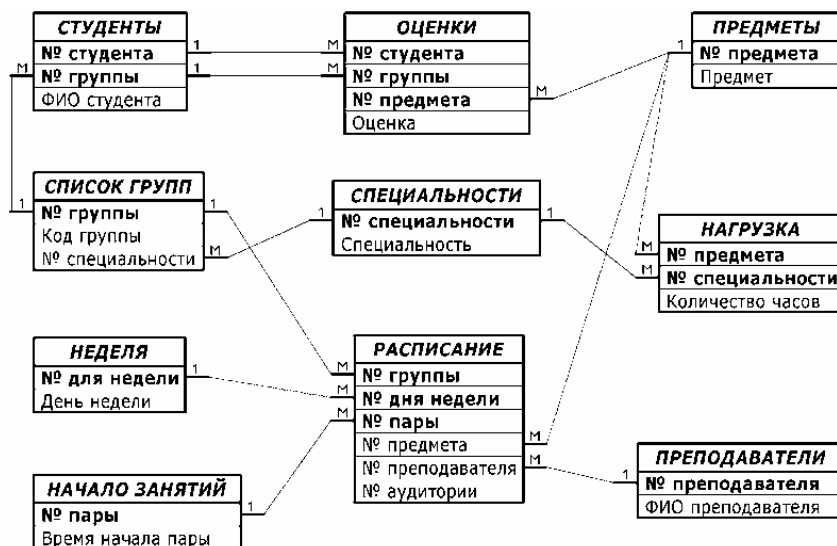


Рис. 3. Результирующая схема БД

боты с другой СУБД, реализован ряд функций, использующих специфические возможности Access. Такие функции при замене СУБД должны быть полностью переработаны под новые возможности.

Функция *str\_key* разработана с целью определения значения первичного ключа в заданном отношении, следовательно, необходимо задать аргументом для данной функции имя отношения, в котором будет выполнен поиск первичного ключа. Входной аргумент выглядит в виде строковой переменной с именем отношения, на выходе — строка из списка атрибутов первичного ключа отношения. Тело функции состоит из тройного цикла, первый из которых проходит по всем таблицам всей БД и ищет таблицу, заданную аргументом рассматриваемой функции *str\_key*. Второй цикл проходит по всем атрибутам текущей таблицы, третий — по каждому символу атрибута. В теле данного тройного цикла в выходную переменную посимвольно записываются значения ключевых атрибутов.

После работы алгоритма построения избыточного набора связей была получена общая схема БД, со всеми ее компонентами, представленными в виде строковых переменных *strtab* — список отношений и *strrel* — список ссылочных ограничений целостности с их атрибутами. Данный метод представления структуры БД в виде строковых переменных используется из-за большой трудоемкости работы с БД напрямую с использованием запросов и, кроме того, существенно облегчает перенос программного обеспечения на другую платформу. Процедура *gen\_rel* под управлением информации из указанных строк формирует фактические связи между отношениями текущей БД, используя при этом методы встроенного языка программирования VBA для вызова SQL-команды ALTER TABLE.

Результат работы программного обеспечения показан на рис. 3. В общей сложности для представленной БД было установлено 12 связей, соответствующих

ссылочным ограничениям целостности. В результате работы процедуры *Связи\_Click* было выявлено, что связь между отношениями «Список групп» и «Оценки» является избыточной. Эта связь автоматически была удалена из схемы БД.

#### Библиографический список

- Gómez-López M. T., Gasca R. M., Pérez-Álvarez J. M. Compliance validation and diagnosis of business data constraints in business processes // *Information Systems*. 2015. Vol. 48. P. 26–43.
- Visser J. Coupled Transformation of Schemas, Documents, Queries, and Constraints // *Electronic Notes in Theoretical Computer Science*. 2008. Vol. 200. № 3. P. 3–23.
- Мейер, Д. Теория реляционных баз данных: моногр. М.: Мир, 1987. 608 с.
- Ульман, Дж. Основы систем баз данных / пер. с англ. М. Когаловского, В. Когутовского; под ред. М. Когаловского. М.: Финансы и статистика, 1983. 334 с.
- Casanova M., Fagin R., Papadimitriou C. Inclusion Dependencies and Their Interaction with Functional Dependencies // *Journal of Computer and System Sciences*. 1984. № 28 (1). P. 29–59.
- Missaoui R., Godin R. The Implication Problem for Inclusion Dependencies: A Graph Approach // *SIGMOD Record*. 1990. Vol. 19. № 1. P. 36–40.
- Levene M., Vincent M. W. Justification for Inclusion Dependency Normal Form // *IEEE Transactions on Knowledge and Data Engineering*. 2000. Vol. 12. № 2. P. 281–291.
- Зыкин, В. С. Ссылочная целостность данных в корпоративных информационных системах // *Информатика и ее применение*. 2015. Т. 9. № 3. С. 119–127.

**ЗЫКИН Владимир Сергеевич**, аспирант, старший преподаватель кафедры прикладной математики и фундаментальной информатики.  
Адрес для переписки: vszykin@mail.ru

Статья поступила в редакцию 09.01.2017 г.

© В. С. Зыкин