



ИНФОРМАТИКА, ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА И УПРАВЛЕНИЕ

УДК 519.2:004.421.5:004.7

В. Н. ЗАДОРЖНЫЙ
Е. Б. ЮДИН
М. Н. ЮДИНА

Омский государственный
технический университет,
г. Омск

Институт математики
им. С. Л. Соболева СО РАН,
Омский филиал

КАЛИБРОВКА ГРАФОВ ПРЕДПОЧТИТЕЛЬНОГО СВЯЗЫВАНИЯ ПО РАСПРЕДЕЛЕНИЯМ СТЕПЕНЕЙ ВЕРШИН И КОЭФФИЦИЕНТАМ КЛАСТЕРИЗАЦИИ

Разрабатывается метод комплексной калибровки случайных графов предпочтительного связывания одновременно по распределениям степеней связности вершин и по коэффициенту кластеризации. В методы теории случайных графов с нелинейным правилом предпочтительного связывания вводится новый прием — использование для выращивания графов сложных стохастических приращений (модулей), состоящих из нескольких взаимосвязанных вершин. Выводятся математические соотношения, позволяющие выполнять комплексную калибровку графа, выращиваемого присоединениями модулей. Полученные результаты демонстрируют принципиальную возможность исследования и использования нового класса случайных графов с нелинейным правилом предпочтительного связывания — случайных графов со сложными стохастическими приращениями.

Ключевые слова: случайные графы, распределения степеней связности вершин, нелинейное правило предпочтительного связывания, коэффициент кластеризации.

Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 16-31-60023 мол_а_дк.

1. Введение. Теория случайных графов предпочтительного связывания [1–3] развивается в работах [4–6] в направлении исследования и использования случайных графов с нелинейным правилом предпочтительного связывания (НППС). Графы

с НППС обладают широкими возможностями их калибровки для построения адекватных моделей реальных больших сетей, содержащих миллионы узлов и связей. Широкие возможности калибровки обеспечиваются решением задач анализа и синтеза

графов с НППС при самых общих предположениях относительно весовой функции $f(k) \geq 0$, используемой в правиле предпочтительного связывания, а также допущением стохастических приращений графа.

Стохастическое приращение представляет собой вершину со случайным числом x исходящих из нее дуг. Приращения поступают на каждом шаге выращивания графа, которое начинается с заданной «затравки» — небольшого начального графа, имеющего, как правило, произвольную конфигурацию. Правило выращивания графа с НППС включает следующие два пункта.

1. Граф растет за счет добавления к нему на каждом шаге времени $t \in \{1, 2, \dots\}$ новой вершины со случайным числом x дуг;

2. Каждая из x дуг новой вершины присоединяется свободным концом к случайно выбираемой вершине графа. Вероятность p_i присоединения дуги к вершине i со степенью связности k_i пропорциональна весу $f(k_i)$ этой вершины:

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

где N — число вершин графа, а 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 будем использовать компактное обозначение f_k . Случайное число x дуг приращения имеет заданное распределение вероятностей $P(x = k) = r_k$, где $g \leq k \leq h$, $r_g + \dots + r_h = 1$. Здесь $g \geq 1$ — наименьшая степень приращения — совпадает с нижней границей интервала определения функции $f(k)$.

После выращивания графа он может быть путем замены дуг ребрами превращен в неориентированный граф.

В последних работах по теории графов с НППС получены результаты, позволяющие рассчитывать динамику изменения характеристик графов в ходе их выращивания [7, 8]. Разработаны методы комплексной калибровки графов по распределению степеней вершин и, одновременно, по двумерному распределению степеней ребер [8], обеспечивающие совпадение этих распределений с соответствующими распределениями узлов и связей в моделируемых сетях. Однако вопрос о коэффициенте кластеризации до сих пор не был в достаточной степени исследован, и генерируемые графы с НППС часто имели слишком низкую по сравнению с моделируемыми сетями его величину.

Коэффициент кластеризации графа определяется формулой

$$C = \frac{3N_{\Delta}}{N_V}, \quad (2)$$

где N_{Δ} — число треугольников в графе (треугольник образуется тремя вершинами, каждые две из которых связаны дугой или ребром), N_V — число вилок, т.е. неориентированных путей длиной две дуги (два ребра). Идеино близкие к коэффициенту кластеризации показатели — мотивы — исследуются в работах [9–14].

Задача повышения коэффициента кластеризации в графах с НППС решается далее путем введения сложных стохастических приращений — модулей, состоящих из нескольких связанных вершин. В статье рассматриваются модули, состоящие

из двух вершин, связанных одной дугой и имеющих свободные дуги, концы которых связываются с вершинами выращиваемого графа. Двухвершинные модули случайным образом чередуются с обычными (одновершинными) приращениями графа.

2. Двухвершинные модули. Двухвершинный модуль (далее — модуль) — это приращение графа, которое состоит из двух вершин — вершины типа A (ведущей) и вершины типа B (ведомой). Из вершины A исходит случайное число x дуг, разыгрываемое в соответствии с тем же распределением $\{r_k\}$, которое используется для разыгрывания числа дуг в обычном приращении. Число дуг, исходящих из вершины B , не разыгрывается, оно в точности равно числу дуг, исходящих из вершины A . При этом одна из дуг, исходящих из вершины B , присоединяется свободным концом к вершине A . Таким образом, у вершины A имеется x свободных дуг для связывания с вершинами графа, а у вершины B — только $(x - 1)$ свободных дуг.

Среднее число $M(x) = \sum_{k=g}^h kr_k$ дуг, исходящих из вершины приращения, обозначим через m , среднюю степень вершин $M(k) = \sum_{k=g}^{\infty} kQ_k$ в выращенном бесконечном графе обозначим через $\langle k \rangle$. Здесь Q_k — вероятность того, что случайно выбранная в выращенном графе вершина имеет степень k . Набор (последовательность) $\{Q_k\}$ всех Q_k для $k = g, g + 1, \dots, M$ — это распределение степеней связности (РСС) вершин графа.

На каждом шаге выращивания графа с вероятностью γ поступает приращение-модуль, с вероятностью $(1 - \gamma)$ — обычное приращение графа. Дуги обычного приращения присоединяются к графу независимо, в соответствии с правилом (1). В тех случаях, когда поступившее приращение является модулем, оно используется для создания (внедрения) треугольников в выращиваемый граф следующим образом. Свободные дуги, исходящие из вершины B модуля, присоединяются независимо к вершинам графа в соответствии с правилом (1). После присоединения каждой из этих дуг к некоторой вершине i графа с заданной вероятностью P , случайно, принимается решение создать треугольник. Если принимается такое решение, то к этой же вершине i присоединяется и одна из дуг, исходящих из вершины A . Тем самым создается треугольник, основанием которого служит дуга, ведущая из вершины B в вершину A . После присоединения всех $(x - 1)$ свободных дуг вершины B остается некоторое число свободных дуг вершины A , не вовлеченных в треугольники. Как минимум, из x дуг вершины A остается одна такая дуга. Эти оставшиеся свободными дуги вершины A присоединяются к вершинам графа независимо, по правилу (1).

В результате присоединения модуля внедряется в среднем $\mu = P(m - 1)$ треугольников. Следовательно, за один шаг выращивания графа в среднем в граф внедряется $(1 - \gamma) \cdot 0 + \gamma \mu = \gamma \mu$ треугольников.

3. Финальное распределение степеней вершин. Для вывода формулы финального распределения степеней вершин воспользуемся подходом, разработанным в [4–6], который основан на составлении уравнения баланса вероятностей для допредельного при числе вершин $N \rightarrow \infty$ состояния графа. При этом будем учитывать, что среднее число треугольников, внедряемых при связывании модуля с графом, равно μ , и, следовательно, среднее число не вовлеченных в треугольники дуг составляет $(2m - 1 - 2\mu)$. Действительно, оно равно среднему

числу $(2m - 1)$ изначально свободных дуг модуля минус среднее число 2μ дуг модуля, вовлеченных в треугольники. Кроме того, будет использоваться выведенная в [4] формула вероятности P_k связывания с вершиной слоя A_k :

$$P_k = \frac{Q_k f_k}{\langle f \rangle}, \quad (3)$$

где A_k — слой, т.е. множество вершин графа, имеющих степень k ; $\langle f \rangle = \sum_{k=g}^{\infty} f_k Q_k$ — средний вес вершин графа.

Рассмотрим, как в среднем изменяется число $|A_k|$ вершин в слое A_k за один шаг выращивания графа.

Если поступает обычное приращение, то $|A_k|$ с вероятностью r_k увеличивается на единицу — с этой вероятностью поступившая вершина приращения имеет k дуг (т.е. ее степень равна k) и, следовательно, вершина попадает в слой A_k . В среднем $|A_k|$ увеличится за счет этого на величину $1 \cdot r_k = r_k$. Кроме того, при связывании дуг приращения с вершинами графа в среднем mP_k таких вершин будет выбрано в слое A_k . Степени выбранных вершин вследствие присоединения новых дуг возрастут на единицу и вершины перейдут в слой A_{k+1} , в результате чего число $|A_k|$ вершин в слое A_k уменьшится в среднем на mP_k . И в среднем mP_{k-1} вершин будет выбрано дугами приращения в слое A_{k-1} . Степень выбранных вершин возрастет на единицу, и они перейдут в слой A_k . За счет этого число вершин в слое A_k возрастет в среднем на mP_{k-1} .

Суммируя перечисленные изменения, найдем среднее приращение $\Delta_1|A_k|$ при поступлении обычного приращения графа:

$$\Delta_1|A_k| = r_k + mP_{k-1} - mP_k. \quad (4)$$

Если поступает приращение-модуль, то в слое A_k происходят следующие изменения.

Вершина B модуля с вероятностью r_k имеет степень k и попадает в слой A_k . За счет этого число вершин в слое A_k возрастет в среднем на r_k .

Вершина A модуля с вероятностью r_{k-1} имеет степень k (это $k - 1$ «своих» дуг и одна дуга, заходящая из вершины A) и попадает в слой A_k . За счет этого число вершин в слое A_k возрастет в среднем на r_{k-1} .

В среднем μ пар дуг модуля, вовлеченных в треугольники, с вероятностями P_{k-2} свяжутся с вершинами слоя A_{k-2} . У соответствующих μ вершин степень возрастет на 2 и они перейдут в слой A_k . Оставшиеся не вовлеченными в треугольники дуги, среднее число которых составляет $(2m - 1 - 2\mu)$, с вероятностями P_{k-1} свяжутся с вершинами слоя A_{k-1} .

У соответствующих вершин степень возрастет на единицу, и они перейдут в слой A_k .

Эти же дуги с вероятностями P_k свяжутся с вершинами слоя A_k , которые в результате возрастания степени уйдут в слой A_{k+1} .

В среднем μ пар дуг модуля, вовлеченных в треугольники, с вероятностями P_k свяжутся с вершинами слоя A_k , и эти вершины перейдут в слой A_{k+2} .

Суммируя средние значения перечисленных изменений числа вершин в слое A_k , находим среднее приращение $\Delta_2|A_k|$ при поступлении модуля:

$$\Delta_2|A_k| = r_k + r_{k-1} + \mu P_{k-2} + (2m - 1 - 2\mu)P_{k-1} - (2m - 1 - 2\mu)P_k - \mu P_k. \quad (5)$$

На каждом шаге выращивания графа модуль поступает с вероятностью γ , обычное приращение — с вероятностью $(1 - \gamma)$. С учетом этого усредняем изменения (4), (5) по видам поступающего приращения и находим среднее изменение Δ числа вершин в слое A_k за один шаг:

$$\begin{aligned} \Delta = & (1 - \gamma)r_k + (1 - \gamma)mP_{k-1} - (1 - \gamma)mP_k + \\ & + \gamma(r_k + r_{k-1}) + \gamma\mu P_{k-2} + \gamma(2m - 1 - 2\mu)P_{k-1} - \\ & - \gamma(2m - 1 - 2\mu)P_k - \gamma\mu P_k = r_k + \gamma r_{k-1} + \\ & + \gamma\mu P_{k-2} + (m + \gamma m - 2\gamma\mu - \gamma)P_{k-1} - \\ & - (m + \gamma m - \gamma\mu - \gamma)P_k. \end{aligned} \quad (6)$$

В пределе при $N \rightarrow \infty$ доля числа вершин в слое A_k в числе всех N вершин графа не изменяется. За один шаг выращивания графа абсолютное число вершин в этом слое возрастает в среднем на величину Δ , полученную в (6), а число вершин в графе возрастает в среднем на $2\gamma + 1 \cdot (1 - \gamma) = 1 + \gamma$.

Приравняв долю вершин в слое A_k до выполнения шага выращивания графа к соответствующей доле после выполнения этого шага, получаем следующее уравнение баланса:

$$\begin{aligned} \frac{|A_k|}{N} = & \frac{|A_k| + r_k + \gamma r_{k-1} + \gamma\mu P_{k-2} + \dots}{N} \rightarrow \\ \rightarrow & \frac{+ (m + \gamma m - 2\gamma\mu - \gamma)P_{k-1} - (m + \gamma m - \gamma\mu - \gamma)P_k}{N + (1 + \gamma)}. \end{aligned} \quad (7)$$

После умножения обеих частей уравнения на N и на $N + \gamma$ можно сократить одинаковые члены $|A_k|N$ в левой и правой частях полученного равенства. Затем обе части равенства делятся на N . В результате оставшееся в левой части выражение $|A_k|(\gamma + 1)$ превращается в $Q_k(\gamma + 1)$, а множители N в правой части сокращаются. Заменяя после этого все P_k в правой части равенства их выражением (3), получаем уравнение баланса вероятностей Q_k , записанное в явной форме:

$$\begin{aligned} Q_k(1 + \gamma) = & r_k + \gamma r_{k-1} + \gamma\mu \frac{Q_{k-2} f_{k-2}}{\langle f \rangle} + \\ & + (m + \gamma m - 2\gamma\mu - \gamma) \frac{Q_{k-1} f_{k-1}}{\langle f \rangle} - (m + \gamma m - \gamma\mu - \gamma) \frac{Q_k f_k}{\langle f \rangle}. \end{aligned} \quad (8)$$

Из этого уравнения можно выразить Q_k следующим образом:

$$\begin{aligned} Q_k = & \frac{r_k \langle f \rangle + \gamma r_{k-1} \langle f \rangle + \gamma\mu Q_{k-2} f_{k-2} + \dots}{(1 + \gamma)\langle f \rangle + (m + \gamma m - \gamma\mu - \gamma)f_k} \rightarrow \\ \rightarrow & \frac{+ (m + \gamma m - 2\gamma\mu - \gamma)Q_{k-1} f_{k-1}}{(1 + \gamma)\langle f \rangle + (m + \gamma m - \gamma\mu - \gamma)f_k}, \end{aligned} \quad (9)$$

$$\text{где } k = g, g + 1, \dots, M, \langle f \rangle = \sum_{k \geq g} f_k Q_k, \quad (10)$$

и при заданных параметрах $\{r_k\}$, $\{f_k\}$, γ , μ и m графа использовать полученную рекуррентную систему уравнений (9), (10) для расчета РСС $\{Q_k\}$ аналогично тому, как это делается в случае применения только простых приращений в [4].

3. Совместная калибровка графа по распределению степеней вершин и по коэффициенту кластеризации. В задаче калибровки графа известно требуемое РСС $\{Q_k\}$ его вершин и требуется найти

параметры $\{r_k\}$, $\{f_k\}$, γ , μ и m , при которых граф будет иметь заданное РСС вершин $\{Q_k\}$. Эту задачу также можно решить с помощью уравнения баланса (8).

Выражая из (8) f_k , получаем рекуррентное соотношение

$$f_k = (r_k + \gamma r_{k-1}) \frac{1}{Q_k} \frac{\langle f \rangle}{a} + \gamma \mu \frac{Q_{k-2} f_{k-2}}{\langle f \rangle} \frac{1}{Q_k} \frac{\langle f \rangle}{a} + (m + \gamma m - 2\gamma \mu - \gamma) \frac{Q_{k-1} f_{k-1}}{\langle f \rangle} \frac{1}{Q_k} \frac{\langle f \rangle}{a} - (1 + \gamma) \frac{\langle f \rangle}{a}, \quad (11)$$

где $a = (m + \gamma m - \gamma \mu - \gamma)$.

Соотношением (11) определяется последовательность $\{f_k\}$, каждый член которой имеет общий мультипликативный множитель $\frac{\langle f \rangle}{a}$. В силу эквивалентности последовательностей $\{f_k\}$ и $\{C f_k\}$, отличающихся общим множителем $C_1 > 0$ (см. правило (1)), мы можем приравнять множитель $\frac{\langle f \rangle}{a}$ в правой части соотношения (11) к единице (определяя тем самым равенство $\langle f \rangle = a$) и записать это соотношение в более компактном виде:

$$f_k = (r_k + \gamma r_{k-1}) \frac{1}{Q_k} + \gamma \mu \frac{Q_{k-2} f_{k-2}}{\langle f \rangle} \frac{1}{Q_k} + (m + \gamma m - 2\gamma \mu - \gamma) \frac{Q_{k-1} f_{k-1}}{\langle f \rangle} \frac{1}{Q_k} - (1 + \gamma). \quad (12)$$

Учитывая, что в результате сделанного выбора мы имеем равенство $\langle f \rangle = a$ и множитель $(m + \gamma m - 2\gamma \mu - \gamma)$ в (12) можем записать в виде $(a - \gamma \mu)$, получаем рекуррентное соотношение

$$f_k = (r_k + \gamma r_{k-1}) \frac{1}{Q_k} + \gamma \mu \frac{Q_{k-2} f_{k-2}}{a} \frac{1}{Q_k} + (a - \gamma \mu) \frac{Q_{k-1} f_{k-1}}{a} \frac{1}{Q_k} - (1 + \gamma), \quad (13)$$

где $k = g, g + 1, \dots, M$, $a = (m + \gamma m - \gamma \mu - \gamma)$.

Формула (13) является решением задачи калибровки графа. Она позволяет для реализации заданного РСС $\{Q_k\}$ получить бесконечно много последовательностей весов $\{f_k\}$ в зависимости от того, какие параметры $\{r_k\}$, γ , μ и m будут нами выбраны.

4. Пример калибровки графа с применением сложных приращений. Целью приводимого примера является демонстрация правильности выведенной формулы (9) для РСС вершин графа, который выращивается поступающими в случайном порядке простыми приращениями и двухвершинными приращениями-модулями, и формулы (13) калибровки такого графа по заданному распределению $\{Q_k\}$ степеней вершин. При этом приращения-модули используются для внедрения в граф некоторого числа дополнительных треугольников для повышения коэффициента кластеризации графа.

Чтобы сократить объем вычислений, воспользуемся примером сети автономных систем (АС), описанной в [15], граф которой калиброван по РСС вершин и РСС ребер в [8]. При этом усложним задачу калибровки дополнительным требованием, чтобы распределение $\{r_k\}$ числа дуг у вершин приращений, полученное в [8] при $h = 20$, не изменилось. Это требование существенно сужает возможности внедрения большого числа треугольников в граф,

но, вместе с тем, позволяет проверить, насколько при сохранении $\{r_k\}$ изменится РСС ребер графа, имеющего заданное РСС $\{Q_k\}$. Таким образом, в условие задачи калибровки наряду с обычным требованием реализовать требуемое РСС $\{Q_k\}$ мы добавляем дополнительное требование сохранить распределение $\{r_k\}$, найденное ранее для графа с простыми приращениями.

Распределение $\{r_k\}$ описано в [8], эмпирическое распределение $\{Q_k\}$ сети АС приводится в [15], его сглаженная версия описана в [8] и в [5]. Как и в [8], здесь мы будем при решении задачи калибровки использовать сглаженное распределение $\{Q_k\}$.

Вначале при калибровке графа рассчитывается по формуле (13) последовательность $\{f_k\}$ весов. При этом используемые в (13) значения $\{Q_k\}$ и $\{r_k\}$ берем из статьи [8]. Этими значениями $\{r_k\}$ определяется среднее число дуг у вершины приращения $m = 2,109055$. Вероятность P вовлечения дуги модуля в треугольник принимаем равной единице, поэтому среднее число μ треугольников, внедряемых модулем, достигает максимального значения $\mu = m - 1 = 1,109055$. При выборе величины вероятности γ имеется ограничение: все веса f_k в (13) должны быть неотрицательными. Поскольку мы наложили дополнительное условие неизменности распределения $\{r_k\}$, полученного в [8], то наибольшее возможное в данных условиях значение γ составляет 0,083894. При перечисленных исходных данных по рекуррентной формуле (13) вычисляем последовательность $\{f_k\}$, начиная со значений $f_0 = 0$, $f_1 = 0$. Тем самым решение задачи калибровки завершается.

После этого целесообразно проверить правильность решения задачи, рассчитывая вероятности Q_k по рекуррентной формуле (9). При этом, учитывая особенности формулы (13), по которой вычислены все используемые f_k , формулу (9) можно привести к следующему более простому виду, заменяя в ней все $\langle f \rangle$ известными $a = (m + \gamma m - \gamma \mu - \gamma)$:

$$Q_k = \frac{r_k a + \gamma r_{k-1} a + \gamma \mu Q_{k-2} f_{k-2} + (a - \gamma \mu) Q_{k-1} f_{k-1}}{(1 + \gamma) a + a f_k}. \quad (14)$$

На рис. 1 показаны графики заданного РСС вершин и распределение степеней вершин графа, выращенного посредством случайно чередующихся простых приращений и приращений-модулей, которое рассчитано по формуле (14) при использовании

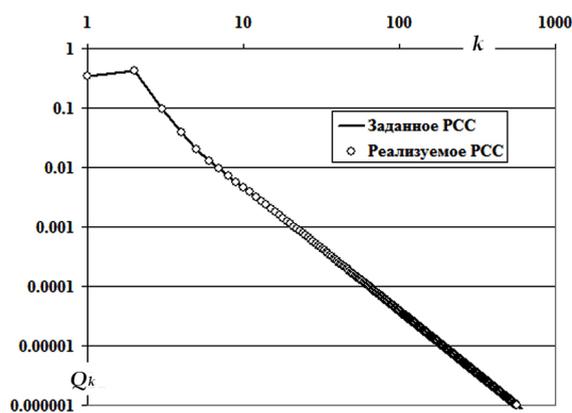


Рис. 1. Сравнение заданного РСС $\{Q_k\}$ с расчетным распределением степеней вершин графа, выращиваемого по калиброванной последовательности весов $\{f_k\}$

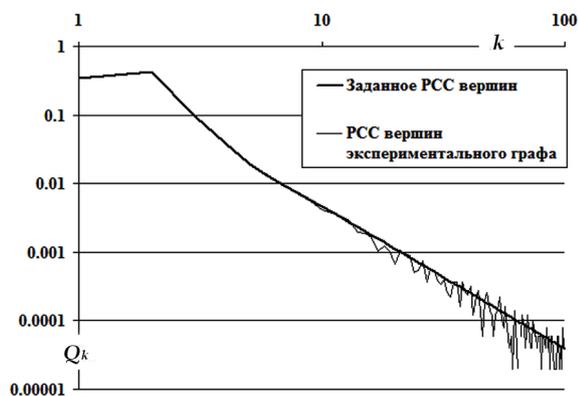


Рис. 2. Сравнение заданного РСС $\{Q_k\}$ с распределением степеней вершин экспериментального графа, выращиваемого по калиброванной последовательности $\{f_k\}$

весов $\{f_k\}$, вычисленных по калибровочной формуле (13). Видно, что эти распределения совпадают.

На рис. 2 заданное РСС вершин сравнивается с РСС вершин графа, действительно выращенного в имитационном эксперименте. Поскольку размер выращенного графа ($N = 22000$), равный размеру моделируемой сети АС, относительно невелик, вероятности старших слоев еще не вполне достигли своих стационарных значений и немного «отстают» от расчетных финальных вероятностей. По этой же причине средняя степень $\langle k \rangle = 4,196$ вершин экспериментального графа немного «отстает» от расчетной финальной средней степени вершин бесконечного графа, равной 4,218.

Достигнутый коэффициент кластеризации экспериментального графа составил небольшую величину $C = 0,003$. Но его радикальное увеличение не было целью рассмотренного примера. Мы хотели увидеть, как изменится РСС ребер графа в результате выполненных «минимальных» изменений механизма его выращивания. Это изменение РСС ребер показано на рис. 3. Как видим, РСС ребер значительно изменяется, хотя и сохраняет некоторые свои черты.

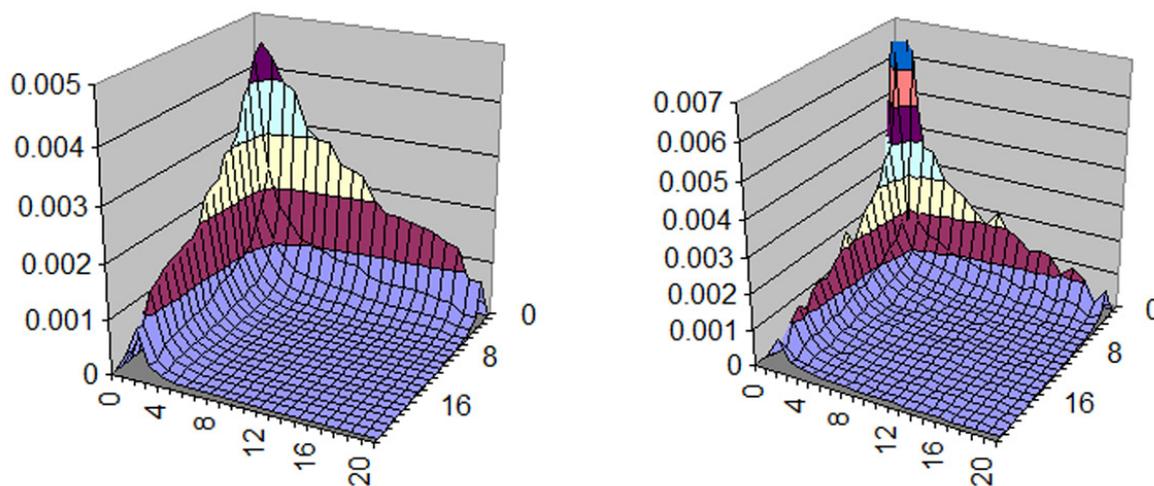


Рис. 3. Сравнение РСС ребер исходного графа (слева) с распределением степеней ребер экспериментального графа, выращиваемого по калиброванной последовательности $\{f_k\}$

Отсюда следует, что требовать сохранения распределения $\{r_k\}$ графа, однажды уже калиброванного по степеням вершин и ребер, не имеет особого смысла.

Варьирование $\{r_k\}$ при $P = 1$, $\gamma = 1$ дает требуемое РСС $\{Q_k\}$ и величину $C = 0,020$, превышающую коэффициент кластеризации $C_{AS} = 0,011$ сети АС. При $P = 0,60$, $\gamma = 0,60$ реализуются требуемые $\{Q_k\}$ и коэффициент $C = 0,012 \approx C_{AS}$.

5. Заключение. В статье предложен обобщенный механизм выращивания графов, основанный на использовании сложных стохастических приращений, состоящих из нескольких взаимосвязанных вершин.

Решена задача расчета финального РСС графов, в ходе выращивания которых наряду с обычными одновершинными приращениями графа используются двухвершинные приращения.

Решена задача калибровки таких графов по степеням связности вершин.

Показано, что при использовании двухвершинных приращений можно внедрять в граф треугольники, тем самым повышая коэффициент кластеризации графа.

Последующее развитие теории сложных стохастических приращений графов предполагает решение задачи совместной калибровки графов по степеням вершин, степеням ребер и коэффициенту кластеризации.

При этом, как видно из результатов исследования, целесообразно использовать такой мощный ресурс калибровки, как варьирование значений и числа вероятностей $\{r_k\}$, описывающих возможное число дуг у вершин приращений.

Кроме того, сложные приращения имеют смысл попытаться использовать для калибровки графов по встречаемости более сложных, чем треугольники, конфигураций — мотивов, что позволит существенно развить теорию растущих сетей [4–8, 16–20].

Библиографический список

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

2. Dorogovtsev S. N., Mendes J. F. F. Effect of the accelerated growth of communications networks on their structure // *Phys. Rev.* 2001. E 63. P. 025101.
3. Krapivsky P. L., Redner S. Organization of growing random networks // *Phys. Rev.* 2001. E 63. P. 066123.
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: 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.
6. Zadorozhnyi V. N., Yudin E. B. Structural properties of the scale-free Barabasi-Albert graph // *Automation and Remote Control.* 2012. Vol. 73, no. 4. P. 702–716. DOI 10.1134/S0005117912040091.
7. Задорожный В. Н. Растущие сети: динамика распределения степеней связности смежных узлов // *Омский научный вестник. Сер. Приборы, машины и технологии.* 2016. № 2 (146). С. 81–86.
8. Задорожный В. Н., Юдин Е. Б. Калибровка случайных графов предпочтительного связывания по распределениям степеней вершин и ребер // *Омский научный вестник. Сер. Приборы, машины и технологии.* 2017. № 1 (151). С. 114–119.
9. 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.
10. Milo R. [et al.]. Network motifs: simple building blocks of complex networks // *Science.* 2002. Vol. 298 (594). P. 824–827.
11. Mangan S., Alon U. Structure and function of the feed-forward loop network motif // *Proc. Natl. Acad. Sci. U.S.A.* 2003. Vol. 100 (21). P. 11980–11985.
12. Cordella L. P. [et al.]. 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.
13. Solnon C. AllDifferent-based Filtering for Subgraph Isomorphism // *Artificial Intelligence.* 2010. Vol. 174 (12–13). P. 850–864.
14. Elenberg E. R. [et al.]. Beyond Triangles: A Distributed Framework for Estimating 3-profiles of Large Graphs // *Proc. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD).* 2015. P. 229–238.
15. Структура автономных систем сети Интернет, воссозданная на основе BGP таблиц, 2006 г. URL: <http://www-personal.umich.edu/~mejn/netdata/as-22july06.zip> (дата обращения: 01.09.2009).
16. Krapivsky P., Redner S., Leyvraz F. Connectivity of growing random networks // *Physical review letters.* 2001. Vol. 85 (21). P. 4629.
17. Antal T., Krapivsky P., Redner S. Dynamics of social balance on networks // *Physical Review E.* 2005. Vol. 72 (3). P. 036121.
18. Growing distributed networks with arbitrary degree distributions // *Eur. Phys. J.* 2007. Vol. 58. P. 175–184.
19. Krapivsky P., Krioukov D. Scale-free networks as preasymptotic regimes of superlinear preferential attachment // *Physical Review E.* 2008. Vol. 78 (2). P. 026114.
20. Ghoshal G., Chi L., Barabasi A.-L. Uncovering the role of elementary processes in network evolution // *Scientific Reports.* 2013. Vol. 3. P. 1–8.

ЗАДОРЖНЫЙ Владимир Николаевич, доктор технических наук, доцент (Россия), профессор кафедры «Автоматизированные системы обработки информации и управления».

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.

ЮДИНА Мария Николаевна, аспирантка кафедры «Автоматизированные системы обработки информации и управления».

Адрес для переписки: mg-and-all@mail.ru

Статья поступила в редакцию 19.04.2017 г.

© В. Н. Задорожный, Е. Б. Юдин, М. Н. Юдина

Книжная полка

Дмитриева, Л. М. Подготовка выпускных квалификационных работ в области разработки и применения информационных технологий в медиаиндустрии : учеб. электрон. изд. локального распространения : учеб. пособие для вузов по направлению подгот. 230400 «Информационные системы и технологии» / Л. М. Дмитриева, О. В. Батенькина. – Омск : Изд-во ОмГТУ, 2017. – 1 эл. опт. диск (CD-ROM).

Содержит требования к структуре, содержанию, выбору тематики и оформлению выпускных квалификационных работ в области разработки и применения информационных технологий в медиаиндустрии. Рассмотрены организационные этапы подготовки выпускной квалификационной работы, проверки, утверждения и защиты в ГАК. Предназначено для студентов высших учебных заведений, обучающихся по направлению подготовки бакалавриата 230400 «Информационные системы и технологии», по профилям «Информационные технологии в дизайне», «Информационные технологии в медиаиндустрии», а также консультантам и преподавателям, осуществляющим руководство выпускными квалификационными работами.

О ВЕРОЯТНОСТИ ПОТЕРЬ В МНОГОЛИНЕЙНЫХ ФРАКТАЛЬНЫХ СИСТЕМАХ МАССОВОГО ОБСЛУЖИВАНИЯ

Разрабатывается и исследуется метод определения вероятностей потерь в многолинейных фрактальных системах массового обслуживания с помощью аппроксимации этих вероятностей стационарными вероятностями состояний и хвостами распределений вероятностей состояний соответствующих бесконечнолинейных систем. Эффективность рассматриваемого подхода к определению вероятностей потерь обосновывается теоретически и с помощью имитационного моделирования. Предложенные в статье аппроксимации позволяют ускорить и упростить поиск оптимального распределения числа каналов по узлам сетей массового обслуживания.

Ключевые слова: вероятность потерь в конечнолинейных СМО, фрактальный трафик, теория массового обслуживания, телекоммуникационные сети.

Исследование выполнено в рамках НИР № 17060В «Разработка и исследование аналитико-имитационных методов расчета и оптимизации систем и сетей массового обслуживания со степенными хвостами распределений» ОмГТУ.

1. Введение. В настоящее время наблюдается процесс конвергенции компьютерных и телекоммуникационных сетей, что приводит к созданию единой мультисервисной сети [1–3], которая предоставляет возможность передачи разнородного трафика. Современные мультисервисные сети строятся на основе принципа коммутации пакетов (когда необходимые для передачи сообщения разбиваются на так называемые «пакеты», которые далее передаются по сети). Отличительной особенностью коммутаторов в сетях с коммутацией пакетов является наличие в них буферов для хранения пакетов [1]. Использование буферов позволяет регулировать скорость продвижения данных по линиям связи, а также предотвратить потери пакетов. Однако существуют ситуации, когда при высокой загрузке буфер переполняется и происходят потери пакетов.

Вероятность потерь заявок является одной из характеристик, влияющих на качество обслуживания (QoS) современных сетей [4]. Поэтому обеспечение требуемой низкой вероятности потерь или даже сведение её к нулю является важной и актуальной задачей при проектировании сетей.

Теория массового обслуживания позволяет математически представлять и описывать мультисервисные сети и их компоненты как сети (СМО) и системы (СМО) массового обслуживания соответственно. Далее будут рассматриваться только системы массового обслуживания. Существует большое количество работ [5–7], в которых показано, что описывающие трафик случайные процессы задаются распределениями со степенными хвостами, поэтому большой интерес представляют фрактальные СМО. Фрактальными будем называть СМО вида $GI/GI/n/m$, функции распределения интервалов

поступления и/или времени обслуживания заявок, в которых имеют степенные хвосты. Функция распределения вероятностей (ф.р.) $F(t)$ имеет степенной хвост $\bar{F}(t) = 1 - F(t)$, если $\bar{F}(t) \sim ct^{-\alpha}$ при $t \rightarrow \infty$. В распределениях, описывающих фрактальные СМО, будем использовать $\alpha \in (1; 2]$; при этом первый начальный момент таких распределений является конечным, а второй — бесконечным.

В данной статье разрабатывается и исследуется метод определения вероятности потерь заявок p_l в многолинейных фрактальных системах массового обслуживания вида $GI/GI/n/0$, т.е. в системах с числом каналов n и без буфера для хранения заявок (размер буфера $m = 0$). Если входящая заявка застаёт все каналы занятыми, то она теряется. Новый метод позволяет получать значения вероятностей потерь p_l в многолинейной фрактальной СМО для любых n путем аппроксимации этих вероятностей соответствующими вероятностями состояний p_n или хвостом $P(k \geq n)$ бесконечнолинейной СМО $GI/GI/\infty$ с аналогичными характеристиками, когда вероятности потерь малы ($10^{-4} - 10^{-5}$ и менее).

Представленные далее результаты получены с помощью аналитико-имитационных методов.

В случае систем массового обслуживания вида $M/GI/\infty$ и $M/GI/n/0$ существуют точные формулы для расчета стационарных вероятностей состояний. Для СМО более общего вида точный анализ затруднителен, поэтому имитационное моделирование является незаменимым инструментом исследования таких СМО [8].

Сама идея определения вероятностей потерь в многолинейных СМО с помощью бесконечнолинейных не является оригинальной [8–11], однако рассматриваемые в литературе задачи аппроксимации вероятностей потерь предполагают наличие

конечной дисперсии распределений интервалов времени поступления и обслуживания заявок, что не соответствует рассматриваемому в данной работе случаю.

2. Постановка задачи. Рассмотрим фрактальную многолинейную систему $A(t)/B(t)/n/0$, где $A(t)$ — ф.р. времени поступления заявок с математическим ожиданием (м.о.) a ; $B(t)$ — ф.р. времени обслуживания заявок с м.о. b .

Определим интенсивность входящего потока заявок $\lambda = 1/a$, интенсивность времени обслуживания заявок $\mu = 1/b$ и коэффициент загрузки $\rho = \lambda b$. В данной работе СМО рассматриваются в стационарном режиме функционирования, когда вероятности их состояний $p_n(t) = p_n$ со временем не меняются.

В качестве ф.р. со степенным хвостом будем использовать распределение Парето $Pa(K, \alpha)$ с параметром масштаба (являющимся одновременно минимальным значением случайной величины) $K > 0$ и параметром формы $\alpha > 0$. Значение параметра α будем рассматривать только в полуинтервале $(1, 2]$ в силу его актуальности. Требуется показать, что вероятности потерь заявок в фрактальной многолинейной системе вида $GI/GI/n/0$ можно заменять стационарными вероятностями состояний p_n или хвостом $P(k \geq n)$ соответствующей бесконечнолинейной фрактальной СМО $GI/GI/\infty$, когда вероятности потерь достаточно малы.

Необходимость такой замены возникает при решении задач оптимизации числа каналов в сетях массового обслуживания. Одна из подобных задач рассмотрена в [12]. Основной недостаток такой оптимизации заключается в том, что для каждого отдельного узла рассматриваемой СеМО, вероятность потерь p_l приходится моделировать по отдельности для любого фиксированного n . Уже при небольшом количестве узлов СеМО проектировщику сетей приходится тратить большое количество времени на имитационное моделирование. Для ускорения процесса поиска оптимального числа каналов в каждом узле СеМО конечнолинейную фрактальную СМО можно заменять на аналогичную бесконечнолинейную, стационарные вероятности состояний p_n и хвост $P(k \geq n)$ которой можно получить для всех n за один прогон имитационной модели.

3. Теория. Аналитическое решение можно получить для фрактальных систем вида $M/GI/n/0$ и $M/GI/\infty$, для которых существуют формулы расчета стационарных вероятностей состояний. В многолинейной СМО $M/GI/n/0$

$$p_l = \frac{(\lambda/\mu)^n}{n!} \cdot \left(\sum_{i=0}^n \frac{(\lambda/\mu)^i}{i!} \right)^{-1} = \frac{\rho^n}{n!} \cdot \left(\sum_{i=0}^n \frac{\rho^i}{i!} \right)^{-1}, \quad (1)$$

в бесконечнолинейной системе $M/GI/\infty$

$$p_n = e^{-\lambda/\mu} \frac{(\lambda/\mu)^n}{n!} = e^{-\rho} \frac{\rho^n}{n!}. \quad (2)$$

Очевидно, что для систем с экспоненциальным распределением времени между приходами заявок при $n \rightarrow \infty$ предел отношения вероятности потери заявки p_l в многолинейной СМО к стационарной вероятности состояния p_n бесконечнолинейной СМО будет равен единице:

$$\lim_{n \rightarrow \infty} \frac{p_l}{p_n} = \lim_{n \rightarrow \infty} \frac{\frac{\rho^n}{n!} \cdot \left(\sum_{i=0}^n \frac{\rho^i}{i!} \right)^{-1}}{e^{-\rho} \frac{\rho^n}{n!}} = 1, \quad (3)$$

так как при увеличении числа каналов $n \rightarrow \infty$ сумма в числителе является разложением экспоненты в ряд Тейлора.

Фрактальные СМО с входящим пуассоновским потоком. Рассмотрим СМО $M/Pa/n/0$ и $M/Pa/\infty$ с интенсивностью входящего потока $\lambda = 8,333$, средним временем обслуживания заявок $b = 3,5$. На рис. 1 показаны вероятности потерь заявок p_l и стационарные вероятности состояний p_n , рассчитанные по формулам (1) и (2). Видно, что при увеличении числа каналов указанные выше вероятности уменьшаются по значению и приближаются друг к другу. Значения вероятностей p_l и p_n для некоторых n приведены в табл. 1.

По данным табл. 1 можно увидеть, что уже при $n = 53$ значения вероятностей потерь и стационарных вероятностей состояний совпадают с точностью до 4 значащих цифр.

Данный результат также подтверждается и имитационным моделированием (ИМ). На рис. 2 для вероятностей p_l и p_n были построены линии тренда, которые обозначены для p_l и p_n как y и y^* соответственно.

Высокие значения коэффициента детерминации R^2 говорят о достаточно хорошем приближении значений линии тренда к исходным данным.

Фрактальные СМО общего вида. Имитационные эксперименты для различных фрактальных СМО

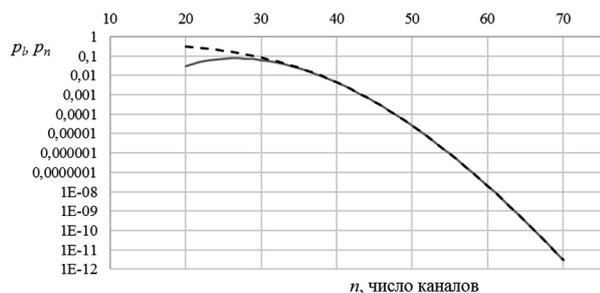


Рис. 1. Вероятности потерь p_l (пунктирная) и стационарные вероятности состояний p_n (сплошная) для систем $M/Pa/n/0$ и $M/Pa/\infty$

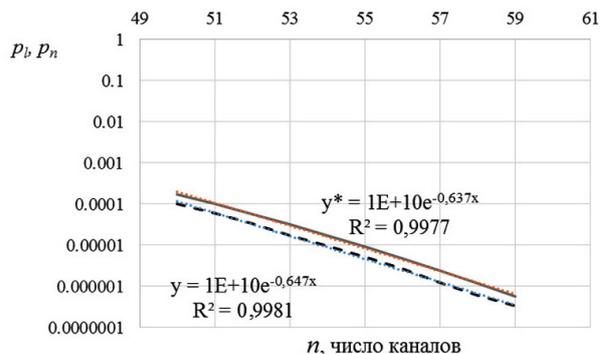


Рис. 2. Полученные путем ИМ вероятности потерь p_l (пунктирная) и стационарные вероятности состояний p_n (сплошная) для систем $M/Pa/n/0$ и $M/Pa/\infty$

Значения p_i и p_n

n	47	48	49	50	51	52	53
p_i	0,000151	8,50964E-05	4,71165E-05	2,55664E-05	1,36011E-05	7,0965E-06	3,63283E-06
p_n	0,000151	8,50879E-05	4,7114E-05	2,55657E-05	1,36009E-05	7,09645E-06	3,63282E-06

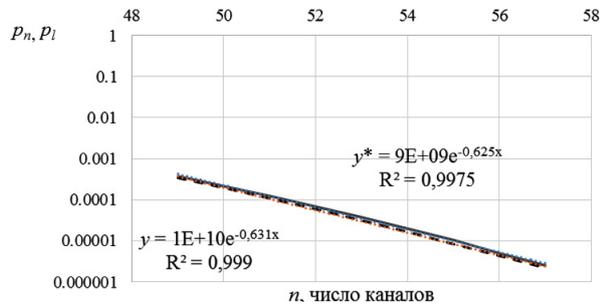


Рис. 3. Вероятности потерь p_i (пунктирная) и стационарные вероятности состояний p_n (сплошная) для систем $\text{Pa}_1/\text{Pa}_2/n/0$ и $\text{Pa}_1/\text{Pa}_2/\infty$

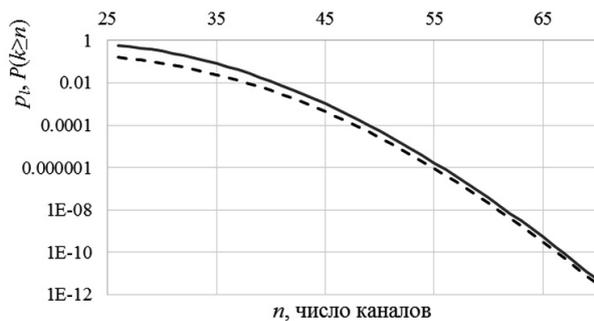


Рис. 4. Вероятности потерь p_i (пунктирная) и хвост $P(k \geq n)$ (сплошная) для систем $M/\text{Pa}/n/0$ и $M/\text{Pa}/\infty$, полученные из (1) и (2)

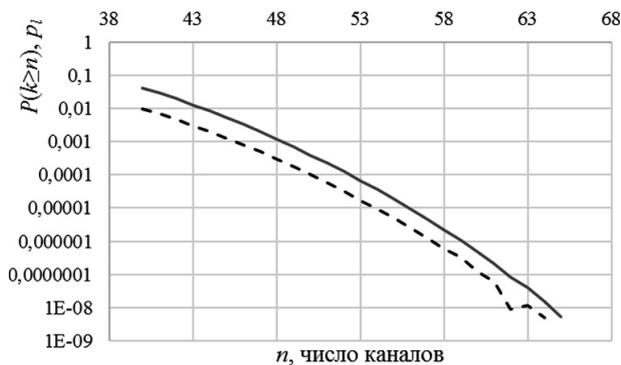


Рис. 5. Вероятности потерь p_i (пунктирная) и хвост $P(k \geq n)$ (сплошная) для систем $M/\text{Pa}/n/0$ и $M/\text{Pa}/\infty$, полученных путем ИМ

вида $GI/GI/n/0$, у которых входящий поток не является пуассоновским, показали, что вероятности потерь p_i заявок и стационарные вероятности состояний p_n также сходятся при увеличении n , когда вероятности потерь малы.

На рис. 3 изображены промоделированные системы $\text{Pa}_1/\text{Pa}_2/n/0$ и $\text{Pa}_1/\text{Pa}_2/\infty$, у которых значения параметров масштаба и формы равны $K_1=0,043$, $\alpha_1=1,5$ для Pa_1 и $K_2=1$, $\alpha_2=1,4$ для Pa_2 , интенсивность входящего потока $\lambda=7,75$, а среднее время обслуживания заявок $b=3,5$.

Стационарные вероятности состояний для фрактальных вероятности определялись как отношение времени пребывания t_n системы в состоянии n , к общему времени моделирования T , что можно записать следующим образом:

$$p_n = \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{i=1}^T I(k=n), \quad (4)$$

где $I(\cdot)$ — ступенчатая функция Хэвисайда, которая принимает значение 1, если условие (\cdot) выполняется, и значение 0 — в противном случае; k — это случайная величина, выражающая число занятых каналов.

Аппроксимация вероятности потерь p_i хвостом $P(k \geq n)$. В литературе встречаются случаи [9, 11], когда в качестве аппроксимации вероятности потерь заявок p_i в многолинейных СМО используют хвост $P(k \geq n)$, который выражает вероятность того, что в системе будет занято не меньше чем n каналов. При пуассоновском входящем потоке такая аппроксимация дает хорошую сходимость хвоста к вероятности потерь при их малых значениях. На рис. 4 для рассмотренных выше систем $M/\text{Pa}/n/0$ и $M/\text{Pa}/\infty$, изображены вероятность потерь p_i , рассчитанная по (1), и хвост $P(k \geq n)$ стационарных вероятностей состояний p_n , полученных по (2):

Данные выводы можно подтвердить аналитически. Рассмотрим сумму $\rho^n/n! + \rho^{n+1}/(n+1)! + \dots = \sum_{k=n}^{\infty} \rho^k/k!$. При всяком фиксированном $\rho > 0$ и достаточно большом n слагаемые данной суммы убывают столь быстро, что она может быть представлена в виде $\sum_{k=n}^{\infty} \rho^k/k! = (\rho^n/n!) \cdot (1 + \varepsilon)$, где ε очень маленькая величина. При $n \rightarrow \infty$ значение $\varepsilon \rightarrow 0$ и тогда можно сказать, что вся сумма равна своему первому члену:

$$\lim_{n \rightarrow \infty} \frac{\left(\frac{\rho^n}{n!}\right)}{\sum_{k=n}^{\infty} \frac{\rho^k}{k!}} = 1. \quad (5)$$

Теперь, на основании вышеизложенного, вычислим предел отношения вероятности $P(k \geq n)$ к вероятности потерь p_i :

$$\lim_{n \rightarrow \infty} \frac{P(k \geq n)}{P(k=n)} = \lim_{n \rightarrow \infty} \frac{\left(\sum_{i=n}^{\infty} \frac{\rho^i}{i!}\right) \cdot e^{-\rho}}{\frac{\rho^n}{n!} \left[\sum_{i=0}^n \frac{\rho^i}{i!}\right]^{-1}} =$$

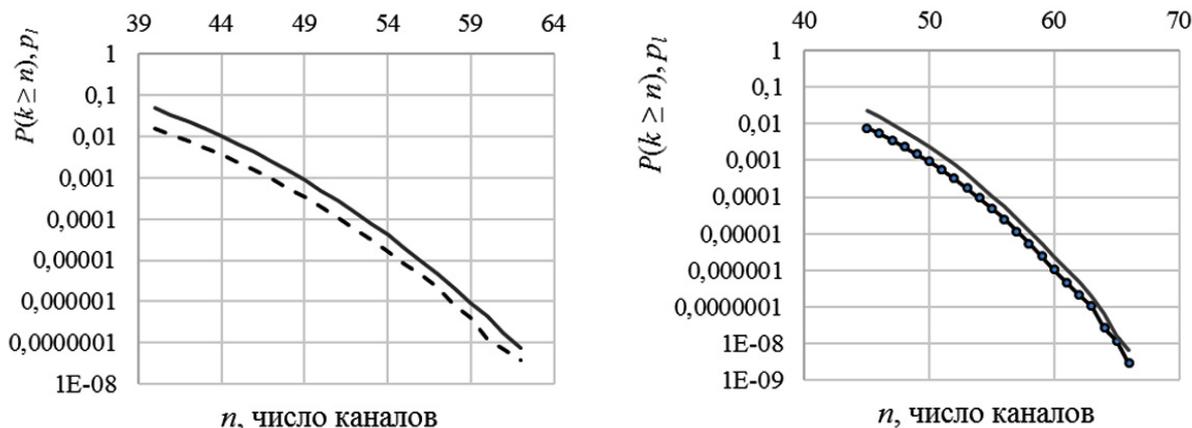


Рис. 6. Слева — вероятности потерь p_l (пунктирная линия) и хвост $P(k \geq n)$ (сплошная) для систем $Pa_1/Pa_2/n/0$ и $Pa_1/Pa_2/\infty$. Справа — p_l (маркированная сплошная) и хвост $P(k \geq n)$ (сплошная) для систем $Pa/\Gamma/n/0$ и $Pa/\Gamma/\infty$

$$\lim_{n \rightarrow \infty} \frac{\frac{\rho^n}{n!} (1 + \varepsilon) \cdot e^{-\rho}}{\frac{\rho^n}{n!}} \cdot \sum_{i=0}^n \frac{\rho^i}{i!} = \lim_{n \rightarrow \infty} (1 + 0) \cdot e^{-\rho} \cdot e^{\rho} = 1. \quad (6)$$

Таким образом, при больших значениях n хвост $P(k \geq n)$ сходится к вероятности $p_l = P(k = n)$ с относительной погрешностью ноль.

На рис. 5 представлены результаты ИМ для $M/Pa/n/0$ и $M/Pa/\infty$, подтверждающие данные выводы.

Как показали имитационные эксперименты, для фрактальных систем общего вида, когда распределение интервалов времени между поступлениями заявок не экспоненциальное, при увеличении n хвост $P(k \geq n)$ также сходится к вероятности p_l . На рис. 6 слева представлены результаты ИМ для рассмотренных выше систем $Pa_1/Pa_2/n/0$ и $Pa_1/Pa_2/\infty$, а справа — для систем $Pa/\Gamma/n/0$ и $Pa/\Gamma/\infty$ с параметрами $K=0,04$, $\alpha=1,5$ для Парето распределения, $\beta=1$ и $\alpha_G=3,6$ для гамма-распределения, и хвост $P(k \geq n)$ с интенсивностью входящего потока $\lambda=7,75$ и средним временем обслуживания $b=3,6$. В работе рассматривается гамма-распределение с плотностью вероятностей

$$f(t) = \frac{t^{\alpha_G}}{\beta^{\alpha_G} \Gamma(\alpha_G)} \cdot e^{-\frac{t}{\beta}}, \text{ где } \Gamma(\alpha_G) \text{ — гамма-функция.}$$

В [12] получено соотношение, выражающее асимптотическую зависимость хвоста $P(k > n)$ от числа каналов n для фрактальных СМО при $n \rightarrow \infty$, имеющую следующий вид:

$$P(k > n) \sim c_0 e^{-cn^2}, \quad (7)$$

где c_0, C — некоторые константы, свои для каждой СМО. В ходе анализа результатов имитационных экспериментов было обнаружено, что при больших n стационарные вероятности состояний p_n имеют аналогичную асимптотику.

Погрешность аппроксимации. При приближенной замене вероятностей потерь p_l вероятностями состояний p_n или хвостом $P(k \geq n)$ появляется погрешность, о которой можно судить по отношению p_n/p_l и $P(k \geq n)/p_l$. Обозначим отношение p_n/p_l как $\delta_1(n)$ и $P(k \geq n)/p_l$ как $\delta_2(n)$. Изучение поведения отношений $\delta_1(n), \delta_2(n)$ с ростом числа каналов n позволяет выработать определенные рекомендации

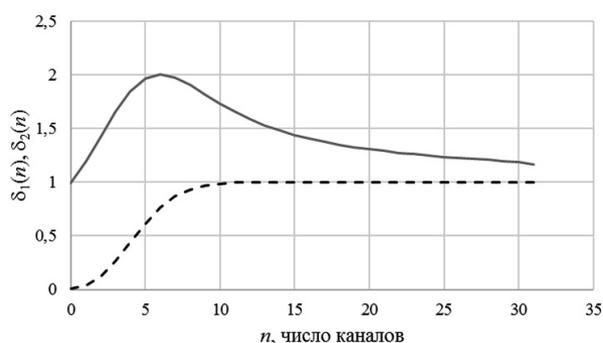


Рис. 7. Отношения $\delta_1(n)$ (пунктирная линия) и $\delta_2(n)$ (сплошная) для СМО вида $M/GI/n/0$ и $M/GI/\infty$ с $\rho=5$

по выбору аппроксимирующего выражения для вероятности потерь p_l . На рис. 7 представлены графики $\delta_1(n), \delta_2(n)$ для СМО вида $M/GI/n/0$ с коэффициентом загрузки $\rho=5$, полученные аналитически с помощью (1) и (2).

Поведение отношений $\delta_1(n), \delta_2(n)$ для других фрактальных СМО, как показывают имитационные эксперименты, имеют схожий характер. Однако для выработки конкретных рекомендаций по использованию $P(k \geq n)$ или p_n вместо p_l необходимы дополнительные исследования.

Заключение. В статье показано, что для определения вероятностей потерь в многолинейных фрактальных СМО, когда эти вероятности малы (от 10^{-4} и менее), можно использовать стационарные вероятности состояний p_n или хвосты $P(k \geq n)$ аналогичных бесконечнолинейных СМО. При этом, для нахождения условий выбора одной из предлагаемых аппроксимаций, требуется дополнительное исследование.

Разработанный метод определения вероятностей потерь p_l заявок в многолинейных фрактальных СМО позволяет значительно ускорять и упрощать процесс оптимизации распределения числа каналов по узлам фрактальных СеМО. Аппроксимация вероятностями состояний и хвостом $P(k \geq n)$ позволяет избегать многократных прогонов имитационных моделей фрактальных СМО для отдельных n , и получать необходимые значения вероятностей потерь за один прогон имитационной модели, что существенно сокращает время моделирования

СеМО, особенно когда число узлов в сети является большим.

В результате исследования обнаружено, что в фрактальных бесконечнолинейных СМО для стационарных вероятностей состояний p_n имеет место соотношение (6), которое установлено для хвоста $P(k > n)$ в [12] эмпирически.

Библиографический список

1. Олифер В. Г. Компьютерные сети. Принципы, технологии, протоколы. 4-е изд. СПб.: Питер, 2012. 944 с.
2. Кузьмин В. В. Модели и процедуры управления трафиком в мультисервисной сети оператора связи: дис. ... канд. техн. наук. Н. Новгород, 2015. 189 с.
3. Степанов С. Н. Основы телеграфика мультисервисных сетей. М.: Эко-Трендз, 2010. 392. ISBN 978-5-88405-092-1.
4. ITU-T Recommendation Y.1540 Internet protocol aspects – Quality of service and network performance; Internet protocol data communication service – IP packet transfer and availability performance parameters. URL: <https://www.itu.int/rec/T-REC-Y.1540/recommendation.asp?lang=en&parent=T-REC-Y.1540-201607-1> (дата обращения: 15.02.2017).
5. Crovella M. E., Taqqu M. S., Bestavros A. Heavy-tailed probability distributions in the World Wide Web // A practical guide to heavy tails. 1998. (1). P. 3–26.
6. Leland W. E. [et al.] On the self-similar nature of Ethernet traffic (extended version) // IEEE. ACM Transactions on networking. 1994. № 1 (2). P. 1–15.

7. Paxson V., Floyd S. Wide area traffic: the failure of Poisson modeling // IEEE. ACM Transactions on networking. 1995. № 3 (3). P. 226–244.

8. Li A., Whitt W. Approximate Blocking Probabilities in Loss Models With Independence and Distribution Assumptions Relaxed // Performance Evaluation. 2014. Vol. 80. P. 82–101.

9. Massey W.A., Whitt W. An analysis of the modified offered-load approximation for the nonstationary Erlang model // The annals of applied probability. 1994. № 4 (4). P. 1145–1160.

10. Srikant R., Whitt W. Simulation run lengths to estimate blocking probabilities // ACM Transactions on modeling and computer simulation. 1996. № 1 (6). P. 7–52.

11. Моисеев А. Н., Назаров А. А. Бесконечнолинейные системы и сети массового обслуживания. Томск: Изд-во НТЛ, 2015. 240 с.

12. Задорожный В. Н., Захаренкова Т. Р. Минимизация риска потери сообщений в сетях с фрактальным трафиком // Омский научный вестник. Сер. Приборы, машины и технологии. 2016. № 5 (149). С. 125–130.

ЗАХАРЕНКОВА Татьяна Романовна, аспирантка кафедры «Автоматизированные системы обработки информации и управления».

Адрес для переписки: ZakharenkovaTatiana@gmail.com

Статья поступила в редакцию 10.04.2017 г.

© Т. Р. Захаренкова

УДК 621.396.96:004.021

Н. Л. КНЯЗЕВ

Омский государственный
технический университет,
г. Омск

ОБРАБОТКА НАВИГАЦИОННЫХ ПАРАМЕТРОВ НА ОСНОВЕ АЛГОРИТМА НЕЧЕТКОЙ КЛАСТЕРИЗАЦИИ

Рассмотрена задача распределения летательных аппаратов (ЛА) по группам (кластерам) на основе обработки навигационных параметров. Предложен алгоритм кластеризации ЛА с использованием математического аппарата нечеткой логики. Разработанный алгоритм предоставляет возможность формировать количество кластеров в процессе своей работы. Показана эффективность предлагаемого алгоритма для точного кластерного анализа обнаруженных летательных аппаратов.

Ключевые слова: кластерный анализ, метрика, нечеткая логика, радиолокация.

1. Введение. Перед современными техническими средствами радиолокации, которые занимаются обнаружением и определением динамических объектов в пространстве, ставится задача обработки и определения наиболее существенных навигационных параметров летательных аппаратов (ЛА) (рассматриваемых в качестве *целей*) для распределения ЛА по группам (*кластерам*). Выделение целей из воздушного пространства происходит на основе анализа пространства и определения параметров движения объектов (*навигационных параметров*)

с использованием радиолокационных станций (РАС) [1]. Сложность решения такой задачи связана с неразличимостью объектов в заданном частотном диапазоне, а также с близостью моментов прихода импульсов от различных ЛА. Для решения проблемы предлагается использовать нечеткую кластеризацию на основе метода *c-means* [2], пригодного для решения подобных задач. Причиной актуальности данной проблемы является обширное развитие сетей мобильной связи поколения 4G, которые затрудняют обработку данных ЛА, т.к. работают

в том же диапазоне частотного канала, что вынуждает разрабатывать более эффективные алгоритмы определения динамических объектов в пространстве.

Летательные аппараты характеризуются следующими основными навигационными параметрами:

- несущей частотой f_c , определяющей, в каком диапазоне частот работает искомая цель;
- мощностью сигнала P , по которой можно оценить дальность объекта;
- длительностью импульса τ ;
- углом крена γ и углом тангажа ψ , необходимыми для определения положения ЛА относительно его продольных и поперечных осей.

2. Задача обработки навигационных параметров.

2.1. Постановка задачи. Пусть имеется последовательность навигационных параметров, представленных в виде вектора свойств объекта $x^p = (x_p^1, x_p^2, \dots, x_p^p, \dots, x_p^N)$, $x^p \in X$, где $p = 1, |X|$ — индекс входного объекта, ЛА, описываемого навигационными параметрами; x_i^p — i -е свойство ЛА с индексом p , $i = 1, N$; N — множество свойств объекта x^p ; X — множество навигационных параметров.

Данная последовательность навигационных параметров разбивается на кластеры, которые заранее не известны и должны формироваться динамически по мере поступления входных данных. Кластеры представлены в виде набора объектов из начальной выборки: $c_m = (x^1, \dots, x^i)$, $x^i \in X$ где $m = 1, M$, M — количество кластеров. Кластеры характеризуются своими центрами (или ядрами). Расстояние между объектом и кластером или между двумя кластерами может быть найдено при помощи евклидовой метрики, представляемой выражением [3]:

$$d(x^p, c_m) = \sqrt{\sum (x_i^p - c_m)^2}. \quad (1)$$

Каждый входной вектор объекта данных x^p должен быть отнесен к одному из имеющихся кластеров c_m .

2.2. Методы решения задачи обработки навигационных параметров. В большинстве современных систем обработки навигационных параметров, получаемых от РЛС, используется стандартная модель кластерного анализа k -means, позволяющая проводить четкую кластеризацию, с начальным указанием требуемого числа кластеров и количества итераций объединения. Однако данный подход не позволит решить проблему, связанную с выделением очень близких целей по несущей частоте f_c или по другим навигационным параметрам. В статье предлагается решение этой проблемы при помощи модификации метода нечеткого кластерного анализа c -means, заключающегося в распределении объектов по кластерам на основе вероятности принадлежности объекта тому или иному объекту [4].

Степень (вероятность) принадлежности объекта к тому или иному кластеру определяется с помощью функции принадлежности, принимающей значения из диапазона $\mu(i) \in [0, 1]$. Наибольшее распространение в технических приложениях получили ступенчатая аппроксимация и аппроксимация симметричной гауссовской функции принадлежности [5]:

$$\mu(x^p) = \exp\left[-\left(\frac{x_i^p - c_m}{2 \cdot \delta}\right)^2\right], \quad (2)$$

где δ — параметр настройки функции принадлежности, равный расчетному количеству кластеров на шаге инициализации. Степень принадлежности вектора x^p кластеру c_m определяется в соответствии с выражением

$$\mu(x^p) = \max[\mu(x_1^p), \mu(x_2^p), \dots, \mu(x_i^p), \dots, \mu(x_N^p)]. \quad (3)$$

2.3. Организация и корректировка кластеров входных данных. Корректировка количества кластеров M происходит после прихода каждого нового кластера. Для того чтобы учесть степень принадлежности объектов к двум ближайшим кластерам c_m и c_k , где $k \neq m$, $k \in M$, введем изменения в формулу (1) с учетом формулы (2). Тогда функция расстояния (метрика) между объектом и кластером примет следующий вид:

$$d(x^m, c_k) = \sqrt{\sum \mu(i) \cdot (x_i^m - c_k)^2}, x^m \notin c_k. \quad (4)$$

Требуется проверить необходимость создания нового кластера c_{M+1} или объединения двух старых. Для этого лучше всего использовать некоторую границу кластера в виде окружности (в двумерном геометрическом пространстве) с радиусом кластера R . *Радиусом кластера* называется расстояние между ядром кластера и максимально удаленным от ядра объектом, принадлежащим этому кластеру. Радиус кластера описывается следующим выражением:

$$R = \max(d(x^m - c_m)), x^m \in c_m.$$

Значения $\mu(c_m)$ возрастают при уменьшении расстояния между вектором данных объекта и кластером. Таким образом, чем больше значение функции принадлежности $\mu(c_m) > \mu(c_k)$, $k \neq m$, тем ближе расположены в пространстве кластеры c_m и c_k , а критерием объединения двух кластеров будет служить радиус кластера.

Чем больше удалены объекты данных от кластеров, тем меньше значение $\mu(c)$ будет для принятия решения об объединении.

2.4. Проблемы остаточных данных кластеризации. Предположим, требуется объединить в кластер два входных объекта с навигационными параметрами несущей частоты $x^1 = (f_{c_1}^1, \dots, f_{c_N}^1)$, $x^2 = (f_{c_1}^2, \dots, f_{c_N}^2)$ и при вычислении функций принадлежности (2), их степени оказались равны, т.е. $\mu(x^1) = \mu(x^2) = 0,5$. При близких значениях можно считать, что мерой близости является выражение: $\tilde{d} = |\mu(x^p) - \mu(x^k)|$.

Критерием объединения кластеров станет выражение:

$$\alpha = (|\min(\mu(x^1), \min(\mu(x^2)))| - 0,5) \geq \tilde{d}.$$

Однако после проведения кластеризации очередного входного вектора x^p некоторые кластеры могут не иметь принадлежащих им векторов объектов данных. Чтобы не загружать результаты выделения кластеризации навигационных параметров, они должны быть удалены из списка, т.е. удалены из множества кластеров (см. табл. 1). Для выявления пустых кластеров вводится переменная $V(m)$, удовлетворяющая равенству

$$\sum_{m=1}^M V(m) = L,$$

где L — длина выборки.

Расчетные значения критерия объединения близких кластеров

Кластеры	Мера близости кластеров \tilde{d}	Критерий объединения α	Принятие решения
Кластеры № 1 и № 2	0,8	$ (0,9 - 0,5) \leq 0,8$ или $ (0,1 - 0,5) \leq 0,8$	Не объединять
Кластеры № 2 и № 3	0,4	$ (0,9 - 0,5) \geq 0,4$ или $ (0,3 - 0,5) \leq 0,4$	Объединить
Кластеры № 1 и № 3	0,7	$V(3) = 0$	Пустой кластер

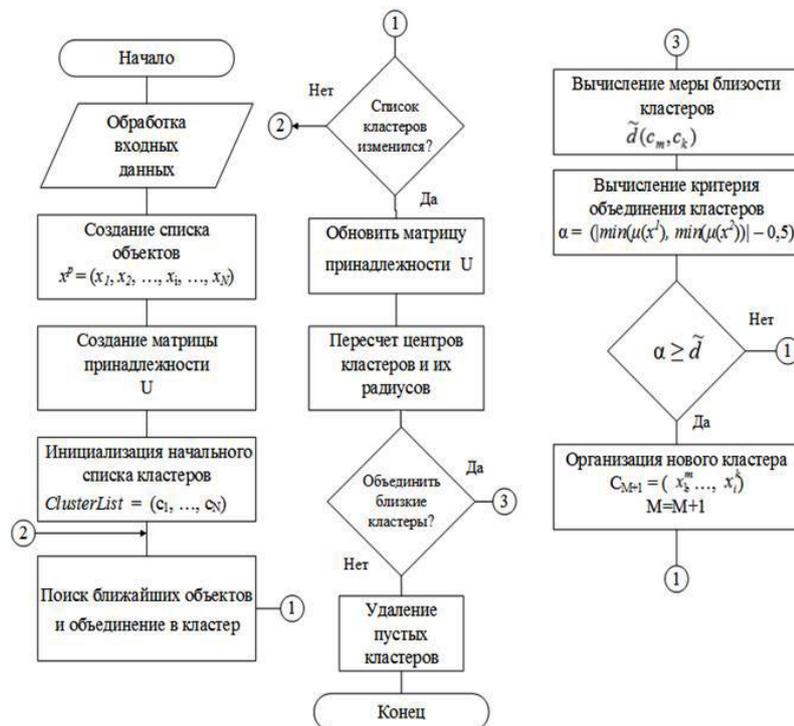


Рис. 1. Схема работы алгоритма CMEANS-CLUSTERVOID

Схема предлагаемого алгоритма CMEANS-CLUSTERVOID приведена на рис. 1, 2.

3. Результаты. Проведен сравнительный анализ работы двух методов кластеризации: k-means и предлагаемого метода CMEANS-CLUSTERVOID. В качестве исходных данных взяты навигационные параметры десяти летательных аппаратов, сформированные при помощи специального программного средства моделирования траектории полета ЛА. Для эксперимента использованы навигационные параметры: угол крена γ (рис. 3) и угол тангажа ψ (рис. 4). Алгоритм CMEANS-CLUSTERVOID реализован на языке программирования C#. Алгоритм k-means реализован при помощи инструментария программы STATISTICA 10 [6]. Проведен однократный расчет и выполнен кластерный анализ при помощи описанных выше методов (табл. 2).

Анализ результатов расчета показал, что различие между кластеризацией по углу крена (рис. 5) и углу тангажа (рис. 6) состоит в одном не найденном летательном аппарате. В первом случае (по углу крена) с использованием метода CMEANS-CLUSTERVOID выделено 10 ЛА, во втором (по углу тангажа) — 9 ЛА. При помощи метода k-means получено, что по углу крена обнаружено 9 ЛА, по углу тангажа — 8 ЛА.

Различие между углом крена и углом тангажа заключается в относительности измерения от оси ЛА. Однако объем выборки экспериментальных данных, полученных для параметра «угол крена», больше, чем для параметра «угол тангажа». Следовательно, угол крена является предпочтительным параметром для анализа. На основе полученных результатов сделан вывод о преимуществе

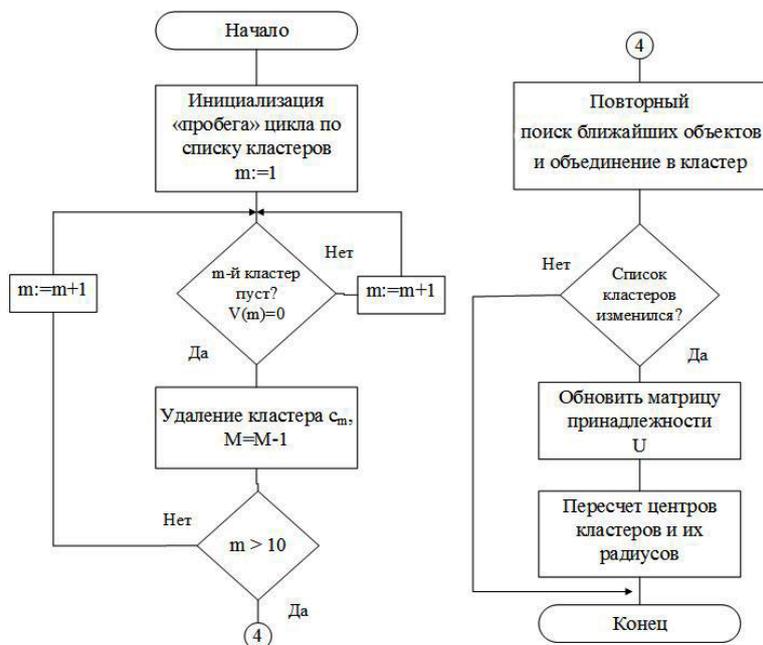


Рис. 2. Алгоритм удаления пустых кластеров

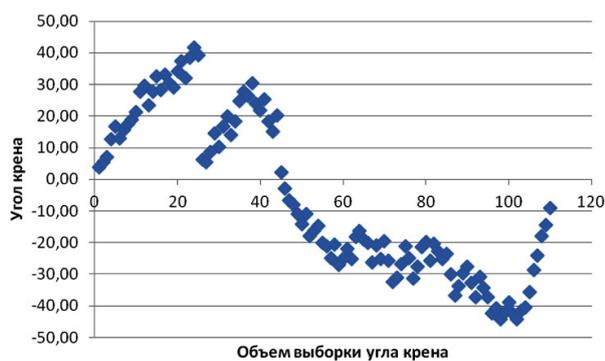


Рис. 3. Начальная выборка параметра угла крена (шаг инициализации)

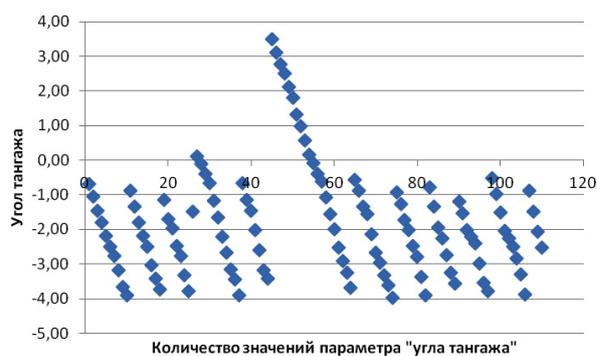


Рис. 4. Начальная выборка параметра угла тангажа (шаг инициализации)

Таблица 2

Сравнительный анализ работы алгоритмов кластеризации

Навигационные параметры	Количество кластеров	Количество итераций
Угол крена (k-means)	9	2
Угол крена (CMEANS-CLUSTERVOID)	10	12
Угол тангажа (k-means)	8	4
Угол тангажа (CMEANS-CLUSTERVOID)	9	13

разработанного алгоритма CMEANS-CLUSTERVOID (по сравнению с алгоритмом k-means), позволяющего определить большее количество ЛА аппаратов.

Заключение. Предложенный алгоритм кластеризации CMEANS-CLUSTERVOID является модификацией алгоритма нечеткой кластеризации с-means. Решена проблема остаточных данных кластеризации

при помощи удаления пустых кластеров и последующей реорганизации кластеров, пересчета их центров, радиусов и повторной проверки на необходимость объединения. Также алгоритм предоставляет возможность определять количество кластеров в процессе работы и завершать цикл кластерного анализа в случае, когда не осталось пустых класте-

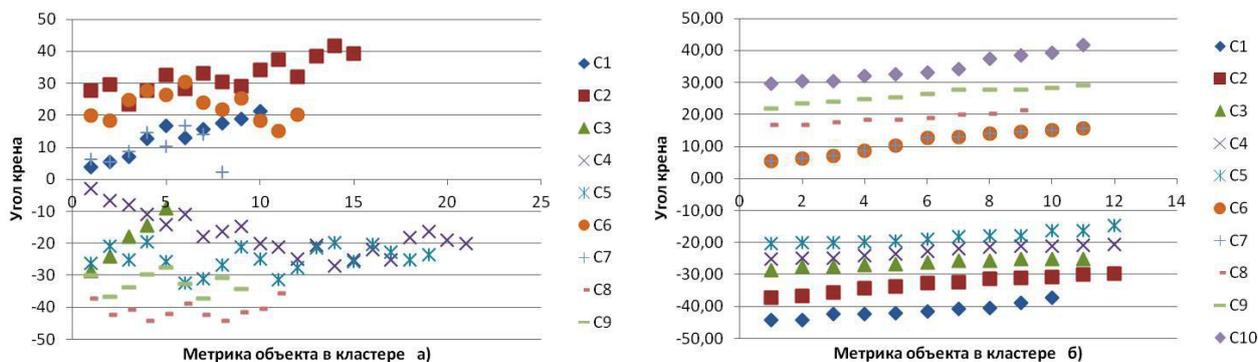


Рис. 5. Кластеризация по углу крена:
а — алгоритм k-means; б — алгоритм SMEANS-CLUSTERVOID

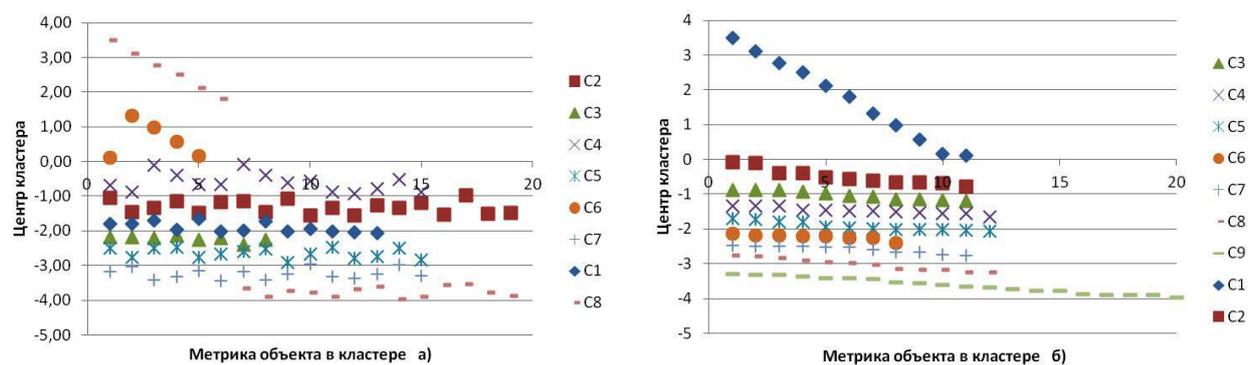


Рис. 6. Кластеризация по углу тангажа:
а — алгоритм k-means; б — алгоритм SMEANS-CLUSTERVOID

ров и отсутствует условие, необходимое для объединения остальных кластеров. С помощью модифицированного алгоритма SMEANS-CLUSTERVOID удалось выделить большее количество летательных аппаратов, чем с помощью алгоритма без модификации s-means, а также при использовании алгоритма k-means.

При анализе полученных результатов установлено, что алгоритм SMEANS-CLUSTERVOID выполняет кластерный анализ за большее число итераций, чем итерационный метод k-means. Данный факт говорит о том, что при наличии выборки большого объема для проведения кластерного анализа требуется больше времени. Однако с учетом темпов развития производительности современных компьютеров можно надеяться, что данный недостаток не будет серьезной проблемой.

Библиографический список

1. Бакулев П. А. Радиолокационные системы. М.: Радиотехника, 2004. 319 с. ISBN 5-931080-27-9.
2. Wolkenhauer O. Fuzzy Clustering. Hard-c-Means, Fuzzy-c-Means, Gustafson-Kessel. Manchester: UMIST, 2000. URL: <http://www.sbi.uni-rostock.de/dokumente/dataengineering/clustering.pdf> (дата обращения: 05.10.2016).

3. Котов А. Н., Красильников Н. Кластеризация данных. 2006. С. 13–15. URL: <http://logic.pdmi.ras.ru/~yura/internet/02ia-seminar-note.pdf> (дата обращения: 15.12.2016).

4. Князев Н. Л., Янишевская А. Г. Модификация алгоритма кластеризации целочисленных данных // Россия молодая: передовые технологии — в промышленность. 2015. № 3. С. 115–117.

5. Вешкурцев Ю. М., Бычков Е. Д., Титов Д. А. Цифровой фильтр на основе теории нечетких множеств с адаптивно изменяемыми функциями принадлежности // Известия вузов России. Радиоэлектротехника. 2007. Вып. 2. С. 43–50.

6. Statistica. Advanced analytics software package / Official WebSite. 2010. URL: <http://www.statistica.io> (дата обращения: 25.12.2016).

КНЯЗЕВ Никита Леонидович, аспирант кафедры «Автоматизированные системы обработки информации и управления» Омского государственного технического университета; инженер-программист АО «ЦКБА», г. Омск.

Адрес для переписки: sakylax@gmail.com

Статья поступила в редакцию 24.03.2017 г.

© Н. Л. Князев

ОПТИМИЗАЦИЯ ТРАНСПОРТНЫХ СЕТЕЙ В РЕЖИМЕ СВОБОДНОГО ДВИЖЕНИЯ АВТОТРАНСПОРТА

Рассматривается задача оптимального распределения ресурса по дорогам транспортной сети в режиме свободного движения. Для поиска решения используется метод множителей Лагранжа. Описывается пример оптимизации. Результаты оптимизации проверяются на имитационной модели транспортной сети с использованием программного продукта TraffOpt.

Ключевые слова: имитационное моделирование, транспортная сеть, ремонт дорог.

Введение. Проблема низкого качества покрытия дорог, а также их своевременного ремонта актуальна не только в России [1], но и во всем мире [2, 3]. Каждый год часть бюджета любого города закладывается на ремонт улично-дорожной сети (УДС). Эта часть бюджета является ресурсом, использовать который можно с различной эффективностью. Одним из критериев эффективности использования ресурса является изменение среднего времени проезда через УДС. При оптимальном распределении ресурса среднее время проезда через сеть приближается к своему минимальному значению. Задача оптимального распределения ресурса является задачей параметрической оптимизации. При этом варьируемыми параметрами являются доли ресурса, выделяемые на дороги УДС. В общем случае данная задача решается с помощью метода направляющих гипербола [4], позволяющего найти оптимальное распределение ресурса за 6–8 итераций [5]. Каждая итерация состоит из двух этапов: аналитическое решение задачи и уточнение найденного решения с помощью имитационного моделирования. В режиме свободного движения автотранспорта решить задачу оптимизации можно с помощью метода множителей Лагранжа [6], который в [7] используется для оптимизации высокорезервированных немарковских сетей с очередями.

Благодаря высокой точности получаемых результатов уточнение с помощью ИМ не требуется, что значительно уменьшает время, затрачиваемое на оптимизацию.

Постановка задачи. Задача оптимального распределения ресурса по дорогам УДС заключается в минимизации среднего времени E проезда через сеть:

$$E(\bar{v}) = \sum_{i=1}^n \alpha_i b_i = \sum_{i=1}^n \alpha_i \frac{l_i}{v_i + \varphi_i} \rightarrow \min_{\bar{\varphi}}, \quad (1)$$

где α_i — среднее число посещений i -й дороги транспортным средством (ТС), заходящим в сеть; b_i — среднее время проезда по i -й дороге; l_i — длина i -й дороги; v_i — средняя базовая скорость на i -й дороге; φ_i — дополнительная скорость на i -й дороге, определяемая количеством выделенного ресурса.

Коэффициенты α_i однозначно определяются из системы уравнений баланса:

$$\alpha_i = \sum_{j=0}^n \alpha_j p_{ji}, \quad i = 0, \dots, n, \quad \alpha_0 \equiv 1,$$

где p_{ij} — вероятность перехода с i -й дороги на j -ю.

Варьируемыми параметрами в задаче (1) являются значения дополнительных скоростей φ_i . При этом любая дорога транспортной сети обладает базовым ресурсом, который обеспечивает на ней скорость v_i . Перераспределять базовый ресурс по дорогам УДС невозможно. При решении задачи оптимизации предполагается, что из-за плохого качества покрытия средняя скорость v_i недостаточна, т.е. ТС не могут передвигаться с желаемой скоростью, и что выделение дополнительного ресурса повышает среднюю скорость на дороге.

Ограничением задачи оптимизации является суммарное количество выделенного на ремонт дорог ресурса Φ^* :

$$\Phi(\bar{\varphi}) = \sum_{i=1}^n l_i c_i \varphi_i^{\beta_i} = \Phi^* = const, \quad (2)$$

где c_i — стоимостный коэффициент, отражающий затраты ресурса на 1 км дороги для увеличения средней скорости на 1 км/ч; β_i — коэффициент нелинейности; $\Phi^* \geq 0$ — количество выделенного ресурса.

Решение задачи. Под режимом свободного движения автотранспорта понимается режим, в котором интенсивность на дороге настолько мала, что ТС не имеют никаких пространственных взаимодействий. Уравнения для оптимальных значений варьируемых параметров и для множителя Лагранжа λ , соответствующего ограничению (2), имеют вид:

$$\begin{cases} \frac{\partial E}{\partial \varphi_1} + \lambda l_1 c_1 \beta_1 \varphi_1^{\beta_1 - 1} = 0, \\ \dots \\ \frac{\partial E}{\partial \varphi_n} + \lambda l_n c_n \beta_n \varphi_n^{\beta_n - 1} = 0, \\ \sum_{i=1}^n l_i c_i \varphi_i^{\beta_i} - \Phi^* = 0. \end{cases} \quad (3)$$

Из (3), выполняя дифференцирование, получаем:

$$\begin{cases} -\frac{\alpha_1 l_1}{(v_1 + \varphi_1)^2} + \lambda l_1 c_1 \beta_1 \varphi_1^{\beta_1 - 1} = 0, \\ \dots \\ -\frac{\alpha_n l_n}{(v_n + \varphi_n)^2} + \lambda l_n c_n \beta_n \varphi_n^{\beta_n - 1} = 0, \\ \sum_{i=1}^n l_i c_i \varphi_i^{\beta_i} - \Phi^* = 0, \end{cases} \quad (4)$$

или, иначе,

$$\begin{cases} \lambda c_1 \beta_1 \varphi_1^{\beta_1 - 1} (v_1 + \varphi_1)^2 = \alpha_1, \\ \dots \\ \lambda c_n \beta_n \varphi_n^{\beta_n - 1} (v_n + \varphi_n)^2 = \alpha_n, \\ \sum_{i=1}^n l_i c_i \varphi_i^{\beta_i} = \Phi^*. \end{cases} \quad (5)$$

Отсюда находим уравнение для долей дополнительного ресурса:

$$\varphi_i = \sqrt{\frac{\alpha_i}{\lambda c_i \beta_i \varphi_i^{\beta_i - 1}} - v_i}, \quad i = 1, \dots, n. \quad (6)$$

В случае, когда $\bar{\beta} = (1, \dots, 1)$, искомое решение принимает вид:

$$\varphi_i = \sqrt{\frac{\alpha_i}{\lambda c_i}} - v_i, \quad i = 1, \dots, n. \quad (7)$$

Подставив выражение (7) и $\beta_i = 1$ в последнее уравнение системы (5), получим

$$\sum_{i=1}^n l_i c_i \left(\sqrt{\frac{\alpha_i}{\lambda c_i}} - v_i \right) = \Phi^*. \quad (8)$$

Таким образом, расчет множителя Лагранжа производится в соответствии со следующим выражением:

$$\lambda = \left(\frac{\sum_{i=1}^n l_i \sqrt{c_i \alpha_i}}{\sum_{i=1}^n l_i c_i v_i + \Phi^*} \right)^2. \quad (9)$$

Описание экспериментов. Решение задачи оптимизации было произведено на участке улично-дорожной сети (УДС) города Омска. Участок УДС включает в себя 9 узлов и 13 дорог. Схема участка приведена на рис. 1. При выполнении оптимизации данный участок УДС можно представить в виде ориентированного графа транспортной сети, вершинами которого являются перекрестки, а ребрами — дороги между перекрестками. Маршрутная матрица данного графа приведена в табл. 1. Для оптимизации данной УДС выделено $\Phi^* = 63820$ руб.

Рассмотрим ситуацию, в которой все стоимостные коэффициенты c_i одинаковы:

$$c_i = \text{const} = 1000 \frac{\text{руб.} \cdot \text{ч}}{\text{км}^2}, \quad i = 1, \dots, n. \quad (10)$$

Выделенный ресурс распределен равномерно на каждый километр транспортной сети, равномерное распределение ресурса по дорогам сети $\Phi_{\text{равн}}$ приведено в табл. 2. Применив распределение ресурса $\Phi_{\text{равн}}$ к УДС, находим по формуле (1), что значение среднего времени проезда через сеть составляет $E = 264,48$ секунд.

По формуле (7) рассчитано оптимальное распределение ресурса $\Phi_{\text{опт}}$ (табл. 2). Применяя распределение ресурса $\Phi_{\text{опт}}$, получаем значение среднего времени проезда через сеть $E = 247,6$ секунд.

Рассмотрим ситуацию, в которой стоимостные коэффициенты c_i принимают неодинаковые значения (табл. 3).

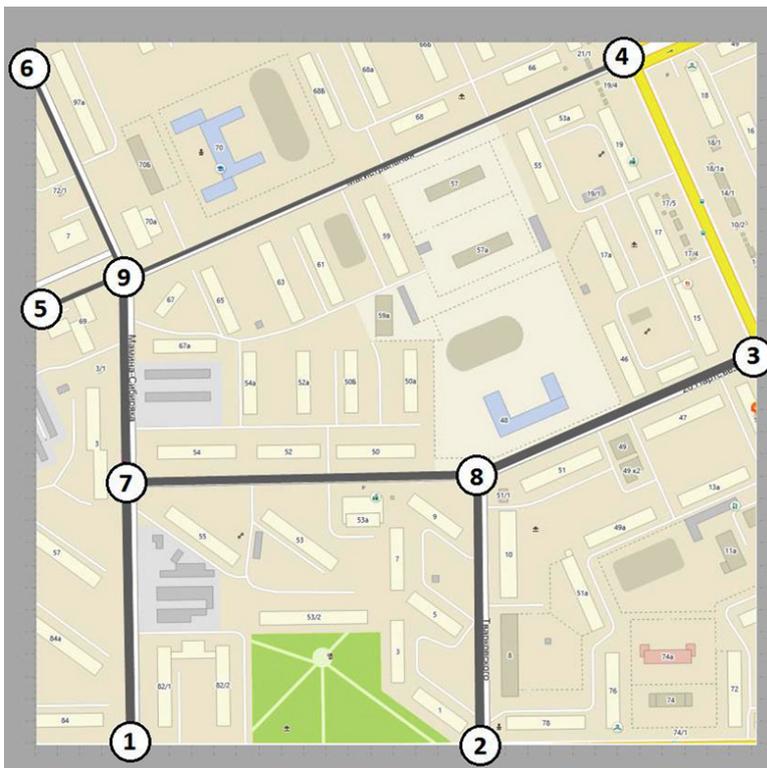


Рис. 1. Пример моделируемого участка УДС

Маршрутная матрица исследуемого участка

Номер вершины графа	1	2	3	4	5	6	7	8	9
1	0	0	0	0	0	0	1	0	0
2	0	0	0	0	0	0	0	1	0
3	0	0	0	0	0	0	0	1	0
4	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0
7	0,3	0	0	0	0	0	0	0,2	0,5
8	0	0,2	0,3	0	0	0	0,5	0	0
9	0	0	0	0,3	0,3	0,2	0,2	0	0

Таблица 2

Результаты оптимизации при одинаковых значениях стоимостных коэффициентов

Дуга графа	Стоимостные коэффициенты c_i , руб.·ч/км ²	Равномерное распределение ресурса $\Phi_{\text{равн}}$, руб.	Оптимальное распределение ресурса $\Phi_{\text{опт}}$, руб.
7,1	1000	4680	7035
2,8	1000	4840	5357
9,7	1000	5140	6837
7,8	1000	4680	5231
3,8	1000	6100	5007
1,7	1000	3520	5591
7,9	1000	4840	3577
8,7	1000	5140	5230
8,2	1000	6100	8900
8,3	1000	9680	6233
9,5	1000	4140	2665
9,6	1000	1440	624
9,4	1000	3520	1527

Таблица 3

Результаты оптимизации при разных значениях стоимостных коэффициентов

Дуга графа	Стоимостные коэффициенты c_i , руб.·ч/км ²	Равномерное распределение ресурса $\Phi_{\text{равн}}$, руб.	Оптимальное распределение ресурса $\Phi_{\text{опт}}$, руб.
7,1	399,42	4680	5600,78
2,8	414,04	4840	4518,12
9,7	440,58	5140	5755,43
7,8	1398,42	4680	6603,2
3,8	523,31	6100	4833,12
1,7	305,67	3520	3945,2
7,9	1412,04	4840	4443,52
8,7	1437,58	5140	6621,51
8,2	524,31	6100	7946,03
8,3	1824,09	9680	7667,38
9,5	1352,45	4140	3275,35
9,6	1122,59	1440	763,57
9,4	1299,67	3520	1846,8

Как и в предыдущем эксперименте до оптимизации, выделенный ресурс распределен равномерно на каждый километр транспортной сети. Равномерное распределение ресурса по дорогам транспортной сети $\Phi_{\text{равн}}$ приведено в табл. 3. Применяв это распределение ресурса, получаем значение среднего времени проезда через сеть $E = 208,6$ секунды.

В результате применения оптимального распределения ресурса $\Phi_{\text{опт}}$ (табл. 3) получаем $E = 202,94$ секунды.

На графиках, приведенных на рис. 2 и рис. 3, для выше описанных экспериментов наглядно показано изменение среднего времени проезда через сеть с ростом входной интенсивности транспортного потока. На этих графиках видно, что разница среднего времени проезда через сеть до и после оптимизации в результате выхода из режима свободного движения не уменьшается. Из этого можно

сделать вывод, что результаты оптимизации могут использоваться не только в режиме свободного движения автотранспорта. Графики на рис. 2 и рис. 3 получены с помощью имитационного моделирования сети средствами TraffOpt [8].

Заключение. В данной статье решена задача поиска оптимального распределения ресурса по дорогам УДС в режиме свободного движения автотранспорта. Для решения этой задачи использован метод множителей Лагранжа.

Показано влияние разных распределений ресурса на среднее время проезда через сеть. С помощью имитационного моделирования установлено сохранение результатов оптимизации, полученных в режиме малой загрузки дорог, после увеличения нагрузки.

В результате оптимизации среднее время проезда через сеть в среднем уменьшалось на 15 %

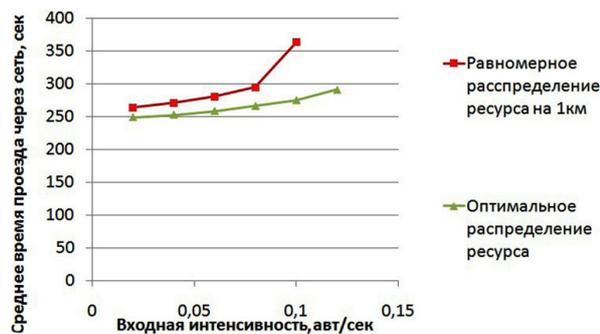


Рис. 2. Влияние входной интенсивности транспортного потока на результаты оптимизации при $c_i = \text{const}$

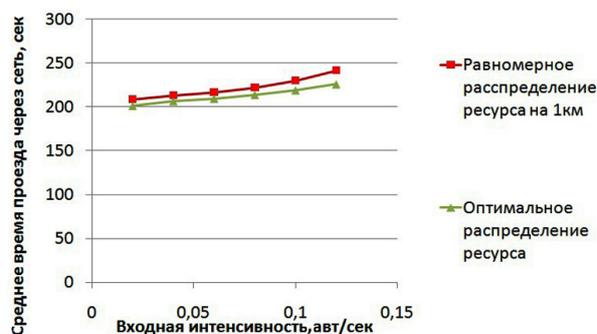


Рис. 3. Влияние входной интенсивности транспортного потока на результаты оптимизации при $c_i \neq \text{const}$

и 3 % по сравнению с равномерным распределением ресурса.

Для проверки результатов оптимизации был использован программный продукт TraffOpt. Транспортная модель создавалась с использованием [9, 10].

Библиографический список

1. Скрыпников А. В., Кондрашова Е. В., Скворцова Т. В. Метод оптимизации планов ремонта участков лесных автомобильных дорог // Современные проблемы науки и образования. 2011. № 6. С. 119.
2. Zhang T. Z., Lu Y. M. Study on simulation and optimization of the road rush-repair model after disaster // Applied Mechanics and Materials. 2011. Vol. 50. P. 298–303. DOI 10.4028/www.scientific.net/AMM.50-51.298.
3. Tang C. H., Yan S., Chang C. W. Short-term work team scheduling models for effective road repair and management // Transportation planning and technology. 2009. Vol. 32. №. 3. P. 289–311. DOI 10.1080/03081060903017150.
4. Zadorozhnyi V. N., Kornach M. A. Optimization of transport queuing networks on the basis of the method of directing hyperbole // International Siberian Conference on Control and Communications (SIBCON). Moscow, 12–14 May, 2016. DOI 10.1109/SIBCON.2016.7491716. URL: <http://ieeexplore.ieee.org/document/7491716> (дата обращения: 08.04.2017).
5. Задорожный В. Н., Корнач М. А., Пендер Е. А., Ганева М. И. Двухуровневый многомодельный подход к задачам

оптимизации транспортной инфраструктуры города // Омский научный вестник. Сер. Приборы, машины и технологии. 2015. № 1 (137). С. 189–193.

6. Корн Г., Корн Т. Справочник по математике для научных работников и инженеров / под. ред. И. Г. Арамановича. М.: Наука, 1973. С. 335–336.

7. Задорожный В. Н. Оптимизация высокорезервированных немарковских сетей с очередями // Омский научный вестник. Сер. Приборы, машины и технологии. 2014. № 3 (133). С. 21–25.

8. Задорожный В. Н., Корнач М. А. Агентный оптимизирующий симулятор TraffOpt: свидетельство о регистрации электронного ресурса. М.: ОФЭРНИО, 2015. № 20801 от 27.02.2015.

9. Хейт Ф. Математическая теория транспортных потоков: пер. с англ. / Под ред. И. Н. Коваленко. М.: Мир, 1966. 287 с.

10. Дрю Д. Теория транспортных потоков и управление ими: пер. с англ. / Под ред. Н. П. Бусленко. М.: Транспорт, 1972. 424 с.

КОРНАЧ Максим Анатольевич, аспирант кафедры «Автоматизированные системы обработки информации и управления».

Адрес для переписки: mkornach@gmail.com

Статья поступила в редакцию 10.04.2017 г.

© М. А. Корнач

ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ В РЕКЛАМНОЙ ДЕЯТЕЛЬНОСТИ

В статье рассмотрены экспертные системы, средства поддержки принятия решений, информационно-аналитические системы и сервисы в области рекламы, медиа- и PR-технологий с точки зрения использования рекламной информации, полученной из общей маркетинговой деятельности предприятия. В зависимости от поставленных целей рекламной кампании выделены функции рекламного менеджмента. На основе анализа современной литературы по маркетинговой деятельности предложен расширенный план проведения рекламной кампании, для каждого пункта этого плана сформулированы задачи рекламной стратегии, поставленные в соответствии выделенным функциям рекламного менеджмента. Проведенное исследование позволяет сделать вывод, что современные информационные системы не решают задачи по определению рекламной идеи (рекламного позиционирования) продукта. Это существенный недостаток, поскольку на основе рекламной идеи строится концепция всей рекламной кампании. Отсутствие рекламного позиционирования продукта ставит под сомнение эффективность вложения финансовых средств в рекламные мероприятия.

Ключевые слова: программное обеспечение, информационные технологии, рекламный менеджмент, интернет-реклама, медиа-планирование, рекламное позиционирование.

Введение. С точки зрения системного подхода реклама является одной из функциональных подсистем маркетинговой деятельности предприятия, которая, в свою очередь, является одной из составных частей целостной системы функционирования организации. Информационный процесс в маркетинговой деятельности включает в себя сбор и обработку информации для дальнейшей ее классификации. Анализ собранной и структурированной информации позволяет принимать эффективные решения в рыночной деятельности фирмы, тем самым уменьшая риски маркетинга. Эта же информация необходима и для осуществления рекламной деятельности. Взаимосвязь подсистемы информационного обеспечения рекламы с системой маркетинговой информации настолько велика, что во многих случаях невозможно выделить рекламную информацию из общей массы маркетинговой информации. В проведенном исследовании рекламная информация выделена из собственно маркетинговой информации и на основе этого проведена классификация соответствующих программных (информационных) систем.

Рекламный менеджмент. Чтобы классифицировать информационные системы и технологии, используемые в области рекламы, необходимо рассмотреть функции и основные задачи рекламной стратегии, определяющие план рекламной кампании (РК).

Основная цель рекламы — убедить потенциального клиента приобрести товары или услугу, предоставляемую рекламодателем, или тем или иным

образом отреагировать на то, что желает рекламодатель.

Основная цель рекламного менеджмента — содействие в решении задач продажи товаров или услуг с учетом условий внешней рыночной среды и возможностей предприятия.

Рассмотрим процесс рекламной деятельности с точки зрения рекламного менеджмента, функции которого раскрывают сущность и содержание процесса управления рекламной деятельностью.

Рекламный менеджмент — отрасль управления рекламой, которая заботится обо всех аспектах, связанных с рекламой и ее организацией, и основывается на применении принципов, методов, концепций и исследований в теории управления, маркетинга, коммуникаций и рекламной деятельности. Рекламный менеджмент включает в себя:

- маркетинговые исследования (которые помогают в создании рекламной стратегии);
- разработку общего плана РК;
- создание конкретного плана действий;
- запуск заверщенного проекта;
- оценку всех затрат и выгод, вовлеченных в РК.

Управление рекламой включает в себя различные специализированные подфункции, такие как медиа-стратегия, стратегия сообщения, медиа-планирование, покупка медиа, расчет эффективности РК.

Рекламный менеджмент, с точки зрения системного подхода, обеспечивает быстрый и выгодный процесс продажи товаров (услуг), решая такие

задачи, как привлечение внимания потенциальных потребителей к фирме или ее товарам и услугам, формирование положительного отношения к фирме или ее товарам и услугам, и приводит убедительные аргументы в пользу выбора потребителем именно этих товаров и услуг.

С точки зрения функционального подхода предлагается выделить в рекламном менеджменте следующие основные функции:

1. Информационная (информационное обеспечение рекламной деятельности).
2. Планирование (планирование рекламной деятельности и рекламных мероприятий).
3. Организационная (организация и руководство реализацией поставленных целей и задач по производству и размещению рекламы).
4. Контрольно-аналитическая (контроль рекламной деятельности, анализ и оценка проведенных рекламных мероприятий).

Главным средством организации, проведения и контроля рекламных мероприятий выступает план РК. В современной литературе по маркетинговой деятельности нет общепринятого плана РК. Многие авторы представляют этапы планирования по-разному. У авторов Е. Голубкова [1], П. Дойля [2], Ф. Котлера [3], Ж.-Ж. Ламбена [4] и М. В. Моховой [5] существуют отличия в определении стадий плана РК и их функционального содержания.

На основе анализа данных подходов [1–5] рассмотрим следующий расширенный и конкретизированный план проведения рекламной кампании:

1. Определение цели РК.
2. Определение и изучение целевой аудитории.
3. Определение предварительной суммы расходов на проведение РК.
4. Определение рекламной идеи, на основе которой строится концепция РК (рекламное позиционирование), объединяющая все мероприятия РК.
5. Определение средств рекламы и каналов ее распространения.
6. Разработка рекламных материалов и мероприятий.
7. Формирование сметы расходов РК.
8. Разработка графика выходов рекламных мероприятий с указанием сроков их проведения и оценка эффективности носителей.
9. Производство рекламоносителей. Закупка мест и эфирного времени средств массовой информации.
10. Проведение мероприятий по РК.
11. Расчет показателей эффективности и анализ проведенных мероприятий.

Этому плану сопоставим задачи, которые решают основные функции рекламного менеджмента:

1. Поиск и сбор информации.
2. Аналитическая обработка информации.
3. Использование информации в целях принятия управленческих решений и творческой разработки РК.
4. Формирование основных идей и целей РК.
5. Оценка бюджета на проведение РК.
6. Творческие разработки форм и видов рекламной продукции.
7. Составление сметы расходов.
8. Выбор конкретных СМИ.
9. Определение сроков проведения РК.
10. Написание рекламных сообщений и производство рекламных материалов.
11. Закупка медиа для распространения рекламной информации.

План рекламной кампании, функции и задачи рекламного менеджмента

№ пункта плана рекламной кампании	№ задачи рекламного менеджмента	Функции			
		Информационная	Планирования	Организационная	Контрольно-аналитическая
1 2	1	+			
	2	+			
	3	+			
4	4		+		
3	5		+		
6	6		+		
7	7		+		
5	8		+		
8	9		+		
9	10			+	
	11			+	
10	12			+	
11	13				+
	14				+
	15				+
	16				+

12. Проведение рекламных мероприятий.
13. Контроль осуществления рекламных мероприятий и расходования средств РК.
14. Комплексная оценка выполнения плана РК и расходования средств.
15. Заключительный анализ правильности выбора группы целевого воздействия.
16. Оценка эффективности РК.

В табл. 1 по пунктам плана РК приведены в соответствие задачи рекламного менеджмента, выполняющие соответствующие функции рекламного менеджмента.

В статье проведен анализ используемых информационных технологий в ПО («ПринтОфис24» и «ПринтОфис», «1С: Управление» «Google Doubleclick», «WS. Рекламное агентство», «Рекламное агентство» и др.). Эти программы разработаны для комплексной автоматизации компаний и предназначены, как правило, для решения задач в управленческой функции, подобно учетным программам BI, CRM (ERP) в маркетинге [6]. Данное ПО специализируется на различных видах рекламы и имеет большой перечень услуг, включающий в себя собственное производство, работу с поставщиками, подрядчиками, а также другие стандартные услуги: сопровождать базы данных с информацией

о клиентах рекламного агентства (контакты, заключенные с ними договоры и т.п.), автоматизировать прием и сопровождение заказов, контролировать платежи и задолженности, управлять своими ресурсами, формировать различного вида отчетность и многое другое.

При анализе информационных технологий в рассмотренных ПО, используемых собственно в рекламном деле, выделим три основных класса программного обеспечения:

- для управления интернет-рекламой;
- для медиа-планирования;
- для творческой составляющей рекламной деятельности.

Рассмотрим каждый класс программного обеспечения более подробно.

Сервисы для управления интернет-рекламой.

Интернет-реклама обладает множеством преимуществ, любое из которых может помочь компании достичь своих целей за меньшие деньги, чем ожидалось. К таким преимуществам относятся: быстрое и эффективное достижение целевой аудитории, интерактивный маркетинг, возможность рекламироваться на местных и национальных рынках, контролируемые расходы, возможность вносить оперативные изменения в рекламную кампанию, информационные ролики, постоянно меняющиеся инновационные идеи и цифровые медиа.

Основные типы интернет-рекламы: баннерная реклама, тизерная реклама, контекстная реклама, рекламные статьи, таргетированная реклама в социальных сетях, push-уведомления, реклама в мобильных приложениях, всплывающие окна (pop-up), реклама в e-mail, реклама на тематических сайтах, поисковые системы, интернет-аукцион, сети мгновенных сообщений [7].

Современной тенденцией является реклама на мобильных устройствах, которая весьма эффективна. Мобильная реклама имеет потенциал раздражать больше, чем большинство других видов рекламы, однако посты в социальных медиа с новостями воспринимаются более благосклонно. Чтобы воспользоваться уникальными преимуществами мобильной рекламы, рекламодатели должны сосредоточиться на взаимодействии с потребителем (вложить личный характер в рекламное сообщение на мобильном устройстве) и актуальности того, что предлагать людям (того, что они на самом деле ценят). Мобильные приложения дают возможность рекламным посланиям достигать пользователей, предлагая им то, что они хотят: хорошие быстрые приложения, которые можно бесплатно и легко скачать и которые будут работать без сбоев для всех версий программ, имеющих четкие описания в магазине приложений.

Программное обеспечение для проведения рекламных кампаний в социальных медиа — программные платформы, многие из которых являются облачными инструментами веб-браузеров и утилиты. С их помощью менеджеры рекламных кампаний в социальных медиа решают три наиболее важные задачи:

1. Мониторинг бренда.
2. Написание постов с рекламной информацией и одновременная их публикация на нескольких социальных площадках.
3. Аналитика и отчетность производительности рекламных кампаний в социальных медиа.

Существует достаточно много сервисов для анализа рекламных кампаний в сети Интернет: Alytics,

GIFTD, AdHands, Яндекс.Аудит, Revendo, Origami, Agency.WG, eLama, Outbrain, Quantcast и многие другие. Задачи, которые решаются данными сервисами:

- анализ ключевых слов, расширенного аудита целевого сегмента;
- автоматизация таргетированной рекламы (предназначенной исключительно той аудитории, которая заинтересована в показе объявления и в получении рекламируемого товара или услуги);
- управление контекстной рекламой;
- сбор данных и увеличение конверсии;
- автоматизация работы с тизерными сетями (в которых рекламодатели могут прорекламировать свои услуги или товары на сайтах веб-мастеров);
- анализ рекламы из социальных сетей;
- управление показами медийной рекламы;
- анализ рекламной статистики;
- e-mail и СМС-рассылка;
- средство организации контента, эффективного управления им;
- измерение демографических показателей аудитории и т.п.

Во всех сервисах для управления интернет-рекламой одним из главных показателей эффективности рекламы является конверсия, показывающая отношение числа пользователей, выполнивших целевые действия, к общему числу посетителей веб-ресурса.

Программы медиа-планирования. В процессе использования компьютерных технологий медиа-планерами можно выделить четыре основных направления развития исследований [8]:

- модели поиска и оценки;
- оптимизационные модели;
- имитационные модели;
- модели покупки площадей в средствах массовой информации.

В статье [9] авторами довольно подробно проведена классификация более тридцати компьютерных программ медиа-планирования, призванных помочь специалистам рекламных и медиа-агентств в планировании и размещении рекламы в средствах массовой коммуникации.

Рассмотрим проведенную классификацию компьютерных программ по следующим параметрам: по СМИ, в которых проводится медиапланирование, по категориям программного обеспечения, по медиа-планированию в медиа-миксе, по рассчитанным показателям охвата в медиа-планировании и по базе расчетов (табл. 2). Таким образом, программы по медиа-планированию предоставляют широкие возможности по созданию и редактированию расписаний выхода рекламы в эфир, используют уникальную технологию прогноза аудитории для созданных расписаний, который позволяет проводить дальнейший анализ и планирование рекламных кампаний.

Однако, как видно из табл. 2, большинство приведенных программных продуктов по медиапланированию привязаны к конкретным базам данных маркетинговых исследований, которые из-за разной методики сбора данных формируются отдельно для каждого типа СМИ и в результате позволяют проводить исследование эффективности только в рамках какого-то одного типа медиа.

Программы для решения творческих задач в рекламе. В данном разделе проведен анализ программных разработок системы консалтинговых фирм «ТРИЗ-ШАНС», основанных в 1992 году

Анализ программ медиа-планирования

№	Название программного продукта/сервиса	Excom Media Planer	Galileo	Media Master	Mediator	Zodiak	OptiMedia	DataFriend	VI	PR-Media	PaloMars Adex
		СМИ, в которых проводится медиа-планирование									
1	Газеты		+	+	+	+	+	+	+	+	+
2	Журналы		+	+	+	+	+	+	+	+	+
3	Радио	+		+	+	+	+	+	+	+	+
4	Телевидение	+		+	+	+	+	+	+	+	+
5	Наружная реклама			+	+						+
6	Интернет				+			+	+	+	
		Категория программного обеспечения									
1	СУБД, работающие с историческими данными	+	+	+		+	+	+			+
2	Медиа-калькуляторы				+						
3	Медиа-планеры	+	+	+		+	+	+		+	
4	СУБД с результатами мониторинга рекл. активности						+			+	+
5	ПО для составления календарных планов графиков размещения	+	+	+	+	+	+		+	+	
		Планирование в медиа-миксе СМИ									
1	Газеты	+		+		+		+	+	+	
2	Журналы	+		+		+		+	+	+	
3	Радио	+		+		+		+	+	+	
4	Телевидение	+		+		+		+	+	+	
		Рассчитываемые показатели									
1	Охват	+	+	+		+		+	+		
2	Количество рекламных контактов	+	+	+		+		+	+	+	
		База расчетов									
1	Анкеты		+	+		+	+				
2	Объявления									+	+
3	Рейтинги	+			+			+	+		
		Разработчик									
	Название фирмы, сайт фирмы	EXCOM	TNS	Arbitron Inc.	А. Г. Чернов	TNS	Клиент Сервис	COMCON-2	Видео Интернешнл	ГОРТИС	TNS

Соответствие ПО задачам рекламного менеджмента

№	Задача рекламного менеджмента	ПО для управленческих функций	ПО для управления интернет-рекламы	ПО для медиа-планирования	ПО для решения творческих задач
Информационная функция					
1	Поиск и сбор информации		+		
2	Аналитическая обработка информации		+		
3	Использование информации в целях принятия управленческих решений и творческой разработки РК		+		
Функция планирования					
4	Формирование основных идей и целей РК				
5	Оценка бюджета на проведение РК	+		+	
6	Творческие разработки форм и видов рекламной продукции				+
7	Составление сметы расходов	+			
8	Выбор конкретных СМИ	+		+	
9	Определение сроков проведения РК	+		+	
Организационная функция					
10	Написание рекламных сообщений и производство рекламных материалов				+
11	Закупка медиа для распространения рекламной информации			+	
12	Проведение рекламных мероприятий			+	+
Контрольно-аналитическая функция					
13	Контроль осуществления рекламных мероприятий и расходования средств РК	+			
14	Комплексная оценка выполнения плана РК и расходования средств	+			
15	Заключительный анализ правильности выбора группы целевого воздействия	+	+	+	
16	Оценка эффективности РК	+	+	+	

профессиональными разработчиками, консультантами и преподавателями Теории решения изобретательских задач (ТРИЗ) из России, Беларуси и Латвии, которые специализируются на разработках методик креатива и на решении нестандартных задач в различных областях деятельности. Были рассмотрены две программы «HeadLiner/Заголовщик» и «EXPO: 1001 Рекламоноситель» для решения творческих задач в рекламе.

Первая разработана с целью повысить эффективность создания автором заголовков, рекламных девизов, слоганов, лозунгов, эхо-фраз, образных фраз, метафор, текстов для баннеров [10].

Учитывая, что слоганы и заголовки часто являются видоизменением известных выражений, программа содержит 10 тематических баз данных с возможностью их просмотра при работе:

- изречения из Ветхого и Нового Завета;
- пословицы и поговорки русского языка;
- фразы из мульт- и кинофильмов;
- крылатые выражения и фразы;
- «перлы» российских политиков XIX – XX веков;
- современный жаргон;
- афоризмы Козьмы Пруткова, Ежи Леца и др.

Модельным объектом программы «EXPO: 1001 Рекламоноситель» является выставка. Эта программа подразумевает наличие уже готовой творческой стратегии рекламной кампании и помогает только в решении организационных задач [11].

Программа позволяет проектировать выставку, PR или промо-акцию «не с нуля», а имея некоторые проверенные ориентиры-решения.

Данные компьютерные программы являются инструментами для решения творческих задач, они помогают переопределять проблемы, находить новые идеи и предпринимать соответствующие действия в дальнейшем, преодолевать концептуальные блоки нашего разума и искать множество решений для эффективного решения проблемы, с которой сталкиваются сотрудники рекламных служб. Но для работы с данным типом ПО предполагается наличие уже готовой творческой стратегии, в рамках которой проходит весь комплекс рекламной кампании. Естественно, что готовую творческую стратегию требуется додумывать самостоятельно и «вручную».

Заключение. Результат проведенного исследования сведен в табл. 3. Задачи 1, 2, 3 (поиск и сбор информации, аналитическая обработка информации, использование информации в целях принятия управленческих решений и творческой разработки РК) относятся к информационным функциям и для их решения используется ПО для проведения маркетинговых исследований и поисковой оптимизации (аналитики) в интернет-рекламе. Задачи 4–9 (формирование основных идей и целей РК, оценка бюджета на проведение РК, творческие разработки форм и видов рекламной продукции, составление сметы расходов, выбор конкретных СМИ, определение сроков проведения РК) относятся к функциям планирования, которые решают ПО для управленческих функций, компьютерные программы для медиа-планирования и решения творческих задач в рекламе. Задачи 10–12 (написание рекламных сообщений и производство рекламных материалов, закупка медиа для распространения рекламной информации, проведение рекламных мероприятий) — из организационной функции и решаются посредством программ для медиа-планирования и решения творческих задач в рекламе. Для задач 13–16 (контроль осуществления рекламных мероприятий и расходования средств РК, комплексная оценка выполнения плана РК и расходования средств, заключительный анализ правильности выбора группы целевого воздействия, оценка эффективности РК), относящихся к контрольно-аналитической функции, используются ПО для управленческих функций, ПО для управления интернет-рекламой и ПО для медиа-планирования.

Как видно из табл. 3, практически для всех пунктов плана проведения рекламной кампании можно подобрать подходящий софт или сервис, кроме пункта, касающегося рекламной идеи (рекламного позиционирования) продукта. Таким образом, формирование основной идеи рекламной кампании существующие программные продукты для управления рекламой не решают, или решают только частично. К примеру, частичное решение этой задачи

возможно при использовании модели Росситера-Перси [12] для позиционирования торговой марки. Частное решение этой модели предложено в работе [13]. Но для решения задачи по определению рекламной идеи, на основе которой строится концепция всей рекламной кампании, требуется дополнительный новый аналитический аппарат.

Библиографический список

1. Голубков Е. П. Маркетинговые исследования: теория, методология и практика. М.: Финпресс, 1998. 416 с. ISBN 5-08001-0003-9.
2. Дойль П. Менеджмент: стратегия и тактика: пер. с англ. / под ред. Ю. Н. Каптуревского. СПб.: Питер, 2002. 560 с.
3. Котлер Ф. Основы маркетинга: пер. с англ. / под общ. ред. Е. М. Пеньковой. М.: Прогресс, 1990. 736 с.
4. Ламбен Ж.-Ж. Менеджмент, ориентированный на рынок: пер. с англ. / под ред. В. Б. Колчанова. 2-е изд. СПб.: Питер, 2014. 718 с. ISBN 978-5-496-00841-9.
5. Мохова М. В. Коммуникативная сущность рекламных кампаний // Современная экономика: проблемы и решения. 2010. № 7. С. 97–106.
6. Кульбида У. Н., Зыкина А. В. Анализ программного обеспечения в маркетинге // Прикладная математика и фундаментальная информатика. 2017. Т. 4. № 1. С. 68–87.
7. Заррелла Д. Интернет-маркетинг по науке: что, где и когда делать для получения максимального эффекта: пер. с англ. М.: Манн, Иванов и Фербер, 2014. 192 с. ISBN 978-5-00057-122-4.
8. Кочеткова А. В. Медиа-планирование. М.: РИП-холдинг, 2003. 174 с. ISBN 5-900045-44-7.
9. Турзай Ю. Н., Климин А. И. Исследование и классификация компьютерных программ медиа-планирования // Аудит и финансовый анализ. 2011. № 2. С. 431–443.
10. Headliner: Коллекция, база слоганов. Словарь крылатых выражений, фраз. URL: http://www.triz-chance.ru/h_demo.html (дата обращения: 27.03.2017).
11. Triz-chance триз-шанс. О программе 1001 Рекламоноситель. URL: <http://www.triz-chance.ru/expo1001.html> (дата обращения: 27.03.2017).
12. Росситер Дж. Р. Реклама и продвижение товаров: пер. с англ. / под ред. Л. А. Волковой. СПб.: Питер, 2001. 656 с. ISBN 5-8046-0115-6, 0-07-053943-X.
13. Кульбида У. Н., Зыкина А. В., Канева О. Н. Оптимизационный подход в медиа-планировании // Омский научный вестник. Серия Приборы, машины и технологии. 2014. № 3 (133). С. 38–40.

КУЛЬБИДА Ульяна Николаевна, ассистент кафедры «Прикладная математика и фундаментальная информатика».

Адрес для переписки: uni_form@mail.ru

ЗЫКИНА Анна Владимировна, доктор физико-математических наук, профессор (Россия), профессор кафедры «Прикладная математика и фундаментальная информатика», заведующая кафедрой «Прикладная математика и фундаментальная информатика».

Адрес для переписки: avzykina@mail.ru

Статья поступила в редакцию 10.04.2017 г.

© У. Н. Кульбида, А. В. Зыкина

МЕТОД МАШИННОГО ОБУЧЕНИЯ РАНЖИРОВАНИЮ НА БАЗЕ ПАРАЛЛЕЛЬНЫХ ГЕНЕТИЧЕСКИХ АЛГОРИТМОВ

Рассмотрена проблема машинного обучения ранжированию документов в поисковой выдаче с целью повышения качества информационного поиска. Предложен метод машинного обучения ранжированию на основе параллельных генетических алгоритмов, направленный на одновременное повышение точности ранжирования и скорости выполнения машинного обучения. Проведены исследования на тестовых коллекциях, подтвердившие эффективность разработанного метода. Исследование выполнено на электронно-вычислительной машине с 16-ядерным процессором, что позволило выполнять до 16 параллельных генетических алгоритмов одновременно. Показано значительное снижение времени, необходимого для выполнения машинного обучения ранжированию.

Ключевые слова: информационный поиск, машинное обучение ранжированию, оптимизация, генетические алгоритмы, параллельные алгоритмы, кооперативная коэволюция.

1. Введение. В современных системах информационного поиска наблюдается постоянный рост количества документов в поисковом индексе (коллекции документов, которыми оперирует система), вследствие чего возрастает актуальность машинного обучения (МО) ранжированию [1, 2]. На основе обучающих данных (набора запросов, каждый из которых связан со списком документов, отсортированным в соответствии с релевантностью), в ходе машинного обучения вычисляются коэффициенты для функции ранжирования [3], которая может быть использована для сортировки результатов поиска в дальнейшей работе поисковой системы.

Хотя повышение точности в ходе МО является основной задачей, скорость выполнения МО также является важным вопросом. Из-за разнообразия запросов и документов обучение включает в себя все большие и большие наборы данных с большим количеством факторов. Кроме того, широкое распространение получили методы искусственного повышения рейтинга документов в поисковой выдаче (множестве документов, которые поисковая система определила релевантными поисковому запросу) за счет оптимизации факторов релевантности, используемых в поисковой системе. Эти методы постоянно развиваются и совершенствуются, и, как правило, такие документы не являются релевантными информационному запросу, а значит, поисковая система должна периодически проходить повторное обучение ранжированию с использованием новых обучающих выборок и новых факторов релевантности.

В данной работе предлагается подход к решению проблемы машинного обучения ранжированию на основе параллельных генетических алгоритмов (ПГА), основанный на кооперативной коэволюции (КК). Предлагаемый алгоритм машинного обучения

ранжированию *LTR-CGA* может напрямую оптимизировать дискретные метрики качества ранжирования, а также за счет параллельного выполнения генетических алгоритмов, может сократить время исполнения процедуры МО в сравнении с традиционными генетическими алгоритмами (ГА) [4].

Кооперативная коэволюция представляет собой особый вид ГА, характеризующийся тем, что приспособленность индивида основывается на том, как хорошо он сотрудничает с особями из других популяций [4]. Алгоритмы КК построены на принципе «разделяй и властвуй», стратегии декомпозиции сложной проблемы на подзадачи и объединения их решений для того, чтобы сформировать полное окончательное решение.

В алгоритмах КК несколько ГА развиваются параллельно. Таким образом, процесс эволюции может быть естественным образом распараллелен, что позволяет добиться существенного повышения скорости МО. Эта возможность не была исследована ранее для задачи МО ранжированию. В данной работе исследуется возможность применения параллельного алгоритма КК в контексте МО ранжированию.

2. Постановка задачи машинного обучения ранжированию. Пусть D — коллекция документов, каждый из которых представлен вектором значений факторов релевантности. В информационно-поисковой системе для запроса q список документов D возвращается в виде поисковой выдачи, где документы отсортированы в соответствии с их релевантностью.

Для данного запроса q релевантность документов (на основе экспертных оценок) определяется как функция $rel: D \rightarrow N$, где N является натуральным числом, характеризующим степень релевантности документа запросу. В некоторых случаях используется

двоичная функция: релевантность документа 0 (не релевантный) или 1 (релевантный). В данной работе были рассмотрены 3 уровня релевантности: 0 (не релевантный), 1 (частично релевантный) и 2 (релевантный).

Пусть F является функцией ранжирования, которая определяет степень релевантности документов поисковому запросу. Качество функции ранжирования может быть оценено с помощью метрик, таких как точность на уровне первых n документов в поисковой выдаче $P@n$, средняя точность MAP (Mean Average Precision) и дискретная метрика $NDCG@n$ (Normalized Discounted Cumulative Gain). Таким образом, задача МО ранжированию формулируется следующим образом.

Дана обучающая выборка T и метрика качества ранжирования $S(f)$, задача машинного обучения ранжированию состоит в том, чтобы оптимизировать функцию ранжирования $f(T)$ так, что $S(f)$ является максимальной.

3. Предлагаемый метод машинного обучения LTR-CGA. Предлагаемый метод *LTR-CGA* адаптирует алгоритм КК к проблеме МО ранжированию, путем двухэтапного построения функции ранжирования на основе обучающей выборки.

Представление решения. В *LTR-CGA* используются древовидные структуры для представления решений [5], особи представлены в виде поддеревьев. Деревья не только имеют достаточную выразительную силу для представления нелинейных функций, но также имеют преимущество, заключающееся в простоте извлечения, внедрения и интерпретации решения.

В частности, для каждого дерева внутренние узлы содержат основные математические операторы $+$, $-$ и $*$. Узлы листьев содержат факторы

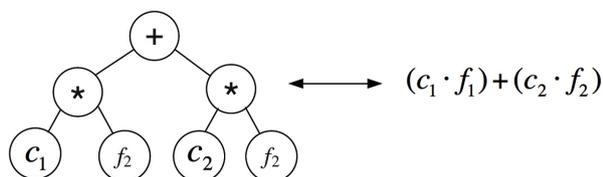


Рис. 1. Представление функции ранжирования в виде дерева

релевантности и константы. Константы служат в качестве весовых коэффициентов факторов релевантности в функции ранжирования f . В *LTR-CGA* весовые коэффициенты представлены рациональными числами на интервале $(0;1)$.

Таким образом, функция ранжирования представляется в виде дерева. Это необходимо для разделения целевой функции на компоненты, которые будут использоваться в параллельно эволюционирующих популяциях. Показано представление дерева (рис. 1) для примера с функцией ранжирования $(c_1 \cdot f_1) + (c_2 \cdot f_2)$.

Глубина d дерева, представляющего полное решение, определяется общим количеством факторов релевантности (nF) и констант (nC). Дерево должно быть достаточно глубоким, для того чтобы число конечных узлов было больше, чем $nF + nC$, т.е. $D = \lceil \log_2(nF + nC) \rceil + 1 = 7 + 1 = 8$. Например, пусть $nF = 46$ и $nC = 19$, то $d = \lceil \log_2(46 + 19) \rceil + 1 = 7 + 1 = 8$. Дерево глубиной 8 имеет $2^8 - 1 = 255 \geq (46 + 19)$ листовых узлов.

Этап декомпозиции. Оптимизационные алгоритмы на основе КК делят общее пространство параметров на подпространства параметров, относящихся к одной подпроблеме. Подобная декомпозиция корректна, потому что их пространство поиска — это декартово произведение параметров [4].

В предлагаемом алгоритме *LTR-CGA* начальных решений представлены в виде древовидных структур, имеющих глубину d . Начальные решения генерируются случайно на всем пространстве поиска. Затем каждое дерево разбивается на N поддеревьев, образуя при этом N популяций, по L особей в каждой.

Каждой популяции присвоен отдельный ГА для эволюции. Показана декомпозиция (рис. 2) одного дерева (слева) в $N = 4$ поддеревья (справа).

Глубина поддеревьев d_i задает верхние пределы пространству параметров особей. Этот параметр используется в *LTR-CGA* всякий раз, когда формируются новые особи. Глубина сборщика определяется как $d_A = \lceil \log_2 nP \rceil$, где nP — число процессоров, используемых в процессе параллельной эволюции. Таким образом, $d_i = d - d_A$, из чего следует, что размер пространства параметров особей ограничен значением $2^{d_i - 1}$. Например, предположим, что глубина деревьев решений $d = 8$. Пусть число процессоров $nP = 8$. Тогда, глубина

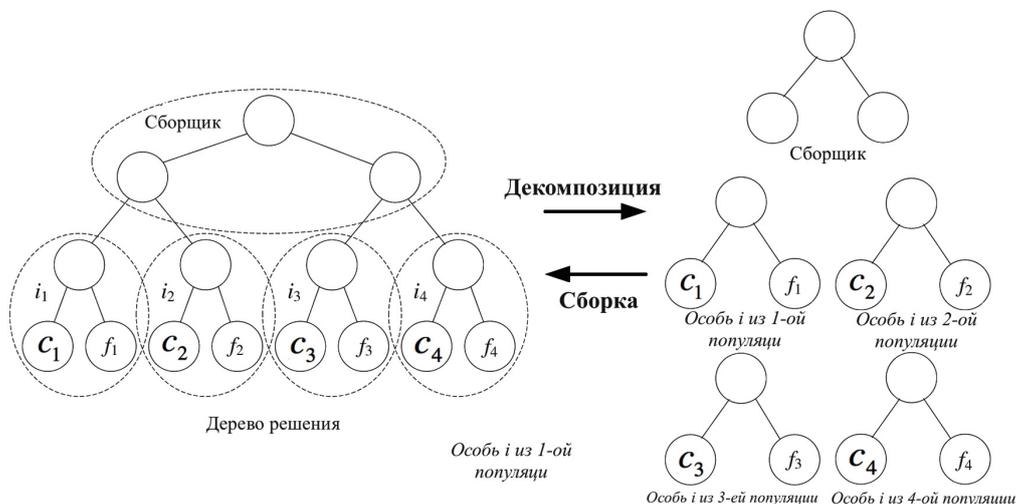


Рис. 2. Декомпозиция и сборка в методе LTR-CGA

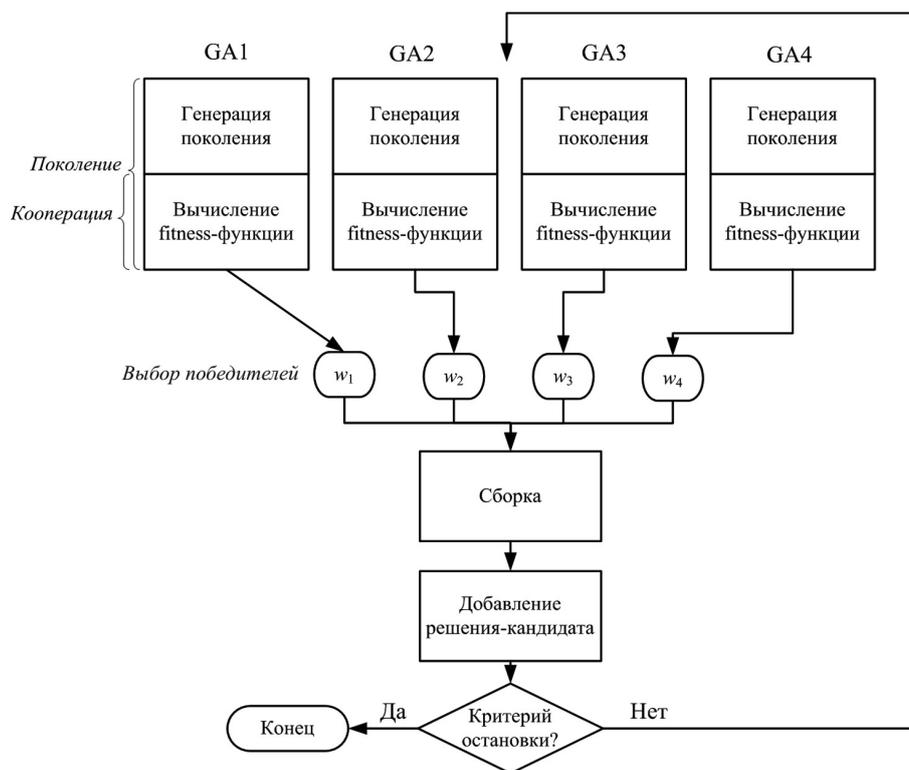


Рис. 3. Параллельные генетические алгоритмы в методе LTR-CGA

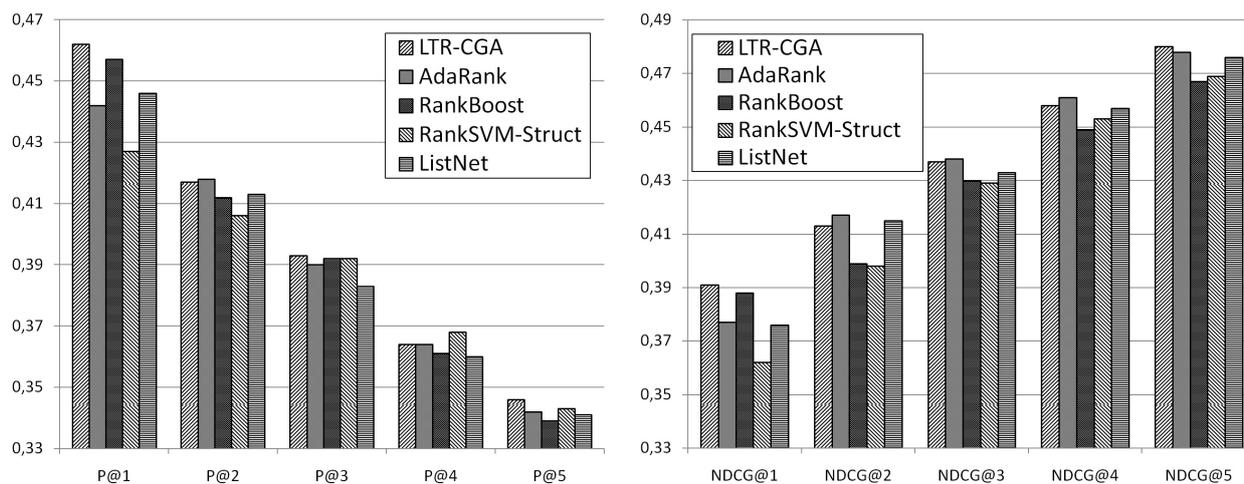


Рис. 4. Сравнительная оценка метрик качества ранжирования по метрикам P@n и NDCG@n на тестовой коллекции MQ2007

сборщика $d_A = \lceil \log_2 8 \rceil = 3$, а глубины особей $d_i = 8 - 3 = 5$.

Этап эволюции. Эволюция в предлагаемом алгоритме *LTR-CGA* выполняется итеративно, а число итераций predetermined установлено в начале параметром ГА. N популяций эволюционируют параллельно, каждая из которых поддерживает набор особей и победителя, который является лучшей особью в поколении с наибольшим значением функции приспособленности.

Показан итеративный процесс эволюции (рис. 3) для 4-х популяций в параллельных генетических алгоритмах *GA1*, *GA2*, *GA3* и *GA4* соответственно. В ходе каждой итерации новое поколение особей в каждой из популяций формируется на основе предыдущего поколения. Затем рассчитываются значения функции приспособленности для всех особей.

После генерации параллельное выполнение временно приостанавливается и создается решение-кандидат для текущего поколения. Вначале выбирается новый победитель в каждой популяции на основе вычисленных значений функции приспособленности. Затем все победители комбинируются с помощью сборщика в качестве кандидата в решение. Если предустановленное ограничение на число итераций не достигнуто, начнется новая итерация и 4 популяции продолжат эволюцию в новом поколении параллельно.

Этап сборки. Сборка происходит в двух случаях: вычисление фитнес-функции (в самом конце поколения, как показано на рис. 4) и генерация кандидата (сразу после окончания поколения, как показано на рис. 3). Сборка представляет собой процесс обратный декомпозиции. Особи (рис. 2) $i_1 \sim i_4$

(справа) объединяются в полное решение (слева) с помощью сборщика. Сборщики используются для объединения особей в общее решение. Таким образом, все узлы сборщика являются внутренними узлами дерева решений и содержат только математические операторы.

Вычисление функции приспособленности.

В рамках алгоритмов КК приспособленность особей основана на том, как хорошо они сотрудничают с другими популяциями. Во-первых, особь i_2 и победители из остальных популяций w_1 , w_3 и w_4 , выбранные из прошлого поколения $GA1$, $GA3$ и $GA4$ соответственно, с помощью сборщика объединяются в решение (справа). Затем на основе тестовых данных рассчитывается метрика качества ранжирования, например, MAP , а результат присваивается особи i в качестве фитнес-функции.

4. Тестовые коллекции. В проведенных исследованиях в качестве обучающей выборки и эталонных значений релевантности использована тестовая коллекция документов и запросов *LETOR 4.0* [6], выпущенная в 2009 году исследователями в области машинного обучения ранжированию из *Microsoft Research Asia* [6]. *LETOR 4.0* использует набор веб-страниц *Gov2* и два набора запросов *MQ2007* и *MQ2008*, которые применялись на конференциях *TREC 2007* и *TREC 2008* соответственно. *MQ2007* содержит около 1700 запросов с 69,623 ссылками, *MQ2008* — около 800 запросов с 15,211 ссылками. Каждый набор данных был разбит на пять частей, чтобы провести 5-кратную перекрестную проверку. Для каждого из исходных наборов три части используются для обучения, одна часть для проверки, а оставшаяся часть — для тестирования.

Факторы релевантности. В тестовой коллекции *LETOR 4.0* для каждого документа предоставляется: 6 факторов релевантности на основе гиперссылок (*PageRank*, количество внешних ссылок, количество внутренних ссылок, количество слешей в URL адресе документа, длина URL адреса и количество дочерних страниц), 40 факторов на основе содержимого документа, состоящие из 20 классических функций, таких как длина документа и частота термов, и 20 высокоуровневых факторов, результаты вероятностной модели *BM25* и алгоритмов *LMIR*.

Метрики качества ранжирования. Для оценки метода *LTR-CGA* из списка общепринятых метрик качества ранжирования документов поисковыми системами выбраны две основные — $MAP@n$ и $NDCG@n$.

1. Средняя точность ранжирования ($MAP@n$).

Точность ранжирования на уровне n первых документов может быть записана как (1):

$$P@n = r/n, \quad (1)$$

где r — количество найденных релевантных документов; b — количество документов, ошибочно принятых за релевантные.

На основе (1) может быть рассчитана средняя точность ранжирования MAP (2):

$$MAP = \frac{\sum_n^N (P@n \cdot rel(n))}{r}, \quad (2)$$

где $rel(n)$ — бинарная функция, принимающая значение 1, если n -й документ релевантен запросу, и 0 — в обратном случае.

2. Дискретная метрика DCG может быть записана как:

$$DCG@n = \sum_{p=1}^n \frac{2^{grade(p)} - 1}{\log_2(2 + p)}, \quad (3)$$

где $grade(p)$ — средняя оценка релевантности, выставленная экспертами документу, расположенному на позиции p в списке результатов, $grade \in [0,3]$; $1/\log_2(2 + p)$ — коэффициент, зависящий от позиций документа (первые документы имеют больший вес);

Тогда нормализованная версия $NDCG$ запишется в виде:

$$NDCG@n = DCG@n / Z, \quad (4)$$

где Z — фактор нормализации, который равен максимально возможному значению $DCG@n$ для данного запроса (т.е. равен DCG идеального ранжирования).

5. Результаты исследований. Проведены две серии экспериментов с использованием тестовых данных для оценки точности ранжирования и скорости работы метода *LTR-CGA*.

Точность ранжирования. В первой серии экспериментов оценивалась точность ранжирования, достигаемая с помощью предлагаемого алгоритма *LTR-CGA*, в сравнении с известными алгоритмами машинного обучения *AdaRank*, *RankBoost*, *RankSVM* и *ListNet* на тестовых коллекциях *MQ2007* [6] и *MQ2008* [6]. В табл. 1 показано сравнение точности по метрике MAP . На тестовой коллекции *MQ2007* предлагаемый алгоритм *LTR-CGA* и *RankBoost* показали лучшие результаты. Для коллекции *MQ2008* алгоритм *LTR-CGA* превзошел остальные алгоритмы на 1,13 %, 0,901 %, 2,60 % и 0,901 % соответственно, что свидетельствует об эффективности предложенного метода.

На рис. 5 и 6 показан результат сравнения алгоритмов по метрикам $P@1-5$ и $NDCG@1-5$ (метрики измерялись на уровне от 1 до 5 первых документов, так как точность ранжирования первых документов поисковой выдачи наиболее важна для поисковой системы [2]). Для этих метрик результаты, анало-

Таблица 1

Сравнение разработанного метода машинного обучения ранжированию с существующими алгоритмами по метрике MAP (средняя точность)

Тестовая коллекция	Средняя точность ранжирования (для алгоритмов)				
	<i>LTR-CGA</i>	<i>AdaRank</i>	<i>RankBoost</i>	<i>RankSVM</i>	<i>ListNet</i>
<i>MQ2007</i>	0,466	0,458	0,466	0,465	0,465
<i>MQ2008</i>	0,482	0,476	0,478	0,47	0,478

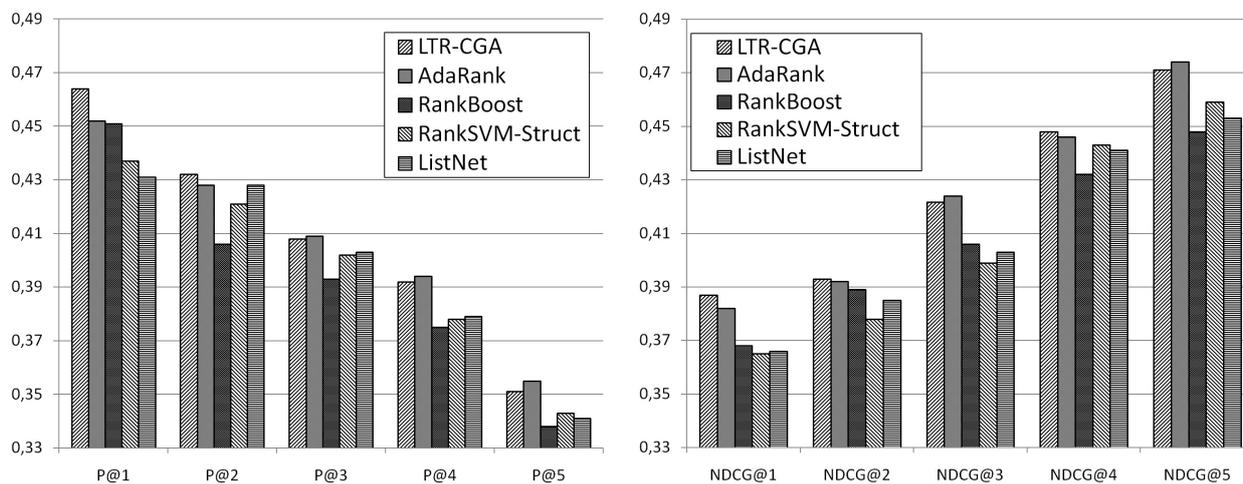


Рис. 5. Сравнительная оценка метрик качества ранжирования по метрикам P@n и NDCG@ на тестовой коллекции MQ2008

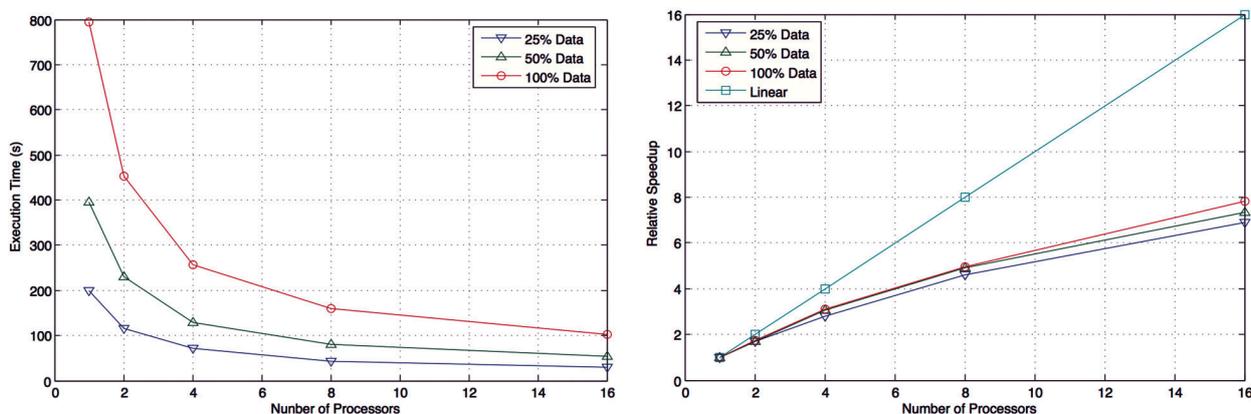


Рис. 6. Оценка времени выполнения машинного обучения методом LTR-MGA

гично результатам по метрике MAP, показывают что *LTR-CGA* является лучшим для обеих метрик, но достигающим того же результата, что и алгоритму *RankBoost* на коллекции *MQ2007* и алгоритму *AdaRank* на коллекции *MQ2008*.

Время выполнения алгоритма. Для демонстрации повышения скорости выполнения МО за счет параллельной эволюции из тестовой коллекции *MQ2008* были последовательно извлечены 25 %, 50 % и 100 % данных, на основе которых подготовлены три обучающие выборки, содержащие 3803, 7606 и 15211 документов соответственно. Затем алгоритм *LTR-CGA* выполнялся для каждого из этих наборов с разным количеством процессоров (1, 2, 4, 8, и 16). Предложенная реализация параллельных алгоритмов для задачи машинного обучения ранжированию использует 8 параллельно выполняющихся генетических алгоритмов, а значит, с ростом количества процессоров до 8 сокращается время выполнения МО (за счет параллельного выполнения вместо последовательного).

Приведены графики (рис. 6) оценок времени выполнения и повышения скорости выполнения машинного обучения алгоритма *LTR-CGA* за счет увеличения числа используемых ядер процессора. Из результатов видно, что параллельная эволюция приводит к значительному приросту скорости в *LTR-CGA*.

По сравнению с использованием одного процессора средний прирост скорости составил 173 %,

299 %, 486% и 736 % для 2, 4, 8, и 16 процессоров соответственно.

Следует отметить, что в идеальной ситуации (без негативного влияния внешних факторов), с увеличением размера набора данных кривая повышения скорости выполнения алгоритма должна приблизиться к линейно нарастающей функции (рис. 6). Однако *LTR-CGA* испытывает трудности в достижении подобных результатов по следующим причинам.

Во-первых, составные части *LTR-CGA* не всегда выполняются параллельно (рис. 3). После каждого поколения, параллельное выполнение частей алгоритма временно приостанавливается для выполнения сборки, чтобы построить предполагаемое решение. Во-вторых, параллельные генетические алгоритмы могут потратить разное количество времени на эволюцию в разных популяциях КК, но сборка может начаться только после того, как итерация завершена во всех ГА. Таким образом, время, которое алгоритм *LTR-CGA* тратит на поколение, равно наибольшему времени, затраченному на эволюцию среди всех параллельных ГА. Однако, несмотря на описанные выше причины, предложенный алгоритм *LTR-CGA*, за счет параллельных вычислений, с увеличением числа процессоров показывает рост скорости выполнения алгоритма, близкий к линейному, и превосходит существующие алгоритмы, которые построены на основе последовательных вычислений.

7. Заключение. В данной работе предложен метод и представлен алгоритм машинного обучения ранжированию *LTR-CGA*, применяющий параллельные генетические алгоритмы на основе кооперативной коэволюции, позволяющий повысить скорость выполнения алгоритма без потерь в точности ранжирования. Также описаны разработанные приемы, позволяющие кооперативной коэволюции в ГА выполняться параллельно.

Проведены испытания на тестовых коллекциях документов и запросов с целью сравнить предлагаемый алгоритм *LTR-CGA* с эталонными алгоритмами МО ранжированию, в ходе чего была продемонстрирована точность и значительное повышение скорости выполнения МО. Предложенный метод *LTR-CGA*, в отличие от существовавших ранее методов, использует преимущество многоядерных процессоров и позволяет значительно снизить время, необходимое на выполнение машинного обучения пропорционально росту числа ядер ЦПУ (в рассмотренном случае — скорость увеличилась примерно в 8 раз с использованием восьмиядерного процессора).

В дальнейшем планируется расширить предложенный метод машинного обучения *LTR-CGA*. В проведенных исследованиях в качестве критериев оптимальности в ходе машинного обучения использовались метрика точности и дискретная метрика *NDCG*.

Для получения наиболее высоких результатов машинного обучения предполагается провести сравнительный анализ метрик качества ранжирования, чтобы определить наиболее эффективные для задачи машинного обучения и определить, какие из них наиболее эффективны для оптимизации в ходе машинного обучения ранжированию.

Книжная полка

Батенькина, О. В. Подготовка и защита выпускных квалификационных работ по магистерской программе «Информационные системы в медиаиндустрии»: учеб. электрон. изд. локального распространения: учеб. пособие для вузов по направлению подгот. магистра 09.04.02 «Информационные системы и технологии (230400)» / О. В. Батенькина, Л. М. Дмитриева. — Омск: Изд-во ОмГТУ, 2017. — 1 эл. опт. диск (CD-ROM).

Приведены требования к структуре, содержанию, выбору тематики и оформлению магистерской диссертации; рекомендации по выполнению и оформлению текстовых и графических материалов диссертационных исследований, а также порядок регистрации интеллектуальной собственности в сфере информационных систем и технологий. Рассмотрены организационные этапы выполнения магистерской диссертации, порядок защиты диссертации в ГАК. Предназначено студентам высших учебных заведений, обучающимся по направлению подготовки магистра 09.04.02 «Информационные системы и технологии» по программе «Информационные системы в медиаиндустрии», а также консультантам и преподавателям, осуществляющим научное руководство магистерской диссертацией.

Компанейц, А. Н. Микропроцессорные системы управления: учеб. электрон. изд. локального распространения: конспект лекций / А. Н. Компанейц. — Омск: Изд-во ОмГТУ, 2017. — 1 эл. опт. диск (CD-ROM).

В работе рассматриваются принципы построения, назначение и состав микропроцессорных систем управления. В качестве элементов МСУ приводятся CISC и RISC микроконтроллеры. Конспект лекций предназначен для студентов очного, очно-заочного, заочного и дистанционного обучения по специальностям 220301 — «Автоматизация технологических процессов и производств», 220401 — «Мехатроника».

Библиографический список

1. Tie-Yan Liu. Learning to Rank for Information Retrieval // Journal Foundations and Trends in Information Retrieval. 2009. March. Vol. 3, issue 3. P. 225–331.
2. Christopher J. C. Burges, Tal Shaked, Erin Renshaw. Learning to Rank using Gradient Descent // Proceeding ICML '05 Proceedings of the 22nd international conference on Machine learning, 2005. P. 89–96.
3. Семенихин С. В., Денисова Л. А. Метод машинного обучения ранжированию на основе модифицированного генетического алгоритма для метрики *NDCG* // Омский научный вестник. Сер. Приборы, машины и технологии. 2016. № 3 (147). С. 100–104.
4. R. Collins and D. Jeerson. Selection in massively parallel genetic algorithms // Proceedings of the Fourth Intl. Conf. on Genetic Algorithms, ICGA-91. Morgan Kaufmann, 1991. P. 249–256.
5. Дональд Э. Кнут. Гл. 2.3. Деревья // Искусство программирования. В 3 т. Т. 2. Основные алгоритмы. 3-е изд. М.: Вильямс, 2000. 832 с. ISBN 5-8459-0081-6, 0-201-89684-2.
6. Tie-Yan Liu, Jun Xu, Tao Qin, Wenying Xiong, and Hang Li. LETOR: Benchmark Dataset for Research on Learning to Rank for Information Retrieval // SIGIR 2007 Workshop on Learning to Rank for Information Retrieval (LR4IR 2007), 2007. P. 3–10. URL: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.93.7195&rep=rep1&type=pdf> (дата обращения 30.03.2017).

СЕМЕНИХИН Святослав Витальевич, аспирант кафедры «Автоматизированные системы обработки информации и управления».
Адрес для переписки: svsemenihin@gmail.com

Статья поступила в редакцию 10.04.2017 г.

© С. В. Семенихин