УДК 004.424.4; 004.424.62

ФУНКЦИЯ РАССТОЯНИЯ МЕЖДУ СТРОКАМИ НА ОСНОВЕ КУСОЧНО-ПОСТОЯННОЙ МОДЕЛИ

В.А. ПРЫТКОВ

Белорусский государственный университет информатики и радиоэлектроники
П.Бровки, 6, Минск, 220013, Беларусь

Поступила в редакцию 23 ноября 2012

Предложена кусочно-постоянная модель строки, которая позволяет определить расстояние между двумя строками. Модель может использоваться в задачах нечеткого поиска и анализа текстов, применима к циклическим строкам.

Ключевые слова: строка символов, последовательность, мера сходства, расстояние, нечеткий поиск, анализ текстов, циклические последовательности.

Введение

Алгоритмы поиска и сравнения последовательностей активно используются при работе с неструктурированными данными, обработке больших объемов информации, поисковых запросов и т.д. Возросший объем информации предъявляет все более высокие требования к качеству и скорости поиска. Как правило, такие алгоритмы разделяются на три больших группы: алгоритмы точного поиска подстрок, нечеткий поиск и поиск наибольшей общей подпоследовательности. Последняя группа имеет важное значение при вычислении расстояний и, таким образом, тесно связана со второй.

Задачи нечеткого поиска чаще всего возникают при коррекции ошибок, фильтрации нежелательных сообщений, обнаружении плагиата, поиске с учетом форм одного и того же слова и основаны на определении расстояния между строками. Эти методы используются также и в генетике. В настоящей работе предлагается модель на основе кусочно-постоянной функции, позволяющая определить меру подобия строк символов.

Теоретический анализ

Хорошие обзоры по теме поиска последовательностей приведены в [1–4]. Рассмотрим вначале основные алгоритмы определения расстояния между строками. Наиболее известны расстояния Хэмминга и Левенштейна, а также *n*-граммы.

Расстояние Хэмминга определяется как число позиций, в которых соответствующие символы двух слов одинаковой длины различны [5, 6]. В [1] приводится альтернативное определение: если две строки A_i и A_j имеют одинаковую длину n, расстояние Хемминга $d_H(A_i, A_j)$ определяется как минимальное количество подстановок (замен), необходимых для преобразования строки A_i в строку A_j.

Расстояние Хэмминга обладает свойствами метрики, удовлетворяя следующим условиям:

$$\begin{aligned} &d(x,y) \geq 0 \\ &d(x,y) = 0 \Leftrightarrow x = y \\ &d(x,y) = d(y,x) \\ &d(x,z) \leq d(x,y) + d(y,z) \end{aligned} \qquad (1)$$

Очевидным недостатком этой меры является требование одинаковой длины строк. Любые искажения в виде пропуска либо лишнего символа не позволяют использовать эту меру. Второй недостаток – ненормированность: в результате и длинная, и короткая пара строк, с одинаковым количеством различающихся символов, имеют одно и то же значение d_H. Эту проблему легко преодолеть, нормируя по длине строки: d_H/n.

Расстояние Левенштейна $d_L(A_i, A_j)$ определяется как минимальное количество операций вставки, удаления либо замены одного символа на другой, необходимых для превращения одной строки в другую [7]. В [1] расстояние Левенштейна учитывает только операции удаления и вставки, а расстояние, учитывающее еще и замену (подстановку) называется расстоянием преобразования $d_E(A_i, A_j)$. Расстояния d_L и d_E являются метриками. Классический алгоритм вычисления расстояния Левенштейна имеет сложность $O(mn)$ для строк длиной m и n. Значительная часть исследований посвящена снижению сложности алгоритмов этого типа, для лучших из них сложность достигает $O(n+m)$. Примером таких работ могут являться [8, 9].

Недостатки этой меры: при перестановке местами слов или частей слов получаются сравнительно большие расстояния; расстояния между совершенно разными короткими словами оказываются меньшими, чем расстояния между очень похожими длинными словами. Второй недостаток также устраняется нормировкой по длине строки.

Обобщением расстояния Левенштейна $d_w(A_i, A_j)$ является использование матрицы весовых коэффициентов для замены символа i символом j. Это расстояние будет являться метрикой только если матрица весовых коэффициентов симметрична [1]. Частным случаем будет вариант, учитывающий вес для каждой из операций, вне зависимости от заменяемого символа. Для вычисления расстояния с использованием весовых коэффициентов, используют алгоритм Вагнера-Фишера.

Еще одной модификацией является расстояние Дамерау-Левенштейна, в котором дополнительно учитываются и операции перестановки (транспозиции) двух соседних символов; n-граммами (q-граммами) называют множество подстрок длины n исходной строки. Оценка расстояния производится на основе подсчета количества различающихся n-грамм данного множества. Одной из базовых работ в этой области является [10].

Помимо рассмотренных, для нечеткого поиска используют также алгоритмы на основе поиска наибольшей общей подпоследовательности, Укконена-Майерса, расширения выборки, триангуляционные, суффиксные, префиксные или trie-деревья, хеширование.

Отдельные алгоритмы поиска подстрок либо их виды, а также варианты реализации рассматриваются в [11–14]. К алгоритмам поиска точного вхождения строк относят алгоритмы Карпа-Рабина, Кнута-Морриса-Пратта, Бойера-Мура и его модификации, Демелки-Бейза-Ятса-Гоннета, поиск регулярных выражений с использованием конечных автоматов, методы, основанные на битовых операциях, алгоритмы поиска множества подстрок Ахо-Корасик, Комменца-Уолтера и др. Поиск наибольшей общей подпоследовательности выполняют алгоритмы Хешберга, Ханта-Шиманского, Машека-Патерсона и др.

Методика

Все указанные алгоритмы рассматривают цепочку символов (строку) как множество символов. Вместе с тем, если каждому символу поставить в соответствие некоторое числовое значение, то строку можно описать кусочно-постоянной функцией

$$A(x) = \begin{cases} k_1, \text{ если } 0 \le x < x_1 \\ k_2, \text{ если } x_1 \le x < x_2 \\ \dots \\ k_m, \text{ если } x_{m-1} \le x < x_m \end{cases}, \qquad (2)$$

где m – количество символов в строке, $x_1, x_2, \dots, x_{m-1}$ – границы смежных символов, и x_m – длина строки. Для возможности сравнения цепочек нормализуем функцию по длине строки:

$$A(x)=\begin{cases} k_1, \text{ если } 0 \le x < \dfrac{x_1}{x_m} \\ k_2, \text{ если } \dfrac{x_1}{x_m} \le x < \dfrac{x_2}{x_m} \\ \dots \\ k_m, \text{ если } \dfrac{x_{m-1}}{x_m} \le x < 1 \end{cases}, \quad (3)$$

где $x = [0, 1)$.

Подобная нормализация делает модель инвариантной к размеру строки. Определим функцию $E_{i,j}$ двух цепочек A_i и A_j как

$$E_{i,j}(x)=\begin{cases} 1, \text{ если } A_i(x)=A_j(x) \\ 0, \text{ если иначе} \end{cases}. \quad (4)$$

Тогда в качестве расстояния можно использовать

$$d_{PC}(A_i, A_j) = 1 - \int_{x=0}^{1} E_{i,j}(A_i(x), A_j(x))\, dx .$$

Однако, для строк разной длины интервалы, соответствующие одному символу, различны по длине. Поэтому функцию расстояния необходимо определить с учетом сдвига A_i и A_j относительно друг друга. Пусть u – сдвиг строки A_j относительно A_i , $u = [-1, 1)$. Определим функцию соответствия $M_{i,j}$ цепочек следующим образом: $M_{i,j}(u) = \int_{x=0}^{1} E_{i,j}(A_i(x), A_j(x+u))\, dx$, тогда расстояние $d_{PC}(A_i, A_j) = 1 - \max(M_{i,j}(u))$.

В таком виде модель уже можно использовать, однако точность вычислений при этом зависит от шага дискретизации. Из определения функции $E_{i,j}$ следует, что она является кусочно-постоянной. Поскольку интеграл от постоянной величины является линейной функцией, то функция соответствия $M_{i,j}(u)$ будет являться кусочно-линейной функцией, при этом можно показать, что она непрерывна. Следовательно, функция соответствия $M_{i,j}(u)$ достигает максимума в точках изменения угловых коэффициентов на концах соответствующих интервалов. Определим эти точки.

Учтем, что длина каждого интервала строки одинакова. При этом обратим внимание на тот факт, что существенными являются границы только тех пар интервалов, значения функций A_i и A_j на которых совпадают. Пусть X – множество точек разрыва функции A: $X = \{x_i \,\big|\, \dfrac{i}{m}, i = 0, 1, \dots, m\}$. Тогда множество $U_{i,j}$ точек изменения угловых коэффициентов функции $M_{i,j}(u)$ определяется следующим образом:

$$\begin{aligned} &U_{i,j} = \{u_{i,j} \,\big|\, x_i - x_j ,\ x_i \in X(A_i), x_j \in X(A_j), A_i(x_i) = A_j(x_j) \\ &\text{либо } A_i(x_{i-1}) = A_j(x_j) \\ &\text{либо } A_i(x_i) = A_j(x_{j-1})\}. \end{aligned} \quad (5)$$

Альтернативы учитывают прерывность функции в точке разрыва. Можно отметить, что

$$U_{i,j} \subseteq \{\pm \frac{k}{\text{НОК}(m,n)},\ k = 0,\ 1, \dots, \text{НОК}(m,n)\}, \quad (6)$$

Здесь НОК – наименьшее общее кратное. В этом случае расстояние будет вычисляться следующим образом: $d_{PC}(A_i, A_j) = 1 - \max(M_{i,j}(U_{i,j}))$. Использование (6) в алгоритме расчета является более простым, но и несколько избыточным способом вычисления расстояния.

Предложенное расстояние будет являться метрикой. Докажем это.

Поскольку $E(x) \le 1$ и $x = [0, 1)$, то $M(u) \le 1$, и, следовательно $d_{PC}(A_i, A_j) \ge 0$. $d_{PC}(A_i, A_j) = 0$ тогда и только тогда, когда $\max(M_{i,j}(U_{i,j})) = 1$. Поскольку при любом u, отличном от 0, строки не полностью сравниваются друг с другом, то $M(u \ne 0) < 1$. Следовательно, равенство единице может иметь место только при $u = 0$. Тогда $\int_{x=0}^{1} E_{i,j}(A_i(x), A_j(x))\,dx = 1$. Из определения функции E (4) следует, что данное равенство выполняется тогда и только тогда, когда $A_i = A_j$.

Равенство $d_{PC}(A_i, A_j) = d_{PC}(A_j, A_i)$ элементарным образом сводится к $E(A_i, A_j) = E(A_j, A_i)$, верность которого следует из определения функции E (4). Докажем свойство треугольника $d(x,z) \le d(x,y) + d(y,z)$.

Независимо от сдвига u, при котором достигается максимум M, все множество точек функции A_x, A_y, A_z будет состоять из точек множеств X, Y, Z не совпадающих ни с одной из двух оставшихся строк, множеств XY, XZ, YZ соответственно из точек, совпавших только для строк x и y, x и z, y и z соответственно и XYZ, состоящее из точек, совпавших у всех строк.

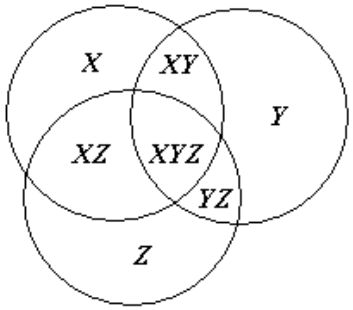


Рис. 1. Множества точек

При этом выполняется: $|X| + |XZ| + |XY| + |XYZ| = 1$, $|Y| + |YZ| + |XY| + |XYZ| = 1$, $|Z| + |XZ| + |YZ| + |XYZ| = 1$. Кроме того , $d_{PC}(x, y) = 1 - |XY| - |XYZ|$, $d_{PC}(x, z) = 1 - |XZ| - |XYZ|$, $d_{PC}(y, z) = 1 - |YZ| - |XYZ|$. Выполнив соответствующую подстановку, получаем: $1 - |XZ| - |XYZ| \le 1 - |XY| - |XYZ| + 1 - |YZ| - |XYZ|$, оттуда $- |XZ| \le 1 - |XY| - |YZ| - |XYZ|$, и, наконец, $- |XZ| \le 1 - |Y|$.

Так как длина любого участка строки не отрицательна и не превышает 1, то правая часть неравенства всегда неотрицательна, левая – не превышает нуля. Таким образом, доказано, что предложенная функция расстояния является метрикой.

Оценим вычислительную сложность полученной модели. Алгоритм включает в себя вычисление множества U и, в соответствии с количеством элементов данного множества, расчета функции M и выбора максимального значения: $f(m) = 2(m+n) + |U|k(m+n) + |U|$.

Здесь m и n – количество символов в сравниваемых цепочках, k – некоторый коэффициент, учитывающий количество операций по вычислению частичной суммы при расчете значения функции M. Мощность множества U не превысит mn для произвольных цепочек. Таким образом, вычислительная сложность предложенного алгоритма $O(m^2n)$.

Экспериментальная часть

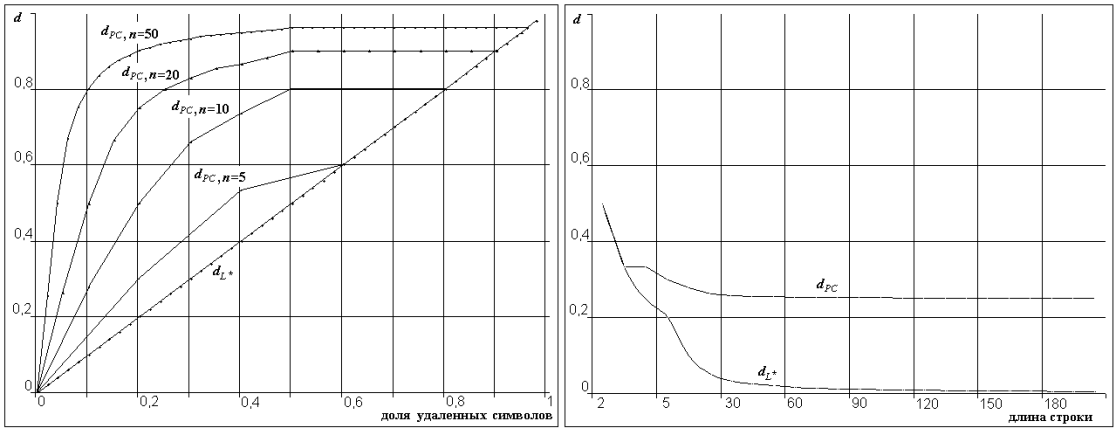
Сравним расстояния d_H, d_L и d_{PC}, а также нормированное расстояние Хэмминга d_{H^*} и нормированное расстояние Левенштейна d_{L^*}. Расстояние Левенштейна будем считать с использованием операций удаления, добавления и замены. Будем выполнять проверки для случаев несовпадающих символов, пропущенных (добавленных) символов, переставленных символов.

Обозначим сравниваемые строки A_i и A_j. Для всех экспериментов будем считать, что все символы каждой из сравниваемых строк отличны друг от друга, и все символы строки A_i совпадают с соответствующими символами строки A_j, исключая проверяемый случай. Например, при проверке отсутствия символа в середине строки, строка $A_i = ABCDEFGHIJKL$, строка $A_j = ABCDEFHIJKL$.

Для строк одинаковой длины $n = |A_i| = |A_j|$, и количества отличных символов k, в одинаковых позициях сравниваемых строк, независимо от их положения в строке расстояние $d_H = d_L = k$, $d_{H*} = d_{L*} = d_{PC} = k / n$.

Пусть $|A_i| = n$, $|A_j| = m$, и $n > m$. Такая ситуация соответствует пропуску (добавлению) символов. Пусть $k = n - m$, пропущенные символы находятся друг за другом и в самом начале строки (либо в самом конце, что эквивалентно). Тогда $d_L = k$, $d_{L*} = k/n$. d_{PC} имеет сложную зависимость от величины k: $d_{PC} = 1–1/n$ при $k = n–1$, $d_{PC} = 1–2/n$ при $k = [n/2, n–2]$, и нелинейный характер на участке $k = [0, n/2)$. Графики зависимостей приведены на рис. 2, *а*.

Пусть $|A_i| = n$, $|A_j| = n–1$. Такая ситуация соответствует пропуску (добавлению) одного символа. Пусть $i = 1, 2, \ldots, n/2$ указывает на позицию пропущенного символа. Тогда $d_L = 1$, $d_{L*} = 1/n$. $d_{PC} = n/4(n+1)$, если n – четно, и $d_{PC} = (n+1)/4n$, если n – нечетно. При этом ни d_L, ни d_{PC} не зависят от позиции пропущенного символа.

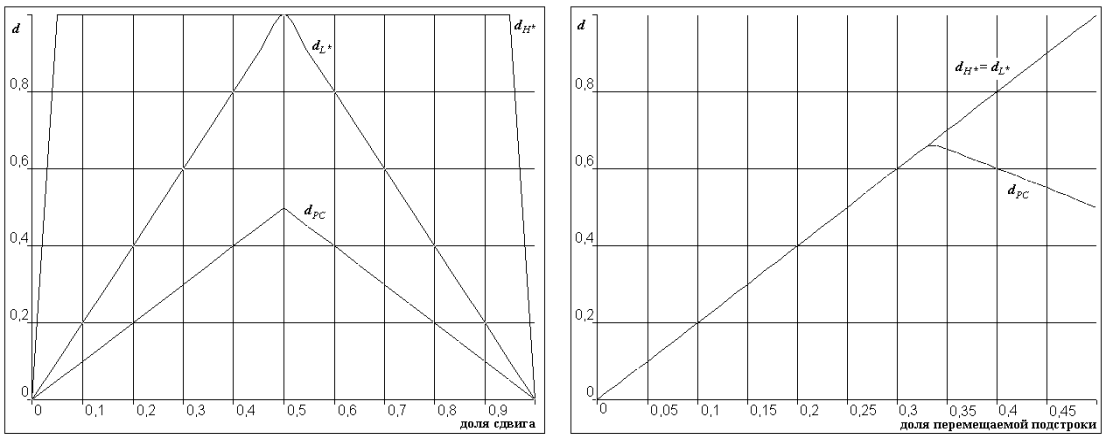


а *б*

Рис. 2. Зависимости d_{L*} и d_{PC} для случаев
k пропущенных символов *(а)*; пропуска (добавления) символа в позиции k *(б)*

Рассмотрим случай циклического сдвига. В этом случае $n = |A_i| = |A_j|$. Пусть $i = 1\ 2, \ldots, n–1$ указывает на количество циклически сдвинутых символов. Тогда $d_H = n$, $d_L = 2 \min(i, n\text{-}i)$. $d_{H*} = 1$, $d_{L*} = 2 \min(i/n, 1–i/n)$. $d_{PC} = \min(i/n, 1–i/n)$. В этом случае d_{PC} симметрично относительно середины строки и ровно вдвое меньше d_{L*}. Графики зависимостей приведены на рис. 3, *а*.

Строку A_i можно представить как конкатенацию подстрок $A_i = \alpha\beta\gamma\delta\varepsilon$. Пусть строка A_j получена из строки A_i путем перестановки подцепочек $A_j = \alpha\delta\gamma\beta\varepsilon$. Для упрощения будем считать, что длина переставляемых подцепочек одинакова $|\delta| = |\beta| = k$, и они расположены симметрично относительно центра строки, т.е. $|\alpha| = |\varepsilon| = i$, причем $|\gamma| = 0$, если n – четно, и $|\gamma| = 1$, если n нечетное. Тогда $d_H = d_L = 2k$. $d_{H*} = d_{L*} = 2k/n$. d_{PC} имеет следующую зависимость: $d_{PC} = 2k/n$ при $k = [0, n/3]$ и $d_{PC} = 1–k/n$ при $k = (n/3, n/2]$.



а *б*

Рис. 3. Зависимости d_{H*}, d_{L*} и d_{PC} для случаев
циклического сдвига k символов *(а)*; перестановки подстрок длиной k символов *(б)*

Результаты и их обсуждение

В случае пропуска (добавления) k символов (рис. 2, *а*) при $k>n/2$ расстояние D_{PC} является постоянной величиной. Объяснить это достаточно легко. При таком k одна строка в два и более раза короче, соответственно каждый фрагмент второй строки при вычислении расстояния перекрывает не менее чем два смежных фрагмента первой. Поскольку все символы отличны друг от друга по условию проведения эксперимента, то в результате независимо от величины сдвига невозможно совпадение более чем на двух соседних участках. Соответственно для больших строк ($n\to\infty$) на этом участке $d_{PC}\to 1$. Этим свойством можно воспользоваться для сокращения времени работы алгоритма на больших строках.

В отличие от расстояния Левенштейна, предложенная модель нелинейна, и уже при отличии 10–20% в длинах строк полученное расстояние близко к максимальному. Это свойство может быть полезным для задач, в которых близкими можно считать только строки с минимальными отличиями.

Позиция одного пропущенного символа не влияет на рассчитанное расстояние (рис. 2, *б)*. Для больших строк ($n\to\infty$) $d_{PC}\to 0{,}25$. Это значение можно условно считать «штрафом» за пропуск (добавление) символа.

В случае циклического сдвига предложенное расстояние меньше расстояния Левенштейна в два раза (рис. 3, *а*), что позволяет рассматривать модель как более подходящую для работы с циклическими последовательностями.

Моделирование перестановок подстрок (рис. 3, *б*) позволяет сделать вывод о том, что предложенная модель в меньшей степени, по сравнению с расстоянием Левенштейна, подвержена тому недостатку, что при перестановке местами слов или частей слов получаются сравнительно большие расстояния.

На слабоподобных участках количество элементов множества U существенно снижается по сравнению с максимальной оценкой, а на близких фрагментах – приближается к ней, поскольку данный параметр зависит от количества одинаковых символов в строке. Таким образом, если вероятность совпадения строк невысока, выгоднее при расчете использовать точную формулу (5) поиска элементов множества U.

Интересным свойством модели является то, что она не требует соблюдения условия равенства всех участков для каждого символа. Фактически, длина участка, соответствующего одному символу, может отличаться от длины участка, соответствующего другому символу. Это свойство делает возможным использование данной модели в задачах распознавания, основанных на структурном (синтаксическом) подходе.

Модель легко адаптировать для сравнения циклических последовательностей, для этого функция (3) должна быть переписана как периодическая с периодом 1. Необходимость в такой модификации может возникнуть в некоторых задачах генетики [4], в задачах распознавания образов и обработки изображений, в частности, в текстурном анализе [15]. Учитывая, что модель менее чувствительна к циклическому сдвигу, а также наличие возможности обобщения на произвольные длины фрагментов, использование периодической модификации модели представляется весьма перспективным.

Заключение

В работе предложена модель строки символов, которая позволяет определить расстояние между двумя строками. Модель основана на представлении строки в виде кусочно-постоянной функции, позволяет сравнивать строки разной длины, легко адаптируется для работы с циклическими строками.

FUNCTION OF DISTANCE BETWEEN THE STRINGS BASED ON PIECEWISE CONSTANT MODEL

V.A. PRYTKOV

Abstract

A piecewise model of a string which allows to define a distance between two strings is proposed. The model can be applied to problems of fuzzy matching and text analysis and the model can be used with cyclic strings.

Список литературы

1. *Смит У.* Методы и алгоритмы вычислений на строках. М., 2006.
2. *Gonzalo Navarro* // ACM Computing Surveys. 2001. Vol. 33, № 1. P. 31–88.
3. *Graham A. Stephen*. String Seaching Algorithms, 1994.
4. *Гасфилд Д.* Строки, деревья и последовательности в алгоритмах. СПб, 2003.
5. *Hamming R.W.* // The Bell System Technical Journal. 1950. Vol. XXIX, № 2. P.147–160.
6. Federal Standard 1037C, 1996.
7. *Левенштейн В.И.* // Докл. Академий Наук СССР. 1965. Т. 163, №4. С. 845–848.
8. *Masek W.J., Paterson M.S.* // Journal of Computer and System Sciences. 1980. Vol. 20, № 1.
9. *Ehrenfeucht A., Haussler D.* // Discrete Applied Mathematics. 1988. Vol. 20, № 3. P. 191–203.
10. *Ukkonen E.* Approximate string-matching with q-grams and maximal matches // Theoretical Computer Science 1992. Vol. 92, № 1. P.191–211.
11. *Hall P.A.V., Dowling G.R.* // Computing Surveys. 1980. Vol. 12, № 4. P. 381–402.
12. *Blumer A., Blumer J., Haussler D. et al.* // Theoretical Computer Science. 1985. Vol. 40. P. 31–55.
13. *Hirschberg D.S.* // Journal of the Assoclauon for Computing Machinery. 1977. Vol 24, № 4. P. 664–675.
14. *Кормен Т., Лейзерсон Ч., Ривест Р. и др.* Алгоритмы: построение и анализ. М., 2005.
15. *Прытков В.А., Барташевич Ю.А., Лукашевич М.М.* // Матер. V межд. конф.-форума «Информационные системы и технологии». Минск, 2009. С. 172–175.