

УДК 621.01

## АВТОМАТИЗАЦИЯ ПОСТРОЕНИЯ СЛОЖНЫХ ЭМПИРИЧЕСКИХ ЗАВИСИМОСТЕЙ ПРИ ИССЛЕДОВАНИЯХ ТЕХНОЛОГИЧЕСКИХ ОБЪЕКТОВ

**В. М. ПАШКЕВИЧ**

*Государственное учреждение высшего профессионального образования «Белорусско-Российский университет»,  
г. Могилев, Республика Беларусь*

**Ключевые слова:** теория резания, эмпирические зависимости, многофакторные модели, кусочно-линейная аппроксимация, многомерная интерполяция.

### Введение

При исследованиях технологических процессов и систем часто возникает необходимость построения эмпирических зависимостей (моделей), приближающих экспериментальные данные. Хорошо известным примером таких зависимостей являются эмпирические формулы теории резания. Обычно соответствующие модели могут включать до 3–5 и более параметров. Их вид, как правило, нелинейный; часто в этой связи для одного и того же процесса при различных сочетаниях параметров используют несколько эмпирических формул. Их построение весьма трудоемко и часто требует нетривиального подхода к выбору не только вида зависимости, но и областей для их построения [1].

Альтернативу такому подходу может составить описанный в статье алгоритм автоматизированного построения эмпирических зависимостей для экспериментальных данных (наблюдений). Для них характерны следующие особенности:

- наблюдения, как правило, нерегулярные (промежутки по аргументам данных переменные);
- наблюдения поступают последовательно и могут иметь значительный объем;
- данные могут быть избыточными, т. е. могут не содержать дополнительной информации о виде зависимости в отношении к уже полученной;
- с поступлением новых наблюдений аппроксимирующая модель требует корректировки.

Указанные обстоятельства снижают эффективность использования классических методов построения эмпирических зависимостей, когда вид модели, приближающей данные, известен или установлен заранее. Альтернативу им могли бы составить методы, использующие алгоритм обучения (например, на основе нейронных сетей). Однако нерегулярность в поступлении данных экспериментов также накладывает определенные ограничения на их применимость, так как новые наблюдения могут потребовать переобучения сети [2].

В этой связи целью настоящей работы является разработка алгоритма, базирующегося на запоминании релевантных обучающих данных и использовании процедуры многомерной интерполяции. При этом модель как таковая не строится, а существует в неявном виде – в форме численного набора обучающих образцов или набора локальных моделей линейного вида.

Представленная работа является развитием направления, связанного с управлением технологическими процессами механической обработки на основе использования функциональных семантических сетей [3].

### Основная часть

Представляемый алгоритм исходит из очевидного представления о том, что для наблюдений в форме многомерных точек  $(x_1, x_2, x_3, \dots, x_n, y)$  может быть предложена сложная аппроксимирующая функция

$$y = f(x_1, x_2, x_3, \dots, x_n), \quad (1)$$

кусочно состоящая из набора функций вида

$$y = a_0 + a_1x_1 + a_2x_2 + \dots + a_nx_n. \quad (2)$$

При этом в силу нелинейности функции (1) зависимости (2) строятся для каждой области аргументов  $x_1, x_2, x_3, \dots, x_n$ , в которой с заранее установленной погрешностью выражение (1) может быть адекватно приближено линейной зависимостью (2). Таким образом, алгоритм представляет собой кусочно-линейную аппроксимацию нелинейной зависимости в многомерном пространстве.

Допустимость такого приближения непосредственно вытекает из теоремы Колмогорова об аппроксимации, в соответствии с которой функция нескольких аргументов может быть приближена суммой, произведением или суперпозицией функций одного аргумента. Так, аналогичный подход используется для объяснения работы нейронных сетей в форме многослойных персептронов [4].

Набор приближающих функций строится в этом случае последовательно, для серий данных (обучающих образцов), образующих в многомерном пространстве выпуклые области (базовые элементы), а с поступлением новых наблюдений корректируется в соответствии с алгоритмом, включающим следующую последовательность действий:

1) провести серию наблюдений, необходимую для построения первого базового элемента в пространстве данных, и включить соответствующие ему точки в базу обучающих образцов;

2) провести новое наблюдение;

3) проверить условие попадания нового наблюдения в область базовых элементов;

4) если наблюдение выходит за пределы области базовых элементов, добавить его в базу обучающих образцов и сформировать новую совокупность базовых элементов, расширяя, таким образом, область покрытия функции (1);

5) перейти к п. 2 алгоритма;

6) если наблюдение попадает в область базовых элементов, провести оценку адекватности базы обучающих образцов новому наблюдению;

7) если база обучающих образцов адекватна новому наблюдению, исключить его из рассмотрения;

8) перейти к п. 2 алгоритма;

9) если база обучающих образцов неадекватна новому наблюдению, включить его в базу и провести ее разбиение на новые базовые элементы с учетом этого наблюдения;

10) перейти к п. 2 алгоритма и т. д. до обработки всей совокупности поступающих данных.

Рассмотрим подробнее реализацию такого алгоритма.

**Операция 1.** На первоначальном этапе работы алгоритма строится база обучающих образцов, включающая минимальное количество наблюдений, требующихся для построения первого выпуклого базового элемента в пространстве задачи.

Очевидно, что так как наблюдения  $(x_1, x_2, x_3, \dots, x_n, y)$  представляют собой точки в  $(n + 1)$ -мерном пространстве, то, соответственно, для построения линейной модели (2) необходимо не менее  $(n + 1)$  таких точек, образующих выпуклую фигуру (базовый элемент) соответствующего пространства.

Так, не менее двух точек потребуется для построения зависимости  $y = a_0 + a_1x_1$ , характеризующей базовый элемент в форме прямой, расположенной в двухмерном пространстве. Не менее трех точек необходимо для определения уравнения плоскости  $y = a_0 + a_1x_1 + a_2x_2$ , проходящей в трехмерном пространстве через эти точки, образующие выпуклый базовый элемент в форме треугольника.

Обобщением понятия такого выпуклого базового элемента на пространства высшей размерности являются: четырехугольник в четырехмерном пространстве; пятиугольник в пятимерном и т. д. По аналогии с понятием гиперплоскости эти фигуры могут быть определены как многомерные гипертреугольники.

Таким образом, база обучающих образцов содержит координаты многомерных точек, совокупности  $(n + 1)$  из которых образуют соответствующие базовые элементы пространства (гипертреугольники), покрывающие область построения модели. База обучающих образцов содержит также список разбиения совокупности точек на соответствующие базовые элементы. Заметим, что в этой связи описываемая процедура может расцениваться как многомерная триангуляция.

Обратим также внимание на принципиальную необходимость использования базовых элементов в форме выпуклых фигур. В геометрическом смысле совокупность таких элементов, составленных из точек базы обучающих образцов, образует набор гиперплоскостей, кусочно (в форме нерегулярной «черепицы») покрывающих нелинейную поверхность отклика (1). Добавление новых наблюдений обеспечит в этом случае построение «покрытия» поверхности отклика  $y = f(x_1, x_2, x_3, \dots, x_n)$  без каких-либо пропусков.

**Операция 2.** Принципиально новые наблюдения могут быть получены в произвольной точке пространства задачи. Однако существенно упрощает процедуру построения поверхности отклика от центра плана, с постепенным расширением этой области к ее периферии.

**Операция 3.** Попадание нового наблюдения в область базовых элементов проверяется последовательно, для каждого элемента списка базы. При этом задача сводится к определению принадлежности точки  $(x_1^0, x_2^0, \dots, x_n^0)$  выпуклому многомерному многоугольнику, заданному координатами образующих его точек.

Отметим, что существует несколько решений такой задачи (методы трассировки луча; суммирования углов; подсчета числа оборотов границы и др.) [5].

Нами использовался упрощенный способ ее решения, который базируется на очевидном свойстве выпуклых многоугольников. Так, из рис. 1 видно, что если некоторая точка  $P_{123}^0$  не принадлежит к области, ограниченной выпуклым многоугольником  $P_1P_2P_3$ , то она всегда может быть разделена с ним в пространстве линией (гиперплоскостью)  $d-d$ .

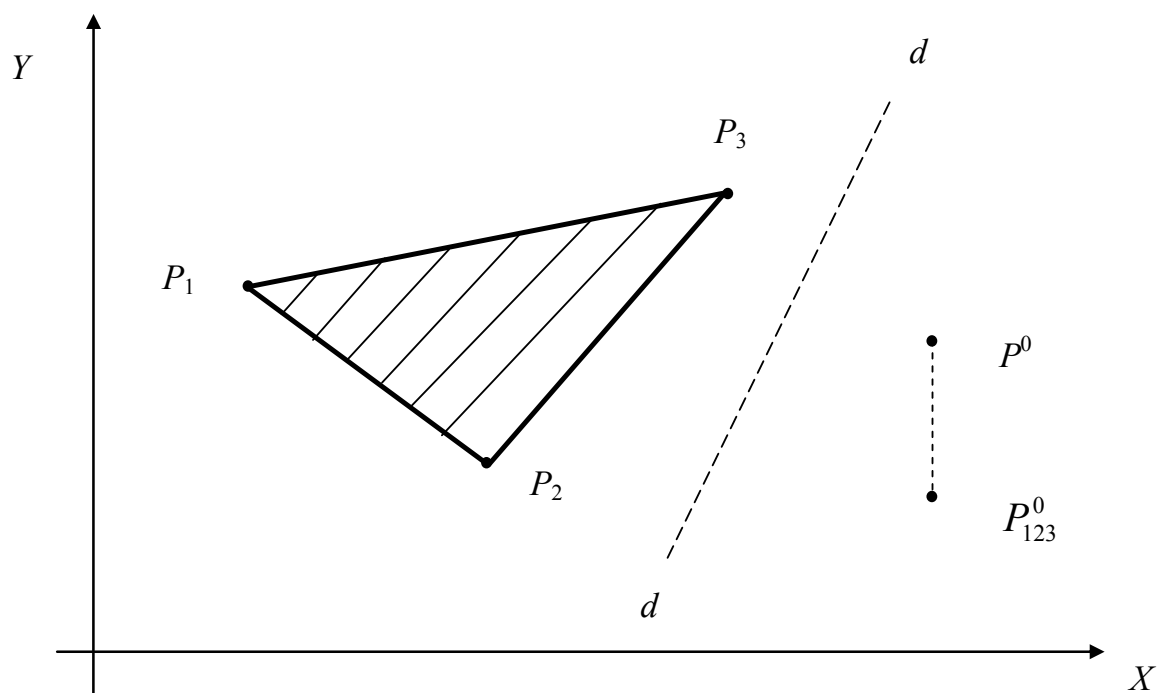


Рис. 1. К оценке принадлежности точки базовому элементу

Заметим, что, применительно к данному примеру, точка  $P_{123}^0$  является проекцией некоторого экспериментального наблюдения  $P^0$  на плоскость, проходящую через точки  $P_1, P_2, P_3$ . Координаты проекции  $P_{123}^0$  могут быть легко найдены по уравнению, описывающему плоскость  $P_1P_2P_3$  (базовый элемент). Построение соответствующих зависимостей хорошо разработано и может базироваться, например, на применении метода наименьших квадратов.

Так, для определения коэффициентов  $a_0, a_1, a_2$ , определяющих положение трехмерной плоскости, может использоваться система уравнений:

$$\begin{cases} ka_0 + a_1 \sum x_1 + a_2 \sum x_2 = \sum y, \\ a_0 \sum x_1 + a_1 \sum x_1^2 + a_2 \sum x_1 x_2 = \sum x_1 y, \\ a_0 \sum x_2 + a_1 \sum x_1 x_2 + a_2 \sum x_2^2 = \sum x_2 y, \end{cases} \quad (3)$$

где  $k$  – количество экспериментальных наблюдений, через которые проведена плоскость (при использовании описанных выпуклых базовых элементов применительно к системе (3) оно равно трем).

Система (3) может быть легко обобщена для построения гиперплоскости в пространстве произвольной размерности.

Разделяющая (дискриминантная) гиперплоскость  $d-d$  может быть найдена на основе алгоритма обучения распознаванию образов в форме зависимости [6]:

$$d(x_0, x_1, \dots, x_n) = w_0 x_0 + w_1 x_1 + \dots + w_n x_n, \quad (4)$$

где  $x_0 = 1$ ;  $w_0, w_1, \dots, w_n$  – весовые коэффициенты.

В соответствии с этим алгоритмом весовым коэффициентам назначаются начальные (как правило, нулевые) значения, а затем последовательно рассчитываются значения дискриминантной функции (4) для точек  $P^0$ ,  $P_1$ ,  $P_2$ ,  $P_3$  и затем классификационного признака:

$$D = \text{sign}(d(x_0, x_1, \dots, x_n)), \quad (5)$$

где  $\text{sign}$  – функция знака, равная +1 для положительных и –1 для отрицательных значений аргумента.

При условии линейной разделимости точек  $P_1$ ,  $P_2$ ,  $P_3$  и точки  $P^0$ , что справедливо для выпуклых базовых элементов, значения признака  $D$  будут иметь разные знаки для класса точек  $P_1$ ,  $P_2$ ,  $P_3$  и класса, содержащего точку  $P^0$ .

Значения весовых коэффициентов в процессе обучения будут корректироваться по формуле

$$w_k^{n+1} = w_k^n + \eta(D_{\text{тр}} - D_{\text{действ}})x_k, \quad (6)$$

где  $w_k^n$  и  $w_k^{n+1}$  – значения коэффициента  $w_k$  на  $n$ -м и  $(n+1)$ -м этапе обучения;  $\eta$  – мера обучения (число от 0 до 1, характеризующее скорость и точность обучения);  $D_{\text{тр}}$  и  $D_{\text{действ}}$  – требуемое и действительное значения классификационного признака (например, требуемое значение для класса точек  $P_1$ ,  $P_2$ ,  $P_3$  – значение +1; для класса, содержащего точку  $P^0$ , требуемое значение: –1);  $x_k$  – значение координаты, соответствующей коэффициенту  $w_k$ .

При малом количестве точек, что характерно для описываемой задачи, итерационные вычисления (6) сходятся очень быстро. В результате после заранее заданного количества повторений классификация оказывается успешной для всех точек обоих классов, если проекция  $P_{123}^0$  не находится внутри базового элемента. И наоборот, классификация будет иметь ошибки, если проекция  $P_{123}^0$  располагается внутри базового элемента.

**Операция 4.** Если наблюдение попадает за пределы области базовых элементов, его добавляют в базу обучающих образцов и формируют новую совокупность базовых элементов (рис. 2).

Так, при добавлении точки  $P^0$  в базу обучающих образцов, содержащую ранее точки  $P_1$ ,  $P_2$ ,  $P_3$  и, соответственно, список базовых элементов, содержащий единственный элемент  $P_1P_2P_3$ , поступают следующим образом.

Для добавляемой точки  $P^0$  в  $n$ -мерном пространстве задачи находят  $n-1$  ближайших к ней точек из базы обучающих образцов и соединяют ее с ними, образуя новый базовый элемент (рис. 2). Этот элемент заносят в соответствующий список, который в соответствии с рисунком станет содержать два базовых элемента –  $P_1P_2P_3$  и  $P^0P_2P_3$ .

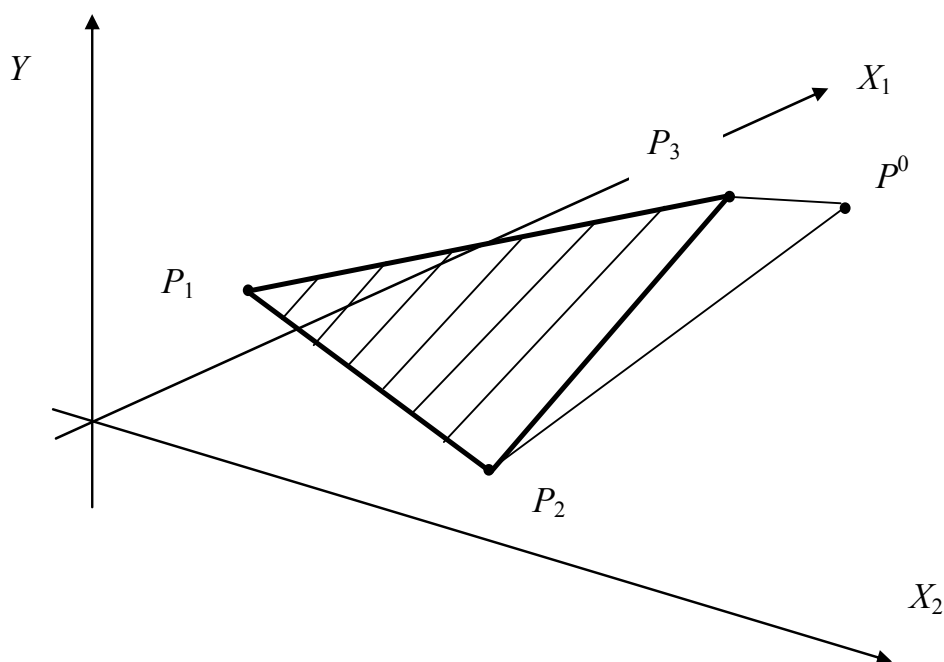


Рис. 2. Формирование нового базового элемента

**Операция 6.** Если наблюдение попадает в область, ограниченную одним из базовых элементов, следует провести оценку адекватности базы обучающих образцов новому наблюдению.

Первоначальный объем наблюдений образует единственный выпуклый базовый элемент в пространстве задачи. Полученное новое наблюдение  $P^0(x_1^0, x_2^0, \dots, x_n^0, y^0)$  при попадании его в область элемента может находиться на большем или меньшем расстоянии  $d^0$  от соответствующей этому базовому элементу гиперплоскости или лежать на ней. Величина такого расстояния может являться мерой адекватности базового элемента текущему наблюдению.

Представленный ниже порядок действий иллюстрируется рис. 3.

Для нахождения расстояния  $d^0$  используется аппроксимирующая зависимость (2), найденная путем решения соответствующей системы уравнений (3).

Если для расстояния  $d^0$  выполняется условие

$$d^0 = |y^0 - y(x_1^0, x_2^0, \dots, x_n^0)| < \varepsilon, \tag{7}$$

где  $y^0$  – фактическое положение наблюдения в пространстве;  $y(x_1^0, x_2^0, \dots, x_n^0)$  – проекция наблюдения на гиперплоскость базового элемента;  $\varepsilon$  – заранее принятая мера неадекватности модели экспериментальным наблюдениям, то точка находится вблизи от гиперплоскости и соответствующий базовый элемент адекватно описывает новое наблюдение. В этой связи добавлять указанное наблюдение в базу обучающих образцов не имеет смысла и соответствующая точка исключается из рассмотрения (операция 7).

Заметим также, что в качестве меры близости (7) могут использоваться и другие критерии (например, относительная, а не абсолютная погрешность модели).

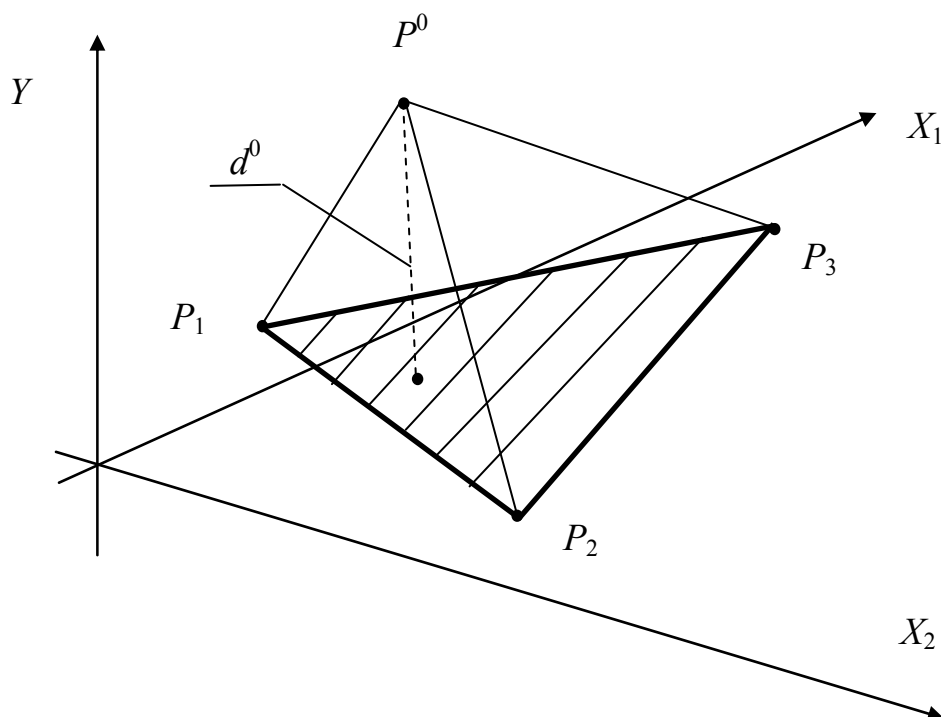


Рис. 3. К оценке адекватности базового элемента

**Операция 9.** Если проверка выявила неадекватность базы обучающих образцов новому наблюдению, следует включить его в базу и провести ее повторное разбиение на базовые элементы с учетом нового наблюдения.

Пусть, например,  $d^0 > \varepsilon$ , следовательно, плоский базовый элемент  $P_1P_2P_3$  неадекватно отражает нелинейную тенденцию изменения зависимости (1) в области, ограниченной точками  $P_1, P_2, P_3$  (рис. 3). Следовательно, на поверхности отклика должны располагаться не только проверенные ранее образцы  $P_1, P_2, P_3$ , но и образец  $P^0$ .

С этой целью для точки  $P^0$  в  $n$ -мерном пространстве задачи находят все возможные комбинации из  $n - 1$  точек неадекватного базового элемента и соединяют ее с ними, образуя, таким образом,  $n$  новых базовых элементов. Эти элементы заносят в соответствующий список, из которого затем удаляется прежний, неадекватный, базовый элемент.

В соответствии с рисунком список, ранее включавший только элемент  $P_1P_2P_3$ , после корректировки станет содержать три базовых элемента –  $P^0P_1P_2, P^0P_1P_3, P^0P_2P_3$ , а кусочно-линейная поверхность отклика станет иметь локальный экстремум в точке  $P^0$ .

Описанные операции алгоритма повторяются применительно ко всей совокупности наблюдений или до получения удовлетворительного результата работы обученной программы. При этом в ее памяти сохраняются списки базовых элементов, а также уравнения, описывающие элементы.

Использование обученной программы для решения практических задач сводится к многомерной интерполяции данных, когда для заданного набора аргументов  $(x_1^0, x_2^0, \dots, x_n^0)$  требуется определить значение выходной переменной  $y^0$ .

В этом случае программой просматривается список базовых элементов и проверяется условие попадания текущего набора аргументов внутрь одного из элементов.

Значение выходной переменной  $y^0$  находится из уравнения, описывающего соответствующий элемент.

### **Заключение**

Представленный алгоритм отличается от традиционно используемых в той части, что общая модель данных (1) не строится. В памяти программы хранятся только координаты наблюдений, а также список их комбинаций, образующих базовые элементы. Такая совокупность в неявном виде представляет собой сетчатую структуру, наложенную на поверхность отклика (1).

В этой связи описанный алгоритм не накладывает каких-либо специфических ограничений на вид модели (1), размер области ее построения, количество аргументов зависимости и т. д., т. е. отличается определенной универсальностью.

Несмотря на внешнюю трудоемкость вычислений, алгоритм работает с линейными зависимостями, поэтому скорость его работы вполне удовлетворительная. Это связано также с тем, что при корректировке базы данных корректируется всего один ее базовый элемент, остальные при этом не изменяются. Таким образом, новые наблюдения не требуют глобального пересчета модели в целом, а используют только локальные процедуры уточнения ее вида.

В то же время следует отметить слабую экстраполяционную способность получаемой модели для данных, выходящих за пределы области ее построения. Это связано с тем, что экстраполяция в этом случае ведется при использовании только крайних базовых элементов, а также только лишь процедуры линейной экстраполяции.

Алгоритм был протестирован на наборе данных, описываемых показательными формулами теории резания с переменными значениями эмпирических коэффициентов и продемонстрировал устойчивую работу при обобщении их результатов, а также в составе функциональных семантических сетей.

### **Литература**

1. Бородина, Н. В. Практикум по теории резания металлов : учеб. пособие : в 2 ч. / Н. В. Бородина. – Екатеринбург : Рос. гос. проф.-пед. ун-т, 2012.
2. Осовский, С. Нейронные сети для обработки информации : пер. с пол. / С. Осовский. – М. : Финансы и статистика, 2004. – 344 с.
3. Пашкевич, В. М. Функциональные семантические сети для обеспечения точности механической обработки : монография / В. М. Пашкевич, М. Н. Миронова. – Могилев : Беларус.-Рос. ун-т, 2015. – 210 с. : ил.
4. Рассел, С. Искусственный интеллект. Современный подход : пер. с англ. / С. Рассел, П. Норвиг. – М. : Вильямс, 2006. – 1408 с.
5. Алгоритмы. Построение и анализ : пер. с англ. / Т. Кормен [и др.]. – 3-е изд. – М. : Вильямс, 2013. – 1328 с.
6. Таунсенд, К. Проектирование и программная реализация экспертных систем на персональных ЭВМ : пер. с англ. / К. Таунсенд, Д. Фохт. – М. : Финансы и статистика, 1999. – 320 с.

*Получено 10.08.2018 г.*