

# Об агрегировании информации в нечетких иерархических системах.

*Рыжов А.П.*

*кафедра математической теории интеллектуальных систем  
механико-математический факультет  
МГУ им. М.В. Ломоносова*

## **Введение.**

Многие реальные процессы имеют иерархическую структуру. Изучение таких структур и применение полученных результатов для анализа реальных объектов и процессов описано, например, в [7, 9]. Методы теории нечетких множеств являются удобным средством моделирования, анализа и синтеза человеко-компьютерных систем [5]. Поэтому изучение нечетких иерархических систем представляется актуальной задачей, имеющей не только теоретической, но и практический интерес. Одним из важных приложений анализа таких систем могут служить системы информационного мониторинга [4, 10].

## **1. Описание проблемы.**

Пусть моделью объекта или процесса является дерево  $D$  с вершинами  $d_j$  ( $j = 0, \dots, N_D$ ), каждой из которых поставлено в соответствие некоторое множество лингвистических значений  $X_j$ , характеризующих состояние вершины. Каждой не концевой вершине также приписан некоторый оператор агрегирования информации, позволяющий на основе оценок состояния подчиненных вершин вычислять ее состояние (то есть выбирать один из элементов соответствующего множества значений). Часто выбор такого оператора определяется свойствами модели. Например, если мы имеем оценки наличия на складе колес, корпусов, двигателей и т.п. и хотим оценить возможность сборки автомобиля, в качестве естественного оператора агрегирования мы должны использовать минимум (при отсутствии хотя бы одного компонента мы не сможем собрать автомобиль). Однако часто этот выбор не является настолько очевидным (например, для задач из области политологии, социологии или медицины). В этом случае необходима разработка методов выбора адекватных операторов агрегирования на основе доступной информации от экспертов и анализа функционирования модели. Подходы к решению этой проблемы описаны в настоящей статье.

## 2. Понятие адекватного оператора агрегирования информации.

Рассмотрим некоторую не концевую вершину  $d_{j_0}$  с подчиненными ей (в смысле рассматриваемого дерева  $D$ ) вершинами  $d_{j_1}, d_{j_2}, \dots, d_{j_{N_0}}$ . Тогда оператор агрегирования информации (ОАИ)  $O_{j_0}$  есть функция, определенная на множестве всех возможных значений подчиненных вершин и принимающая значения в множестве  $X_{j_0}$ :

$$O_{j_0}: X_{j_1} \times X_{j_2} \times \dots \times X_{j_{N_0}} \rightarrow X_{j_0}. \quad (1)$$

Множества  $X_j$  ( $j = 0, \dots, N_D$ ) представляет собой набор лингвистических значений  $a_j^1, a_j^2, \dots, a_j^{N_j}$ .

Обозначим множество операторов агрегирования информации для вершины  $d_{j_0}$  через  $M[O_{j_0}]$ . Ясно, что для конкретного элемента модели проблемы  $d_{j_0}$  число возможных ОАИ является большим: из (1) непосредственно следует, что

$$|M[O_{j_0}]| = |X_{j_0}| \times |X_{j_1}| \times |X_{j_2}| \times \dots \times |X_{j_{N_0}}|.$$

Нашей задачей является выбор конкретного оператора  $o_j \in M[O_j]$  для всех не концевых вершин  $d_j$  дерева-модели  $D$ .

Этот выбор базируется на некоторой информации  $I_j$  об “идеальном” ОАИ  $o_j \in M[O_j]$ . Эта информация представляет собой два множества:

$$I_j = I_j^{(1)} \cup I_j^{(2)},$$

где  $I_j^{(1)}$  - множество высказываний экспертов о “правильном поведении”  $\hat{o}_j$ ;

$I_j^{(2)}$  - множество результатов работы выбранного ОАИ.

Примерами элементов  $I_j^{(1)}$  могут служить высказывания типа “При  $d_{j_1} = a_{j_1}^1$  и  $d_{j_2} = a_{j_2}^1$  и ... и  $d_{j_{N_0}} = a_{j_{N_0}}^1$ , значение  $d_{j_0} = a_{j_0}^1$ ”, “При сильном возрастании  $d_{j_1}$  значение  $d_{j_0}$  убывает”, “Значение  $d_{j_0}$  монотонно изменяется по всем аргументам” и т.п.

Множество  $I_j^{(2)}$  представляет собой таблицу вида табл. 1.

Таблица 1.

$(d_{j_1}, d_{j_2}, \dots, d_{j_{N_0}})$	$d_{j_0}$
$a_{j_1}^1, a_{j_2}^1, \dots, a_{j_{N_0}}^1$	$\tilde{o}(a_{j_1}^1, a_{j_2}^1, \dots, a_{j_{N_0}}^1)$
$a_{j_1}^1, a_{j_2}^1, \dots, a_{j_{N_0}}^2$	$\tilde{o}(a_{j_1}^1, a_{j_2}^1, \dots, a_{j_{N_0}}^2)$
$\vdots$	$\vdots$
$a_{j_1}^1, a_{j_2}^1, \dots, a_{j_{N_0}}^{N_{N_0}}$	$\tilde{o}(a_{j_1}^1, a_{j_2}^1, \dots, a_{j_{N_0}}^{N_{N_0}})$
$\vdots$	$\vdots$
$a_{j_1}^{N_{j_1}}, a_{j_2}^{N_{j_2}}, \dots, a_{j_{N_0}}^{N_{N_0}}$	$\tilde{o}(a_{j_1}^{N_{j_1}}, a_{j_2}^{N_{j_2}}, \dots, a_{j_{N_0}}^{N_{N_0}})$

В левом столбце таблицы расположены все попарно различные значения  $d_{j_1}, d_{j_2}, \dots, d_{j_{N_0}}$ , в правом - значения  $d_{j_0}$ , полученные на основе информации  $I_j^{(1)}$  (то есть для этих строк  $\tilde{o}_j = \hat{o}_j$ ), или значения  $d_{j_0}$ , полученные в ходе работы с системой, или пустые значения.

В начале работы с системой таблица содержит только значения  $d_{j_0}$  первого типа (полученные на основе информации  $I_j^{(1)}$ ). По мере получения и оценки пользователем информации таблица заполняется на основе вычислений для выбранного оператора  $\hat{o}_j$  до момента, когда пользователь не согласен с “теоретическим” значением  $d_{j_0}$ . Если такое противоречие не возникает, значит оператор  $\hat{o}_j$  выбран удачно и является адекватным ОАИ для данного узла дерева - модели. Если противоречие возникает - необходимо повторить процедуру выбора адекватного ОАИ, но на основе дополненной и, быть может, уточненной с экспертом информации  $I_j'^{(1)}$  и  $I_j^{(2)}$ . Этот процесс повторяется до тех пор, пока вся таблица не будет заполнена. Полученная заполненная таблица и есть адекватный ОАИ для данной вершины  $d_{j_0}$ .

### 3. Выбор адекватных операторов агрегирования информации.

Описанная укрупненная схемы выбора ОАИ допускает несколько реализаций в зависимости от интерпретаций множества  $X_j$ . Мы можем интерпретировать данное множество как набор дискретных или как набор нечетких значений элемента  $d_j$ . Эта интерпретация зависит от свойств предметной области (проблемы). Так, если при оценке

информации набора значений  $a_j^1, a_j^2, \dots, a_j^{N_j}$  достаточно (ситуации, когда оценка находится “между” соседними значениями, либо отсутствуют, либо их мало), то мы можем говорить о дискретной модели выбора ОАИ. Если же часто при оценке поступающей в систему информации возникают ситуации, когда оценка находится “между” соседними значениями  $a_j^i, a_j^{i+1}$ , причем пользователь может говорить, что она более близка, например, к  $a_j^i$ , чем к  $a_j^{i+1}$ , то мы должны использовать нечеткую модель выбора ОАИ. Заметим, что дискретная модель является частным случаем нечеткой и может быть получена при замене соответствующих нечетких множеств на множества уровня 0,5 для полных ортогональных семантических пространств. Мы ее выделяем, однако, потому, что в этом случае возможна разработка специальных алгоритмов выбора ОАИ. Заметим также, что, как показывает практика, дискретная модель является допустимой при степени нечеткости  $\xi(d_j) \leq 0,2$  [5].

В зависимости от доступной информации  $I_j^{(1)}$  в рамках дискретной модели можно выделить два подхода к выбору ОАИ: геометрический и логический. Первый применим тогда, когда эксперт может только определить значение  $\hat{d}_j$  на некоторых наборах значений  $(d_{j_1}, d_{j_2}, \dots, d_{j_{N_0}})$ . Второй - когда кроме этого возможна формулировка некоторых условий на “поведение”  $\hat{d}_j$ . Эти подходы описаны в разделах 3.1 и 3.2 соответственно.

В рамках нечеткого подхода также предполагается, что известны значения  $\hat{d}_j$  на некоторых наборах значений  $(d_{j_1}, d_{j_2}, \dots, d_{j_{N_0}})$ . Эта информация представляется в виде логических высказываний вида “Если  $d_{j_1} = a_{j_1}^{k_1}$  и  $d_{j_2} = a_{j_2}^{k_2}$  и ... и  $d_{j_{N_0}} = a_{j_{N_0}}^{k_{N_0}}$ , то  $d_{j_0} = \hat{d}_{j_0}(a_{j_1}^{k_1}, a_{j_2}^{k_2}, \dots, a_{j_{N_0}}^{k_{N_0}})$ ” для всех известных наборов  $(d_{j_1}, d_{j_2}, \dots, d_{j_{N_0}})$ . Такое представление  $I_j^{(1)}$  позволяет использовать обучение либо на основе генетических алгоритмов (при выполнении достаточно естественных условий монотонности, коммутативности и ассоциативности ОАИ), либо на основе нейронных сетей (при выполнении данных условий). Эти подходы описаны в разделах 3.3 и 3.4 соответственно.

### 3.1. Геометрический подход.

Данный подход является исторически первым и основан на представлении оператора агрегирования информации в узле, имеющем  $n$  потомков, как некоторой поверхности в  $(n+1)$ - мерном пространстве [1].

Предполагается, что эксперт может определить значение в узле - родителе на некоторых наборах значений узлов - потомков. Эти наборы значений и значение в узле - родителе интерпретируются как точки в  $(n+1)$  - пространстве. Далее, исходя из этих точек и некоторых предположений строится поверхность. Примерами таких предположений могут быть:

- поскольку мы не имеем информации о значениях оператора на не определенных экспертом наборах, будем считать искомую поверхность кусочно - линейной (в смысле  $n$  - мерных гиперплоскостей);
- желательно удобное хранение оператора агрегирования, поэтому будем считать, что искомая поверхность является гладкой и описывается полиномом  $(k+1)$  степени, где  $k$  - число известных точек в нашем пространстве.

Несомненными достоинствами данного подхода является его простота и наглядность.

Недостатками является необходимость знать значение оператора агрегирования хотя бы на  $(n+1)$  наборе значений и невозможность в его рамках использования дополнительной информации от экспертов о поведении оператора агрегирования.

### **3.2. Логический подход.**

Если мы имеем конечное множество значений оценок в каждом узле дерева модели  $k$ , то мы можем представить оператор агрегирования информации как некоторую функцию  $k$  - значной логики. Если количество входов узла равно  $n$ , то в нем может быть использована в качестве оператора агрегирования одна из функций  $k$  - значной логики от  $n$  переменных. Обозначим множество всех таких функций через  $P_n^k$ .

Таких функций очень много ( $k^{k^n}$ ). Как выбрать одну из них для использования в качестве оператора агрегирования информации? Обычно эксперт может определить значение функции на некоторых наборах ее аргументов. В этом случае мы говорим о частично определенной функции. Пусть известны значения функции на  $t$  наборах. Обозначим класс таких функций через  $P_{n,t}^k$ . Число таких функций при большой разнице  $k^n - t$  также является необозримым. Если эксперт может сформулировать содержательные условия на поведение искомой функции типа “При сильном возрастании первого аргумента значение функции слегка убывает”, “При совместном возрастании аргументов 3 и 5 значение функции сильно возрастает” и т.п., мы можем говорить о нечетко заданных подклассах  $k$  - значной логики.

Рассмотрим  $P_{n,t}^k$  и обозначим через  $S$  множество нечетких условий на поведение функций из  $P_{n,t}^k$ . Можно сформулировать следующие задачи.

**Задача 1.** Являются ли условия  $S$  на поведение конкретной функции  $f \in P_{n,t}^k$  совместимыми или противоречивыми?

**Задача 2.** Если условия не противоречивы, можно ли каким-либо образом описать класс функций  $S(P_{n,t}^k)$ , им удовлетворяющих? В частности, можно ли предложить процедуру вычисления степени принадлежности любой  $f \in P_{n,t}^k$  нечетким условиям  $S$ ?

**Задача 3.** Если условия  $S$  противоречивы, можно ли сформулировать условия  $S^-$  ( $S^- \subset S$ ), максимально похожие на  $S$  и являющиеся непротиворечивыми?

Перечисленные задачи взаимосвязаны. Удобнее сначала привести решение задачи 2, так как решение задачи 1 может быть получено как некоторое свойство решения задачи 2.

Идея алгоритма решения задачи 2 описана в [3]. Она базируется на представлении нечеткого условия в виде некоторого нечеткого отношения. Данное отношение определяется на декартовом квадрате области определения функции и описывает поведение функции, ему удовлетворяющей, на соседних значениях области определения.

Для простоты сначала рассмотрим простейший случай. Пусть у нас есть одно нечеткое условие на поведение функции от одной переменной.

Нечеткое отношение  $\tilde{S}$ , соответствующее нечеткому условию  $S$ , описывает принадлежность функции к данному классу в на основе значений функции в точках  $i$  и  $i+1$  ( $1 \leq i \leq k-1$ ). Значение  $\mu_{\tilde{S}}(p, q)$  есть степень принадлежности функции к данному классу при условии, что  $f(i) = p, f(i+1) = q$  ( $0 \leq p, q \leq k-1$ ). Приведем примеры такой формализации обычных и нечетких условий.

**Пример 3.1.** Рассмотрим обычное, не нечеткое, условие возрастания функции. Оно разбивается на следующие локальные требования:

$$\forall i (1 \leq i \leq k-1) f(i) < f(i+1).$$

Матрица, описывающая это условие, имеет следующий вид:

f(i)\f(i+1)	0	1	2	3	4	5
0	0	1	1	1	1	1
1	0	0	1	1	1	1
2	0	0	0	1	1	1
3	0	0	0	0	1	1
4	0	0	0	0	0	1
5	0	0	0	0	0	0

Наружные строка и столбец содержат все возможные значения функции в двух соседних точках ( $i$  и  $i+1$ ). На пересечении  $p$ -ой строки и  $q$ -го столбца матрицы находится число из  $\{0,1\}$ , характеризующее степень принадлежности функции к описываемому классу при условии, что  $f(i) = p$ ,  $f(i+1) = q$ .

Ясно, что такую матрицу можно построить и для условия убывания функции.

**Пример 3.2.** Рассмотрим следующее нечеткое условие "*При возрастании  $x$  функция  $f(x)$  слегка возрастает*". Матрица, описывающая это условие, может иметь следующий вид:

$f(i)\backslash f(i+1)$	0	1	2	3	4	5
0	0.6	1	0.6	0.2	0	0
1	0	0.6	1	0.6	0.2	0
2	0	0	0.6	1	0.6	0.2
3	0	0	0	0.6	1	0.6
4	0	0	0	0	0.6	1
5	0	0	0	0	0	0.6

Так же как и в предыдущем примере, наружные строка и столбец содержат все возможные значения функции в двух соседних точках ( $i$  и  $i+1$ ). На пересечении  $p$ -ой строки и  $q$ -го столбца матрицы находится число из  $[0,1]$ , характеризующее степень принадлежности функции к описываемому классу при условии, что  $f(i) = p$ ,  $f(i+1) = q$ .

Функция удовлетворяет нечеткому условию, если она удовлетворяет ему для всех значений  $i$  ( $1 \leq i \leq k-1$ ). Таким образом, по матрице нечеткого отношения  $\tilde{S}$  степень принадлежности любой функции  $f \in P^k$  этому условию вычисляется однозначно. Она будет равна некоторой  $t$ -норме соответствующих степеней принадлежности из матрицы:

$$\mu_S(f) = \prod_{i=1}^{k-1} \mu_{\tilde{S}}(f(i), f(i+1)).$$

**Пример 3.3.** Рассмотрим две функции: возрастающую и “похожую” на возрастающую ( $f_1$  и  $f_2$  соответственно):

$x$	$f_1$	$f_2$
0	0	0
1	1	1
2	2	2
3	3	2
4	4	4
5	5	5

В качестве  $t$ -нормы возьмем умножение. Это представляется удобным, так как в этом случае при подсчете степени принадлежности функции мы будем учитывать ее поведение во всех соседних точках. Пусть  $S_1$  - условие строгого возрастания функции (таблица из примера 5.1),  $S_2$  - условие “слегка возрастает”, задаваемое таблицей из примера 5.2. Тогда

$$\mu_{S_1}(f_1) = \mu_{\tilde{S}_1}(0,1) \cdot \mu_{\tilde{S}_1}(1,2) \cdot \mu_{\tilde{S}_1}(2,3) \cdot \mu_{\tilde{S}_1}(3,4) \cdot \mu_{\tilde{S}_1}(4,5) = 1 \cdot 1 \cdot 1 \cdot 1 \cdot 1 = 1;$$

$$\mu_{S_1}(f_2) = \mu_{\tilde{S}_1}(0,1) \cdot \mu_{\tilde{S}_1}(1,2) \cdot \mu_{\tilde{S}_1}(2,2) \cdot \mu_{\tilde{S}_1}(2,4) \cdot \mu_{\tilde{S}_1}(4,5) = 1 \cdot 1 \cdot 0 \cdot 1 \cdot 1 = 0;$$

$$\mu_{S_2}(f_1) = \mu_{\tilde{S}_2}(0,1) \cdot \mu_{\tilde{S}_2}(1,2) \cdot \mu_{\tilde{S}_2}(2,3) \cdot \mu_{\tilde{S}_2}(3,4) \cdot \mu_{\tilde{S}_2}(4,5) = 1 \cdot 1 \cdot 1 \cdot 1 \cdot 1 = 1;$$

$$\mu_{S_2}(f_2) = \mu_{\tilde{S}_2}(0,1) \cdot \mu_{\tilde{S}_2}(1,2) \cdot \mu_{\tilde{S}_2}(2,2) \cdot \mu_{\tilde{S}_2}(2,4) \cdot \mu_{\tilde{S}_2}(4,5) = 1 \cdot 1 \cdot 0.6 \cdot 0.6 \cdot 1 = 0.36.$$

Нечеткое отношение  $\tilde{S}$ , соответствующее нечеткому условию  $S$ , может служить основой для построения нечеткого отношения  $\tilde{\tilde{S}}$ , описывающего все функции, удовлетворяющие условию  $S$  и некоторому начальному условию.

Рассмотрим для простоты конкретный класс функций  $P_{1,1}^k$ . Отношение  $\tilde{\tilde{S}}$  также как и отношение  $\tilde{S}$ , определяется на декартовом квадрате области определения функций рассматриваемого класса. Пусть, для определенности, начальным условием является  $f(0) = 0$ . Тогда первой строкой матрицы нечеткого отношения  $\tilde{\tilde{S}}$ , соответствующего значению аргумента 0, является строка  $(1, 0, \dots, 0)$  - как выражение начального условия.

Рассмотрим значение  $\mu_{\tilde{\tilde{S}}}(2,1)$ . Оно соответствует ситуации, когда  $f(1) = 0$ .

Учитывая начальное условие, мы можем сформулировать вопрос по другому: с какой



степенью функция  $f$  удовлетворяет условию  $S$ , если  $f(0) = 0$  и  $f(1) = 0$ ? Очевидно, это элемент матрицы отношения  $\tilde{S}$  с координатами (1,1).

Рассмотрим значение  $\mu_{\tilde{S}}(2,2)$ . Оно соответствует ситуации, когда  $f(1) = 1$ .

Учитывая начальное условие, мы можем сформулировать вопрос по другому: с какой степенью функция  $f$  удовлетворяет условию  $S$ , если  $f(0) = 0$  и  $f(1) = 1$ ? Очевидно, это элемент матрицы отношения  $\tilde{S}$  с координатами (1,2).

Повторив эти рассуждения  $k$  раз, мы получим, что вторая строка матрицы отношения  $\tilde{S}$  в точности совпадает с первой строкой матрицы отношения  $\tilde{S}$ .

Рассмотрим значение  $\mu_{\tilde{S}}(3,1)$ . Оно соответствует ситуации, когда  $f(2) = 0$ .

Учитывая начальное условие, мы можем сформулировать вопрос по другому: с какой степенью функция  $f$  удовлетворяет условию  $S$ , если  $f(0) = 0$  и  $f(2) = 0$ ? Ответ на этот вопрос можно сформулировать следующим образом:

$$\left. \begin{array}{l} \text{если } f(1) = 0, \quad \text{то } f(2) = 0 \quad \text{со степенью уверенности } \mu_S(1,1) \quad \text{или} \\ \text{если } f(1) = 1, \quad \text{то } f(2) = 0 \quad \text{со степенью уверенности } \mu_S(2,1) \quad \text{или} \\ \vdots \\ \text{если } f(1) = k - 1, \quad \text{то } f(2) = 0 \quad \text{со степенью уверенности } \mu_S(k,1) \end{array} \right\} \quad (2)$$

Заметим, что степень выполнимости условий  $f(1) = i$  ( $0 \leq i \leq k-1$ ) правой части приведенных высказываний в точности совпадает со второй строкой матрицы отношения  $\tilde{S}$  при анализируемом начальном условии. Это означает, что мы можем переписать условия (2) следующим образом:

$$\mu_{\tilde{S}}(3,1) = \bigwedge_{i=1}^k \mu_{\tilde{S}}(2,i) \times \mu_{\tilde{S}}(i,1) \quad (3)$$

Повторив приведенные рассуждения для  $\mu_{\tilde{S}}(3,j)$  ( $2 \leq j \leq k$ ), получим следующие аналоги (3):

$$\mu_{\tilde{S}}(3,j) = \bigwedge_{i=1}^k \mu_{\tilde{S}}(2,i) \times \mu_{\tilde{S}}(i,j), \quad (1 \leq j \leq k). \quad (4)$$

Рассмотрим значение  $\mu_{\tilde{S}}(l,1)$  ( $3 < l \leq k$ ). Повторив рассуждения для  $\mu_{\tilde{S}}(3,1)$ , получим следующий аналог (3):

$$\mu_{\tilde{S}}(l,1) = \bigoplus_{i=1}^k \mu_{\tilde{S}}(l-1,i) \times \mu_{\tilde{S}}(i,1) \quad (5)$$

и, аналогично (4), его обобщение:

$$\mu_{\tilde{S}}(l,j) = \bigoplus_{i=1}^k \mu_{\tilde{S}}(l-1,i) \times \mu_{\tilde{S}}(i,j), \quad (1 \leq j \leq k; 2 \leq l \leq k). \quad (6)$$

Формулу (6) можно интерпретировать как своеобразное умножение предыдущей строки матрицы отношения  $\tilde{S}$  на столбцы матрицы отношения  $\tilde{S}$ , где вместо операции сложения используется операция взятия  $t$ -конормы. Заметим при этом, что первый столбец матрицы нечеткого отношения  $\tilde{S}$  совпадает с начальным условием.

Рассмотрим ситуацию, когда начальное условие сформулировано не для  $f(0)$ , а для  $f(k-1)$ . Пусть, для определенности, это будет условие  $f(k-1) = q$  ( $1 \leq q \leq k-1$ ). Это условие, аналогично рассмотренному выше случаю, определяет  $k$ - строку матрицы отношения  $\tilde{S}$  - строка содержит одни нули за исключением 1 в  $q$  столбце.

Рассмотрим значение  $\mu_{\tilde{S}}(k-2,1)$ . Оно соответствует ситуации, когда  $f(k-2) = 0$ . Учитывая начальное условие, мы можем сформулировать вопрос по другому: с какой степенью функция  $f$  удовлетворяет условию  $S$ , если  $f(k-2) = 0$  и  $f(k-1) = q$ ? Очевидно, это элемент матрицы отношения  $\tilde{S}$  с координатами  $(1, q)$ .

Рассмотрим значение  $\mu_{\tilde{S}}(k-2,2)$ . Оно соответствует ситуации, когда  $f(k-2) = 1$ . Учитывая начальное условие, мы можем сформулировать вопрос по другому: с какой степенью функция  $f$  удовлетворяет условию  $S$ , если  $f(k-2) = 1$  и  $f(k-1) = q$ ? Очевидно, это элемент матрицы отношения  $\tilde{S}$  с координатами  $(2, q)$ .

Повторив эти рассуждения  $k$  раз, мы получим, что  $(k-1)$  - строка матрицы отношения  $\tilde{S}$  в точности совпадает с  $q$ -столбцом матрицы отношения  $\tilde{S}$ .

Рассмотрим значение  $\mu_{\tilde{S}}(k-3,1)$ . Оно соответствует ситуации, когда  $f(k-3) = 0$ . Учитывая начальное условие, мы можем сформулировать вопрос по другому: с какой

степенью функция  $f$  удовлетворяет условию  $S$ , если  $f(k-1) = q$  и  $f(k-3) = 0$ ? Ответ на этот вопрос, аналогично рассмотренному выше случаю с начальным условием на  $f(0)$ , можно сформулировать следующим образом:

$$\left. \begin{array}{l} \text{если } f(k-2) = 0, \quad \text{то } f(k-3) = 0 \quad \text{со степенью уверенности } \mu_S(1,1) \quad \text{или} \\ \text{если } f(k-2) = 1, \quad \text{то } f(k-3) = 0 \quad \text{со степенью уверенности } \mu_S(1,2) \quad \text{или} \\ \vdots \\ \text{если } f(k-2) = k-1, \quad \text{то } f(k-3) = 0 \quad \text{со степенью уверенности } \mu_S(1,k) \end{array} \right\} (7)$$

Заметим, что степень выполнимости условий  $f(k-2) = i$  ( $0 \leq i \leq k-1$ ) правой части приведенных высказываний в точности совпадает с  $(k-1)$  строкой матрицы отношения  $\tilde{S}$  при анализируемом начальном условии. Это означает, что мы можем переписать условия (7) следующим образом:

$$\mu_{\tilde{S}}^{\approx}(k-3,1) = \bigoplus_{i=1}^k \mu_{\tilde{S}}^{\approx}(k-2,i) \times \mu_{\tilde{S}}(1,i) \quad (8)$$

Повторив рассуждения вывода формулы (5.6), получаем следующий аналог этой формулы для начального условия на  $f(k-1)$ :

$$\mu_{\tilde{S}}^{\approx}(l,j) = \bigoplus_{i=1}^k \mu_{\tilde{S}}^{\approx}(l+1,i) \times \mu_{\tilde{S}}(j,i), \quad (1 \leq j \leq k; 1 \leq l \leq k-2). \quad (9)$$

Формулу (9) можно также интерпретировать как умножение предыдущей строки матрицы отношения  $\tilde{S}$  на строки матрицы отношения  $\tilde{S}$ , где вместо операции сложения используется операция взятия  $t$ -конормы. Заметим, что данное отличие в построении матрицы отношения  $\tilde{S}$  для начальных условий на  $f(0)$  и  $f(k)$  видно уже из сравнения условий (2) и (7). Будем называть процедуру построения  $\tilde{S}$  для начального условия на  $f(0)$  построением сверху вниз, а для условия на  $f(k)$  - построением снизу вверх.

Рассмотрим теперь ситуацию, когда начальное условие сформулировано для некоторого промежуточного значения  $f(l^*)$  ( $1 < l^* < k-1$ ). Пусть, для определенности, это будет условие  $f(l^*) = q$  ( $1 \leq q \leq k-1$ ). Это условие, аналогично рассмотренным выше

случаям, определяет  $l$  - строку матрицы отношения  $\tilde{S}$  - строка содержит одни нули за исключением 1 в  $q$  столбце. Достаточно очевидно, что в этом случае строки матрицы отношения  $\tilde{S}$ , расположенные выше  $l$  - строки, строятся снизу вверх, а строки, расположенные ниже ее - сверху вниз.

Таким образом, общий метод построения матрицы отношения  $\tilde{S}$  определяется следующей формулой:

$$\mu_{\tilde{S}}(l, j) = \begin{cases} \bigwedge_{i=1}^k \mu_{\tilde{S}}(l+1, i) \times \mu_{\tilde{S}}(j, i) & \text{при } 1 \leq l < l^* \\ \bigwedge_{i=1}^k \mu_{\tilde{S}}(l-1, i) \times \mu_{\tilde{S}}(i, j) & \text{при } l^* < l \leq k \end{cases}, (1 \leq l, j \leq k). \quad (10)$$

Так построенная матрица отношения  $\tilde{S}$  позволяет определить все функции  $f \in P_{l,l}^k$ , удовлетворяющие нечеткому условию  $S$  со степенью принадлежности не ниже заданной. Действительно, любая такая функция определяет путь в  $\tilde{S}$ , вершины которого лежат по одной на каждой строке матрицы. Каждому из  $k-1$  ребра этого пути приписывается число, равное значению соответствующего элемента матрицы. Таким образом, для того чтобы выписать все функции, удовлетворяющие нашему нечеткому условию со степенью принадлежности не ниже заданной, необходимо выписать из матрицы отношения  $\tilde{S}$  все пути, проходящие через вершины, значения которых не ниже заданного числа. После этого степень принадлежности каждой функции считается как  $t$  - норма значений ребер. Это позволяет утверждать, что задача 2 нами решена полностью.

Данное свойство отношения  $\tilde{S}$  позволит нам в дальнейшем решить задачу 1 - выяснение противоречивости нечетких условий на поведение функций из  $P_{n,t}^k$ .

Сначала рассмотрим ситуацию, когда для функции от одной переменной задано одно нечеткое условие и  $t$  начальных условий, то есть ситуацию  $|S| = 1$  и класс  $P_{l,t}^k$ ,  $t > 1$ . Это означает требование для функции выполнения всех начальных условий. В рамках теории нечетких множеств данное требование означает взятие некоторой  $t$  - нормы. Таким образом, если начальных условий несколько, то матрица  $\tilde{S}_i$  строится для каждого  $i$  - условия в отдельности ( $1 < i \leq t$ ), а итоговая матрица  $\tilde{S}$  получается следующим образом:

$$\tilde{S} = \prod_{i=1}^t \tilde{S}_i. \quad (5.11)$$

В заключение изучения класса  $P_{l,t}^k$  рассмотрим случай  $|S| = s > 1$ . Для каждого из нечетких условий  $S^r$  ( $1 \leq r \leq s$ ) мы можем построить матрицу отношения  $\tilde{S}^r$ , описывающую все функции, ему удовлетворяющие. Чтобы определить совместимость этих условий, необходимо получить обобщенную матрицу  $\tilde{S}_s$ , которая бы описывала функции, удовлетворяющие всем условиям  $S^r$  ( $1 \leq r \leq s$ ). В данной ситуации это означает, что  $f \in S(P_{l,t}^k) \Leftrightarrow \forall r (1 \leq r \leq s) f \in S^r(P_{l,t}^k)$ . Аналогом этого высказывания для матриц отношений  $S^r$  является следующее выражение:

$$\tilde{S}_s = \prod_{r=1}^s \tilde{S}^r. \quad (12)$$

На основе сформулированных свойств матрицы отношения  $\tilde{S}$ , мы можем утверждать, что если полученная в соответствии с (12) матрица  $\tilde{S}_s$  имеет хотя бы одну нулевую строку, то множество условий  $\{S^r\}_{r=1}^s$  является противоречивым, так как любая функция  $f$  из  $P_{l,t}^k$  удовлетворяет им с нулевой степенью (в силу свойства ограниченности  $t$ -нормы). Более того, используя матрицу отношения  $\tilde{S}_s$ , мы, аналогично случаю с одним нечетким условием, сможем описать все функции, удовлетворяющие нечетким условиям  $\{S^r\}_{r=1}^s$ . Таким образом, любое количество нечетких условий по одной переменной формально легко сводится к одному нечеткому условию по этой переменной. Учитывая данное замечание, в последующих рассуждениях будем считать, что по каждой переменной больше одного условия не дается.

Описанную технику легко обобщить на случай функций от нескольких переменных. Рассмотрим класс функций  $P_{n,t}^k$  и множество нечетких условий  $S$ . Для простоты изложения положим  $n = 2$  и  $|S| = 2$ . Пусть первое нечеткое условие определено по первой переменной, второе условие - по второй переменной.

Мы можем описанным выше способом построить матрицы отношений  $\tilde{S}^1$  и  $\tilde{S}^2$ . Удовлетворение условиям  $S$  означает одновременное удовлетворение условий  $S^1$  и  $S^2$ . Это в свою очередь означает, что на наборе  $(i_1, i_2)$  значений переменных  $x_1$  и  $x_2$  ( $i_1, i_2 \in \{0, 1, \dots, k-1\}$ ) мы в качестве значения строки матрицы отношения  $\tilde{S}$  должны взять  $t$

- норму  $(i_1 + 1)$  строки матрицы отношения  $\tilde{S}^1$  и  $(i_2 + 1)$  строки  $\tilde{S}^2$ . Полученная таким образом матрица размером  $(k^2 \times k)$  и будет искомой.

Данная матрица обладает всеми свойствами матриц отношений для одной переменной. Мы точно также можем поставить в соответствие любой функции из анализируемого класса путь в матрице и определить согласованность/противоречивость нечетких условий на поведение функций и вычислить степень удовлетворения любой функции этим условиям. Таким образом, задачи 1 и 2 решены и в общем случае.

Рассмотрим задачу 3. Пусть у нас есть противоречивая система  $S$  из  $n$  нечетких условий. Для решения задачи можно предложить следующий достаточно очевидный и переборный по своей природе алгоритм.

Проверим на противоречивость все нечеткие условия системы попарно. Отметим противоречивые пары. Далее проверяем на противоречивость все тройки нечетких условий, в которые не входят противоречивые пары. Отметим противоречивые тройки. Эта операция повторяется, пока на шаге  $s$ ,  $1 \leq s \leq n-1$  не окажется, что все кортежи длины  $s+1$  нечетких условий противоречивы. Тогда максимально похожими на  $S$  непротиворечивыми подсистемами окажутся подсистемы, выписанные на шаге  $s-1$  и состоящие из  $s$  нечетких условий.

### 3.3. Обучение на основе генетических алгоритмов.

Обучение на основе генетических алгоритмов применимо для нечеткой модели выбора операторов агрегирования информации, то есть когда при оценке поступающей в систему информации возникают ситуации, когда оценка находится “между” соседними значениями  $a_j^i$ ,  $a_j^{i+1}$ , причем пользователь может говорить, что она более близка, например, к  $a_j^i$ , чем к  $a_j^{i+1}$ . Напомним, что в рамках нечеткого подхода также предполагается, что известны значения  $\hat{\delta}_j$  на некоторых наборах значений  $(d_{j_1}, d_{j_2}, \dots, d_{j_{N_0}})$ . Эта информация представляется в виде логических высказываний вида “Если  $d_{j_1} = a_{j_1}^{k_1}$  и  $d_{j_2} = a_{j_2}^{k_2}$  и ... и  $d_{j_{N_0}} = a_{j_{N_0}}^{k_{N_0}}$ , то  $d_{j_0} = \hat{\delta}_{j_0}(a_{j_1}^{k_1}, a_{j_2}^{k_2}, \dots, a_{j_{N_0}}^{k_{N_0}})$ ” для всех известных наборов  $(d_{j_1}, d_{j_2}, \dots, d_{j_{N_0}})$ . Кроме этого, предполагается, что ОАИ является монотонным, коммутативным и ассоциативным (то есть имеет смысл обобщенных операций “И”, “ИЛИ”). В рамках теории нечетких множеств эти операторы в наиболее общем виде выражаются через  $t$ -нормы (“И”) и  $t$ -конормы (“ИЛИ”) [5]. Таких  $t$ -норм и  $t$ -конорм бесконечно много. Какую из них взять в качестве оператора агрегирования, снимающего противоречие в системе?

Предлагается для выбора такого оператора использовать процедуру обучения на основе генетических алгоритмов [6].

Наиболее активно процедуры обучения используются при разработке нечетких контроллеров [5], то есть аппаратно реализованных систем правил нечеткого логического вывода. Нечеткие контроллеры в течение значительного времени используются во многих областях от бытовой техники до систем вооружений и показали свою простоту, надежность и эффективность. Обучение производится путем коррекции по определенным правилам функций принадлежности используемых нечетких переменных.

Непосредственное использование такого типа процедур при обучении нечетких иерархических систем не представляется естественным, так как приводит к разной формализации одних и тех же лингвистических значений в разных узлах модели. Поэтому предлагается в качестве параметра обучения использовать не функции принадлежности, а параметры  $t$ -норм и  $t$ -конорм. Раньше так задача обучения не ставилась. В качестве средства ее решения было предложено использовать генетические алгоритмы (см., например, [8]).

Для оценки реализуемости такой процедуры обучения была разработана система, описание которой приводится ниже.

Возьмем в качестве параметрической  $t$ -нормы строгую архимедову  $t$ -норму:

$$H_\lambda = \frac{M_A \cdot M_B}{1 + (1 - \lambda)(M_A + M_B - M_A \cdot M_B)}, \quad \text{где } 0 \leq \lambda \leq \infty.$$

В частных случаях имеем известные в рамках теории нечетких множеств формализации операции пересечения :

$$\begin{aligned} \text{- при } \lambda = 0 & \quad H_0 = \frac{M_A \cdot M_B}{M_A + M_B - M_A \cdot M_B}, \\ \text{- при } \lambda = 1 & \quad H_1 = M_A \cdot M_B, \\ \text{- при } \lambda = \infty & \quad H_\infty = \begin{cases} \mu_A, & \text{если } \mu_B = 1, \\ \mu_B, & \text{если } \mu_A = 1, \\ 0, & \text{иначе.} \end{cases} \end{aligned}$$

Для вычисления выхода системы правил используется строгая архимедова  $t$ -конорма:

$$H^*_{\lambda} = \frac{M_A + M_B - (2 - \lambda) \cdot M_A \cdot M_B}{1 - (1 - \lambda) \cdot (M_A \cdot M_B)}$$

Для дефазификации выхода системы правил возьмем метод нахождения центра тяжести композиции  $t$ -нормы и  $t$ -конормы, наиболее часто используемый и самый типичный.

Для поиска параметров первоначально был опробован классический генетический алгоритм.

Хромосомы – бинарные строки, представляющие двоичный код вещественных параметров, размерности  $32 \cdot (R+1)$  (по числу битов отвечающих за параметр на число правил в системе и один параметр  $t$ -конормы). Был реализован одноточечное скрещивание, оставляющее двух потомков, и мутация – бинарный код случайного вещественного числа, лежащего в области определения параметра. Функция пригодности вычислялась как среднеквадратичная ошибка, определяемая расстоянием между  $\tilde{Y}$  - выходом контроллера и  $Y^*$  - выходом обучающей выборки выходом:

$$\tilde{Y}^1 = \sqrt{\sum_{i=1}^D (Y_i - Y_i^*)^2 / D},$$

где  $D$  - размерность выборки.

Однако, при экспериментах с программой возникали ситуации ее непозволительно долгой (до нескольких часов) работы или не достижении требуемой ошибки. Для оптимизации стандартного генетического алгоритма было опробовано усиление мутации (как увеличение ее вероятности, так и разрешение мутации не в одном, а в нескольких генах хромосомы), выживание только одного потомка скрещивания их комбинации. Однако это практически не повлияло на его сходимость (некоторые результаты экспериментов приведены в таблице 2).

Для преодоления указанных трудностей было предпринято изменение самого принципа кодирования данных. Использовалось кодирование, отражающее структуру пространства поиска - представление хромосомы, как массива вещественных чисел размерности  $R+1$  (параметры  $t$ -нормы каждого правила и параметр  $t$ -конормы). Это повлекло за собой необходимость разработать соответствующие операторы рекомбинации.

*Мутация* была реализована как получение случайного числа принадлежащего области определения параметра  $\lambda$  (от 0 до  $\infty$ ). Вероятность мутации просчитывалась для каждой хромосомы популяции.



Таблица 2.

Число поколений	Число запусков алгоритма	Средняя минимальная ошибка последнего поколения	
		Классический алгоритм	Модифицированный алгоритм
3000	5	-	0,00000000005
2000	5	-	0,0000000001
1000	10	0,000009412333	0,0000000042
500	10	0,00007227655	0,0000059195
200	10	0,000136715	0,0000554
100	10	0,00014429	0,0001553

В качестве аналога скрещивания использовался новый оператор - “почкование”, включающий в себя следующие этапы:

1. выбор случайного числа близкого к гену “родительской” хромосомы (около 10% от величины самого числа);
2. повторение первого шага для каждого гена хромосомы.

В программной реализации этот оператор представляет собой добавление к гену нормально распределенной, близкой к нулю случайной величины (положительной или отрицательной). Управление “масштабом” осуществлялось изменением дисперсии этого случайного числа. Вероятность почкования также просчитывалась для каждой хромосомы популяции.

В качестве оператора отбора использовался простой (не элитный) пропорциональный отбор по функции качества.

Угруппенная схема программы выглядит следующим образом:

#### Инициализация:

На этом этапе работы алгоритма инициализируется случайный набор хромосом. Определяется качество каждой хромосомы, в соответствии с которым хромосомы ранжируются.

#### Шаг:

1. В новом поколении с вероятностью воспроизведения, рассчитанной оператором отбора, воспроизводятся дубликаты лучших хромосом и генерируются хромосомы, полученные почкованием и мутацией.

2. Проверяется “качественность” - минимальная среднеквадратичная ошибка хромосомы – для каждой хромосомы популяции.
3. Отмирают “неприспособленные” индивидуумы популяции.
4. Если качество лучшей хромосомы популяции удовлетворяет запросам задачи, то лучшая хромосома и является решением задачи - гены ее составляющие являются параметрами искомой  $t$ -нормы.

Иначе шаг алгоритма повторяется.

Итак, задачу выбора адекватных ОАИ при выполнении описанных выше предположений удалось сформулировать в терминах генетических алгоритмов и предложить ее эффективное решение, доведенное до уровня компьютерной реализации.

Выбор адекватного ОАИ в этом случае выглядит следующим образом.

1. При начальной инициализации системы для каждой не концевой вершины  $d_j$  дерева - модели на основе информации  $I_j^{(1)}$  формируется таблица вида таблицы 1.
2. Полученная таблица рассматривается как обучающая выборка. Предложенным методом выбираются параметры  $t$  - нормы.
3. Во время работы с системой все изменения значений вершины  $d_j$  и подчиненных ей вершин, не приводящие к возникновению конфликтной ситуации, заносятся в таблицу. Полученные на данном шаге записи таблицы образуют  $I_j^{(2)}$ .
4. В случае возникновения конфликта переходим к шагу 2.

### 3.4. Обучение на основе нейронных сетей.

Обучение на основе нейронных сетей, так же как и обучение на основе генетических алгоритмов, используется для нечеткой модели выбора операторов агрегирования информации. Отличие в применимости этих подходов состоит в том, что для данного подхода не требуется выполнения свойств монотонности, коммутативности и ассоциативности оператора агрегирования информации.

Здесь также, как и в рамках нечеткого подхода, предполагается, что известны значения  $\hat{o}_j$  на некоторых наборах значений  $(d_{j_1}, d_{j_2}, \dots, d_{j_{N_0}})$ . Эта информация представляется в виде логических высказываний вида “Если  $d_{j_1} = a_{j_1}^{k_1}$  и  $d_{j_2} = a_{j_2}^{k_2}$  и ... и  $d_{j_{N_0}} = a_{j_{N_0}}^{k_{N_0}}$ , то  $d_{j_0} = \hat{o}_{j_0}(a_{j_1}^{k_1}, a_{j_2}^{k_2}, \dots, a_{j_{N_0}}^{k_{N_0}})$ ” для всех известных наборов  $(d_{j_1}, d_{j_2}, \dots, d_{j_{N_0}})$ .

Задачу выбора адекватного ОАИ можно сформулировать как задачу его распознавания.

Будем считать, что объект описывается конечным набором признаков  $A = \{A_1, \dots, A_n\}$ . Каждому признаку  $A_i$  ставится в соответствие множество  $U_i$  его “физических” значений и множество  $\{a_{1i}, \dots, a_{n_i i}\}$  лингвистических значений ( $1 \leq i \leq n$ ). Каждому такому лингвистическому значению  $a_{ji}$  ставится в соответствие функция принадлежности  $\mu_{a_{ji}}(u_i)$  в универсальном множестве  $U_i$  ( $1 \leq j \leq n_i$ ). Пусть, кроме этого, есть  $K$  классов  $C_k, k \in \{1 \dots K\}$ . Информация о классах в терминологии [2] задана в виде совокупности  $r$  правил:

$$\left. \begin{array}{l} \text{Если } A_1 = a_{i_1 1}^1 \text{ и } A_2 = a_{i_2 2}^1 \text{ и } \dots \text{ и } A_n = a_{i_n n}^1 \text{ то } \mu_{C_1}(O) = \eta_1^1 \text{ и } \dots \text{ и } \mu_{C_k}(O) = \eta_k^1 \\ \text{Если } A_1 = a_{i_1 1}^2 \text{ и } A_2 = a_{i_2 2}^2 \text{ и } \dots \text{ и } A_n = a_{i_n n}^2 \text{ то } \mu_{C_1}(O) = \eta_1^2 \text{ и } \dots \text{ и } \mu_{C_k}(O) = \eta_k^2 \\ \vdots \\ \text{Если } A_1 = a_{i_1 1}^r \text{ и } A_2 = a_{i_2 2}^r \text{ и } \dots \text{ и } A_n = a_{i_n n}^r \text{ то } \mu_{C_1}(O) = \eta_1^r \text{ и } \dots \text{ и } \mu_{C_k}(O) = \eta_k^r \end{array} \right\} (13)$$

где объект  $O$  (правая часть правил) имеет значения признаков, перечисленных в левой части правил.

На вход классификатора подается вектор вещественных чисел  $u^* = (u_1, \dots, u_N)$  (“физические” значения признаков объекта). Выходом является вектор  $M^* = (M_1, \dots, M_K)$ , описывающий принадлежность объекта классам  $C_1 \dots C_K$ .

Для решения данной задачи использовалась следующая нейронная сеть (рис. 1).

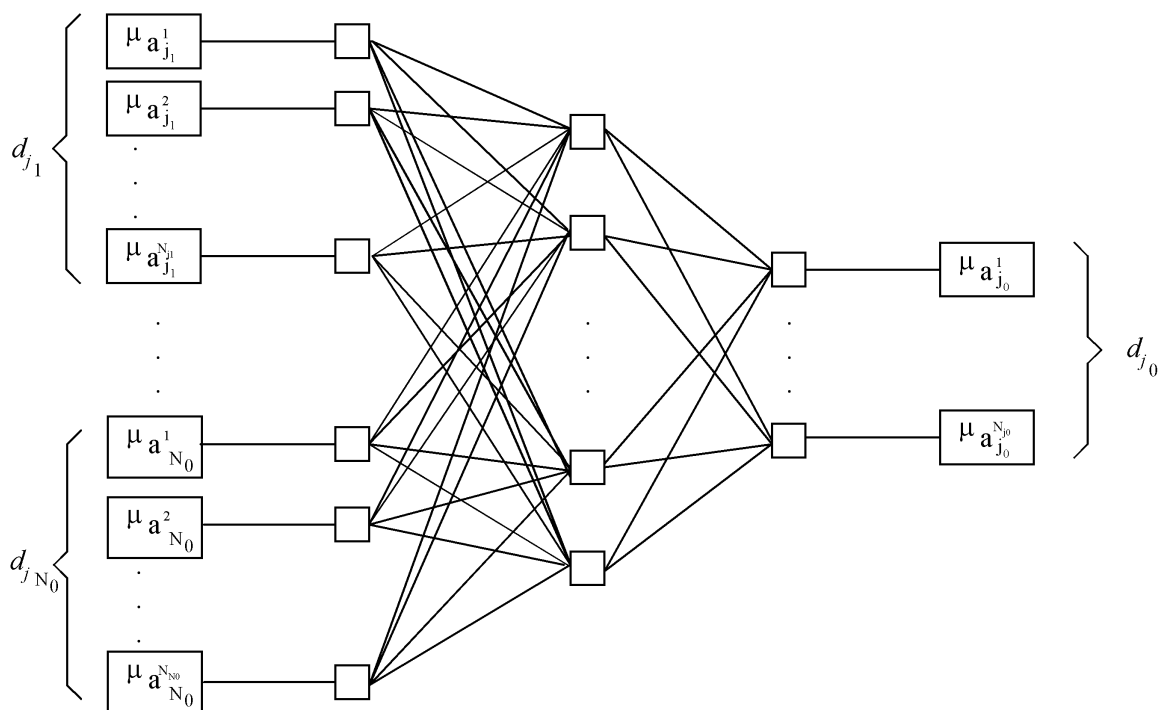


Рис. 1.

Входной слой состоит из  $|X_{j_1}| \times |X_{j_2}| \times \dots \times |X_{j_{N_0}}|$  нейронов (то есть каждому возможному значению подчиненной вершины соответствует один нейрон слоя). Нейроны во втором слое соответствуют четким правилам, задаваемым экспертом, то есть таким правилам, которые однозначно относят объект к тому или иному классу. Эти правила соответствуют строкам таблицы 1. Нейроны в третьем слое соответствуют значениям признака в вершине  $d_{j_0}$ . Все нейроны первого слоя соединены с нейронами второго слоя, и все нейроны второго слоя - с нейронами третьего слоя.

Начальные веса задаются следующим образом. Нейроны первого слоя соединяются с нейронами во втором слое весом равным единице, если соответствующее правило содержит в себе это значение входа. То же самое верно и для весов идущих к последнему слою: если правило относит объект к тому или иному значению признака в вершине  $d_{j_0}$ , то вес между соответствующими нейронами будет равен единице. Оставшиеся веса инициализируются маленькими числами порядка 0.1 (это необходимо, для того чтобы у них была возможность обучаться).

Функционирование происходит по следующим правилам. Для каждой входной переменной вычисляется степень принадлежности ее физического значения к множествам значений признака (это значение соответствующей функции принадлежности). Эти числа подаются на вход первого слоя. Нейроны во втором и третьем слоях вычисляют выход по формуле

$$Out^k = f\left(\sum_{i=1}^r w_i^k In_i\right)$$

где  $f(x) = 2 / (1 + e^{-2x}) - 1$  - (активационная функция нейрона)

$k$  - номер нейрона в слое.

$Out^k$  - выход  $k$  нейрона.

$i$  - номер нейрона в предшествующем слое.

$w_i^k$  - вес от нейрона с номером  $i$  в предшествующем слое к нейрону с номером  $k$ .

$In_i$  - сигнал от  $i$  нейрона в предшествующем слое.

Начальная обучающая выборка состоит из правил, задаваемые экспертом (аналог таблицы

1). Она представляет собой набор  $t$  пар

$$T_k = (I_k, O_k) \quad (1 \leq k \leq t),$$

где  $t$  - мощность обучающей выборки;

$I_k$  - вектор длины  $|X_{j_1}| \times |X_{j_2}| \times \dots \times |X_{j_{N_0}}|$ , содержащий  $N_0$  единиц, соответствующих значениям признаков подчиненных вершин (одна единица в одной из позиций с номером от 1 до  $|X_{j_1}|$ , одна единица в одной из позиций с номером от  $|X_{j_1}| + 1$  до  $|X_{j_2}|$  и т.д.);

$O_k$  - вектор длины  $|X_{j_0}|$ , содержащий одну единицу, соответствующую значению признака в вершине  $d_{j_0}$ .

Таким образом, структура сети и начальной обучающей выборки моделирует ситуацию, в которой множества значений входных переменных определены четко.

Процедура обучения нейронной сети выглядит следующим образом.

1. Выбирается вектор  $T_k$  из обучающей выборки ( $1 \leq k \leq t$ ) и подается на вход первого слоя;

2. По описанным выше правилам вычисляются выход второго слоя

$$O^2 = (Out_1^2, Out_2^2, \dots, Out_t^2) \text{ и выход сети } O^3 = (Out_1^3, \dots, Out_{|X_{j_0}|}^3);$$

3. Вычисляется вектор ошибок  $E^3 = (e_1^3, \dots, e_{|X_{j_0}|}^3)$ , где  $e_i^3 = O_i^k - Out_i^3$ , ( $1 \leq i \leq |X_{j_0}|$ );

4. Вычисляется вектор  $\Delta_3 = (\delta_1^3, \dots, \delta_{|X_{j_0}|}^3)$ , где  $\delta_i^3 = \frac{df}{dx}(Out_i^3) e_i^3 = (Out_i^3 + 1)(1 - Out_i^3) e_i^3$  ( $1 \leq i \leq |X_{j_0}|$ );

5. Вычисляется вектор ошибок второго слоя  $E^2 = (e_1^2, \dots, e_t^2)$ , где  $e_i^2 = \sum_{k=1}^{|X_{j_0}|} w_{2i}^k e_k^3 = E^3 W_2$ ;

$w_{2i}^k$  - вес от  $i$  - нейрона во втором слое к  $k$  - нейрону в третьем слое;  $W_2$  матрица весов второго слоя (элементами ее являются  $w_{2i}^k$ ); ( $1 \leq i \leq t$ );

6. Вычисляется вектор  $\Delta_2 = (\delta_1^2, \dots, \delta_t^2)$ , где  $\delta_i^2 = \frac{df}{dx}(Out_i^2) e_i^2 = (Out_i^2 + 1)(1 - Out_i^2) e_i^2$  ( $1 \leq i \leq t$ );

7. Веса сети корректируются по формуле  $w_{p-1,i}^j(n+1) = w_{p-1,i}^j(n) + S \delta_j^p \alpha$  ( $p \in \{2,3\}$ ), где

- $w_{p,i}^j$  вес связи от нейрона с номером  $i$  в  $p$ - слое к нейрону с номером  $j$  в слое с номером  $p+1$ ,
- $S$  - сигнал проходивший по этой связи (для нейронов второго слоя это входной сигнал - соответствующий элемент вектора  $I_k$ );
- $\alpha$  - коэффициент скорости обучения (константа от 0.1 до 1).

Обучение происходит до тех пор, пока среднеквадратичная ошибка по всем элементам выборки не станет меньше заранее заданной величины

Выбор адекватного ОАИ в этом случае выглядит следующим образом.

1. При начальной инициализации системы для каждой не концевой вершины  $d_j$  дерева - модели на основе информации  $I_j^{(1)}$  формируется обучающая выборка  $T_k^j = (I_k^j, O_k^j)$  ( $1 \leq k \leq t_j$ ).
2. Описанным методом проводится обучение сети.
3. Во время работы с системой все изменения значений вершины  $d_j$  и подчиненных ей вершин, не приводящие к возникновению конфликтной ситуации, заносятся в таблицу. Полученные на данном шаге записи таблицы образуют  $I_j^{(2)}$ .
4. В случае возникновения конфликта из  $I_j^{(2)}$  формируется новая обучающая выборка и повторяется шаг 2.

## Литература.

1. Ананич И.С., Беленький А.Г., Пронин Л.Б., Рыжов А.П. Агрегирование информации в системах информационного мониторинга. *Труды Международного семинара "Мягкие вычисления - 96"*. Казань, 3-6 октября 1996, с. 43 - 46.
2. Журавлев Ю.И. Об алгебраическом подходе к решению задач распознавания и классификации. *Проблемы кибернетики*. 1978, вып. 33, с. 28 - 57.
3. Рогожин С.В., Рыжов А.П. О нечетко заданных классах функций k-значной логики. *V Всероссийская конференция «Нейрокомпьютеры и их применение»*. Сборник докладов. Москва, 17-19 февраля 1999 года, с.460-463.
4. Рыжов А.П. О системах информационного мониторинга сложных объектов и процессов. *VI Международная конференция по математическому моделированию*. 24 июня – 1 июля 2000 г. Москва, Московский государственный технологический университет (СТАНКИН), с. 33-40.
5. Рыжов А.П. Элементы теории нечетких множеств и измерения нечеткости. М., Диалог-МГУ, 1998, 116 с.
6. Рыжов А.П., Федорова М.С. Генетические алгоритмы в задаче выбора операторов агрегирования информации в системах информационного мониторинга. *V Всероссийская конференция "Нейрокомпьютеры и их применение"*. Сборник докладов. Москва, 17 - 19 февраля 1999 года, с. 267-270.
7. Саати Т. *Анализ иерархических процессов*. М., Радио и связь, 1993 - 315 с.
8. Скурихин А.Н. Генетические алгоритмы. *Новости искусственного интеллекта*. № 4, с. 6 - 46.
9. Messarovich M.D., Macko D., Takahara Y. *Theory of hierarchical multilevel systems*. Academic Press, N.Y.- London 1970 - 344 p.
10. Ryjov, A., Belenki, A., Hooper, R., Pouchkarev, V., Fattah, A. and Zadeh, L.A., *Development of an Intelligent System for Monitoring and Evaluation of Peaceful Nuclear Activities (DISNA)*, IAEA, STR-310, Vienna, 1998, 122 p.