Синтез операторов агрегирования информации по экспертным описаниям

А. А. Лебедев

Задача выбора оператора агрегирования информации – определение функции, характеризующей зависимость некоторой величины от наблюдаемых параметров – возникает при разработке большинства систем сбора и обработки информации. В работе развивается подход к описанию функций k-значной логики на основе нечётких условий. Доказывается NP-полнота задачи выбора функции по нечётким описаниям в общем случае, а также приводятся полиномиальные алгоритмы решения этой задачи для некоторых частных случаев.

Ключевые слова: выбор операторов агрегирования, нечеткие условия.

1. Введение

Задача выбора оператора агрегирования информации — определение функции, характеризующей зависимость некоторой величины от наблюдаемых параметров — возникает при разработке большинства систем сбора и обработки информации. Особую роль эта задача играет для систем информационного мониторинга [9]. В зависимости от ситуации, для решения этой задачи применяются методы математической статистики, data mining [10], нейронные сети и т. д.

В работах [3, 4] был впервые рассмотрен подход на основе нечётких условий: на множестве функций k-значной логики от n переменных на основе экспертных описаний задавалась функция принадлежности, и в качестве оператора агрегирования выбиралась та функция k-значной логики, степень принадлежности которой максимальна. Однако в этих работах было описано всего два типа условий («значение в точке» и «локальное поведение по одной переменной»), и алгоритмы поиска функции были переборными.

В настоящей работе мы предлагаем более общий подход к описанию функций k-значной логики нечёткими условиями. Мы покажем, как в этом подходе можно реализовать предыдущие результаты, докажем NP-трудность задачи выбора оператора по нечетким описаниям, а также приведем полиномиальные алгоритмы решения этой задачи в некоторых частных случаях.

2. Постановка задачи и формулировка результатов

Функции $E_K^n \to E_k$, где $E_l = \{0, 1, \ldots, l-1\}$, будем называть функциями K, k-значной логики. Множество всех таких функций от n фиксированных переменных обозначим $P_{K,k}(n)$.

Нечётким условием на функции K, k-значной логики от n переменных мы называем отображение $\mu : P_{K,k} \to [0; 1]$. Значение $\mu(f)$ называем степенью выполнения условия для функции $f \in P_{K,k}(n)$.

В нашей работе мы предлагаем задавать нечёткие условия с помощью управляющей системы, которую мы назовём *граф нечёткого условия.*

Граф нечёткого условия C — это тройка $\langle G, \rho, T\rangle$, где:
$G = \langle V, E\rangle$ — неориентированный граф без кратных рёбер с множеством вершин $V = \{(\alpha, a), \alpha \in E_K^n, a \in E_k\}$ и некоторым множеством рёбер E;
$\rho : E \to [0; 1)$ — веса рёбер;
$T : [0; 1]^2 \to [0; 1]$ — Т-норма — бинарная операция, удовлетворяющая свойствам коммутативности, ассоциативности, монотонного неубывания и ограниченности: $T(0, x) = 0$, $T(1, x) = x$ (Т-нормы — обобщения операций «И» в нечёткой логике. Наиболее простыми примерами Т-норм являются функции минимума и произведения. Подробнее см., например, [5])

Граф нечёткого условия C реализует нечёткое условие $\mu : P_{K,k} \to [0;1]$, степень выполнения которого для произвольной функции $f(x_1, \ldots, x_n) \in P_{K,k}(n)$ вычисляется следующим образом:

1) из множества вершин V графа G выделяется подмножество $V_f = \{(\alpha, f(\alpha)), \alpha \in E_K^n\}$;
2) в графе G выделяется подграф $G_f = \langle V_f, E_f \rangle$, индуцированный подмножеством V_f;
3) значением $\mu_C(f)$ является значение T-нормы от весов всех рёбер подграфа G_f, то есть $\mu(G) = \underset{e \in E_f}{\mathrm{T}} \mu(e)$. Если E_f пусто, полагаем $\mu_C(f) = 1$.

Класс всех графов нечёткого условия для фиксированных n, K, k и T обозначим $\Phi(K, k, n, T)$.

Степенью выполнимости нечёткого условия μ назовём величину $\mu^{\max} = \max\limits_{f \in P_{K,k}(n)} \mu(f)$.

Задача определения степени выполнимости заключается в том, чтобы для данного нечёткого условия μ_C, заданного графом нечёткого условия C, и данного рационального $0 < \alpha \leqslant 1$ проверить, верно ли, что $\mu_C^{\max} \geqslant \alpha$.

Задача нахождения оптимальной функции заключается в нахождении для данного нечёткого условия μ_C, заданного графом нечёткого условия C, хотя бы одной функции $f \in P_{K,k}(n)$, такой, что $\mu_C(f) = \mu_C^{\max}$.

Заметим, что решение задачи нахождения оптимальной функции позволяет найти решение задачи определения степени выполнимости. Обратное неверно.

Теорема 1. *При $C \in \Phi(K, k, n, T)$ задача определения степени выполнимости нечёткого условия μ_C является NP-трудной для любого фиксированного $K \geqslant 2$, любого фиксированного $k \geqslant 3$, любого фиксированного $0 < \alpha \leqslant 1$ и любой фиксированной T-нормы T.*

Теорема 2. *При справедливости гипотезы $P \neq NP$ не существует приближённого полиномиального алгоритма, решающего задачу* *определения степени выполнимости с результатом, отклоняющимся от истинного значения не более чем в фиксированное число раз.*

Дальнейшие результаты работы посвящены выявлению случаев, разрешимых за полиномиальное время.

Временем работы алгоритма мы считаем общее число элементарных операций, таких как:

- операции над графом:
  - добавление / удаление ребра;
  - добавление / удаление вершины;
  - присвоение вершине метки / чтение метки вершины;
  - присвоение ребру метки / чтение метки ребра;
  - получение пары вершин, инцидентных ребру;
  - получение списка рёбер, инцидентных вершине;
- арифметические бинарные и унарные операции;
- логические бинарные и унарные операции.

Теорема 3. *Если $C \in \Phi(K, 2, n, \min)$, то задача нахождения оптимальной функции разрешима за время $O(K^{2n})$ при $n \to \infty$.*

Замечание. Так как входом алгоритма является граф нечёткого условия, имеющий $2K^n$ вершин, то такая оценка сложности является полиномиальной.

Граф нечёткого условия назовём *параллельным*, если все рёбра в нём проводятся через пары вершин вида $(\alpha, a), (\alpha+\delta, b)$, где $\delta \in E_K^n$ — фиксированный вектор. Класс всех параллельных графов нечёткого условия обозначим $\Pi(K, k, n, T)$.

Теорема 4. *Если $C \in \Pi(K, k, n, T)$, то задача нахождения оптимальной функции разрешима за время $O(K^{n+k+1})$ при $n \to \infty$, $K \to \infty$.*

Граф нечёткого условия назовём *локальным*, если в нём ребро между вершинами (α, a) и (β, b) может быть проведено, только если $\max\limits_i |\alpha_i - \beta_i| \leqslant 1$. Класс всех локальных графов нечёткого условия обозначим $\Lambda(K, k, n, T)$.

Теорема 5. *Если $C \in \Lambda(K, k, n, T)$, то при $n = 1$ задача нахождения оптимальной функции разрешима за время $O(k^2 K)$ при $k \to \infty$, $K \to \infty$, а для любого фиксированного $n \geqslant 2$ задача определения степени выполнимости условия является NP-трудной.*

3. Труднорешаемость задачи выбора оптимальной функции

В этом разделе мы докажем Теоремы 1, 2 и второе утверждение теоремы 5. Сначала переформулируем задачу в терминах графов.

Итак, нечёткое условие на функцию K, k-значной логики $f(x_1, \ldots, x_n)$ (k фиксировано, n может быть произвольно большим) задано графом C с множеством вершин $V = \{(\alpha, a), \alpha \in E_K^n, a \in E_k\}$, некоторым множеством рёбер E, некоторыми весами рёбер $\rho(e) \in [0; 1)$, $e \in E$ и Т-нормой T. Определим функционал на графах с взвешенными рёбрами: $\mu(G) = \underset{e \in E_G}{\mathrm{T}} \rho(e)$, где T — выбранная Т-норма. Тогда задача нахождения оптимальной функции эквивалентна задаче выбора в этом графе подмножества вершин V' со следующими свойствами:

1) для любого $\alpha \in E_K^n$ существует единственное $a \in E_k$ такое, что $(\alpha, a) \in V'$;
2) из всех графов, индуцированных подмножествами вершин, удовлетворяющих свойству 1, граф G', индуцированный подмножеством V', максимизирует функционал μ.

Эту задачу мы назовём *задачей выбора оптимального подмножества.*

В случае, когда веса всех рёбер в графе нечёткого условия имеют только значение 0, задачу выбора оптимального подмножества можно сформулировать следующим образом:

для данного простого графа G, множество вершин которого имеет вид $V = \{(\alpha, a), \alpha \in E_K^n, a \in E_k\}$, а множество рёбер произвольно, проверить, существует ли такое подмножество вершин V', содержащее по одному элементу (α, a) для каждого $\alpha \in E_K^n$, что индуцированный им подграф не имеет ни одного ребра. Эту задачу мы назовём *задачей выбора оптимального подмножества для 0,1-графов.*

Лемма 1. *Для любого фиксированного $K \geqslant 2$ и любого фиксированного $k \geqslant 3$ задача выбора оптимального подмножества для 0,1-графов является NP-трудной.*

Доказательство. Для доказательства NP-трудности сведем к нашей задаче классическую NP-полную задачу k-РАСКРАШИВАЕМОСТЬ ГРАФА [8] — для данного графа $\Gamma = \langle B, P \rangle$ и натурального k проверить, существует ли у графа правильная вершинная раскраска в k цветов. Эта задача эквивалентна задаче о выполнимости следующей «конъюнктивной» формы с k-значными аргументами:

$$\bigwedge_{\alpha\beta \in E} (x_\alpha \neq x_\beta)$$

(значение переменной x_α соответствует цвету вершины α, множитель $(x_\alpha \neq x_\beta)$ соответствует ребру $\alpha\beta$). С помощью тождества

$$(x_\alpha \neq x_\beta) \equiv \bigwedge_{i \in E_k} ((x_\alpha \neq i) \vee (x_\beta \neq i))$$

сведем задачу о выполнимости этой формы к задаче о выполнимости формы

$$\bigwedge_{\alpha\beta \in E} ((x_\alpha \neq a_\alpha) \vee (x_\beta \neq b_\beta))$$

которая эквивалента нашей задаче (значение переменной x_α соответствует выбору вершины (α, x_α), а множитель $((x_\alpha \neq a_\alpha) \vee (x_\beta \neq b_\beta))$ соответствует ребру между вершинами (α, a) и (β, b). Функция, реализуемая формой, принимает значение 1 только на тех наборах значений переменных $c_\alpha, \alpha \in E_K^n$, для которых соответствующий подграф, индуцированный подмножеством вершин $\{(\alpha, c_\alpha), \alpha \in E_K^n\}$ не имеет ни одного ребра). Лемма доказана.

Непосредственными следствиями этой леммы являются теоремы 1 и 2.

Доказательство теоремы 1. Утверждение теоремы следует из того, что задача выбора оптимального подмножества для 0,1-графов является частным случаем задачей выбора оптимального подмножества, которая эквивалентна задаче определения степени выполнимости.

Доказательство теоремы 2. Утверждение теоремы следует из того, что, согласно лемме 1, при выполнении гипотезы любой полиномиальный алгоритм ошибочно присвоит степень выполнимости 0 некоторой последовательности нечётких условий, имеющих степень выполнимости 1, или наоборот, тем самым ошибившись в бесконечное число раз.

Теперь докажем вторую часть теоремы 5. В основе доказательства лежит следующая лемма.

Лемма 2. *Для любого фиксированного $k \geqslant 3$ задачей выбора оптимального подмножества для 0,1-графов с множеством вершин $\{(x, y, a),\ \ x, y \in E_K,\, a \in E_k\}$ и множеством рёбер E, где рёбра могут проводиться только между такими вершинами (x_1, y_1, a) и (x_2, y_2, a), для которых $|x_1 - x_2| \leqslant 1$ и $|y_1 - y_2| \leqslant 1$, является NP-трудной.*

Доказательство. Для доказательства NP-трудности сведём к этой задаче задачу k-РАСКРАШИВАЕМОСТЬ ГРАФА — для данного графа $\Gamma = \langle B, P\rangle$ и натурального k проверить, существует ли у графа правильная вершинная раскраска в k цветов. Данному графу сопоставим граф $G = \langle V, E\rangle$, удовлетворяющий условиям леммы, следующим образом:

1) Пусть $B = \{v_1, \ldots, v_l\}$. Тогда положим $K = l^2$ (соответственно, множество вершин графа G есть $V = \{(x, y, a), x, y \in E_K, a \in E_k\}$). Каждой вершине v_i сопоставим множество вершин $V_i = \{(x_i^j, y_i^j, a), j = 1, \ldots, K; a \in E_k\}$, где $x_i^j = j - 1$, $y_i^j = 2(i-1) + j \bmod 2$.
Через каждую пару вершин вида $(x_i^j, y_i^j, a)(x_i^{j+1}, y_i^{j+1}, b), a \neq b$ проходит ребро.

2) Каждому ребру $v_i v_m, i < m$ графа Г сопоставим множество вершин
$V_{i,m} = \{(x^j_{i,m}, y^j_{i,m}, a), j = 1, \ldots, 2(m-i); a \in E_k\}$, где
$y^j_{i,m} = 2(i-1) + j, j = 1, \ldots, 2(m-i)$,
$x^j_{i,m} = 2(i \cdot l + m) + j \bmod 2, j = 2, \ldots, 2(m-i) - 1$, $x^1_{i,m} = x^2_{i,m}, x^{2(m-i)}_{i,m} = x^{2(m-i)-1}_{i,m}$.
Через каждую пару вершин вида $(x^j_{i,m}, y^j_{i,m}, a)(x^{j+1}_{i,m}, y^{j+1}_{i,m}, b)$, $a \neq b$, $j > 1$ и через все пары вершин вида $(x^1_{i,m}, y^1_{i,m}, a)(x^2_{i,m}, y^2_{i,m}, a)$ проходят ребра.

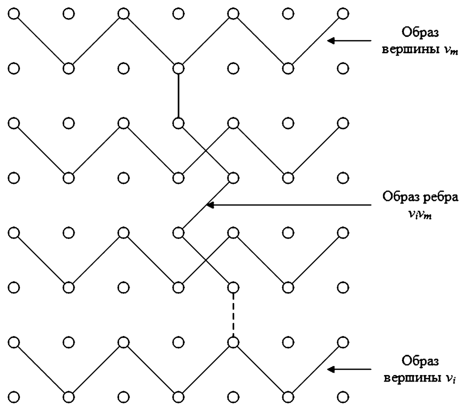

Рис. 1. Реализация графа Г.

Фрагмент такого графа изображён на рис. 1. Кружкам соответствуют множества вершин $\{(x, y, a), a \in E_k\}$ для некоторых x и y, сплошным линиям соответствуют семейства рёбер между парами вершин $(x_1, y_1, a)(x_1, y_1, b), a \neq b$, пунктирной линии — семейства рёбер между парами вершин $(x_1, y_1, a)(x_2, y_2, a)$. Таким образом, горизонтальные ломанные соответствуют образам вершин, вертикальная ломанная — образу ребра.

Заметим, что построенный граф удовлетворяет условиям леммы, а также имеет размер, полиномиально зависящий от размера графа Г. Также заметим, что для множеств V_i и $V_{i,j}$ выполняются свойства: $V_i \cap V_j = \varnothing$, $i \neq j$, $V_{i,j} \cap V_{l,m} = \varnothing$, $(i,j) \neq (l,m)$, $V_i \cap V_{l,m} = \varnothing$, $i \neq l$, $i \neq m)$, более того, все подмножества указанных видов попарно не связаны путями.

Решение задачи определения степени выполнимости для такого графа с ответом «Да» будет означать существование функции $f(x,y) : E_K^2 \to E_k$, график которой $V_f = \{(x,y,f(x,y)),(x,y) \in E_K^2\}$ индуцирует в графе G подграф G', не имеющий ни одного ребра. Но рёбра в графе G были проведены таким образом, что в подграфе G' должны выполняться следующие свойства:

1) для каждого $v_i \in V$ все элементы множества $V_f \cap V_i$ имеют одинаковую третью компоненту (обозначим её a_i);
2) для каждого ребра $v_i v_m, i < m$, все элементы множества $V_f \cap V_{i,m}$, кроме единственного элемента, составляющего множество $V_f \cap V_{i,m} \cap V_i$, имеют одинаковую третью компоненту, которая совпадает с a_m, но отличается от a_i.

Таким образом, на основе функции $f(x,y)$ можно получить правильную вершинную раскраску графа Γ — вершине v_i сопоставляем цвет a_i. И обратно, если у графа Γ существует правильная вершинная раскраска, то, пользуясь ей, можно решить задачу о выборе оптимальной функции для графа G — выбирая из множеств V_i и $V_{i,j}$ вершины с третьей компонентой a_i, а в остальных случаях произвольно. Отсюда следует, что задача k-РАСКРАШИВАЕМОСТЬ ГРАФА сводится к нашей задаче, что доказывает Лемму 2.

Лемма 3. *Если $C \in \Lambda(K,k,n,T)$, то для любого фиксированного $n \geqslant 2$ задача определения степени выполнимости нечёткого условия μ_C NP-полна.*

Доказательство. Для доказательства этой леммы нужно лишь заметить, что:

1) для $n = 2$ задача выбора оптимального подмножества для 0,1-графов является частным случаем задачи выбора оптимального подмножества, которая эквивалентна задаче определения степени выполнимости;

2) задача для $C \in \Lambda(K,k,2,T)$ тривиальным образом полиномиально сводится к задаче для $C \in \Lambda(K,k,n,T)$ при любом фиксированном $n \geqslant 3$.

4. Примеры условий

В этом разделе мы приведем некоторые наиболее распространенные типы нечетких условий, в том числе и описанные в работе [4], и способ их реализации в нашем подходе.

4.1. Условие на значение в точке

Это условие определяет желаемое значение функции в некоторой точке вне зависимости от принимаемых ей значений в других точках. Примерами условий такого типа могут быть: «при максимальных значениях всех аргументов значение функции максимально», «значение функции не превосходит значения третьего аргумента» (это условие представляет собой конъюнкцию нескольких точечных условий).

Условие задаётся точкой α и функцией принадлежности $\mu : E_k \to [0;1]$. Для произвольной функции k-значной логики $f(x_1,\ldots,x_n)$ степень выполнения такого условия равна $\mu(f(\alpha))$. Условие может быть строгим — функция μ принимает значение 1 в единственной точке и равна нулю в остальных, или нестрогим (μ принимает положительные значения в нескольких точках).

Например, условие «на нулевом наборе функция должна принимать нулевое значение» будет задаваться набором $\alpha = (0,\ldots,0)$ и функцией $\mu(x)$, принимающей значение 1 при $x = 0$ и значение 0 в остальных случаях. Условие «на нулевом наборе функция должна принимать малое значение» может задаваться набором $\alpha = (0,\ldots,0)$ и функцией $\mu(x)$ следующего вида:

x	0	1	2	3	$\ldots$	$k-1$
$\mu(x)$	1	0.8	0.4	0	$\ldots$	0

В нашей графовой модели условие задается петлями на вершинах $\{(\alpha, a), a \in E_k\}$. с весами $\mu(a)$ (если $\mu(a) = 1$, петля не добавляется).

4.2. Локальное условие по одной переменной

Это условие определяет желаемое поведение функции на паре наборов, отличающихся на единицу по одной выбранной переменной. Условия такого типа могут задаваться высказываниями: «функция не убывает по первой переменной», «функция слабо изменяется при изменении второго аргумента» и т. п.

Условие описывается матрицей $\mu_{ij}, i, j, \in E_k$. Элемент матрицы μ_{ij} определяет степень, с которой пара значений функции $f(a)$ и $f(a+1)$ удовлетворяет условию, если $f(a) = i$ и $f(a+1) = j$. Например, условие «функция $f(x)$ строго монотонна» задается следующей матрицей:

$f(i+1)$ $f(i)$	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

а условие «При увеличении x функция f слабо возрастает» может быть задано следующей матрицей ($k = 6$):

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

Степень выполнения условия для функции будет вычисляться как некоторая Т-норма от степеней выполнимости условия на всех наборах, отличающихся на единицу по соответствующей переменной. Например, если $T(x, y) = xy$, то для функции от одной переменной $f(x)$

x	0	1	2	3	...	$k-1$
$f(x)$	1	0.8	0.4	0	...	0

степень выполнимости условия «слабого возрастания», заданного выше, будет равна: $\mu(f) = \mu_{f(0)f(1)} \cdot \mu_{f(1)f(2)} \cdot \mu_{f(2)f(3)} \cdot \mu_{f(3)f(4)} \cdot \mu_{f(4)f(5)} = \mu_{01} \cdot \mu_{12} \cdot \mu_{22} \cdot \mu_{24} \cdot \mu_{45} = 1 \cdot 1 \cdot 0.2 \cdot 1 \cdot 1 = 0.2$.

В графовой модели условие задается следующим образом: для каждой пары наборов $(\alpha, \beta) : \alpha, \beta \in E_k^n, \alpha_i = \beta_i, i \neq j, \beta_j = \alpha_j + 1$, где j — номер переменной, по которой строится условие, и каждой пары $(a, b) : a, b \in E_k$ проводится ребро, соединяющее вершины (α, a) и (β, b) и имеющее вес $\mu_{a,b}$.

Получаемый граф нечёткого условия является локальным и параллельным.

4.3. Локальное условие по нескольким переменным

Это условие определяет желаемое поведение функции на наборах, отличающихся на единицу по нескольким выбранным переменным. Условия такого типа могут задаваться высказываниями: «при совместном возрастании второй и четвертой переменных функция слабо возрастает» и т. п.

Условие задается аналогично предыдущему с той лишь разницей, что ребра проводятся между вершинами с несколькими отличающимися на единицу координатами.

Получаемый граф нечёткого условия является локальным и параллельным.

4.4. Устойчивость

Условие A-устойчивости и его важность при проектировании иерархических систем рассмотрены в работе [1]. Оно формулируется

следующим образом:
функция K, k-значной логики $f(x_1, \dots, x_n)$ называется A-устойчивой $(1 \leqslant A \leqslant \min(K, k) - 2)$, если выполняется условие:

$$|f(\alpha_1, \dots, \alpha_n) - f(\beta_1, \dots, \beta_n)| \leqslant A \forall \alpha, \beta \in E_K^n : \max_{i=1,\dots,n} |\alpha_i - \beta_i| \leqslant A.$$

В графовой модели условие задаётся следующим образом: для каждой пары наборов $(\alpha, \beta) : \alpha, \beta \in E_K^n : \max\limits_{i=1,\dots,n} |\alpha_i - \beta_i| \leqslant A$ и каждой пары $(a, b) : a, b \in E_k : |a - b| \leqslant A$ проводятся ребро, соединяющее вершины (α, a) и (β, b) и имеющее вес 0 (в случае, когда выполнение условия не обязательно, а «желательно», можно использовать веса из интервала $(0; 1)$).

Заметим, что граф нечёткого условия для условия 1-устойчивости является локальным. Для $A \geqslant 2$ свойство локальности не выполняется.

4.5. Логические операции

Мы можем комбинировать условия с помощью *логических операций* «И» и «ИЛИ».

Итак, пусть задано множество вершин $\{(\alpha, a), \alpha \in E_k^n, a \in E_k\}$ и два множества взвешенных рёбер, соответствующих нечётким условиям.

Конъюнкции двух условий будет соответствовать объединение множеств ребер. В случае если ребро входило в оба исходных множества, его весом в конъюнкции будет некоторая Т-норма исходных весов.

Дизъюнкции двух условий будет соответствовать пересечение множеств ребер. Веса ребер в дизъюнкции будут вычисляться как некоторая Т-конорма весов соответствующих рёбер условий-аргументов. (Т-конорма — бинарная операция, удовлетворяющая свойствам коммутативности, ассоциативности, монотонного неубывания и ограниченности: $T(0, x) = x$, $T(1, x) = 1$. Т-конормы — обобщения операций «ИЛИ» в нечёткой логике. Наиболее простыми примерами Т-конорм являются функции максимума и алгебраической суммы: $T(x, y) = x + y - xy$).

Заметим, что логические операции сохраняют свойство локальности графов нечётких условий.

5. Полиномиально разрешимые случаи

5.1. Одномерный локальный случай

В этом разделе мы докажем первую часть теоремы 5.

Лемма 4. *Если $C \in \Lambda(K, k, 1, T)$, то задача нахождения оптимальной функции разрешима за время $O(k^2 \cdot K)$ при $K \to \infty$, $k \to \infty$.*

Замечание. При отсутствии условия локальности доказательство теоремы об NP-полноте общего случая задачи определения степени выполнимости нечёткого условия без изменений переносится и на одномерный случай, так как при доказательстве нигде не использовалась топология вершин графа нечёткого условия.

Доказательство леммы. Рассмотрим задачу выбора функции одной переменной $f : E_K \to E_k$ по графу нечёткого условия $C = \langle G, \rho, T\rangle$. Множество вершин графа G будет иметь вид $\{(a, b), a \in E_K, b \in E_k\}$, а рёбра проводятся только между парами вершин вида $(a, b), (a + 1, c)$.

Алгоритм, решающий задачу, будет иметь следующий вид:

1) По графу нечёткого условия построим ориентированный граф G' следующего вида: его вершинами будут все вершины графа G, а также две дополнительные вершины s и f. Множество рёбер нового графа состоит из:
   а) рёбер, соединяющих вершину s с вершинами вида $(0, a)$ и вершины $(K - 1, b)$ с вершиной f. Эти рёбра имеют вес 1;
   б) рёбер графа G, не являющихся петлями. В графе G' они ориентированны от (a, b) к $(a + 1, c)$;
   в) рёбер, соединяющих вершины вида (a, b) и $(a + 1, c)$, не соединённые рёбрами предыдущего типа. Таким рёбрам присваивается вес 0. Рёбра ориентированны от (a, b) к $(a+1, c)$;

г) Если у вершины (a, b) в графе G была петля с весом w, то вес v каждого ребра, входящего в вершину (a, b), заменяется на $T(w, v)$.

Вес ребра от вершины α к вершине β, обозначим $w(\alpha, \beta)$, а вес пути от вершины s к вершине α обозначим $\rho(\alpha)$.

2) Алгоритм вычисляет веса $\rho(\alpha)$ для всех вершин следующим образом:
$\rho((0, a)) = 1, a \in E_k$;
Цикл по i от 0 до $K - 2$

    Цикл по $a \in E_k$

        $\rho((i+1, a)) = \max\limits_{b \in E_k} T(\rho((i, b), w((i, a), (i+1, b)))$

        вершине $(i + 1, a)$ добавляется метка со значением (i, b) — вершины, на которой был достигнут максимум.

    Конец цикла

Конец цикла

3) $V' = \varnothing$, $x = f$;
Цикл (пока $m(x) \neq s$)

    Добавляем $m(x)$ в множество V'

    $x = m(x)$

Конец цикла

4) Множество V' — искомое подмножество вершин, $\rho(f)$ — значение степени выполнимости нечёткого условия

Оценим время работы алгоритма:
Время работы шага 1(a) алгоритма равно $O(K)$, шагов 1(b) и 1(c) — $O(k^2 \cdot K)$, шага 1(d) — $O(k \cdot K)$.
На шаге 2 внешний цикл выполняется $K - 1$ раз, внутренний — k раз, одна итерация выполняется за $O(k)$ операций. Итого, требуется $O(k^2 \cdot K)$ операций.
Шаг 3 требует $O(K)$ операций.
В результате получаем, что время работы алгоритма составляет $O(k^2 \cdot K)$ операций. Лемма 4 доказана.

Теорема 5 следует из Лемм 4 и 5.

5.2. Булев случай

В этом разделе мы докажем теорему 3.

В случае $T = \min$ задачу можно свести к задаче выбора оптимального подмножества для 0,1-графов следующим способом: всем рёбрам графа G, вес которых строго меньше α, присваивается вес 0, все остальные рёбра удаляются. Для полученного графа G' решается задаче выбора оптимального подмножества для 0,1-графов.

Как было показано в разделе 3, эта задача эквивалентна задаче о выполнимости формы вида $\bigwedge\limits_{\alpha\beta\in E}((x_\alpha \neq a_\alpha) \vee (x_\beta \neq b_\beta))$. Эта форма в булевом случае приобретает вид $\bigwedge\limits_{\alpha\beta\in E}(x_\alpha^{\overline{a_\alpha}} \vee x_\beta^{\overline{b_\beta}})$. Итак, задача становится эквивалентна задаче 2-ВЫПОЛНИМОСТЬ, которая, как известно ([7]), разрешима за полиномиальное время.

Приведем алгоритм, работающий за линейное (от суммы числа вершин и рёбер графа) время, построенный на основе алгоритма, приведённого в работе [6]:

Нам задан граф нечёткого условия $C \in \Phi(K, 2, n, \min)$ и число $\alpha \in [0; 1]$

1) Удаляем из графа G графа нечёткого условия C все рёбра $e \in E$, для которых $\rho(e) \geqslant \alpha$.
2) По изменённому графу G построим ориентированный граф G' следующего вида: его вершинами будут все вершины графа G. Каждому ребру $(\alpha, a)(\beta, b)$ графа G в графе G' соответствует пара рёбер $(\alpha, a) \to (\beta, \overline{b})$ и $(\beta, b) \to (\alpha, \overline{a})$.
3) С помощью алгоритма поиска в глубину разбиваем граф G' на компоненты сильной связности.
4) Если ни для какого $\alpha \in E_K^n$ пара вершин (α, a) и $(\alpha, \overline{a})$ не окажется в одной и той же компоненте, то задача имеет ответ «Да». В противном случае — ответ «Нет».

Каждый шаг этого алгоритма требует $O(|V| + |E|)$ шагов. Так как $|V| = 2K^n$ а $|V| \leqslant 2K^{2n}$, то общее время работы алгоритма не превосходит $O(K^{2n})$. Теорема 3 доказана.

5.3. Случай параллельных графов

Для доказательства теоремы 4 приведём алгоритм, решающий задачу поиска оптимальной функции.

1) Разбиваем множество вершин V графа G на подмножества $V_i, i = 1, \ldots, m$ следующим образом:
   - Вершины вида (α, a) и (α, b) принадлежат одному подмножеству;
   - Вершины, соединённые ребром, принадлежат одному подмножеству.

   Таким образом, каждое подмножество V_i имеет вид $V_i = \widetilde{V}_i \times E_k$, где $\widetilde{V}_i \subseteq E_K^n$
2) Цикл по всем V_i

   $\mu^{\max} = 0$, $V_i^{\max} = \varnothing$

   Цикл по всем возможным отображениям $f_i : \widetilde{V}_i \to E_k$

   Вычисляем значение $\mu = \mathop{\mathrm{T}}\limits_{e \in E_i'} \rho(e)$, где E_i' — множество рёбер подграфа, индуцированного множеством вершин $V_i' = \{(\alpha, f_i(\alpha)), \alpha \in \widetilde{V}_i\}$

   Если $\mu > \mu^{\max}$, то $V_i^{\max} = V_i'$

   Конец цикла

   $G_i^{\max}$ — подграф, индуцированный множеством вершин $V_i^{\max}$.

   Конец цикла
3) Подграф $G' = \bigcup\limits_i G_i^{\max}$ будет искомым.

Время работы алгоритма на шагах 1 и 3 линейно зависит от размера (суммы числа вершин и числа рёбер) графа G, а на шаге 2 не превосходит произведения числа подмножеств V_i на максимальное время работы алгоритма на одном подмножестве. В случае параллельных графов каждое подмножество имеет вид $V_i = \widetilde{V}_i \times E_k$, где $\widetilde{V}_i = \{\alpha + j\delta, j \in E_K\}$. Следовательно, число итераций внешнего цикла (число подмножеств V_i) равно K^{n-1}, число итераций внутреннего цикла (число отображений f_i) равно K^k, а время выполнения одной итерации равно $O(K^2)$. Таким образом, общее время работы алгоритма имеет порядок $O(K^{n+k})$ при $k \to \infty$. Теорема 4 доказана.

6. Пример

Для лучшего понимания функционирования графа нечёткого условия, приведём пример, демонстрирующий задание условий на функцию графом нечёткого условия и нахождение оптимальной функции по графу нечёткого условия.

Будем задавать функцию от двух переменных $f(x_1, x_2)$ в трёхзначной логике ($K = k = 3$). Наложим на функцию следующие условия:

- функция 1-устойчива (строго);
- при совместном возрастании переменных функция строго возрастает;
- функция слабо возрастает по переменной x_1;

В качестве Т-нормы в этом примере мы будем использовать операцию взятия минимума.

6.1. Построение графов элементарных условий

Для всех графов условий множеством вершин будет множество $E_K^2 \times E_k = E_3^3$.

Условие 1-устойчивости запрещает функции принимать значения, отличающиеся больше, чем на единицу (в случае $k = 3$ это могут быть только значения 0 и 2), на наборах, находящихся в пределах одного подкуба с длиной стороны 1. В нашем случае таких подкубов четыре: их диагонали $(0;0) - (1;1)$, $(1;0) - (2;1)$, $(0;1) - (1;2)$ и $(1;1) - (2;2)$. В каждом из таких подкубов по 6 пар наборов. Соответственно, множество рёбер графа условия будет состоять из 48 рёбер вида $(\alpha, 0) - (\beta, 2)$, где α и β — наборы, принадлежащие одному их подкубов. На рис. 2 показан фрагмент графа этого условия. Так как условие строгое, все рёбра будут иметь вес 0.

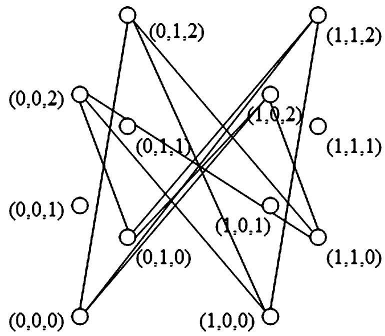

Рис. 2. Фрагмент графа условия 1-устойчивости.

Условие строгого возрастания по обеим переменным задаётся следующей матрицей:

$f(i+1)$ $f(i)$	0	1	2
0	0	1	1
1	0	0	1
2	0	0	0

Множество рёбер графа условия, соответствующего этой матрице, состоит их рёбер вида: $(a, b, 1) - (a+1, b+1, 0)$, $(a, b, 2) - (a+1, b+1, 0)$, $(a, b, 2) - (a+1, b+1, 1)$ и $(a, b, c) - (a+1, b+1, c)$ где $a, b \in \{0, 1\}, c \in E_3$ — всего 48 рёбер. В силу строгости условия все они имеют вес 0. На рис. 3 показан фрагмент графа этого условия.

Условие слабого возрастания по одной переменной задаётся, например, следующей матрицей:

$f(i+1)$ $f(i)$	0	1	2
0	0.4	1	0.8
1	0	0.4	1
2	0	0	0.4

Множество рёбер графа условия по переменной x_1, соответствующего этой матрице, состоит их рёбер вида:

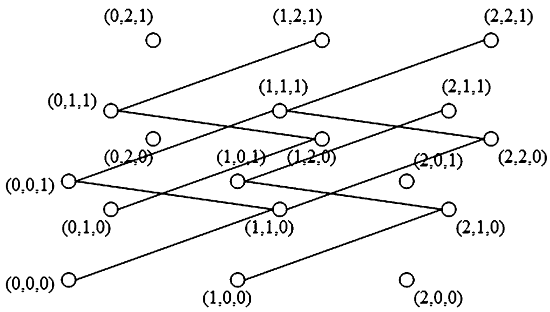

Рис. 3. Фрагмент графа условия строгого возрастания при совместном возрастании обеих переменных.

- $(a, b, 1) - (a+1, b, 0)$, $(a, b, 2) - (a+1, b, 0)$,$(a, b, 2) - (a+1, b+1, 1)$, имеющих вес 0;
- $(a, b, c) - (a+1, b, c)$, имеющих вес 0,4;
- $(a, b, 0) - (a+1, b, 2)$, имеющих вес 0,8.
  $(a \in \{0, 1\}; b, c \in E_3)$

На рис. 4 показан фрагмент графа этого условия. Рёбра различного веса изображены различными стилями.

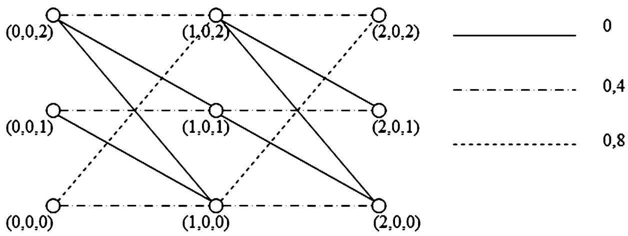

Рис. 4. Фрагмент графа условия слабого возрастания по переменной x_1.

6.2. Построение графа совокупного условия

По этим трём графам строим граф совокупного условия на функцию: в соответствии с правилом конъюнкции условий, его рёбрами будут все рёбра, входящие хотя бы в один граф. Если ребро входило в два или три графа, в граф совокупного условия оно войдёт с минимальным весом (в качестве T-нормы мы рассматриваем минимум). Например, рёбра вида $(a, b, 0) - (a + 1, b, 2)$ входят в граф условия 1-устойчивости с весом 0 и в граф условия слабого возрастания по переменной x_1 с весом 0,8. Следовательно, в граф совокупного условия такие рёбра войдут с весом 0.

6.3. Восстановление функции по графу условия

Как было показано выше, в общем случае задача восстановления функции по графу условия NP-полна, поэтому существование для неё быстрого алгоритма маловероятно. Приводимые далее рассуждения будут относиться к конкретному случаю, хотя, возможно, они будут формализованы в будущем.

Итак, нам нужно выбрать в нашем графе подмножество вершин с условиями, описанными в разделе 2. Будем постепенно удалять из графа вершины, которые заведомо не войдут в это подмножество.

Сначала опишем все функции, удовлетворяющие условию с положительной степенью. При этом мы можем игнорировать рёбра, имеющие положительный вес.

Рассмотрим рёбра, соответствующие условию строгого возрастания по обеим переменным. Единственный способ выбрать вершины вида $(0, 0, a)$, $(1, 1, b)$, $(2, 2, c)$ так, чтобы они не были соединены этими рёбрами, — это положить $a = 0$, $b = 1$, $c = 2$. Тем самым мы однозначно определили значения строимых функций на наборах $(0, 0)$, $(1, 1)$ и $(2, 2)$ и можем удалить вершины $(0, 0, 1)$, $(0, 0, 1)$, $(1, 1, 0)$, $(1, 1, 2)$, $(2, 2, 0)$, $(2, 2, 1)$.

Вершина $(0, 0, 0)$, которая обязательно войдёт в выбираемый подграф, соединена рёбрами с вершинами $(1, 0, 2)$ и $(0, 1, 2)$ (эти рёбра входили в граф условия 1-устойчивости). Следовательно, мы можем удалить эти вершины. Аналогичным образом удаляются вершины $(1, 2, 0)$ и $(2, 1, 0)$ (соединённые с вершиной $(2, 2, 2)$).

Возможные значения функции на данном этапе можно описать следующей таблицей:

x_1 / x_2	0	1	2
0	0	0;1	0;1;2
1	0;1	1	1;2
2	0;1;2	1;2	2

Теперь рассмотрим рёбра, ограничивающие значения функции на наборах $(1,0)$, $(2,0)$ и $(2,1)$. Ребро из графа условия строгого возрастания по двум переменным запрещает сочетание $(1,0,1)-(2,1,1)$, ребро графа условия слабого возрастания по переменной x_1 запрещает сочетание $(1,0,1)-(2,0,0)$, а рёбра графа условия 1-устойчивости запрещают сочетания $(1,0,0)-(2,1,2)$, $(1,0,0)-(2,0,2)$ и $(2,0,0)-(2,1,2)$. В результате имеется всего четыре допустимых сочетания: $(1,0,0)-(2,0,0/1)-(2,1,1)$ и $(1,0,1)-(2,0,1/2)-(2,1,2)$. Аналогично для тройки наборов $(0,1)$, $(0,2)$ и $(2,1)$ получим возможные сочетания $(0,1,0)-(0,2,0/1)-(1,2,1)$ и $(0,1,1)-(0,2,1/2)-(1,2,2)$. Всего получается 16 функций, каждая из которых удовлетворяет совокупному условию с ненулевой степенью.

Теперь обратим внимание на рёбра, имеющие положительный вес и выберем из «допустимых» функций ту, которая будет иметь наибольшую степень выполнимости условия. Ей окажется функция $f(x_1,x_2) = x_1$ — единственная, удовлетворяющая условию со степенью 1. К этому же результату мы могли бы прийти, сразу ограничив свой поиск только функциями, удовлетворяющими условию со степенью 1 — для этого нужно было присвоить всем рёбрам в графе совокупного условия вес 0.

7. Заключение

В работе был предложен графовый подход к описанию функций k-значной логики нечёткими условиями. Была доказана NP-полнота и неразрешимость приближённым полиномиальным алгоритмом задачи выбора оператора по графу нечёткого условия, а также приведены полиномиальные алгоритмы решения этой задачи в некоторых частных случаях.

Заметим, что предлагаемый подход без каких-либо изменений переносится на более общий случай неоднородных функций k-значной логики — дискретные функции, у которых различные аргументы могут принимать значения из различных множеств, подробнее см., например, [2].

Дальнейшие исследования будут направлены на увеличение числа типов описываемых методом условий, расширение множества полиномиально разрешимых случаев для задачи выбора оператора, использование вероятностных алгоритмов, эвристических методов сокращения перебора (например, метод ветвей и границ) и другие вопросы.

Автор выражает признательность А. П. Рыжову за постановку задачи, а также В. Б. Кудрявцеву и Э. Э. Гасанову за обсуждение работы и ценные рекомендации.

Список литературы

[1] Лебедев А. А. О задачах оптимального распределения ресурсов и проверки устойчивости для схем функциональных элементов в k-значной логике // Интеллектуальные системы. 2011. Т. 15, вып. 1–4. С.

[2] Кудрявцев В. Б. Функциональные системы. М.: Изд-во МГУ, 1982.

[3] Рогожин С. В., Рыжов А. П. О нечетко заданных классах функций k-значной логики // V Всероссийская конференция «Нейрокомпьютеры и их применение». Москва, 17–19 февраля 1999 г. Сб. докладов. С. 460–463.

[4] Рыжов А. П. Разработка методов агрегирования информации в нечетких иерархических системах с использованием методов мягких вычислений // Интеллектуальные системы. 2001. Т. 6, вып. 1–4. С. 341–364.

[5] Рыжов А. П. Элементы теории нечетких множеств и измерения нечеткости. М.: Диалог-МГУ, 1998.

[6] Aspvall B., Plass M. F., Tarjan R. E. A linear-time algorithm for testing the truth of certain quantified boolean formulas // Information Processing Letters. 8 (3). 1979. P. 121–123.

[7] Even S., Itai A., Shamir A. On the complexity of timetable and multicommodity flow problems // SIAM J: Comput. 5. 1976. P. 691–703.

[8] Karp R. M. Reducibility among combinatorial problems // Complexity of computer computations. NY: Plenum Press, 1974. P. 85–103.

[9] Ryjov A. P. Basic principles and foundations of information monitoring systems // Monitoring, Security, and Rescue Techniques in Multi-agent Systems. Springer, 2005. P. 147–160.

[10] Usama M. Fayyad (Ed.) Advances in Knowledge Discovery and Data Mining. MIT Press, 1996.