

РОЗДІЛ «ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ»

УДК 004.057.3

ЖУЛЬКОВСКАЯ И.И., к.т.н., доцент
ЖУЛЬКОВСКИЙ О.А., к.т.н., доцент
НИКОЛАЕНКО Ю.В., студент

Днепродзержинский государственный технический университет

ВЫЧИСЛЕНИЕ ГРАНИЧНЫХ ЗНАЧЕНИЙ ДЕЙСТВИТЕЛЬНЫХ ЧИСЛОВЫХ ДАННЫХ В IEEE-СТАНДАРТЕ

Введение. Все большее значение в современном обществе приобретают информационные системы и технологии. На них сейчас перекалдывается колоссальное число функций по представлению, хранению и обработке информации, что в значительной степени повышает актуальность контроля достоверности и точности вычислительных процессов.

Известно [1], что многие техногенные катастрофы были обусловлены различного рода вычислительными ошибками, и особое значение этот фактор приобрел в известных [2, 3] катастрофах в военной и космической областях.

Нарастание масштабов вычислений и задействованных при этом ресурсов ЭВМ, а также постоянно растущие требования к точности результатов теоретических исследований (математического моделирования) и проектирования сложных технологических систем и процессов повышает значимость проблемы машинного представления и хранения числовой информации в современных ЭВМ (персональных компьютерах).

Численное решение большинства практических задач сопряжено, главным образом, с выполнением операций над действительными (вещественными) числами или так называемыми числами с плавающей запятой (точкой).

Ассоциацией *IEEE (Institute of Electrical and Electronics Engineers)* был разработан и внедрен единый стандарт для представления в двоичном коде чисел с плавающей запятой. Полное название этого стандарта – *IEEE Standard for Binary Floating-Point Arithmetic, ANSI/IEEE Std 754-1985 (ANSI – American National Standards Institute)* [4-6]. Этот, наиболее распространенный и используемый стандарт, используется многими современными аппаратными и программными средствами ЭВМ. В частности, на его базе основывается архитектура математических сопроцессоров фирмы *Intel*.

На данный момент ассоциация *IEEE* реорганизована из международной общественной инженерной организации в коммерческую структуру, в связи с чем авторские стандарты *IEEE* стали коммерческим продуктом и не находятся в свободном распространении и обращении.

Ограничение доступности указанных стандартов не могло не сказаться на компетентности постановки вычислительных задач и достоверности результатов численного экспериментирования.

Известные и доступные публикации на тему представления и хранения действительных чисел в памяти компьютера [7-10] рассматривают лишь вопросы погрешностей результатов при вычислении функций, пути минимизации ошибок, связанных с округлением, создания тестов для реализаций математических функций и т.п.

Постановка задачи. Ранее [11] авторами данной работы исследован и описан алгоритм современного подхода к формированию машинного представления и хранения числовой информации в формате с плавающей запятой.

Целями же настоящего исследования являются рассмотрение особенностей представления, а также вычисление граничных (максимальных и минимальных) значений числовых данных в стандарте *IEEE 754*.

Результаты работы. Для того, чтобы использовать действительные числа в машинных вычислениях, необходимо понимать, каким образом произвольное число хранится в ограниченном количестве ячеек памяти.

Двоичное число с плавающей запятой (*binary floating point number*) – битовая строка, характеризующаяся тремя составляющими: знаком (*sign*), знаковым порядком или экспонентой (*signed exponent*) и мантиссой (*significand*). Знаковый бит равен 0 для положительных чисел и 1 – для отрицательных.

Так, число с плавающей запятой можно представить в следующем виде:

$$A = \pm a_0.a_1a_2a_3\dots a_{n-1} \times S^e, \tag{1}$$

где S – основа системы счисления;

a_i – цифры ($0 \leq a_i < S$);

e – порядок или экспонента (не путать с числом e).

В нормализованном двоичном числе старший разряд всегда равен 1, поэтому в памяти его можно не хранить. Это позволяет при сохранении действительных чисел не записывать в память заведомо известный старший разряд, а за счет освободившегося бита сохранить еще один дополнительный разряд мантиссы (перед вычислениями в процессоре «отброшенная» единица восстанавливается). Такой метод хранения носит название «скрытая единица» (*hidden bit*).

В стандарте *IEEE 754* для того, чтобы не хранить в ячейке памяти знак порядка, он представляется не как целое число со знаком в явном виде, а в виде беззнакового числа, называемого смещенным порядком или характеристикой. При этом характеристика (X) отличается от порядка на некоторую фиксированную для данного формата величину, называемую смещением (b):

$$X = e + b. \tag{2}$$

В общем виде, если экспонента действительного числа занимает k бит и находится в промежутке от 0 до 2^{k-1} , величина смещения порядка или смещения экспоненты (*exponent bias*) определяется по формуле:

$$b = 2^{k-1} - 1. \tag{3}$$

Стандарт *IEEE 754* определяют несколько возможных типов чисел с плавающей запятой, из которых чаще всего используются числа одинарной точности (*single precision*), числа двойной точности (*double precision*) и числа двойной расширенной точности (*double-extended precision*). Они отличаются диапазоном представимых в них значений. Параметры базовых форматов приведены в табл. 1.

Таблица 1 – Параметры базовых форматов чисел с плавающей запятой

Формат	Всего бит (n)	Бит в порядке (k)	Бит в мантиссе	«Скрытый» бит	Смещение порядка (b)
<i>single</i>	32	8	23	+	127
<i>double</i>	64	11	52	+	1023
<i>extended</i>	80	15	64	–	16383

В стандарті *IEEE 754* існують засоби, які дозволяють отримувати правильні результати або результати з невеликою помилкою навіть в разі виходу отриманого за час обчислень значення за межі розрядної сітки. Для цього передбачено формування спеціальних значень.

В стандарті *IEEE 754* мінімальний $(000\dots00_2)$ і максимальний $(111\dots11_2)$ n -бітові машинні порядки не служать для кодування порядків звичайних чисел, а використовуються в якості признаку особливих спеціальних значень.

Суть цього підходу полягає в трактуванні кодів з нульовим порядком $(000\dots00_2)$ і ненульовою мантиссою як спеціальних значень, які прийнято називати денормалізованими числами. При цьому використовується наступне правило: якщо код містить всі нулі в відведених під порядок розрядах і ненульову мантиссу, то вважається, що порядок числа дорівнює $000\dots001_2$, а явно не вказана в полі ціла частина мантисси приймається рівною нулю. Отже, мантисса – денормалізована, і її код фактично збігається з числом. Використання денормалізованих чисел – це спосіб збільшити кількість представимих чисел з плаваючою комою для підвищення точності обчислень.

Таким чином, в обговорюваному стандарті мінімально можливий n -бітовий машинний порядок нормалізованих чисел дорівнює $000\dots01_2$, а максимально можливий – $111\dots10_2$.

Для обчислення граничних значень необхідно представити, як вони зберігаються в коміях пам'яті. Наприклад, мінімальне нормалізоване число в форматі *single* можна представити в вигляді, показаному на рис. 1.

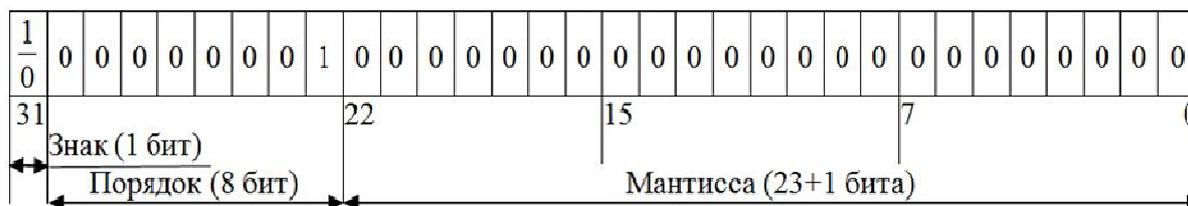


Рисунок 1 – Представлення мінімального нормалізованого числа в форматі *single*

Як видно, мантисса мінімального нормалізованого числа в форматі *single* складається з 23 нулів. Восстановлюючи в мантиссі цілу частину, отримаємо її вихідне значення $1,0_2 = 1,0_{10}$.

Смещений порядок мінімального нормалізованого числа рівняється одиниці в двоичному коді і відрізняється від порядку на величину зміщення. Використовуючи формули (2) і (3), можна розрахувати значення порядку в десятичній системі числення:

$$e = 2^0 - (2^{8-1} - 1) = -126.$$

Відповідно, використовуючи формулу (1), можна підрахувати значення мінімального нормалізованого числа в форматі *single* в десятичній системі числення:

$$A_{\min} = \pm 1,0 \cdot 2^{-126} \approx \pm 1.175494E - 38 .$$

Таким чином, можна отримати загальну формулу для розрахунку значень мінімальних нормалізованих чисел в різних форматах:

$$A_{\min} = \pm 2^{1-b} . \tag{4}$$

Використовуючи формулу (4), отримаємо значення мінімальних нормалізованих чисел для залишених форматів:

няється тем, що десятибайтова разрядная сетка используется не только для хранения действительных чисел, но и для выполнения над ними различных операций. Если первоначально число в формате с плавающей запятой хранится в четырех- или восьмибайтовом поле, то перед выполнением какого-либо действия в коде мантиисы этого числа восстанавливается единица целой части и производится расширение кода до десятибайтовой сетке. Таким приемом обеспечиваются правильность выполнения всех арифметических операций над числами в формате с плавающей запятой и наивысшая возможная точность вычислений.

Мантииса максимального нормализованного числа в формате *extended* состоит из 64 единиц. С учетом указанного отличия, мантииса наибольшего числа в десятибайтовом поле равняется:

$$m = 1 + (1 - 2^{-63}) = 2 - 2^{-63}.$$

Смещенный порядок максимального нормализованного числа рассматриваемого формата равен 111111111111110_2 . Воспользовавшись формулами (2) и (3), можно рассчитать значение порядка в десятичной системе счисления:

$$e = 32766 - (2^{15-1} - 1) = 16383.$$

Наибольшее, представимое в формате *extended* число равно:

$$A_{\max} = \pm(2 - 2^{-63}) \cdot 2^{16383} \approx \pm 1.189731E + 4932.$$

Полученные граничные значения сведены в табл.2.

Таблица 2 – Граничные значения нормализованных чисел с плавающей запятой

Формат	Максимальное значение	Минимальное значение
<i>single</i>	$\pm 3.402823E + 38$	$\pm 1.175494E - 38$
<i>double</i>	$\pm 1.797693E + 308$	$\pm 2.225073E - 308$
<i>extended</i>	$\pm 1.189731E + 4932$	$\pm 3.362103E - 4932$

Выводы. В работе описано машинное представление и хранение граничных (максимальных и минимальных) значений чисел с плавающей запятой в стандарте *IEEE 754*. Получены формулы для вычисления и рассчитаны граничные значения в десятичной системе счисления для различных форматов этого стандарта.

Так как действительные числа – это бесконечное множество, в то время как их машинная реализация числами с плавающей запятой представляет собой конечное множество, то для создания корректных программ, а также минимизации ошибок вычислений важно понимать особенности представления и хранения в ЭВМ граничных значений таких чисел.

ЛИТЕРАТУРА

1. Петров Ю.П. Обеспечение надежности и достоверности компьютерных расчетов / Петров Ю.П. – СПб: БХВ-Петербург, 2008. – 160с.
2. Patriot Missile Defense: Software Problem Led to System Failure at Dhahran. [Электронный ресурс] – Режим доступа: <http://www.gao.gov/products/IMTEC-92-26>.
3. Report by the Inquiry Board. [Электронный ресурс] – Режим доступа: <http://www.ima.umn.edu/~arnold/disasters/ariane5rep.html>.
4. IEEE Standard for Binary Floating-Point Arithmetic. – New York, 1985. – 23p.

5. IEEE Standard for Floating-Point Arithmetic. – New York, 2008. – 70p.
6. IEEE floating point – Wikipedia, the free encyclopedia. [Электронный ресурс] – Режим доступа: http://en.wikipedia.org/wiki/IEEE_floating_point.
7. Ніконов О.Я. Оцінка точності обчислень спеціальних функцій при розробці комп'ютерних програм математичного моделювання / О.Я.Ніконов, О.В.Мнушка, В.М.Савченко // Вісник НТУ «ХП». Тематичний випуск: Інформатика і моделювання. – Харків: НТУ. – 2011. – №17. – С.115-121.
8. Кулямин В.В. Формальные подходы к тестированию математических функций / Кулямин В.В. // Труды ИСП РАН. – 2006. – №10. – С.69-114.
9. Аноприенко А.Я. Гибкая разрядность и постбинарные форматы представления вещественных чисел / Аноприенко А.Я., Иваница С.В. // Вестник Инженерной Академии Украины. Теоретический и научно-практический журнал Инженерной Академии Украины. – Киев. – 2012. – Вып.1. – С.92-98.
10. Кулямин В.В. Стандартизация и тестирование выполнения математических функций в вычислениях с плавающей точкой / Кулямин В.В. // Программирование. – 2007. – №3. 33(3). – С.44-72.
11. Жульковская И.И. Алгоритм формирования машинного представления числовых данных в формате с плавающей запятой / Жульковская И.И., Жульковский О.А. // Математичне моделювання. – Дніпродзержинськ: ДДТУ. – 2013. – №2 (29). – С.69-72.

Поступила в редколлегию 22.12.2014.

УДК 519.8

КАДОЧНИКОВА Я.Е., к.ф.-м.н., ст. преподаватель

Днепродзержинский государственный технический университет

РЕШЕНИЕ НЕПРЕРЫВНЫХ ЗАДАЧ РАЗМЕЩЕНИЯ-РАЗБИЕНИЯ В УСЛОВИЯХ РИСКА

Введение. Под задачей размещения-разбиения (location-allocation problem) обычно понимают необходимость размещения набора «производителей» в заданной области и при этом распределения всех «потребителей» по «производителям», т.е. каждый потребитель должен быть отнесён к одному или нескольким производителям таким образом, чтобы издержки на транспортировку товара от «производителей» к «потребителям» были минимальны, а количество «производителей», размещённых в области, было достаточным для удовлетворения спроса всех «потребителей».

В качестве производителей могут выступать пожарные депо, госпитали, полицейские участки, склады, торговые точки, автозаправки, сервисные центры, банки и многое другое. Соответственно, потребителями являются люди или организации, приобретающие товары или услуги.

Отметим, что классификация задач размещения проводится согласно [1] по основным компонентам этих задач, которыми являются производители, их местоположение и потребители.

Одной из важных характеристик является количество размещаемых производителей. Самый простой случай – размещение только одного предприятия, более общий случай предполагает размещение одного и более центров.

Тип предприятия – другая важная характеристика. В простейшем случае все предприятия считаются одинаковыми в плане своих размеров и наборов сервисов, которые они предлагают. Однако часто требуется размещать производства, которые отличаются друг от друга, например, больницы и пункты скорой помощи.

Моделі розміщення-розбиення можуть також бути розділені на однопродуктові та багатопродуктові моделі в залежності від кількості товарів або сервісів, які виробляє або надає виробник.

Також на увазі приймається, чи можуть виробники забезпечувати нескінченний попит або ж мають обмеження на об'єми виробництва. В цьому відношенні їх можна розділити на задачі з обмеженнями на виробничі потужності та без обмежень на виробничі потужності.

Множина допустимих розміщень виробника може бути одного з трьох типів: дискретна, неперервна або задана на графі, в залежності від чого будують наступні моделі:

- неперервна модель, в якій дозволяється розміщати виробників в будь-якій точці допустимого множини;
- дискретна модель, в якій можливі місця розміщення виробників фіксовані та рахункові;
- мережна модель, яка базується на теорії графів.

В деяких моделях задачі розміщення-розбиення розглядаються не як детерміновані, з чітко заданими початковими даними, параметрами та залежностями, а в умовах певної невизначеності. Таке розширення є природним, так як при розв'язанні реальних завдань дуже складно, а іноді й неможливо, витративши великі кошти або час, отримати достовірну, чітко визначену інформацію. Тому деякі функції можуть бути детермінованими або ймовірнісними, що дозволяє також класифікувати моделі як детерміновані та стохастичні.

В цій роботі розглядається, згідно зведеної вище класифікації, детермінована неперервна задача розміщення-розбиення з неперервним попитом без обмежень з урахуванням вартості розміщення виробництва. Названа задача узагальнена та зведена до детермінованої задачі оптимального розбиення множин. Побудована стохастична модель задачі розміщення-розбиення на випадок неможливості отримати повну та точну інформацію про функції попиту та метрики, описаних в детермінованій задачі.

Постановка задачі. Нижче наведено постановку задачі розміщення-розбиення з неперервною функцією попиту, яка сформульована в роботі [2].

В певній прямокутній області $M \subseteq R^2$, кожна точка $x = (x_1, x_2) \in M$ характеризується функцією попиту $\rho(x)$, потрібно розмістити N виробств τ_i , $i = 1, \dots, N$.

Кожне виробництво задовольняє потреби попитового регіону в одному обслуговуваному регіоні $A_i \subseteq M$, $i = 1, \dots, N$.

Річні фіксовані витрати підприємств рівні $G_i = G(\tau_i)$.

Загальний попит в кожному регіоні A_i визначається через $w_i = \int_{A_i} \rho(x) dx$.

Виробничі витрати визначаються через $f_i = f(\tau_i, w_i) = k(\tau_i) + a(\tau_i)w_i$, $i = 1, \dots, N$, де $k(\tau_i)$ – фіксовані виробничі витрати, $a(\tau_i)$ – річні змінні витрати на виробництво одиниці продукції.

Продукція реалізовується в регіонах шляхом доставки кожному споживачеві, тому транспортні витрати залежать від подолюваного відстані. Таким чином, витрати на реалізацію продукції в кожному регіоні A_i становлять $C_i = s \int_{A_i} c(x, \tau_i) \rho(x) dx$, де s – вартість доставки одиниці продукції, $c(x, \tau_i)$ – евкли-

дова метрика.

Тогда задача размещения производств в некоторой области с одновременным разбиением заданной области, непрерывно заполненной потребителями, на зоны обслуживания с целью минимизации суммарных транспортных и производственных затрат имеет следующий вид.

$$\text{Задача А.} \quad \underset{A_i, \tau_i, i=1, \dots, N}{\text{Minimize}} \quad Z = \sum_{i=1}^N z_i = \sum_{i=1}^N G_i + f_i + C_i,$$

где регионы $A_i, i=1, \dots, N$, образуют разбиение области M .

Результаты работы. Легко показать, что задача A является частным случаем непрерывной задачи оптимального разбиения множеств при ограничениях в форме равенств и неравенств из [3], постановка которой приведена ниже.

Пусть Ω – ограниченное, замкнутое, выпуклое, измеримое по Лебегу множество в n -мерном евклидовом пространстве E_n . Требуется найти такое разбиение множества Ω на N измеримых по Лебегу подмножеств $\Omega_1^*, \dots, \Omega_N^*$ (среди которых могут быть и пустые) и такие координаты центров $\tau_1^*, \dots, \tau_N^*$ этих подмножеств в области Ω , которые являются решением следующей оптимизационной задачи.

Задача Б. Найти

$$\min_{(\{\Omega_1, \dots, \Omega_N\}, \{\tau_1, \dots, \tau_N\})} F(\{\Omega_1, \dots, \Omega_N\}, \{\tau_1, \dots, \tau_N\})$$

при условиях

$$\bigcup_{i=1}^N \Omega_i = \Omega, \quad \text{mes}(\Omega_i \cap \Omega_k) = 0, \quad i \neq k, \quad i, k = 1, \dots, N, \quad \text{где } \text{mes}(\cdot) \text{ – мера Лебега,}$$

$$\int_{\Omega_i} \rho(x) dx = b_i, \quad i = 1, \dots, p,$$

$$\int_{\Omega_i} \rho(x) dx \leq b_i, \quad i = p+1, \dots, N,$$

$$\int_{\Omega_i} c(x, \tau_i) \rho(x) dx \leq l_i, \quad i = 1, \dots, N,$$

$$\text{где } F(\{\Omega_1, \dots, \Omega_N\}, \{\tau_1, \dots, \tau_N\}) = \sum_{i=1}^N \left\{ a_i \int_{\Omega_i} \rho(x) dx + s \int_{\Omega_i} c(x, \tau_i) \rho(x) dx \right\},$$

$\rho(x)$ – действительная, ограниченная, измеримая и неотрицательная на Ω функция, которая характеризует потребительский спрос,

функции $c(x, \tau_i)$ – действительные, ограниченные, измеримые по аргументу x на некотором открытом, ограниченном, выпуклом множестве Ω , и выпуклые по τ_i на Ω для всех $i = 1, \dots, N$, являющиеся метриками,

$\tau_i = (\tau_i^{(1)}, \dots, \tau_i^{(n)})$ – неизвестная заранее точка подмножества $\Omega_i, i = 1, \dots, N$, которая является i -ым центром области Ω ,

b_1, \dots, b_N – заданные действительные неотрицательные числа, ограничивающие производственные мощности производств $\tau_i, i = 1, \dots, N$,

l_1, \dots, l_N – заданные действительные неотрицательные числа, ограничивающие пропускные способности производств $\tau_i, i = 1, \dots, N$.

Здесь и в дальнейшем интегралы понимаются в смысле Лебега, мера множества граничных точек подмножеств $\Omega_i, i = 1, \dots, N$, равна нулю.

Отметим, что задача *B* является обобщением задачи *A* на случай введения ограничений-равенств и ограничений-неравенств на производственные мощности, а также дополнительных ограничений на пропускные коммуникации производств.

Решение задачи *B* основывается на едином подходе, суть которого состоит в сведении исходной задачи оптимизации определённым образом к негладкой конечно-мерной задаче оптимизации, для решения которой применяется метод обобщённых псевдоградиентов с растяжением пространства, близкий к *r*-алгоритму Шора. Подробное обоснование метода решения и разработанный алгоритм приведены в [3].

Дальнейшее усложнение модели подразумевает постановку задачи *A* в условиях риска, т.е. когда известны некоторые вероятностно-статистические параметры η_i , $i = 0, 1, \dots, N$, функций потребительского спроса $\rho(x, \eta_0)$ и метрики $c(x, \tau_i, \eta_i)$, $i = 1, \dots, N$. В таком случае получаемая стохастическая задача размещения-разбиения может быть сведена к следующей задаче.

Задача В. Найти
$$\min_{(\{\Omega_1, \dots, \Omega_N\}, (\tau_1, \dots, \tau_N))} F_M(\{\Omega_1, \dots, \Omega_N\}, (\tau_1, \dots, \tau_N))$$

при условиях

$$\begin{aligned} M \int_{\Omega_i} \rho(x, \eta_0) dx &= b_i, \quad i = 1, \dots, p, \\ M \int_{\Omega_i} \rho(x, \eta_0) dx &\leq b_i, \quad i = p + 1, \dots, N, \\ M \int_{\Omega_i} c(x, \tau_i, \eta_i) \rho(x, \eta_0) dx &\leq l_i, \quad i = 1, \dots, N, \\ \{\Omega_1, \dots, \Omega_N\} &\in \{(\Omega_1, \dots, \Omega_N) : \bigcup_{i=1}^N \Omega_i = \Omega, \\ &mes(\Omega_i \cap \Omega_k) = 0, \quad i \neq k, \quad i, k = 1, \dots, N\}, \\ \tau &= (\tau_1, \dots, \tau_N) \in \Omega^N, \end{aligned}$$

где $F_M(\{\Omega_1, \dots, \Omega_N\}, (\tau_1, \dots, \tau_N)) = M \sum_{i=1}^N \int_{\Omega_i} [c(x, \tau_i, \eta_i) + a_i] \rho(x, \eta_0) dx$,

$x = (x^{(1)}, \dots, x^{(n)}) \in \Omega$, $\tau_i = (\tau_i^{(1)}, \dots, \tau_i^{(n)}) \in \Omega$; $b_1, \dots, b_N, l_1, \dots, l_N$ – заданные действительные числа, a_1, \dots, a_N – заданные действительные неотрицательные числа,

$\eta_i = \eta_i(\theta) : \Theta \rightarrow R$ ($i = 0, 1, \dots, N$) – случайные величины на вероятностном пространстве $(\Theta, \mathfrak{F}, P)$ с заданными конечными математическими ожиданиями $\bar{\eta}_0, \dots, \bar{\eta}_N$ и дисперсиями $\check{\eta}_0, \dots, \check{\eta}_N$.

Функции $c(x, \tau_i, y_i)$ – действительные, ограниченные, измеримые по аргументу x на некотором открытом, ограниченном, выпуклом множестве W из E_n , содержащем Ω , выпуклые по τ_i на W и борелевские по y_i на множестве значений случайной величины $\eta_i(\theta)$ для всех $i = 1, \dots, N$; функция $\rho(x, y_0)$ – действительная, ограниченная, неотрицательная, измеримая по аргументу x на Ω , и борелевская по y_0 на множестве значений случайной величины $\eta_0(\theta)$.

Для решения задачи *B* можно использовать непрямой метод решения из [4], который основан на замене исходной стохастической задачи детерминированным эквивалентом, точным для линейных и квадратичных относительно случайных параметров

функцій, входящих в постановку задачи, и приближённым в остальных случаях. При использовании такого подхода отыскивается детерминированное оптимальное разбиение.

Переход к детерминированному эквиваленту можно осуществить с помощью представления случайных функций от параметров состояния, входящих в целевой функционал и ограничения задачи, в виде разложения в ряд Тейлора в окрестности математических ожиданий случайных параметров с сохранением линейного и квадратичного членов разложения. При этом для построения детерминированного эквивалента используется субъективная информация о вероятностных характеристиках случайных параметров задачи.

Такой простой подход к построению детерминированного эквивалента требует всего лишь задания субъективных математических ожиданий и дисперсий конечного числа случайных параметров задачи и позволяет избежать решения сложной задачи определения многомерной совместной плотности распределения вероятностей, к которой формально сводится определение математических ожиданий из целевого функционала и функционалов ограничений.

Для решения полученного детерминированного эквивалента применяется метод аналогичный методу решения непрерывных линейных задач оптимального разбиения множеств из [3].

Полное описание метода решения задачи B приведено в [4]. Ниже приведён алгоритм решения детерминированного эквивалента.

Алгоритм.

0-й шаг. Область Ω заключаем в прямоугольный параллелепипед Π , стороны которого параллельны осям декартовой системы координат, полагаем $\rho(x, \bar{\eta}_0) = 0$ при $x \in \Pi \setminus \Omega$. Параллелепипед Π покрываем прямоугольной сеткой и задаём начальное приближение $(\tau, \psi, \xi) = (\tau^{(0)}, \psi^{(0)}, \xi^{(0)})$. Задаём параметры $\alpha, q_1, q_2, n_h, \varepsilon$ модификации $r(\alpha)$ -алгоритма.

Вычисляем значения $\lambda^{(0)}(x)$ в узлах сетки по формулам

$$\lambda_i(x) = \begin{cases} 1, & \text{если } \varphi_i(x, \tau_i, \psi_i, \xi_i) \leq \varphi_k(x, \tau_k, \psi_k, \xi_k), \\ & k = 1, \dots, N, i \neq k \text{ п. в. для } x \in \Omega, \\ 0 & \text{в остальных случаях.} \end{cases} \quad (1)$$

где $\varphi_i(x, \tau_i, \psi_i, \xi_i) = \left[(1 + \xi_i) \left(c(x, \tau_i, \bar{\eta}_i) + \frac{1}{2} c''_{\eta_i \eta_i}(x, \tau_i, \bar{\eta}_i) \check{\eta}_i \right) + a_i + \psi_i \right] \rho(x, \bar{\eta}_0) +$

$+\frac{1}{2} \left[(1 + \xi_i) c(x, \tau_i, \bar{\eta}_i) + a_i + \psi_i \right] \rho''_{\eta_0 \eta_0}(x, \bar{\eta}_0) \check{\eta}_0 + c'_{\eta_i}(x, \tau_i, \bar{\eta}_i) \rho'_{\eta_0}(x, \bar{\eta}_0) \text{cov}(\eta_0, \eta_i)$

при $\tau = \tau^{(0)}, \psi = \psi^{(0)}, \xi = \xi^{(0)}$. Вычисляем значения вектора $g_p(\tau^{(0)}, \psi^{(0)}, \xi^{(0)})$ в узлах сетки по формулам

$$g_p^{w_i}(\tau, \psi, \xi) = \int_{\Omega} \left[\rho(x, \bar{\eta}_0) + \frac{1}{2} \rho''_{\eta_0 \eta_0}(x, \bar{\eta}_0) \check{\eta}_0 \right] \lambda_i(x) dx - b_i, \quad i = 1, \dots, p, \quad (2)$$

$$g_p^{w_i}(\tau, \psi, \xi) = \int_{\Omega} \left[\rho(x, \bar{\eta}_0) + \frac{1}{2} \rho''_{\eta_0 \eta_0}(x, \bar{\eta}_0) \check{\eta}_0 \right] \lambda_i(x) dx - b_i + S_1 \max(0, \text{sign}(-\psi_i)), \quad i = p+1, \dots, N, \quad (3)$$

$$g_P^{\xi_i}(\tau, \psi, \xi) = \int_{\Omega} \left\{ \left(c(x, \tau_i, \bar{\eta}_i) + \frac{1}{2} c''_{\eta_i \eta_i}(x, \tau_i, \bar{\eta}_i) \check{\eta}_i \right) \rho(x, \bar{\eta}_0) + \right. \\ \left. + \frac{1}{2} c(x, \tau_i, \bar{\eta}_i) \rho''_{\eta_0 \eta_0}(x, \check{\eta}_0) \check{\eta}_0 \right\} \lambda_i(x) dx - l_i + S_2 \max(0, \text{sign}(-\xi_i)), i = 1, \dots, N, \quad (4)$$

$$g_P^{\tau_i}(\tau, \psi, \xi) = \int_{\Omega} \left\{ (1 + \xi_i) \left(c'_{\tau_i}(x, \tau_i, \bar{\eta}_i) + \frac{1}{2} c'''_{\eta_i \eta_i \tau_i}(x, \tau_i, \bar{\eta}_i) \check{\eta}_i \right) \rho(x, \bar{\eta}_0) + \right. \\ \left. + \frac{1}{2} (1 + \xi_i) c'_{\tau_i}(x, \tau_i, \bar{\eta}_i) \rho''_{\eta_0 \eta_0}(x, \bar{\eta}_0) \check{\eta}_0 + c''_{\eta_i \tau_i}(x, \tau_i, \bar{\eta}_i) \rho'_{\eta_0}(x, \bar{\eta}_0) \text{cov}(\eta_0, \eta_i) \right\} \lambda_i(x) dx - \\ - S_3 \max\{0, \text{sign}(x_{\min}^k - \tau_i^k)\} - S_4 \max\{0, \text{sign}(\tau_i^k - x_{\max}^k)\}, i = 1, \dots, N, \quad (5)$$

при $\tau = \tau^{(0)}$, $\psi = \psi^{(0)}$, $\xi = \xi^{(0)}$, $\lambda(x) = \lambda^{(0)}(x)$.

Выбираем начальный пробный шаг $h_0 > 0$, полагаем $B_0^{\tau} = I_{nN}$, $B_0^{\psi} = I_N$, $B_0^{\xi} = I_N$ - квадратные матрицы размерностей $nN \times nN$, $N \times N$ и $N \times N$ соответственно. Находим

$$\begin{aligned} \tau^{(1)} &= \tau^{(0)} - h_0 g_P^{\tau}(\tau^{(0)}, \psi^{(0)}, \xi^{(0)}), \\ \psi^{(1)} &= \psi^{(0)} - h_0 g_P^{\psi}(\tau^{(0)}, \psi^{(0)}, \xi^{(0)}), \\ \xi^{(1)} &= \xi^{(0)} - h_0 g_P^{\xi}(\tau^{(0)}, \psi^{(0)}, \xi^{(0)}). \end{aligned}$$

Переходим ко второму шагу.

Пусть в результате вычислений после k , $k = 1, 2, \dots$ шагов алгоритма получены величины $\tau^{(k)}$, $\psi^{(k)}$, $\xi^{(k)}$, $\lambda^{(k-1)}(x)$ в узлах сетки, матрицы B_k^{τ} , B_k^{ψ} , B_k^{ξ} .

Опишем $(k+1)$ -й шаг.

1. Вычисляем значения $\lambda^{(k)}(x)$ в узлах сетки по формулам (1) при $\tau = \tau^{(k)}$, $\psi = \psi^{(k)}$, $\xi = \xi^{(k)}$.

2. Вычисляем значения $g_P(\tau^{(k)}, \psi^{(k)}, \xi^{(k)})$ в узлах сетки по формулам (2)-(5) при $\tau = \tau^{(k)}$, $\psi = \psi^{(k)}$, $\xi = \xi^{(k)}$, $\lambda(x) = \lambda^{(k)}(x)$.

3. Проводим $(k+1)$ -й шаг $r(\alpha)$ -алгоритма обобщённых псевдоградиентов с растяжением пространства, близкого к r -алгоритму Шора, итерационная формула которого имеет вид

$$\begin{aligned} \tau^{(k+1)} &= \tau^{(k)} - h_k B_{k+1}^{\tau} \frac{(B_{k+1}^{\tau})^T g_P^{\tau}(\tau^{(k)}, \psi^{(k)}, \xi^{(k)})}{\|(B_{k+1}^{\tau})^T g_P^{\tau}(\tau^{(k)}, \psi^{(k)}, \xi^{(k)})\|}, \\ \psi^{(k+1)} &= \psi^{(k)} - h_k B_{k+1}^{\psi} \frac{(B_{k+1}^{\psi})^T g_P^{\psi}(\tau^{(k)}, \psi^{(k)}, \xi^{(k)})}{\|(B_{k+1}^{\psi})^T g_P^{\psi}(\tau^{(k)}, \psi^{(k)}, \xi^{(k)})\|}, \\ \xi^{(k+1)} &= \xi^{(k)} - h_k B_{k+1}^{\xi} \frac{(B_{k+1}^{\xi})^T g_P^{\xi}(\tau^{(k)}, \psi^{(k)}, \xi^{(k)})}{\|(B_{k+1}^{\xi})^T g_P^{\xi}(\tau^{(k)}, \psi^{(k)}, \xi^{(k)})\|}. \end{aligned}$$

4. Если условие

$$\|(\tau^{(k+1)}, \psi^{(k+1)}, \xi^{(k+1)}) - (\tau^{(k)}, \psi^{(k)}, \xi^{(k)})\| \leq \varepsilon, \varepsilon > 0, \quad (6)$$

не выполняется, переходим к $(k+2)$ -му шагу алгоритма с новыми значениями величин $\tau^{(k+1)}$, $\psi^{(k+1)}$, $\xi^{(k+1)}$, $\lambda^{(k)}(x)$ в узлах сетки и матриц B_{k+1}^{τ} , B_{k+1}^{ψ} , B_{k+1}^{ξ} , если выполняется, то переходим на п.5.

5. Полагаем $\tau_* = \tau^{(l)}$, $\psi_* = \psi^{(l)}$, $\xi_* = \xi^{(l)}$, $\lambda_*(x) = \lambda^{(l)}(x)$, где l – номер итерации, на которой выполнилось условие (6).

6. Вычисляем оптимальное значение двойственного функционала по формуле

$$\tilde{G}_M(\tau, \Psi) = -\sum_{i=1}^N \psi_i b_i - \sum_{i=1}^N \xi_i l_i + \int_{\Omega} \min_{k=1, \dots, N} \{ \varphi_k(x, \tau_k, \psi_k, \xi_k) \},$$

где $\varphi_i(x, \tau_i, \psi_i, \xi_i)$, $i=1, \dots, N$, $j=1, \dots, M$, задаются формулами

$$\varphi_i(x, \tau_i, \psi_i, \xi_i) = \left[(1 + \xi_i) \left(c(x, \tau_i, \bar{\eta}_i) + \frac{1}{2} c''_{\eta_i \eta_i}(x, \tau_i, \bar{\eta}_i) \check{\eta}_i \right) + a_i + \psi_i \right] \rho(x, \bar{\eta}_0) +$$

$$+ \frac{1}{2} \left[(1 + \xi_i) c(x, \tau_i, \bar{\eta}_i) + a_i + \psi_i \right] \rho''_{\eta_0 \eta_0}(x, \bar{\eta}_0) \check{\eta}_0 +$$

$+ c'_{\eta_i}(x, \tau_i, \bar{\eta}_i) \rho'_{\eta_0}(x, \bar{\eta}_0) \text{cov}(\eta_0, \eta_i)$, $i=1, \dots, N$ при $\tau = \tau_*$, $\psi = \psi_*$, $\xi = \xi_*$, и для контроля правильности счёта по формуле

$$I(\lambda_*(\cdot), \tau_*) = \int_{\Omega} \sum_{i=1}^N \left\{ \left[c(x, \tau_{*i}, \bar{\eta}_i) + a_i + \frac{1}{2} c''_{\eta_i \eta_i}(x, \tau_{*i}, \bar{\eta}_i) \check{\eta}_i \right] \rho(x, \bar{\eta}_0) + \right.$$

$$\left. + \frac{1}{2} \left[c(x, \tau_{*i}, \bar{\eta}_i) + a_i \right] \rho''_{\eta_0 \eta_0}(x, \bar{\eta}_0) \check{\eta}_0 + c'_{\eta_i}(x, \tau_{*i}, \bar{\eta}_i) \rho'_{\eta_0}(x, \bar{\eta}_0) \text{cov}(\eta_0, \eta_i) \right\} \lambda_{*i}(x) dx.$$

Алгоритм описан.

Выводы. В данной работе рассмотрена классификация задач размещения-разбиения. Приведена постановка детерминированной непрерывной задачи размещения-разбиения с непрерывным спросом без ограничений. Названная задача сведена к задаче оптимального разбиения множеств с учётом обобщения на случай введения ограничений на производственные мощности и пропускные коммуникации.

Даная задача размещения-разбиения сформулирована в условиях риска, т.е. когда известны некоторые вероятностно-статистические параметры функций потребительского спроса и метрики. Решение стохастической задачи получено непрямым методом, идея которого состоит в сведении стохастической задачи к её детерминированному эквиваленту, для решения которого необходимо знание лишь субъективных математических ожиданий и дисперсий случайных параметров задачи. Описан алгоритм решения задачи размещения-разбиения в условиях риска, который программно реализован и успешно протестирован на ряде модельных задач, где функция метрики имеет вероятностно-статистическую природу.

ЛИТЕРАТУРА

1. Farahani R.Z. Facility Location: Concepts, Models, Algorithms and Case Studies / R.Z.Farahani, M.Hekmatfar. – Berlin: Springer, 2009. – 549pp.
2. Murat A.A Continuous Analysis Framework for the Solution of Location-Allocation Problems with Dense Demand / A.Murat, V.Verter, G.Laporte // Computers & Operations Research. – 2010. – Vol. 37. – P.123-136.
3. Киселёва Е.М. Решение непрерывной однопродуктовой задачи оптимального разбиения с дополнительными ограничениями / Е.М.Киселёва, Я.Е.Кадочникова // Проблемы управления и информатики. – 2009. – №4. – С.47-61.
4. Кадочникова Я.Е. Решение одной задачи оптимального разбиения множеств при дополнительных ограничениях в условиях неопределённости / Я.Е.Кадочникова // Збірник наукових праць Дніпродзержинського державного технічного університету. – Дніпродзержинськ: ДДТУ. – 2010. – Випуск 2(15). – С.147-152.

Поступила в редколлегию 18.02.2015.

Вінницький національний технічний університет

УДОСКОНАЛЕНИЙ МЕТОД ФОРЕЛ ДЛЯ КЛАСТЕРИЗАЦІЇ СТАНІВ КОМП'ЮТЕРНОЇ ТЕХНІКИ

Вступ. Сучасний стан розвитку суспільства характеризується різким зростанням обсягів інформації у різних галузях, що зумовило необхідність широкого застосування комп'ютерної техніки (КТ) та, відповідно, спричинило зростання попиту на неї.

У зв'язку із зростанням кількості та різноманіття комп'ютерної техніки зростає і складність задач прийняття рішень щодо експлуатації такої техніки при виникненні несправностей в процесі її роботи, оскільки кожен новий її вид має свої особливості функціонування, що відображається в її станах. Тому доцільною є розробка і удосконалення методів та засобів кластеризації станів комп'ютерної техніки з метою формування в подальшому обґрунтованих рекомендацій щодо забезпечення коректного функціонування такої техніки [1].

Постановка задачі. Множину станів КТ X необхідно розбити на k непересічних підмножин, які називаються таксонами (кластерами), таким чином, щоб кожен з них складався із станів, при яких формуються подібні рекомендації щодо подальших дій при їх виникненні.

Результати роботи. Розробка удосконаленого методу *ФОРЕЛ* для кластеризації станів комп'ютерної техніки. Серед множини методів кластеризації для аналізу станів КТ доцільно застосувати метод *ФОРЕЛ*, який потрібно вдосконалити для усунення недоліків існуючих його модифікацій та адаптації до відповідної предметної області.

До особливостей відомого методу кластеризації станів КТ *ФОРЕЛ* [2-5] відносять такі:

- його продуктивність є невисокою;
- метод є збіжним за скінченне число кроків;
- в лінійному просторі центром кластера може виступати як будь-яка точка, так і стан КТ;
- на першому етапі обирається один із станів КТ як початковий об'єкт, від якого проводитиметься кластеризація, що, в свою чергу, впливатиме на якість розбиття;
- наявність апріорних знань про діаметри кластерів;
- відомий метод не враховує особливостей станів КТ, що негативно впливає на якість їхньої кластеризації.

Метод *ФОРЕЛ* та його відомі модифікації передбачають: необхідність задання радіуса кластера R ; можливість віднесення до кластера об'єктів з інших таксонів через неправильний вибір радіуса, що негативно впливає на достовірність декомпозиції множини станів КТ, оскільки користувач не має інформації про розміри кластерів. У зв'язку з цим запропоновано удосконалення методу, що дозволяє підвищити якість розбиття множини станів КТ за рахунок аналізу отриманих кластерів і виявлення станів, які є викидами, а також визначення радіуса кластера R , що враховує допустимі відхилення значень параметрів та характеристик станів КТ і показника якості кластеризації η , яка б задовольнила користувача

$$R = \frac{(MAX - MIN) \times (100 - \eta)}{100} + MIN - \Delta, \quad 1)$$

де MAX, MIN – максимальна та мінімальна відстані між двома станами КТ у множині;

η – показник якості кластеризації $0 \leq \eta \leq 100$;

Δ – значення допустимого відхилення відстані між станами КТ.

Після визначення радіуса центр кластера поміщається в будь-який з некластеризованих станів КТ та відносяться до таксона стани, відстань до яких від центра менша за R . Після чого визначається новий центр (новим центром кластера стає центр мас знайденого таксона) та відбувається повторне формування кластера, до якого відносяться стани КТ, відстань від знайденого центра до яких менша за радіус R . Такі дії виконуються поки центр мас не стабілізується.

Для унеможливлення появи викидів серед станів всередині кластерів після формування кожного з них проводиться відповідний аналіз та віднесення їх до множини некластеризованих.

Стани КТ, які належать новому таксону, вилучаються з некластеризованих, а робота методу повторюється, поки множина некластеризованих станів не буде пустою.

У формалізованому вигляді алгоритм кластеризації станів КТ ФОРЕЛ (рис.1), в основу якого покладено запропонований удосконалений метод, містить такі основні кроки.

1. Формування множини некластеризованих станів КТ

$$U = \{X_1, X_2, \dots, X_n\}, \quad (2)$$

де U – множина некластеризованих станів КТ;

n – кількість станів КТ, що підлягають кластеризації.

2. Знаходження значень відстаней між некластеризованими точками:

а) мінімальної

$$MIN = \min_{i,j=1,\dots,n,i \neq j} a_{3E}(X_i, X_j), \quad (3)$$

де $a_{3E}(X_i, X_j)$ – відстань між станами X_i та X_j ;

б) максимальної

$$MAX = \max_{i,j=1,\dots,n,i \neq j} a_{3E}(X_i, X_j) \quad (4)$$

3. Визначення допустимого відхилення відстані Δ , з використанням зваженої евклідової відстані.

4. Знаходження значення радіуса кластера з використанням формули (1).

5. За умови $U \neq \emptyset$ (у вибірці є некластеризовані стани КТ):

5.1. Обрати довільний стан $X_0 \in U$ випадковим чином.

5.2. Сформувати кластер станів КТ – сферу з центром в X_0 і радіусом R

$$K_0 = \{X_i \in U \mid a_{3E}(X_i, X_0) \leq R\}, \quad (5)$$

де K_0 – сформований кластер станів КТ;

$a_{3E}(X_i, X_0)$ – відстань від стану КТ X_i до центра кластера X_0 .

- 5.3. Помістити центр кластера в його центр мас

$$X_0 = \operatorname{argmin}_{X_i \in K_0} \sum_{X_j \in K_0} a_{3E}(X_i, X_j) \quad (6)$$

де X_0 – центр мас кластера.

5.4. Виконувати пункти 5.1–5.3, поки центр X_0 не стабілізується.

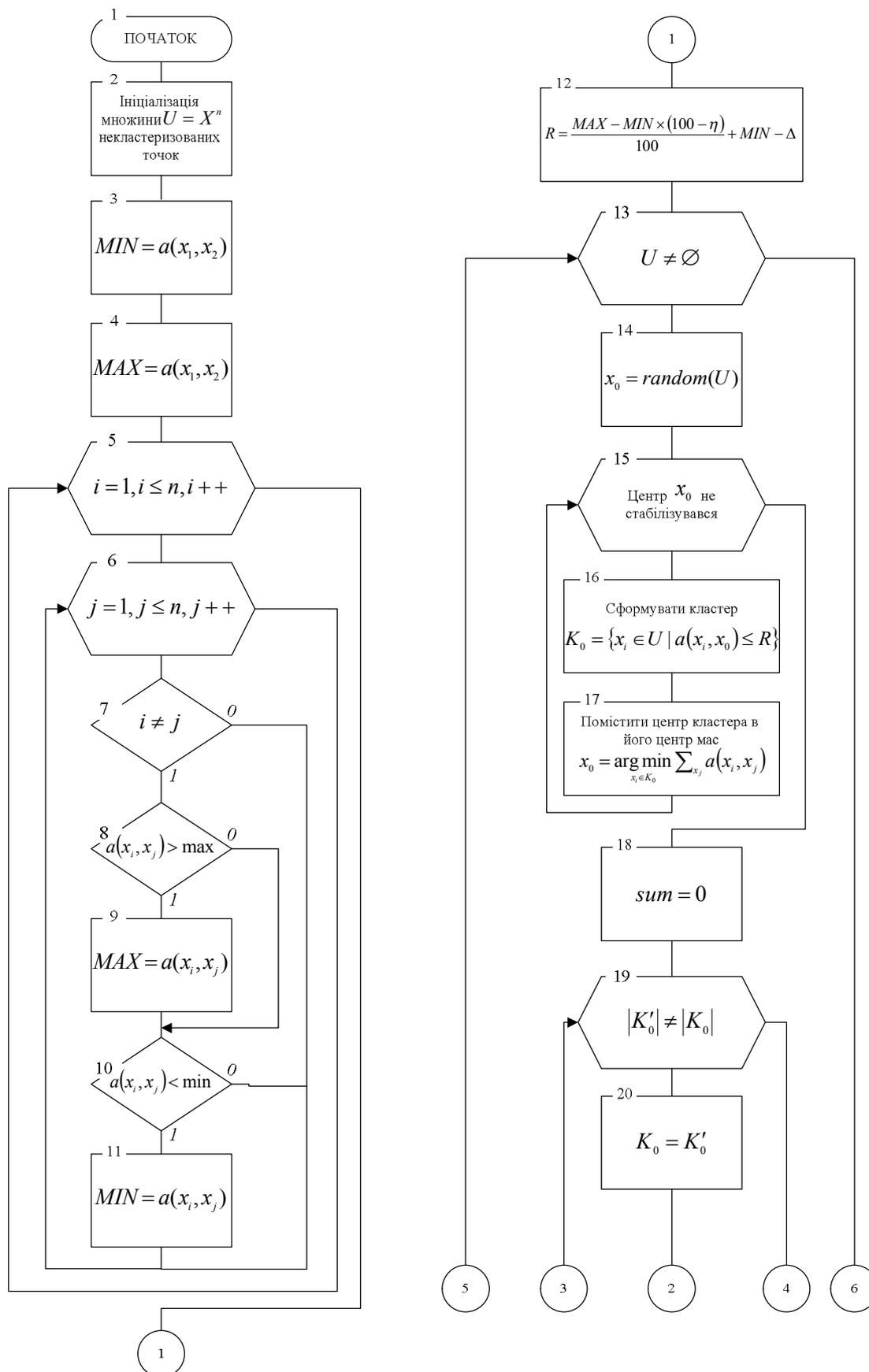


Рисунок 1 – Схема удосконаленого алгоритму кластеризації станів КТ ФОРЕЛ

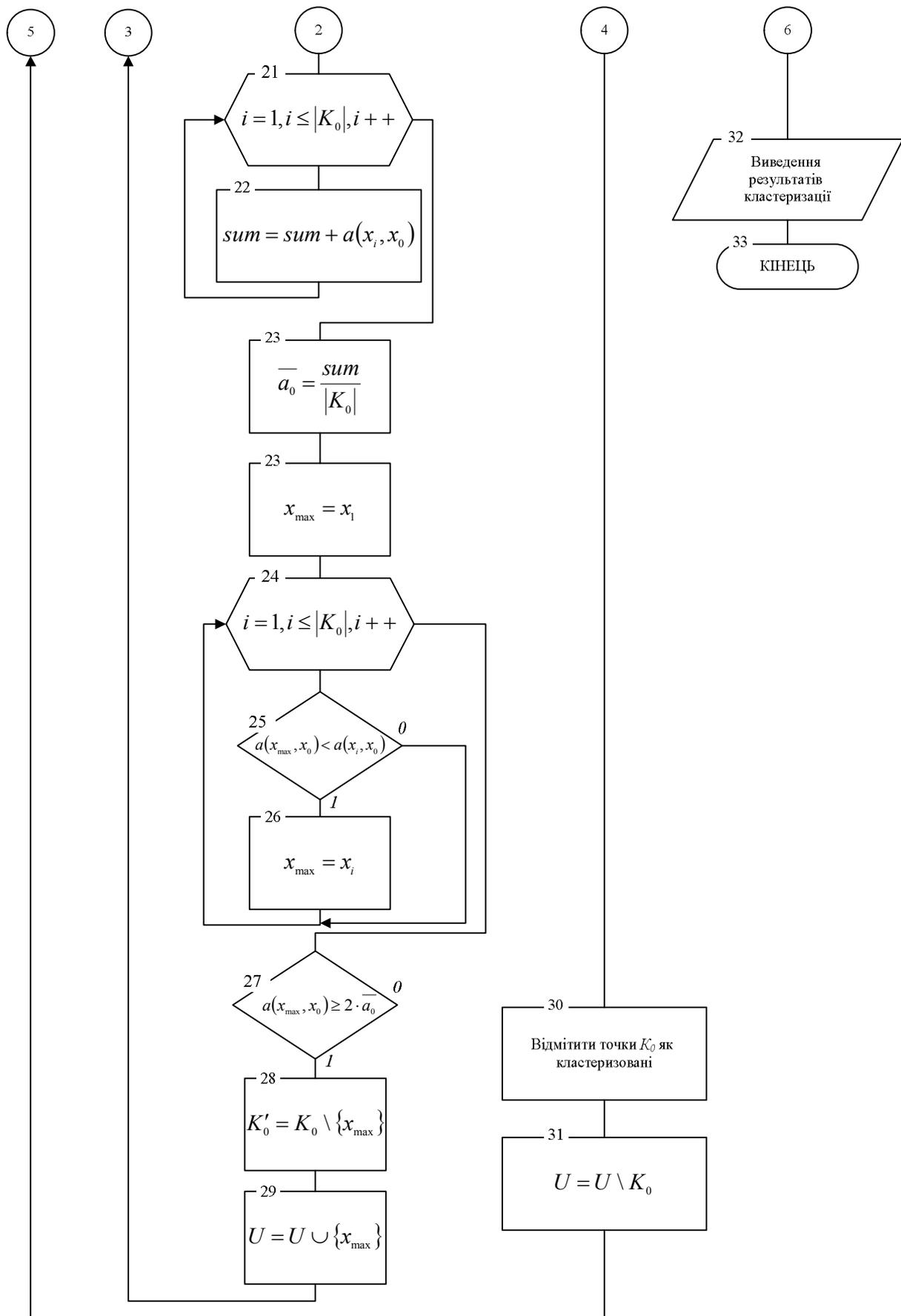


Рисунок 1, аркуш 2

5.5. Знайти

$$\bar{a}_0 = \frac{\sum_{X_0, X_i \in K_0} a_{zE}(X_i, X_0)}{|K_0|}, i = 1, \dots, |K_0|, \quad 7)$$

де \bar{a}_0 – середнє значення відстані між центром кластера та станами КТ, які йому належать.

5.6. Знайти

$$X_{\max} = \arg \max_{X_i \in K_0} a_{zE}(X_i, X_0), i = 1, \dots, |K_0|, \quad 8)$$

де X_{\max} – максимально віддалений від центра кластера стан.

5.7. Якщо $(a_{zE}(X_{\max}, X_0) \geq 2 \cdot \bar{a}_0)$, то

X_{\max} вилучити з K_0

$$K_0 = K_0 \setminus \{X_{\max}\} \quad 9)$$

та X_{\max} додати до U

$$U = U \cup \{X_{\max}\}. \quad 10)$$

5.8. Виконувати пункти 5.6 та 5.7, поки не буде вилучень X_{\max} з K_0 .

5.9. Вилучити стани, що віднесені до кластера K_0 (як кластеризовані)

$$U = U \setminus K_0. \quad 11)$$

6. Виконувати п. 5, поки $U = \emptyset$ (всі стани КТ кластеризовані).

Отже, удосконалений метод кластеризації станів КТ ФОРЕЛ розраховує радіуси кластерів і визначає стани, які є викидами, та вилучає їх з кластерів до множини некластеризованих станів, що дозволило зменшити кількість викидів в кластерах та підвищити якість кластеризації. Також на основі удосконаленого методу розроблено удосконалений алгоритм ФОРЕЛ.

Висновки. Таким чином, удосконалено метод кластеризації станів комп'ютерної техніки ФОРЕЛ, який передбачає розрахунок радіусів кластерів з урахуванням особливостей предметної області, визначає стани, що є викидами та вилучає їх з кластерів. Це дозволило підвищити якість кластеризації таких станів на 6%.

ЛІТЕРАТУРА

1. Перспективы развития компьютерных систем [Электронный ресурс] – режим доступа: <http://www.itstan.ru/komp-tehnika/perspektivy-razvitija-kompjuternyh-sistem.html>.
2. Загоруйко Н.Г. Прикладные методы анализа данных и знаний / Н.Г.Загоруйко. – Новосибирск: ИМ СО РАН, 1999. – 270с. – ISBN 5-86134-060-9.
3. Загоруйко Н.Г. Алгоритмы обнаружения эмпирических закономерностей / Н.Г.Загоруйко, В.Н.Ёлкина, Г.С.Лбов. – Новосибирск: Наука, 1985. – 112с. – ISBN 5-1601212-А.
4. Interpretation of FOREL clusters [Electronic resource]. – Access mode: <http://www.biomedcentral.com/content/supplementary/1471-2164-7-318-s1.pdf>.
5. Крюков А.В. Кластерный анализ режимов систем тягового электроснабжения для целей ситуационного управления [Электронный ресурс] / А.В.Крюков, В.П.Закарюкин, Н.А.Абрамов // Винеровские чтения: 4-й Всерос. конф.: материалы конф. – Режим доступа: <http://berestneva.am.tpu.ru/Papers/KONF2009/%F7%C9%CE%C5%>

D2%CF%D7%D3%CB%C9%C5%20%DE%D4%C5%CE%C9%D1/2009%20(F)/fscomm
and/pdf/042.pdf.

Надійшла до редколегії 10.03.2015.