УДК 004.8

ДОСЛІДЖЕННЯ МОЖЛИВОСТЕЙ ГЕНЕТИЧНОГО АЛГОРИТМУ В ЗАДАЧІ КЛАСТЕРИЗАЦІЇ КОРИСТУВАЧІВ МЕРЕЖІ INTERNET

С.М. Захарченко, Н.Р. Кондратенко, О.О. Манаєва

Вступ

На сьогоднішній день мережа Інтернет є універсальним глобальним інформаційним середовищем, а отже, невід’ємною частиною життя суспільства. Перед будь-яким Інтернет-провайдером постає задача тарифікації власних послуг відповідно до деяких критеріїв. Розв’язання цієї та інших задач покладається, як правило, на білінгову систему – програмний комплекс, що здійснює облік обсягу спожитих абонентом послуг, розрахунок та списання коштів відповідно до тарифів компанії [1].

Таким чином, однією з важливих складових будь-якої білінгової системи є розробка тарифних планів. Вона вимагає ретельного дослідження та формування компактних груп абонентів із подібними потребами.

Виділенням серед множини об’єктів деякої кількості однорідних підмножин, таких, щоб об’єкти всередині підмножин були в певному сенсі подібними, а об’єкти з різних підмножин – відмінними, займається задача кластеризації. Елементами підмножини можуть бути довільні об’єкти, які можна задати векторами характеристик. Самі ж групи прийнято називати кластерами.

Кластеризація має велику кількість практичних застосувань як в інформатиці, так і в інших галузях. Одним із найважливіших напрямків досліджень є її використання для аналізу даних та виділення прихованих закономірностей. До цього класу проблем належить і задача розподілу користувачів мережі Інтернет за їхніми вимогами до якості та характеру послуг, що надає провайдер.

Відомо, що для її розв’язання не існує універсальних методів, що дозволяють швидко знайти абсолютно точні рішення [2].

За способом розбиття на кластери розрізняють два типи алгоритмів: ієрархічні та неієрархічні [3]. Класичні ієрархічні алгоритми працюють тільки з категорійний атрибутами, коли будується повне дерево вкладених кластерів. Тут поширені агломеративні методи побудови ієрархій кластерів - в них проводиться послідовне об'єднання вихідних об'єктів і відповідне зменшення числа кластерів. Ієрархічні алгоритми забезпечують порівняно високу якість кластеризації і не вимагають попереднього задання кількості кластерів.

Неієрархічні алгоритми ґрунтуються на оптимізації деякої цільової функції, що визначає оптимальне в певному сенсі розбиття множини об'єктів на кластери. У цій групі популярні алгоритми родини k-середніх (k-means), які в якості цільової функції використовують суму квадратів зважених відхилень координат об'єктів від центрів шуканих кластерів. Кластери шукаються сферичної або еліпсоїдної форми. У канонічній реалізації мінімізація функції здійснюється на основі методу множників Лагранжа і дозволяє знайти тільки найближчий локальний мінімум.

Серед неієрархічних алгоритмів, не заснованих на відстані, слід виділити EM-алгоритм (Expectation-Maximization). У ньому замість центрів кластерів передбачається наявність функції щільності ймовірності для кожного кластеру з відповідним значенням математичного сподівання і дисперсією [3, 4].

Основним недоліком усіх цих алгоритмів є настільки великий обсяг обчислень, що, не зважаючи на високу швидкодію сучасних обчислювальних машин, розв’язання задачі залишається складним.

Тому для розв’язання цієї задачі доцільно використати інтелектуальні технології [5], зокрема генетичні алгоритми.

Постановка задачі

Поставимо задачу розбиття деякої множини користувачів провайдера інтернет-послуг на групи відповідно до певного набору характеристик: швидкості доступу, обсягу спожитого вхідного та вихідного трафіку. Таке розбиття дасть змогу оцінити потреби абонентів та відповідно до цього організувати максимально гнучку та пристосовану до ситуації на ринку систему тарифів провайдера.

Для розв’язання поставленої задачі запропонуємо генетичний алгоритм, що здійснює кластеризацію користувачів мережі Інтернет відповідно до виділених вище ознак та дослідимо його ефективність і швидкодію в порівнянні з алгоритмом граничного перебору на основі динамічного програмування.

Математична модель та методика дослідження

Нехай маємо набір абонентів $I=\{I_1, I_2, \ldots, I_n\}$ певного провайдера телекомунікаційних послуг. На основі множини показників, що описують цих абонентів, згрупуємо їх у підмножини (кластери)

таким чином, щоб абоненти, що входять до одного класу, були більш однорідними за характером споживання ними трафіку, схожими між собою більше, ніж із абонентами, що належать до інших класів.

Кожному з n абонентів поставимо у відповідність множину ознак (вимірів) $X_i=\{x_1, x_2, \ldots, x_p\}$, які будуть нести інформацію про характер споживання ним Інтернет-трафіку за деякий період. Таким чином, для множини I маємо множину векторів ознак $X=\{X_1, X_2, \ldots, X_n\}$, які повністю її характеризують. У такому записі окремих користувачів з множини I зручно представити у вигляді n точок у p-вимірному просторі ознак. Під подібністю/відмінністю будемо розуміти відносну геометричну близькість між такими точками у багатомірному просторі.

Нехай m – ціле число, менше за n. Задача кластеризації полягає в тому, щоб на основі даних, що містяться у множині I, розбити множину абонентів I на m кластерів.

Розв’язком задачі кластерного аналізу є розбиття, що задовольняє певному критерію оптимальності. Цей критерій може являти собою деякий функціонал, що відображає рівень бажаності різних розбиттів і групувань. Цей функціонал часто називають цільовою функцією [2]. За цільову функцію приймемо традиційну для задач кластеризації суму квадратів Евклідових відстаней між об’єктами в межах одного кластера:

$$d_{Eij} = \left(\sum_{l=1}^{m} \left(x_i^l - x_j^l \right)^2 \right)^{\frac{1}{2}}$$

В основі генетичних алгоритмів лежить теорія еволюції. Згідно з нею, кожен біологічний вид розвивається, поступово змінюється задля того, щоб якнайкращим чином пристосуватися до навколишнього середовища. Класичний генетичний алгоритм являє собою ітераційний процес, на кожній ітерації якого популяція послідовно піддається операціям відбору, схрещування та мутації (рисунок 1). Зупинивши ітераційний процес у певний момент та вибравши кращу особину з популяції, можна отримати достатньо прийнятний розв’язок задачі [5].

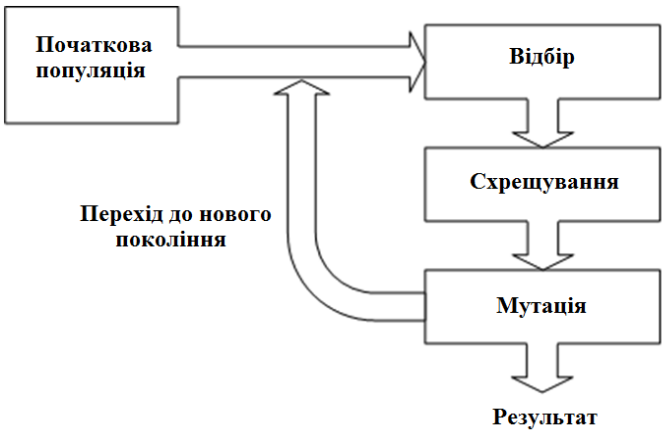

Рисунок 1 - Робота класичного генетичного алгоритму

Для формалізованого опису особини запропонуємо неоднорідну хромосому (рисунок 2):

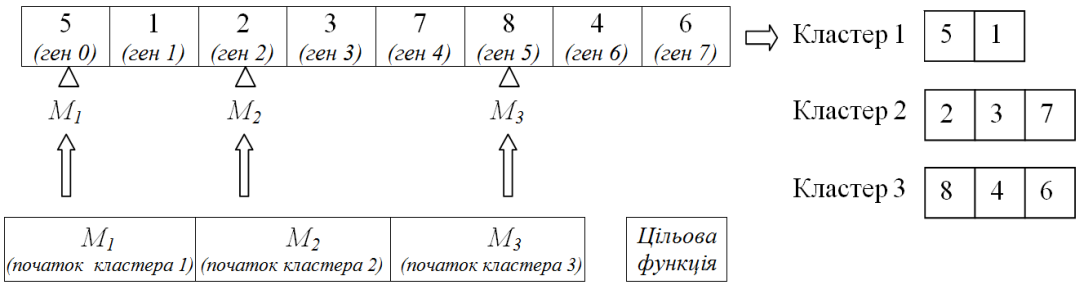

Рисунок 2 – Неоднорідна хромосома

Така хромосома містить дві складові, які попри зовнішню подібність мають принципово різний зміст. Перша складова являє собою лінійний масив розмірності n, що містить гени - номери об’єктів - у певній послідовності. Для виділення в межах хромосоми окремих кластерів введемо до складу нашої хромосоми другу складову – набір маркерів розбиття $M_1, \ldots M_m$. Значення маркеру $M_i = k$ сигналізує про

те, що з k-того гену основної складової починається новий кластер. Окрім того, кожній хромосомі з простору можливих розв’язків ставиться у відповідність значення цільової функції. В процесі отримання розв’язку задачі такі набори даних (проміжні розв’язки) будемо зберігати в робочій матриці.

Закономірно, що для хромосоми такого вигляду операції схрещування та мутації мають певні особливості. Їх буде розглянуто нижче при покроковому описі алгоритму.

Щоб розв’язати задачу кластеризації в описаній вище постановці, необхідно розробити послідовність операцій, які моделюють еволюційні процеси на основі математичних аналогів генетичної спадковості, мінливості та природного відбору.

1) Генерування початкової популяції. Початкова популяція містить n хромосом, згенерованих випадковим чином. При цьому як послідовність генів, так і характер (форма) розбиття задається випадково. Для кожного з n отриманих розбиттів розраховується цільова функція, і початкова популяція записується в робочу матрицю, займаючи її елементи з індексами від 0 до n-1.

2) Створення нащадків:

а) двоточкове схрещування (рисунок 3): хромосома випадковим чином розбивається на три частини, як показано на рисунку. Міняючи місцями крайні ділянки хромосоми, отримуємо новий розв’язок:

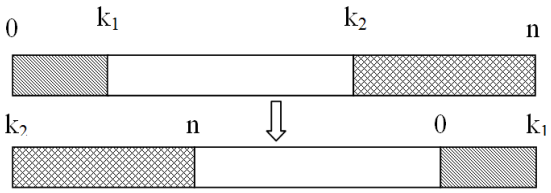

Рисунок 3 – Двоточкове схрещування

Хромосоми, отримані за механізмом двоточкового схрещування, записуються в робочу матрицю з індексами від n до 2n-1.

б) мутація 1: два випадково взяті гени однієї хромосоми міняються місцями (рисунок 4).

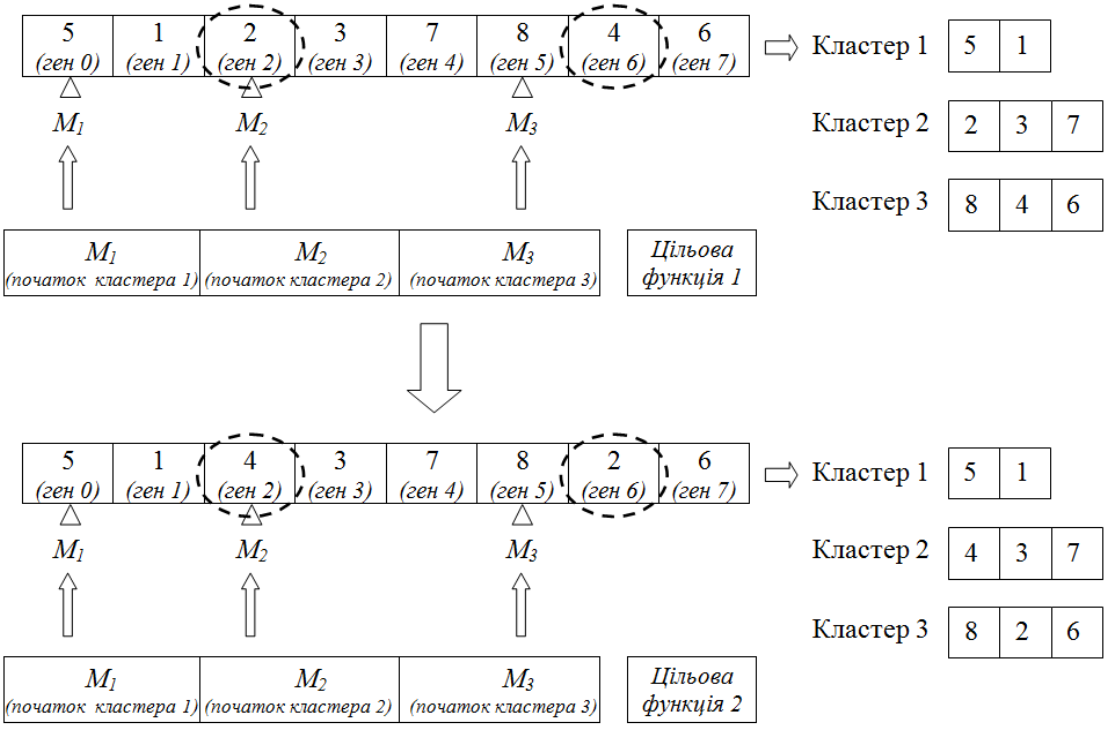

Рисунок 4 – Мутація в межах однієї форми розбиття

Таким чином, ми змінюємо положення лише одного об’єкту відносно кластерів. Цей вид мутації забезпечує мінливість у межах однієї форми розбиття. Мутанти, отримані за першим видом, розташовуються в робочій матриці з індексами від 2n до 3n-1.

в) мутація 2: для сформованої батьківської хромосоми випадковим чином генерується новий масив маркерів розбиття, при цьому порядок слідування об’єктів залишається незмінним (рисунок 5).

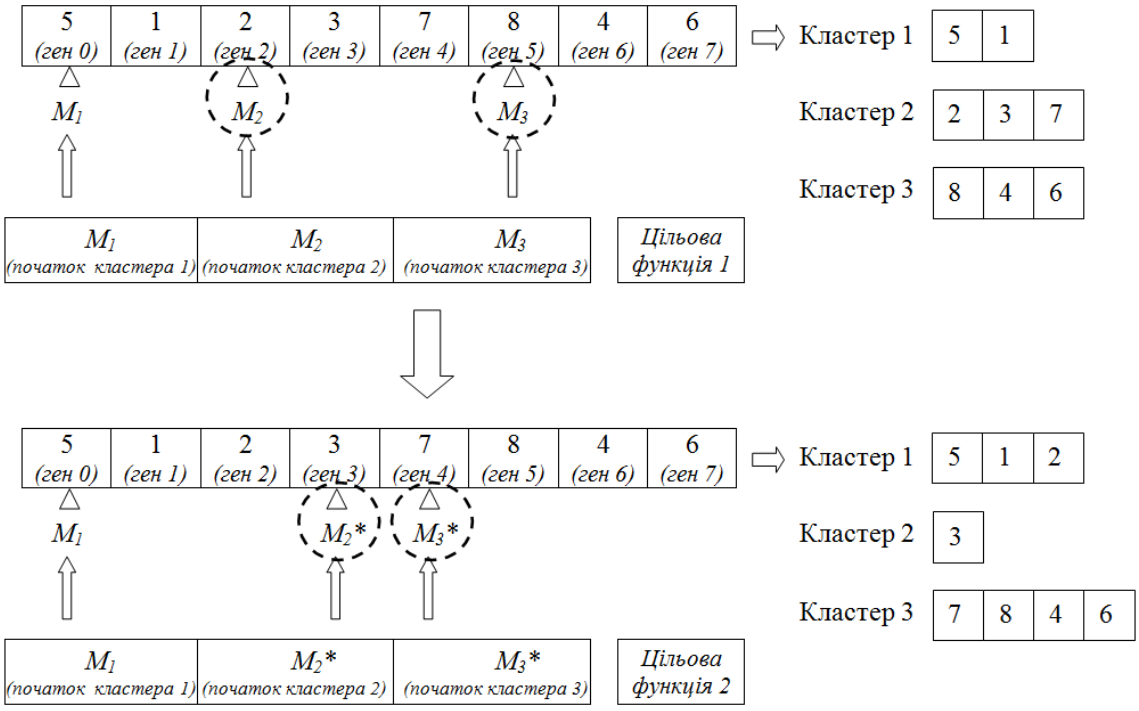

Рисунок 5 – Мутація форм розбиття

Цей вид мутації забезпечує змінність форм розбиття та дозволяє розширити область пошуку оптимального розв’язку. Мутанти за другим видом займають позиції з 3n до 4n-1 робочої матриці.

Для кожного з отриманих таким чином нащадків також підраховується цільова функція.

3) На цьому етапі реалізується механізм природного відбору в межах популяції. Застосуємо стратегію елітизму, яка полягає в тому, що особини з найбільшою пристосованістю гарантовано переходять у нову популяцію. Використання елітизму дозволяє прискорити збіжність генетичного алгоритму. Тому серед 4n особин популяції, хромосоми яких зберігаються в робочій матриці, відбираємо n за критерієм мінімальності цільової функції. Вони формують наступне покоління.

4) Якщо не виконується умова зупинки, очищуємо робочу матрицю та повторюємо попередню процедуру, починаючи з кроку 2. Ітераційний процес вважаємо завершеним, коли поточна сума значень цільової функції для кращих n особин перестає зменшуватись.

Таким чином, із покоління в покоління, сприятливі характеристики розповсюджуються по всій популяції. Схрещування найбільш пристосованих особин приводить до того, що досліджуються найбільш перспективні ділянки простору пошуку. В загальному підсумку популяція буде сходитись до оптимального розв’язку задачі або близького до оптимального. Перевага розробленого генетичного алгоритму полягає в тому, що він знаходить приблизні оптимальні розв’язки за відносно короткий час.

Комп’ютерний експеримент

Для дослідження було взято дані фірми-провайдера телекомунікаційних послуг «Хорс-Телеком». Розглядалася статистика доступу за добу 14.06.2010 р. для 50 користувачів. Для характеристики абонентів було обрано такі показники:

- швидкість передачі даних;
- повний обсяг вхідного трафіку за заданий період часу;
- повний обсяг вихідного трафіку за заданий період часу.

В результаті роботи програми отримуємо варіант розбиття набору вхідних векторів на 4 однорідні групи (таблиця 1):

Таблиця 1 – Результат розбиття множини абонентів на кластери

Номер кластеру	Кількість користувачів	Швидкість доступу	Співвідношення вхідного (Inbound) та вихідного (Outbound) трафіку
1	6	1 - 10 M	
2	23	128 k – 2 M	Inbound > Outbound
3	12	128 k – 1 M	Inbound >> Outbound
4	9	128 k – 1 M	Inbound ≤ Outbound

Таким чином, у вихідних даних можна виділити:

кластер 1 – невеликий сегмент користувачів із найвищою по вибірці швидкістю доступу;

кластер 2 – найбільший; у ньому представлені користувачі з середньою швидкістю доступу, в яких вхідний трафік перевищує вихідний;

кластер 3 – до нього потрапили переважно абоненти з невисокою швидкістю, що використовували лише вхідний напрямок передачі даних;

кластер 4 – складається з абонентів, для яких характерна перевага обсягу вихідного трафіку над вхідним; швидкість доступу переважно невисока.

Як уже зазначалося, дане розбиття є наближеним. Не зважаючи на це, в отриманих даних чітко прослідковується наявність компактних скупчень об’єктів, і на їх основі можна судити про характер логічних закономірностей, закладених у вихідних даних.

Проведемо дослідження часу роботи розробленого генетичного алгоритму порівняно з часом виконання алгоритму граничного перебору для різної кількості об’єктів у вихідній множині. Результати порівняння представимо на графіку, наведеному на рис. 6. По осі абсцис будемо відкладати кількість об’єктів у вихідній множині, по осі ординат – час роботи програми.

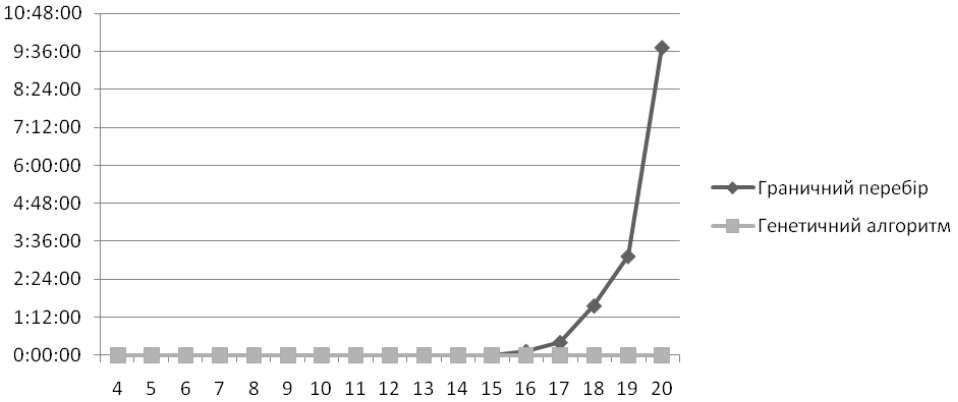

Рисунок 6 – Порівняння алгоритмів за часом виконання

Бачимо, що в задачах невеликої розмірності (до 20 користувачів) генетичний алгоритм за часом виконання показує результат, неспіврозмірно малий порівняно з алгоритмом граничного перебору. Час виконання останнього починає стрімко зростати вже при кількості n=15 користувачів.

При цьому в задачах більшої розмірності генетичний алгоритм поводить себе таким чином, як показано на рис. 7: помітне зростання часу виконання програми спостерігається лише тоді, коли кількість користувачів перевищує 50. Проте навіть для значно більшого обсягу вхідних даних (100 користувачів і більше) час виконання надзвичайно малий порівняно з аналогічним показником для алгоритму граничного перебору.

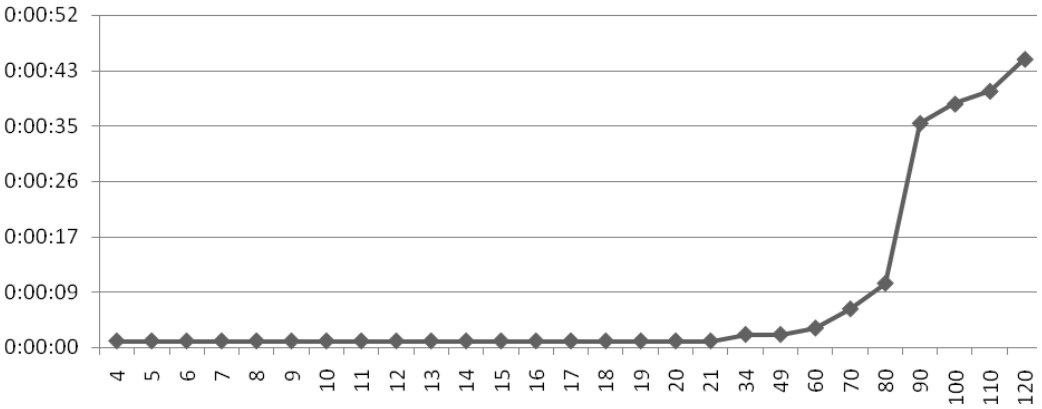

Рисунок 7 – Залежність часу виконання генетичного алгоритму від розмірності задачі

Слід зазначити, що експеримент поставлено на ПЕОМ із середніми на сьогоднішній день характеристиками: AMD Athlon 64 X2, 2,4 ГГц; 2 ГБ ОЗП.

Отримані дані дають змогу стверджувати, що за показником швидкодії генетичний алгоритм кластеризації має значну перевагу над алгоритмом граничного перебору на основі динамічного

програмування. Вже для 21 об’єкта вихідної множини час розв’язання задачі за допомогою методу граничного перебору не вкладається в жодні розумні межі. Водночас розроблений генетичний алгоритм дає розв’язок при значно більшій розмірності задачі за цілком прийнятний час.

Специфіка Інтернет-технологій порівняно з іншими вимагає передусім високої швидкодії програмних засобів, а також можливості нарощування розмірності задачі в широких межах. За таких умов доцільно надати перевагу запропонованому наближеному методу кластеризації.

Висновки

У статті запропоновано підхід для розв’язання задач кластеризації користувачів мережі Інтернет. Розроблено генетичний алгоритм для розв’язання поставленої задачі та відповідне програмне забезпечення для його практичного застосування. Задля врахування специфіки задачі та підвищення ефективності роботи генетичного алгоритму було застосовано неоднорідні хромосоми. Відповідно до цього було внесено суттєві модифікації до перебігу класичних процедур схрещування та мутації.

Розроблений алгоритм було досліджено на швидкодію в порівнянні з алгоритмом граничного перебору та показано значну його перевагу за цим показником.

Подальші дослідження будуть пов’язані з розвитком інтелектуального підходу до розв’язання даної задачі та використанням нечітких множин із метою подолання зашумленості вхідних даних.

Список літератури

1. Муссель К. Предоставление и биллинг услуг связи. Системная интеграция. – М.: Эко-Трендз, 2003. – 319 с.
2. Дюран Б., Оделл П. Кластерный анализ. - М.: Статистика, 1977. – 128 с.
3. Матвеев Ю.Н. Основы теории систем и системного анализа: учебное пособие. – Тверь: ТГТУ, 2007. – 100 с.
4. Мандель И.Д. Кластерный анализ. – М.: Статистика, 1988. – 176 с.
5. Ротштейн А.П. Интеллектуальные технологии идентификации: нечеткие множества, генетические алгоритмы, нейронные сети. – Винница: «УНИВЕРСУМ-Винница», 1999. – 320с.

Відомості про авторів

Захарченко Сергій Михайлович, к.т.н., доцент кафедри ОТ, Вінницький національний технічний університет, Хмельницьке шосе, 95, Вінниця, 21021, Україна, тел.: (0432) 43-70-41.

Кондратенко Наталія Романівна, к.т.н., доцент, професор кафедри ЗІ, Вінницький національний технічний університет, Хмельницьке шосе, 95, Вінниця, 21021, Україна, тел.: (0432) 59-83-79, e-mail: kondrn@yandex.ru.

Манаєва Ольга Олексіївна, студентка, Вінницький національний технічний університет, Хмельницьке шосе, 95, Вінниця, 21021, Україна, тел.: (0432) 56-33-42, e-mail: sleeepy@meta.ua.