УДК 004.942
DOI: 10.15587/1729-4061.2019.170824

Дослідження математичного апарату z-апроксимації функцій для побудови адаптивного алгоритму

О. О. Кряжич, О. В. Коваленко

Проведеними дослідженнями запропоновано математичний апарат та методика побудови адаптивного алгоритму, основаного на Z-апроксимації функцій. Це необхідно для вдосконалення підходів до побудови алгоритмів, які змінюють свою поведінку в залежності від зміни вхідної інформації. Зазначене, у свою чергу, значно покращує результати виконання завдання, що реалізуються за допомогою такого алгоритму. Наприклад, рішення нелінійних задач, опис складних поверхонь, пошуку інформації.

Показано, що отримані на цьому рішення узгоджуються із застосуванням однакових алгоритмів для окремих груп функцій, які використовуються для апроксимації. Ці функції використовуються при побудові напряму для пошуку та дають можливість розробити модель похибки Z-апроксимації з використанням початкових або заключних наближень.

Наведене визначення Z_m-апроксимації, як апроксимації з багатократним зменшенням інтервалу, що призводить до спрощення рекурентних формул і є особливістю представленого підходу. Запропонована методика та базовий алгоритм дозволяють безпосередньо визначати ряд загальних та гіперболічних функцій з використанням Z_m-апроксимацій та паралельних обчислень. За підсумками досліджень представлений адаптивний алгоритм обчислення arctg *x як функції, що є оберненою до* tg *x.*

Представлене може бути використане при створенні адаптивного алгоритму пошуку в масивах неструктурованої та слабо систематизованої інформації. Подібний пошук застосовується для книг та підручників, які були викладені в мережу інтернет у форматах jpeg, pdf, *або у вигляді фрагментів обох форматів. У цьому випадку на основі адаптивного алгоритму розробляється спеціальна модель, реалізація якої може бути виконана за декількома варіантами зі зміною напрямів руху*

Ключові слова: пошуковий алгоритм, розподіл процесів, рекурентний запис, нев'язка, апроксимація

1. Вступ

Пошук інформації для рішення задачі чи відповіді на питання вимагає обґрунтованого вибору сукупності джерел за деякими ознаками. Це витікає із загального розуміння інформації як наукової категорії [1], що визначає відомості, знання або данні за певними властивостями [2]. За допомогою математичної логіки можна вивести такі важливі властивості інформації, як повнота та несуперечність, а своєчасність визначити, як математичну модель події у точці часового відрізку. За схожою схемою відбувається вирішення і математичної задачі, особливо складної, коли є ряд обмежень та існує необхідність округлення чи іншої трансформації даних. Це може викликати виникнення та накопичування похибки. У цьому випадку рішення в певній точці інтервалу, на якому розглядається задача, вважатиметься вірним, коли відповідатиме критерію повноти і несуперечності.

Сам пошук рішення або отримання відповіді на інформаційний запит можна представити як процес вибору масиву даних за двома паралельними напрямами. За першим напрямом відповідь розглядається як число і важливість на даний момент часу одержаних відомостей. Другий напрям дозволяє з використанням визначення загальної міри кількості інформації К. Шеннона і ентропії [3] отримати відповідь, що має прагматичну цінність [1].

В наведеному слід підкреслити цікаву обставину. Кількість результатів та важливість отриманої за цими джерелами інформації на фіксований момент часу можуть принести нерелевантну інформацію при виході за межі часового відрізку. А от прагматична цінність дозволяє побудувати логічний ланцюг стосовно об'єкту запиту, у тому числі, при його розвитку у часі. Особливо гостро це питання відчувається при виконанні задач і алгоритмів, що реалізуються в інтерактивному середовищі, особливо – в мережі Інтернет [4, 5]. Як при вирішенні математичної задачі, так і при дослідженні деякого питання чи проблеми шляхом пошуку відповідей в Інтернеті, існуватимуть обмеження [6], що ускладнюватимуть пошук. У цьому випадку для здійснення пошуку потрібні спеціальні алгоритми, які здатні змінювати поведінку під час виконання в залежності від вхідних даних, тобто – адаптивні алгоритми. Все зазначене вказує на актуальність досліджень, пов'язаних з розробкою адаптивних алгоритмів.

2. Аналіз літературних даних та постановка проблеми

В дослідженні [7] зазначено, що перетворювачами інформації можуть виступати різні системи, які отримують інформацію з навколишнього світу і переробляють її з метою розкриття закономірностей та отримання знань. Набір пошукових образів документів або записів фактів (даних) буде являти собою інформаційний масив [8] великого обсягу [9]. Наведене можна вважати основоположним в інформаційному підході [10], оскільки тут розкривається суть трансформації інформації в дані з акцентом, що це – «великі дані».

У цьому контексті слід зазначити, що відокремленої «проблеми» обробки великих обсягів інформації не існувало. Це розглядалося як закономірний розвиток [11] інформаційних систем, пов'язаний з переведенням технологій збору, передачі, обробки та зберігання інформації на безпаперову основу з метою подальшого надання цієї інформації на запити користувачів. Для задач обробки великих обсягів інформації з урахуванням постійного збільшення швидкості приросту та можливостей використання спеціальних алгоритмів було запропоновано просте рішення. Тобто, слід розподілити процеси та виділити функції інформаційного забезпечення в самостійні інформаційні потоки. Потім – об'єднати ці потоки за певними ознаками окремих даних після обробки та тематично розмістити в базах і розподілених автоматизованих банках даних.

Використаний у [12] підхід базується на критеріях повноти, несуперечності та своєчасності отриманої інформації з точки зору цінності інформації. Повноту інформації можна представити, як міру достатності інформації для вирішення певної задачі або як можливість подати всі значення істини з множини логічних операцій за допомогою формул з елементів цієї множини [2]. Несуперечність можна визначити у якості властивості системи, з якої неможна вивести поняття суперечності [13], коли одна множина параметрів заперечує іншу. У логічному виражені це можна представити як хибу [2] або неоднозначність. Своєчасність може означати надходження інформації в зручний або призначений час [12]. Вона може бути представлена за підходом [14] математичною залежністю інформації від відрізку часу, на якому ця інформація є актуальною. Тоді, враховуючи наведене, цінність інформації можна представити через категорію релевантності. Але при цьому релевантність слід розглядати не просто як відповідність інформації якомусь запиту [15], а як можливість представити повну та несуперечну інформацію в межах зміни в часі.

Саме про таке, називаючи це адаптацією до змін навколишнього середовища будь-якої життєздатної системи, визначається в роботі [16]. Проте, на відміну від [7], де вказується на необхідність чіткого математичного апарату для автоматизації переробки інформації, у [16] наведена схематична візуалізація, без математичного обґрунтування [17].

Але обчислювальна техніка сприймає процес роботи з інформацією у формалізованому вигляді. Саме тому в світі активно використовується підхід [18], який має перевірений математичний апарат для побудови алгоритмів, у тому числі – і пошукових.

При здійсненні пошуку рішення задачі або пошуку певних відомостей в Інтернеті шлях до відповіді не завжди може бути описаний лінійно. Найчастіше відповіді будуть розташовані на якійсь кривій, описаній функцією або рядом функцій. Метою пошуку є знаходження певного набору змінних з переходом між деякими точками-джерелами інформації за певним алгоритмом, який знаходить інформацію визначеної структури даних і залежить напряму від структури даних, для якої він реалізований [19]. Такий пошук може бути реалізований [20] за методом можливих напрямків Дж. Зойтендейка [21], коли отримується деяка кількість полюсних точок, які надають інформацію з дотриманням визначених обмежень. Але бувають задачі, коли обмеження можуть бути різними в залежності від умов її реалізації. Наприклад, при математичному описі опуклої чи вгнутої поверхні, що може бути вирішено за методом Дж. Зойтендейка. Або при здійсненній пошуку в Інтернеті, коли результатом очікується як структурована, так і не структурована або слабо систематизована інформація, і алгоритм повинен перебудувати пошук в залежності від типу даних. У цьому випадку більш універсальним може стати адаптивний алгоритм, який додаватиме нові точки-джерела інформації в залежності від розподілу вхідних даних, у зв'язку з чим цікавість представляє застосування *Z*-апроксимації, як, наприклад, у [22].

Метод *Z*-апроксимації функцій початково базується на адаптивних алгоритмах, здатних змінювати свої функціональні особливості і при цьому надавати змінну точність обчислень. Крім того, особливістю *Z*-апроксимації виступає можливість зміни структурних особливостей самого алгоритму за рахунок початкових та кінцевих наближень та власних параметрів алгоритму.

Аналіз результатів наведених досліджень дозволяє вважати, що існуючі підходи для вирішення задач, які потребують оперування різними за структурою даними, є недостатніми для отримання релевантної відповіді на поставлене питання. Останнім часом ці питання найбільше проявляються в мережі Інтернет. Пов'язано це з накопиченням великих обсягів неструктурованої та слабо систематизованої інформації. А це призводить до обмеження пошуку [6] та ускладнень при формуванні масиву відповідей на запит, неповноти та суперечності таких відповідей.

Зазначена частина проблеми може бути вирішена шляхом організації процесу розв'язання задач на основі побудови моделей для пошуку рішень, що базуються на адаптивних алгоритмах.

3. Мета і завдання дослідження

Метою роботи є аналіз математичного апарату Z-апроксимації функцій для побудови адаптивного алгоритму, тобто алгоритму, який змінює свій хід виконання в залежності від отриманих або уточнених на попередніх етапах виконання вхідних даних. Зокрема, задачі виконання адаптивних алгоритмів виконуються при здійсненні пошуку – знаходження рішення функцій, вибору кращого варіанту (задачі мінімаксу), відповідності отриманих даних поставленим обмеженням при пошуку текстів і т. і.

Для досягнення мети були поставлені такі завдання:

– проаналізувати вимоги, які висуваються до розробки адаптивних алгоритмів;

– довести ефективність застосування *Z*-апроксимації для розробки адаптивних алгоритмів за методикою, що пропонується в роботі;

– навести базову методику створення адаптивного алгоритму на основі *Z*-апроксимації.

4. Вимоги до адаптивних алгоритмів

Припустимо, що побудована деяка модель пошуку рішення, ключові значення якої лежать в різних точках системи координат. Між цими точками-джерелами інформації вірогідні шукані рішення. А перехід для пошуку цих рішень можна представити невеличкими відрізками ламаної кривої, які в результаті згладжування описуються за допомогою тригонометричних і гіперболічних функцій, експоненти, логарифму. Тоді, враховуючи зазначене та особливості алгоритмів [23], здатних змінюватися в залежності від вхідних параметрів, можна викласти основні вимоги до побудови адаптивного алгоритму для реалізації описаної моделі пошуку рішення:

– алгоритм повинен забезпечувати можливість рекурентного запису, тобто, вираховувати значення на основі попередніх членів послідовності;

– константи, що використовуються в таких алгоритмах, повинні бути представлені або малою кількістю цифр, або легко вираховуватися з довільною точністю;

– можливість заміняти початкові або кінцеві наближення – тобто обраний механізм адаптації для різних видів запиту.

Для виконання переліченого можна застосовувати однакові алгоритми для окремих груп функцій, що використовуються. Наприклад, не можна використовувати деякі методи, що мають менше $n/2$ констант з довільною розрядністю для розрахунку прямих та зворотних тригонометричних і гіперболічних функцій, експоненти і логарифму. А вирахування таких констант з довільною точністю вимагає певного часу та відповідних апаратних засобів, що вже не відповідає вимогі щодо роботи з різними апаратними засобами. А от функції, типу y/x, $1/x$, $\sqrt{x}$, $\sqrt[3]{x}$ не вимагають вирахування таких констант і до них може бути застосований один окремий алгоритм розрахунку.

Також слід взяти до уваги, що вимога відносно констант повинна виконуватися як в алгоритмі щодо початкових, так і кінцевих наближень. А виходячи з цього, для виконання поставлених в роботі задач та реалізації адаптивного алгоритму, більш підходять наступні методи:

– рекурентний запис розкладань в ряди. При цьому початкове наближення y_0 і відповідне йому x_0 бажано представити, як константи або вирази, що легко обчислювати;

– рекурентний запис розкладань за многочленами. В деяких випадках є раціональним, хоча не завжди простим при здійсненні програмування такого алгоритму;

– обчислення ланцюгового дробу, включаючи розкладання за нев'язками, що є доволі зручним для представлення у вигляді таблиці бази даних;

– цікавий метод для реалізації пошуку – вирахування значень на основі попередніх членів послідовності для обчислення нескінченних творів;

– методи Z-апроксимації, що також є цікавими для реалізації пошуку, коли використовується заданий великий набір методів для початкових і завершальних наближень і є можливість вибору співвідношення складності рекурентного відношення та початкового або завершального наближення;

– звичайні ітераційні формули, які отримані заздалегідь. Знов таки, зручні тим, що є великий набір методів для отримання ітераційних формул і початкових наближень, що може бути представлено в таблиці бази даних.

Зазначені методи можуть бути реалізовані у алгоритмах послідовно-паралельної архітектури.

Всі наведені методи мають значну перевагу – можливість скоротити час обчислень за рахунок використання змінної точності обчислень. З цієї точки зору найцікавішими є методи Z-апроксимації, де при кожній наступній ітерації збільшується точність розрахунків.

5. Ефективність застосування Z-апроксимації для розробки адаптивних алгоритмів

Щоб довести ефективність застосування Z-апроксимації та адаптивних алгоритмів на цій основі, можна розглянути модель погрішності Z-апроксимації функції з використанням початкових або заключних наближень. В загальному вигляді таку модель можна записати, як

$$\Delta \approx C_n \left(x / N^m \right)^n \approx \left\lceil x / N^{mn+l} \right\rceil,$$

де C_n – константи, які в своїй більшості залежать від параметру n; N – число, яке визначає величину зменшення інтервалу; m – число ітерацій, які виконані за рекурентною формулою; n – порядковий номер члена виразу, який віднімається при виконанні наближень $l = -\log_N |C_n|$.

Базуючись на цій моделі, можна визначити значення параметрів m та n.

Слід зауважити, що для багатьох функцій, типу e^x, $\sin x$, $\cos x$ та подібних, значення C_n надасть значного внеску в зменшення погрішності методу. Але з деяких значень n та m більший вплив отримає величина N^{mn}. Це обумовлюється тим, що, наприклад, у разі, коли коефіцієнт є пропорційним величині $1/(n+m)!$, то з деяких значень n та m функція N^{mn} зростає швидше, ніж $(n+m)!$. Так для $N=2$, $n=4$, $m=4$ буде отримано $(n+m)!=40320$, $2^{mn}=65536$, а при $m=10$, $n=4$, $(n+m)!=87178291200$, $2^{mn}=1099511627776$.

Також слід зазначити, що використання Z-апроксимації еквівалентне багатократному зменшенню інтервалу, пропорційному значенню величини $1/N^m$. Збільшення величини N призводить до ускладнення рекурентних формул, основаних на Z-апроксимації, у зв'язку з чим визначення використовуваної для обраної адаптивної алгоритмізації апроксимацію та сам алгоритм можна вживати з префіксом Z_m. Крім того, зі збільшенням N збільшується можливість розпаралелювання рекурентних відношень на основі Z_m-апроксимації. Для дослідження цього, початкову модель погрішності можна представити у вигляді

$$\Delta \approx C'_n \left(x / \left(N + \Delta N \right)^{m+\Delta m} \right)^{n+\Delta n},$$

де ΔN, Δm, Δn – прирощення відповідних параметрів N, m, n.

Логарифм цього виразу:

$$\ln \Delta \approx \ln C'_n + \left(n + \Delta n \right) \left(\ln x - \left(m + \Delta m \right) \ln \left(N + \Delta N \right) \right).$$

З останнього випливає, що збільшення величин Δm та Δn впливає на зменшення погрішності Δ. Але при послідовних і паралельних розрахунках алгоритми вибору параметрів будуть розрізнятися. Тому слід зауважити, що збільшення параметру m призводить до збільшення кількості ітерацій за рекурентною формулою та зростання погрішності. Збільшення параметру n

призведе до збільшення кількості членів при початковому чи кінцевому наближенні, а також зменшенню C'_n. Тому для кожного фіксованого N необхідно обрати співвідношення між m та n.

Для рекурентних співвідношень слід ввести поняття базової послідовності. Під базовою послідовністю рекурентних співвідношень розуміється послідовність рекурентних формул, p-член якої має параметр n_p більшим, ніж n_{p-1} у виразі $Z(nx_m)=f[Z(x_m)]$, де $x_m=x/n^m$.

Наведемо приклади таких послідовностей.

Для вирахування функції $sin\ x$:

$$Z_{m-1}=\sum_{k=0}^{n}\left(-1\right)^k C_{2n+1}^{2k+1} Z_m^{2k+1}\left(1-Z_m\right)^{n-2k},$$

де

$$Z_m=\sin\left[x/\left(2n+1\right)^m\right],\ m=m_0,\ m_0-1,\ ...,\ 0,$$

звідки $Z_0=\sin x$.

Це найбільш простіше можна представити у загальному вигляді:

$$Z_{m-1}=T_{2n+1}\left(Z_m\right),$$

де $T_{2n-1}(Z_m)$ – многочлен Чебишова.

Для обчислення tg$x$ можна використати метод Ейлера [24]:

$$Z_{m-1}=\frac{nZ_m}{1-}\,\frac{\left(n^2-1\right)Z_m^2}{3-}\,\frac{\left(n^2-4\right)Z_m^2}{5-...}\,\frac{\left(n^2-p\right)Z_m^2}{-2p+1-...},$$

де $Z_m=\operatorname{tg}\frac{x}{n^m},\ Z_0=\operatorname{tg}x.$

Аналогічно можна записати рекурентні співвідношення породжуваних дрібно-раціональних апроксимацій.

Для обчислення ctg$x$:

$$Z_m=\frac{\sum\limits_{k=0}^{[n/2]}\left(-1\right)^k C_n^{2k} Z_{m-1}^{n-2k}}{\sum\limits_{k=0}^{(n-1)/2}\left(-1\right)^k C_n^{2k+1} Z_{m-1}^{n-2k-1}},$$

де $Z_m=\operatorname{ctg}\frac{x}{n^m},\ Z_0=\operatorname{ctg}x.$

Використовуючи [25], можна прийняти твердження щодо помилки округлення: коли для функції y=e^x використовувати рекурентне відношення

$$Z_m = 2Z_{m-1}, \quad m = \overline{n,1},$$

то абсолютна погрішність, що викликана похибкою округлення, буде дорівнювати

$$\Delta_n = 2^n \Delta_0 \left(e^{x/2^n} + \overline{\Delta_0} \right)^{2n-1},$$

де $\left|\overline{\Delta_0}\right| \le \Delta_0$, а величина x визначається з рівняння e^x=$2^p l^x$.

Тобто, погрішність може бути більшою за величину 2^{n-t-1}, де t – розрядність чисел. Якщо замість функції y=e^x розглядати функцію $v=e^x-1$, то величина погрішності не буде більшою за величину $O(n2^{-t})$.

Подібне стосується і функцій cos$x$, ln$x$, arctg$x$. Інші тригонометричні функції не потребують подібних перетворень, що випливає, зокрема, з [26].

Для побудови адаптивних алгоритмів за допомогою Z_m-апроксимації слід ввести математичне представлення Z_m-функції. Під Z_m-функцією розуміється пряме чи зворотне рекурентне співвідношення виду

$$Z_{m+1} = f\left(Z_m\right) \tag{1}$$

або

$$Z_{m-1} = f\left(Z_m\right). \tag{2}$$

У (1) задається деяке початкове наближення Z_0, а функція Z_m виступає шуканою функцією. У випадку (2) задане деяке початкове наближення Z_{m0}, а функція Z_0 виступає шуканою функцією.

Формула (2) може бути отримана з виразу, виду:

$$Z\left(x_m/n\right) = f\left[Z\left(x_m\right)\right] \tag{3}$$

шляхом заміни

$$x_m = x/n^m \tag{4}$$

при визначенні Z_m як

$$Z_m = Z\left(x_m\right), \tag{5}$$

де n – можна представити як базис системи розрахунку.

Формула (5) може бути отримана з виразу

$$Z\left(nx_m\right) = f\left[Z\left(x_m\right)\right] \tag{6}$$

шляхом заміни (4) при прийнятому визначенні (5).

У якості початкового наближення для (2) можна взяти декілька членів розкладення функції Z_m в ряд Тейлора.

Для оцінки абсолютної погрішності методу до формул (1) або (2) слід підставити вираз:

$$Z_m = \overline{Z_m} + \overline{\Delta_m}, \tag{7}$$

де $\overline{Z_m}$ – точне значення Z_m; $\overline{\Delta_m}$ – абсолютна похибка Z_m.

У підсумку отримується оцінка:

$$\overline{\Delta}_{m+1} \le k_1 \overline{\Delta}_m \tag{8}$$

або

$$\overline{\Delta}_{m-1} \le k_2 \overline{\Delta}_m,$$

де k_1, k_2 – константи.

З виразу (8) випливає, що оцінка абсолютної погрішності методу у випадку прямої рекурентної послідовності має вигляд:

$$\overline{\Delta_m} \le k_1^m \overline{\Delta}_0. \tag{10}$$

З виразу (9) можна зробити висновок, що оцінка абсолютної погрішності методу у випадку зворотної рекурентної послідовності має вид:

$$\overline{\Delta_0} \le k_2^m \overline{\Delta}_m. \tag{11}$$

А враховуючи, що існує деяка помилка початкових даних і якась оцінка цієї помилки, яка може бути представлена, як

$$\overline{\Delta}_0 \le p,$$

де p – константа, отримаємо з (10) наступне

$$\overline{\Delta_m} \le k_1^m p, \tag{12}$$

а для зворотної рекурентної послідовності з (11):

$$\overline{\Delta}_0 \le k_2^m p.$$

В загальному випадку Z_m-апроксимацію можна представити у вигляді

$$Z\left(\phi\left(x_m\right)\right) = f\left[Z\left(x_m\right)\right],$$

де $x_m = \phi^{-1}\left(x\right)$, $\phi^{-1}\left(x\right)$ – функція обернена до функції φ(x).

Наприклад, для функції y=ln$x$, виконуючи m разів операцію видобування кореню зі ступенем n, буде отримано $\lim\limits_{m\to\infty} x_m \to 1$, тобто, $\ln x_m = 0$ при $m \to \infty$.

У цьому випадку ln$x$=n^m lnx_m.

Тому можна представити запис як

$$\ln x_{m+1} = n \ln \sqrt[n]{x_m},$$

тобто Z_{m+1}=nZ_m.

Проте у останньому випадку виникає можливість значного накопичення погрішності. Але у випадку, коли n=2, можна використати формулу

$$\ln(1+x) = 2\ln(0 + x / 1 + \sqrt{1+x}).$$

На основі викладеного можна вивести ряд методів для отримання рекурентних формул з метою обчислення ряду функцій із застосуванням Z_m-апроксимації, наприклад, для найбільш застосовуваних cos$x$, sin$x$, a^x та деяких гіперболічних функцій.

6. Базова методика створення адаптивного алгоритму на основі Z_m-апроксимації

Алгоритм буде реалізовано коректно у тому випадку, коли для будь-яких вхідних даних, які належать заданій області, рішення є єдиним і стійким. Формально це можна представити наступним чином.

Приймаємо, що для вирішення задачі існує деякий алгоритм A, який забезпечує отримання значення y, що відповідає вхідним даним x. Тобто:

$$y = A\left(x\right).$$

У випадку, коли вхідні дані були задані з деякою погрішністю Δx, значенню x+Δx буде відповідати значення

$$y + \Delta y = A\left(x + \Delta x\right)$$

або

$$\Delta y = A(x + \Delta x) - A(x).$$

Алгоритм буде визначений, як стійкий за вхідними даними, якщо при $\|\Delta x\| \to 0$ завжди буде забезпечуватися $\|\Delta y\| \to 0.$ В іншому випадку алгоритм буде нестійким за вхідними даними.

Якщо алгоритм нестійкий, то навіть незначні похибки Δx приводитимуть до виникнення значних погрішностей Δy, що у підсумку призведе до спотворення результату. У цьому випадку можна визначити глобальну та локальну стійкість. У разі глобальної стійкості процес буде завжди стійким, незалежно від похибок окремих операцій в процесі реалізації алгоритму. Локальна стійкість спостерігається у тому разі, коли виникає похибка, більша за деяке критичне значення. При алгоритмізації, особливо процесів пошуку інформації, виникають слабко стійкі завдання. Це відповідає випадку $\|\Delta y\| \le M\|\Delta x\|,$ де константа *M* достатньо велика. І хоча алгоритм формально стійкий, похибка може досягти неприпустимо великого значення.

Виконаний аналіз погрішностей Z_m-апроксимацій функцій показав, що у випадку застосування рекурентних формул типу $Z_{m-1}=f(x,Z_m)$, спостерігається стійкість за початковими даними. Погрішність методу з врахуванням початкових даних лежить в межах $|\xi_0| \le M^m L < C,$ де C – константа, M – визначається на основі $|\Delta_{m-1}| \le M|\Delta_m|$, а L – погрішністю початкового наближення $|\xi_{m_0}| \le L.$

Виходячи з отриманих оцінок, для кожного конкретного випадку можна обирати параметри *m* та *n* для отримання необхідної точності у проміжних обчисленнях та ослаблення впливу похибок округлення. Це є теоретичним доказом використання методів Z_m-апроксимацій для побудови адаптивних алгоритмів.

Під адаптивним Z_m-алгоритмом мається на увазі деякий загальний алгоритм, який дозволяє отримувати два та більше окремих алгоритмів. Причому кожен цих алгоритмів є достатньо ефективним для вирішення конкретних задач на основі обчислення Z_m-функції.

Побудова адаптивного Z_m-алгоритму відбувається шляхом додавання/видалення параметрів деяких значень, а також використання декількох видів апроксимацій шуканих функцій.

Як приклад побудови такого алгоритму можна навести наступний алгоритм обчислення *tg x* та *arctg x* з довільною розрядністю.

Використати алгоритм обчислення *arctg x* на основі рекурентної формули

$$Z_{m+1} = Z_m \Big/ \left(1 + \sqrt{1 + Z_m^2}\right), .$$

де $Z_1 = x$, $\operatorname{arc}\operatorname{tg} x \approx 2^m Z_{m+1}$, $Z_m = \operatorname{tg}\left(x/2^m\right)$ є недоцільним з причини розрахунку *m* разів квадратного кореня.

Тому алгоритм обчислення arctg*x* доцільно побудувати як функцію, що є оберненою до *tg x* за ітераційною формулою

$$y_{i+1} = y_i - \sum_{k=0}^{p} (-1)^k Z_i^{2k+1} \Big/ (2k+1), \; Z_i = \frac{tg\, y_i - x}{1 + x tg\, y_i}.$$

Власне Z_m-алгоритм обчислення можна представити наступним чином:

1. Обчислення y_0 – значення arctg*x*, вирахуване з фіксованою розрядністю.

2. *p*=0, *p*=1, *p*=2, *p*=3.

Якщо взяти *p*=0, буде отримана ітераційна формула третього порядку, тобто точність обчислення підвищується. У разі *p*=3, буде отримана ітераційна формула п'ятого порядку і далі порядок може зростати. Але у цьому випадку для обчислення Z_i необхідно буде вичислити зворотну функцію tgy_i з довільним порядком.

3. Зворотна функція tgy_i – в загальному випадку розраховується за допомогою рекурентного відношення

$$Z_{m-1} = T_{2n+1}\left(Z_m\right) \Big/ T_{2n}\left(Z_m\right),$$

де $T_n\left(Z\right)$ – поліном Чебишова;

$$Z_m = tg\left(y_i \big/ N^m\right), \; N = 2n+1, \; n = 0,1,2..., \; Z_0 = \operatorname{tg} x.$$

4. Початкове наближення. Може бути виконане за трьома окремими алгоритмами.

4. 1. В якості початкового наближення z_{m_0} використовується відповідне число членів siny_m та cosy_m, тобто:

$$tg\, Z_m = \frac{\sin y_m}{\cos y_m} = \frac{\sum_{k=0}^{\infty} (-1)^k y_m^{2k+1} \Big/ (2k+1)!}{\sum_{k=0}^{\infty} (-1)^k y_m^{2k} \Big/ (2k)!},$$

де $y_m = y_i \big/ N^m$.

4. 2. При виконанні в цій частині алгоритму послідовного розрахунку *sin* y_m та *cos* y_m, обчислення відбувається за рекурентними формулами:

$$u_{k+1} = -u_k y_m^2 \big/ 2k\left(2k+1\right), \; t_{k+1} = t_k + u_{k+1}, \; u_1 = t_1 = y_m,$$

$$v_{k+1} = -v_k x_m^2 \big/ (2k-1)2k,\ Z_{k+1} = Z_k + v_{k+1},\ Z_1 = v_1 = 1,\ k = 1, 2, \ldots$$

4. 3. Безпосередньо розрахунок початкового наближення Z_{m0} може бути виконаний також у вигляді ланцюгового дробу:

$$Z_{m_0} = tg\, y_m = \frac{y_m}{1-}\,\frac{y_m^2}{3-}\,\frac{y_m^2}{5-}\ldots\frac{y_m^2}{-2n+1}.$$

5. Ще один окремий алгоритм для обчислення arctg$x$ за формулою:

$$arctg\, x = y_0 - arctg\, Z_0 = y_0 - \left(\frac{Z_0}{1+}\,\frac{Z_0^2}{3+}\,\frac{4Z_0^2}{1+}\ldots\frac{n^2 Z_0^2}{+2n+1} \right),$$

де $Z_0 = (tg\, y_0 - x) \big/ (1 + x\, tg\, y_0)$.

5. 1. Наведений ланцюговий дріб може бути вирахуваний на основі рекурентних співвідношень:

$$R_{n-1} = (A_n - B_{n-1}) Z_0^2 \big/ \left((B_n - 2) + R_n \right),\ n = k,\ k-1, .., 2;$$

$$\mathrm{arc\,tg}\, x = y_0 - Z_0 \big/ (1 + R_1),$$

де $A_n = n^2,\ B_n = 2n+1$.

На цьому алгоритм завершено.

Наведений алгоритм дозволяє представити формалізацію задачі пошуку в масиві неструктурованої інформації. Задача може бути поставлена в термінах нечіткої логіки і вирішена через кроки, перехід між якими в системі координат може бути описаний елементарними та/або тригонометричними функціями:

1) Початковою точкою пошуку обираємо назву видання, наприклад, монографія «АВС».

2) Обираємо терми лінгвістичних змінних, які відповідають підмножині *A*.

3) Для кожного терму обираємо значення, яке його найкраще характеризує.

4) Робимо крок до підмножини В та відшукуємо там ідентичні терми, чітко дотримуючись правила комутативності щодо термів з множин *A* і *B*.

5) Здійснюємо пошук та перебір відповідних значень з обраних на етапі 4 характеристик у підмножині *B* з присвоєнням «1» або «0» кожному з отриманих значень.

6) Після отримання екстремальних значень обираємо проміжні значення, кроки до яких можуть відбуватися не по прямій, а описуватися різноманітними функціями.

7) Присвоюємо «1» або «0» кожному з отриманих проміжних значень.

8) Всім значенням присвоюємо відповідні функції стандартних приналежностей.

9) Формуємо інформаційний масив для пошуку.

10) Визначаємо продукційні правила пошуку з перебором значень, як на кроці 5.

При роботі з Google Scholar продукційні правила пошуку можуть бути задані наступним чином:

– Якщо монографія «ABC» = ISBN 978-966-0000-00-0;

– Якщо монографія «ABC» = ББК (номер);

– Якщо монографія «ABC» = УДК (номер);

– Якщо автор Z = Наукова установа Q;

– Якщо наукова установа Q = Рішення Вченої ради (номер, дата),

та інші, які виконуються послідовно, у тому числі із заміщенням лівої та правої частини та перебором окремих даних (зокрема, частин номерів та прізвищ).

Результат пошуку буде обумовлений відповідністю хоча б за одним продукційним правилом або відповідністю частин продукційних правил пошуку.

Фактично, вирішуючи цю задачу, проводиться доповнення нечіткої підмножини A в множині X підмножиною $\neg B$ з функцією приналежності:

$$\mu_{\neg B}(x) = 1 - \mu_{\neg A}(x), \forall x \in X.$$

Функція $\mu_{(A:B)} : X \to [0; n]$ є функцією приналежності підмножини A (B) до базової множини X (Google).

Запропонований підхід дозволяє знаходити відповідні за структурою фрагменти з масиву неструктурованої інформації, де в одному записі одночасно представлена інформація різної структури, що значно підвищує кількість варіантів отримання точної відповіді.

Наведений підхід був реалізований при пошуку посилань на монографію [27] в Google Scholar. Книга була представлена на одному з сайтів в мережі Інтернет у pdf-форматі єдиним файлом, який утримував в собі неструктуровану інформацію у вигляді малюнку з фрагментами текстового поля, де знаходилися текстові символи без смислового змісту. І на цю книгу було зроблено ряд посилань, які не відбивалися в профілі науковців в Google Scholar.

З усього масиву неструктурованої інформації була вибрана для реалізації алгоритму друга сторінка (рис. 1), де розташовані коди УДК, ББК, ISBN – за якими можна чітко ідентифікувати видання як оригінальне, а також прізвища авторів та назва наукової установи, що дозволяє розширити запит для пошуку та додатково підтвердити відповідність результатів запиту у разі, якщо перелічені коди будуть представлені лише як елемент графічної інформації.

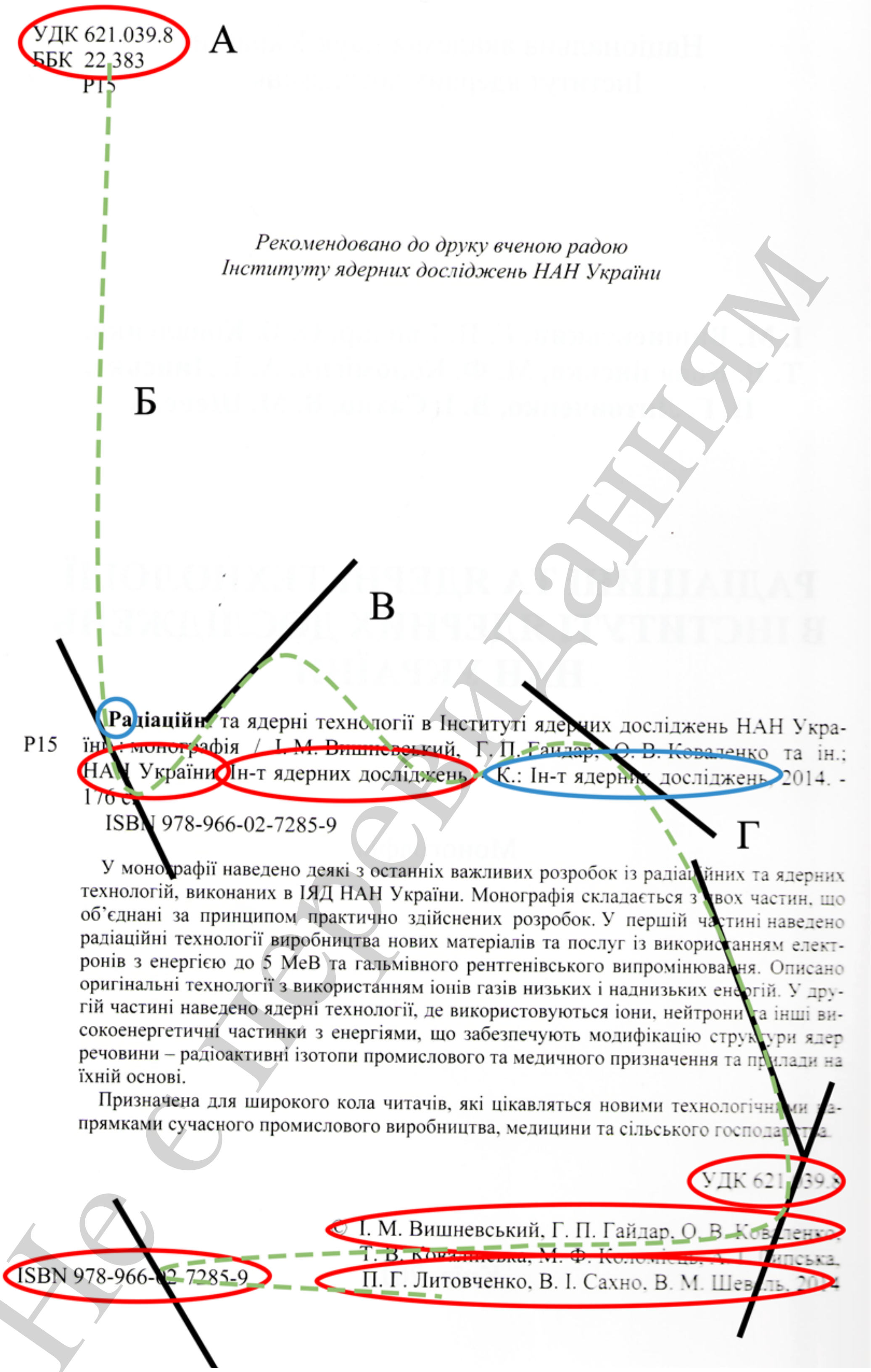

Рис. 1. Алгоритм пошуку в масиві неструктурованої інформації

Перелічене приймається як базові точки (виділено червоним контуром), від верхнього краю сторінки, з початковою точкою А. Сторінка умовно ділиться на прямокутні комірки (клітини). Чим менше сторона комірки, тим відбудеться більш точний пошук за кроками для отримання полюсних точок, які у даному дослідженні представлені словом або фрагментом слова. Пунктирною лінією Б представлена побудована функція за напрямами, представленими векторами В. За цими напрямами в полюсних точках з обраними словами для пошуку обчисляються значення для переходу з повторенням кроків алгоритму. У даному прикладі «від'ємним значенням» виступає відсутність слова чи фрагменту слова в комірці. У цьому випадку вибір слова для формування запиту за напрямом зупиняється.

Далі було сформовано ряд запитів з використанням правил побудови адаптивних алгоритмів, що складалися з слова або коду в кожній базовій точці з словами, словом або фрагментом слова у полюсних точках, що належали області цієї базової точки. З цих запитів була створена вибірка найкращого результату, де були представлені за максимумом слова або коди, які дозволили ідентифікувати відповідність посилання до монографії.

Результат, за яким були знайдені посилання та проведена ідентифікація монографії, представлено контурами Г. За таким алгоритмом було знайдено шість раніше незнайдених посилань.

7. Обговорення результатів дослідження побудови адаптивного алгоритму

Звичайно, для обчислення функцій можна використати відомі розклади тригонометричних функцій, як це зазначено у [25, 26]. Але у разі алгоритмізації та машинного обчислення складних розрахунків з побудовою графіків, виникає задача розбиття кривих на відрізки [21], проведення згладжувань та знаходження точок [20]. У таких випадках слід дотриматися певної точності. І запропонований алгоритм дозволяє застосувати різні підходи до обчислення зі збереженням максимальної точності.

На основі наведеного алгоритму є можливість безпосередньо визначити $\sin x$, $\cos x$, $\mathrm{tg} x$ через визначення $y_m = x$, або $\mathrm{tg} x$ з використанням Z_m-апроксимацій. А от $\mathrm{arctg} x$ за цим алгоритмом можна розрахувати з використанням ітераційної формули для обчислення обернених функцій.

Цей загальний алгоритм можна розширити за допомогою паралельних обчислень $\mathrm{tg} x_m$ як відношення двох сум, а також розпаралелювання обчислення ітераційної формули $\mathrm{arctg} x$.

Аналогічно наведеному можуть бути представлені інші функції, наприклад, гіперболічні.

Наведений математичний апарат значно полегшує реалізацію методу у вигляді алгоритму, на відміну від [26]. Це пояснюється тим, що ряд функцій, які використовуються при розрахунках, можуть бути задані таблично і представлені у базі даних. Тоді при виконанні адаптивного алгоритму частина рішень за напрямом розрахунку, якщо вони задовольняють вхідні умови, береться з таблиці.

В перспективі подібний метод може бути реалізований саме для пошукового алгоритму. Найбільш цікава задача для реалізації такого алгоритму – пошук неструктурованої та слабо систематизовано інформації. Наприклад, при відцифруванні книг, можуть бути отримані документи у форматах jpeg, pdf, або у вигляді фрагментів обох форматів. Подібне значно ускладнює наступний пошук [6]. Це призводить до того, що багато цікавих першоджерел губляться у великих масивах інформації в мережі Інтернет. Адаптивний алгоритм для пошуку таких документів може бути реалізований шляхом розробки спеціальної моделі, реалізація якої може бути виконана за декількома варіантами. Зокрема, на основі методу можливих напрямків Дж. Зойтендейка [21]. Але при використанні цього методу виникають нелінійні обмеження, які можна обійти, знов таки, за допомогою наближень. Тобто, через реалізацію розгалуженого адаптивного алгоритму для вирішення окремих пошукових задач. Тож, в цілому модель пошуку буде представляти собою адаптивний алгоритм, який будуватиме шлях між ключовими словами, заданими для пошуку. Але порівняння проводитиме на основі символів чи лексем, виявлених в точках рішення функції, якою описаний шлях пошуку.

8. Висновки

1. В запропонованому підході визначені та доведені вимоги до адаптивних алгоритмів, які можуть бути практично застосовані у прикладних програмах та веб-розробках. Визначним є можливість застосування однакових алгоритмів для окремих груп функцій, що використовуються для апроксимації при побудові напряму для пошуку.

2. Доведена ефективність застосування Z-апроксимації та адаптивних алгоритмів на цій основі для вирішення задачі здійснення адаптивного пошуку шляхом розробки моделі погрішності Z-апроксимації функції з використанням початкових або заключних наближень. Наведене визначення Z_m-апроксимації, як апроксимації з багатократним зменшенням інтервалу, пропорційному значенню величини $1/N^m$, що є особливістю наведеного підходу.

3. Запропонована базова методика створення адаптивного алгоритму на основі Z_m-апроксимації та на прикладі обчислення tg$x$ та arctg$x$ з довільною розрядністю представлений загальний Z_m-алгоритм. На основі представленого алгоритму показана можливість безпосереднього визначення ряду загальних та гіперболічних функцій з використанням Z_m-апроксимацій та паралельних обчислень.

Література

1. Энциклопедия кибернетики / ред. В. М. Глушков и др. К.: Главная редакция украинской советской энциклопедии, 1974. 1228 с.

2. Філософський енциклопедичний словник. Київ: Абрис, 2002. 742 с.

3. Шеннон К. Работы по теории информации и кибернетике. М.: Иностранная литература, 1963. 832 с.

4. Searching the web: The public and their queries / Spink A., Wolfram D., Jansen M. B. J., Saracevic T. // Journal of the American Society for Information Science and Technology. 2001. Vol. 52, Issue 3. P. 226–234. doi: https://doi.org/10.1002/1097-4571(2000)9999:9999<::aid-asi1591>3.0.co;2-r

5. Figueroa A. Exploring effective features for recognizing the user intent behind web queries // Computers in Industry. 2015. Vol. 68. P. 162–169. doi: https://doi.org/10.1016/j.compind.2015.01.005

6. Ашманов И., Иванов А. Оптимизация и продвижение сайтов в поисковых системах. СПб.: Питер, 2011. 464 с.

7. Глушков В. М. Кибернетика. Вычислительная техника. Информатика. Избранные труды. Т. 3. Кибернетика и ее применение в народном хозяйстве / ред. В. С. Михалевич и др. К.: Наук. думка, 1990. 224 с.

8. Глушков В. М. Кибернетика. Вопросы теории и практики. М.: Наука, 1986. 488 с.

9. Глушков В. М., Стогний А. А., Афанасьев В. Н. Автоматизированные информационные системы. М.: Знание, 1973. 64 с.

10. Академик В. М. Глушков – пионер кибернетики. К.: Издательство Юниор, 2003. 384 с.

11. Глушков В. М. Основы безбумажной информатики. изд. 2-е, испр. М.: Наука, Гл. Ред. физ.-мат. лит., 1987. 552 с.

12. Плескач В. Л, Затонацька Т. Г. Інформаційні системи і технології на підприємствах: підручник. К.: Знання, 2011. 718 с.

13. Encyclopaedia of Mathematics (set) / M. Hazewinkel (Ed.). Springer, 1994.

14. Коваленко О. В. Концептуальні основи створення бази даних наукового експерименту та спостереження // Математичні машини і системи. 2016. № 2. С. 91–101.

15. Diaz F. Autocorrelation and Regularization of Query-Based Retrieval Scores. Chap. 3: PhD thesis. Amherst, 2008.

16. Beer S. Brain of the Firm, Allen Lane. London: Herder and Herder, 1972. 416 p.

17. Beer S. Designing Freedom. House of Anansi Press, 1993. 110 p.

18. Algorithms and Data Structures. URL: https://people.inf.ethz.ch/wirth/AD.pdf

19. Knuth D. E. The Art of Computer Programming. Vo. 1. Fundamental Algorithms. 3-rd ed. Reading, Massachusetts: Addison-Wesley, 1997. 664 p.

20. Кряжич О. О., Коваленко О. В., Іванченко В. В. Спосіб опису забрудненої території: програмна реалізація // Математичне моделювання в економіці. 2016. № 2. С. 22–35.

21. Zoutendijk G. Methods of feasible directions. A study in linear and non-linear programming. Elsevier Pub. Co., 1960. 178 p.

22. A Resolver-to-Digital Conversion Method Based on Third-Order Rational Fraction Polynomial Approximation for PMSM Control / Wang S., Kang J., Degano M., Buticchi G. // IEEE Transactions on Industrial Electronics. 2019. Vol. 66, Issue 8. P. 6383–6392. doi: https://doi.org/10.1109/tie.2018.2884209

23. Knuth D. E. The Art of Computer Programming. Vol. 3: Sorting and Searching. 2-nd ed. Reading, Massachusetts: Addison-Wesley, 1998. 780 p.

24. Эйлер Л. Интегральное исчисление. Т. 1. М.: ГИТТЛ, 1956. 416 с.

25. Бахвалов Н. С., Жидков Н. П., Кобельков Г. М. Численные методы. 6-е изд. М.: БИНОМ. Лаборатория знаний, 2008. 636 с.
26. Попов Б. А., Теслер Г. С. Приближение функций для технических приложений. К.: Наукова думка, 1980. 352 с.
27. Радіаційні та ядерні технології в Інституті ядерних досліджень НАН України: монографія / Вишневський І. М., Гайдар Г. П., Коваленко О. В. та ін. К.: Ін-т ядерних досліджень, 2014. 176 с.