

УДК: 004.043

О.М. Трофимчук, О.О. Кряжич

МЕТОД ОБРОБКИ НЕСТРУКТУРОВАНОЇ ІНФОРМАЦІЇ НА ВЕБ-РЕСУРСАХ

Трофимчук Олександр Миколайович

Інститут телекомунікацій і глобального інформаційного простору НАН України,
м. Київ,

itgis@nas.gov.ua

Кряжич Ольга Олександрівна

Інститут телекомунікацій і глобального інформаційного простору НАН України,
м. Київ; Тернопільський національний технічний університет імені Івана Пулюя,

economconsult@gmail.com

З часом стає складно знайти інформацію в мережі Інтернет, яка зацікавила раніше, навіть якщо відомо, на якому ресурсі вона була розміщена. Пошук відбувається за заданими параметрами, якщо інформація, що знаходиться на ресурсі, є структурованою та систематизованою. Якщо ж це не так, пошук значно ускладнюється, часто й уповільнюється, через що користувач перериває сесію пошуку та оновлює параметри вибору. Тому багато інформації, що знаходиться в мережі, не потрапляє до результатів запитів користувачів. Задачею цієї статті є представлення методу обробки обсягу неструктурованої неперервної, дискретизованої, регулярної та випадкової інформації на веб-ресурсах у вигляді чітко структурованого набору розподілених даних. У роботі аналізується можливість формування масиву інформації як кількості точок — джерел інформації. У цьому випадку для реалізації процесу структуризації найбільш універсальним є адаптивний алгоритм, який додаватиме нові точки — джерела інформації для її обробки веб-сервісом або для пошуку в масивах неструктурованої та слабо систематизованої інформації в залежності від розподілу вхідних даних. Зокрема, пропонується Z-перетворення, оскільки методи Z-апроксимації початково базуються на адаптивних алгоритмах, здатних змінювати свої функціональні особливості та при цьому надавати змінну точність обчислень. Особливістю вирішення поставленої задачі є приведення неперервної, дискретизованої, регулярної та випадкової інформації з її обробкою в цьому процесі до необхідного формату, що математично можна описати окремими функціями, які й використовуватимуться в алгоритмах обробки. Зазначене може бути використане для розробки веб-сервісів обробки інформації для довідкових, пошукових, рекомендаційних систем та платформ дистанційного навчання, а також для вдосконалення алгоритмів обробки та відображення інформації для прикладного програмного забезпечення роботи браузерів.

Ключові слова: алгоритм, перетворення, розкладання, формалізація, точність, сервіс, живучість, параметр.

Вступ

Інформація в мережі Інтернет зберігається нескінченно довго [1], але цікавить користувача приблизно протягом шести годин з моменту публікації [2]. Після цього вона не зникає, але перестає відображатись у топових новинах пошукових машин. І навіть при створенні уточненого запиту на пошук можуть бути видані матеріали, серед яких не завжди буде посилання на шукане. Виникає пи-

тання «Як швидко знайти інформацію, яка зацікавила раніше?» Споживач інформації часто має класичні умови [3], коли мета, обмеження та наслідки дій, виконаних на основі отриманої інформації, невідомі. І простим веб-серфінгом [4] це питання вирішити неможливо, бо пошукові та довідкові системи видають першими посилання на рейтингові ресурси або нещодавно оновлену і змінену інформацію зі спеціалізованих довідкових і пошукових сервісів.

Якщо ж припустити, що ситуація змінилася, користувач задає новий пошук, але до інформаційних масивів пошукових та довідкових систем необхідні зміни і доповнення не вносяться. Такий розвиток взаємодії користувача та машини щодо пошуку релевантної інформації можна представити за Д.О. Поспеловим [5], а саме — використати його погляди на ситуаційне управління, коли інформаційний контекст побудовано двома системоутворюючими компонентами — ситуаційною моделлю об'єкта та алгоритму виділення і порівняння ознак розвитку ситуації. А в залежності від того, яка це система — пошуку чи отримання довідки, — відповідно можна використати «лабіринтну» гіпотезу мислення у вигляді прямого перебору варіантів ситуації, щоб знайти відповідь на запитання з максимально можливим забезпеченням повноти, несуперечності та своєчасності, або «модельну» гіпотезу з перебором комбінацій ознак для вибору ознак, які найкраще підходять за запитом користувача і можуть бути визначені як релевантні дані. У цьому випадку живучість сервісу обробки інформації, реалізованого у мережі Інтернет на веб-платформі, можна визначити як виконання певного алгоритму, що повинен вибрати з певних масивів інформації ті дані, які максимально відповідають запиту користувача. Обрання такого алгоритму передбачає дослідження множин щодо ознак, властивостей та критеріїв. При цьому формування алгоритмічного базису для вирішення конкретної задачі залежить від різноманітності алгоритмів, які представлено у інформаційній системі, масовості набору алгоритмів для виконання різноманітних задач та можливості їх адаптації до умов використання. Як зазначається у Г.С. Теслера [6], останні три таксони є основою класифікації алгоритмічного базису як фактора процесу обчислень і можуть виступати критеріями оцінки живучості таких інформаційних систем. Зазначене можна продемонструвати на прикладах Вікіпедії, яка є інформаційно-довідковою системою, або бібліографічної бази даних Google Scholar, яка є інформаційно-пошуковою системою. Живучість таких систем оцінюється частотою оновлення та актуалізації інформації.

Проте вказані особливості реалізуються лише за умов, що інформація у довідковій чи пошуковій системі чітко структурована та систематизована. Якщо ж це не так, отримати адекватні результати можна лише за умов обробки інформації та її приведення до структурованого набору даних.

Постановка задачі

Задача роботи — представити можливий метод обробки неструктурованої, неперервної, дискретизованої, регулярної та випадкової інформації на веб-ресурсах у вигляді чітко структурованого набору розподілених даних. Це можна пояснити як перетворення інформації в масив даних за адаптивною схемою за двома паралельними напрямками. Перший — кількість інформації розглядається саме як число і важливість одержуваних відомостей із їхніми семантичними і прагматичними аспектами. Це — розуміння обсягу інформації з точки зору одержувача. Другий напрямок — використовуючи визначення загальної міри кількості інформації К. Шеннона і ентропію [7], що дозволяє дослідити технічні питання кодування, передачі та зберігання. Цей аспект дозволяє зрозуміти обсяг одержуваної інформації та пов'язати його з поведінкою отримувача, який вирішує якісь завдання, що призводить до розуміння кількості семантичної інформації, яка слугує

для оцінки прагматичної цінності та отримується і переробляється в масив матеріалу [8]. Масив інформації можна розглянути як кількість полюсних точок та точок — джерел інформації, представлених із використанням методу Дж. Зойтендейка [9, 10], які надають дані для побудови адекватної моделі. У такому разі для реалізації запропонованого найбільш універсальним виступатиме адаптивний алгоритм, який додаватиме нові точки — джерела інформації для її обробки веб-сервісом або для пошуку в масивах неструктурованої та слабо систематизованої інформації в залежності від розподілу вхідних даних. І більш ефективним виглядає Z-перетворення [11], оскільки методи Z-апроксимації початково базуються на адаптивних алгоритмах, здатних змінювати свої функціональні особливості та при цьому надавати змінну точність обчислень. Ще одна цікава особливість Z-апроксимації полягає в тому, що можна змінювати структурні особливості самого алгоритму за рахунок початкових та кінцевих наближень та власних параметрів алгоритму. Для вирішення поставлених задач це цікаво з точки зору того, що, наприклад, змінивши деякі параметри $N: a, b, t$ та інші для алгоритмів $\cos x, \sin x, a^x$ і подібних, можна отримати зміни початкового або заключного алгоритму за допомогою багаточленів або розкладення в ряд Тейлора, нев'язок та дробових розкладень.

Вирішення поставленої задачі

Особливістю вирішення поставленої задачі є розподілення неперервної, дискретизованої, регулярної та випадкової інформації з її обробкою в цьому процесі до отримання необхідного формату, що математично можна описати окремими функціями, які й використовуватимуться в алгоритмах обробки.

Будемо вважати, що непушта множина об'єктів не задовольняє вимогам заданого пошуку, зокрема: немає визначеної ієрархічної структури скінченної множини понять щодо предмета дослідження, існує деяка вільна інтерпретація понять і відношень, функції інтерпретації не формалізовані, аксіоми не визначені. Тобто масив інформації необроблений, і до початку формалізації слід вирішити задачу забезпечення структури інформації таким чином, щоб вона відповідала вимогам інформаційної системи та могла бути представлена у вигляді певних залежностей.

У такому випадку обробку інформації та формалізацію можна провести за допомогою адаптивних алгоритмів [12], які дозволяють подавати та структурувати інформацію за певними правилами. Для цього спочатку виконується вибірка інформації, яка явно чи неявно стосується теми запиту, а потім проводяться функціональні перетворення та застосовуються методи породжуваних алгоритмів [13, 14] у системі генерування алгоритмів [15] із використанням нечіткої логіки, подальшої формалізації з застосуванням одного з базових методів наближення для отримання результату за максимумом чи мінімумом відповідності.

За такого підходу в кінцевому підсумку веб-сервіс обробки інформації буде утримувати не просто масиви інформації, структуровані за тематикою, а:

- сформовані масиви інформації на засадах символічних перетворень;
- використання як загальних, так і окремих схем виведення інформації на запит;
- переформулювання задач і запитів для виведення максимально повної та різної за структурою інформації за запитом.

Базуючись на зазначеному, можна викласти основні вимоги до побудови адаптивного алгоритму для побудови веб-сервісу обробки неструктурованої та слабо систематизованої інформації, враховуючи, зокрема, вимогу, що такі алгоритми повинні стабільно працювати на різній за характеристиками техніці серверів і адекватно функціонувати у різних браузерах:

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

— константи, що використовуються в таких алгоритмах, повинні або бути представлені малою кількістю цифр, або легко вираховуватись із довільною точністю;

— повинна забезпечуватись можливість заміняти початкові або кінцеві наближення, тобто обраний механізм адаптації для різних видів запиту.

Для виконання переліченого можна застосовувати однакові алгоритми для окремих груп використовуваних функцій. Наприклад, не можна використовувати деякі методи, що мають менше $n/2$ констант із довільною розрядністю, для розрахунку прямих та зворотних тригонометричних і гіперболічних функцій, експоненти і логарифму. А вирахування таких констант із довільною точністю вимагає певного часу та відповідних апаратних засобів, що вже не відповідає вимозі щодо роботи з різними апаратними засобами. А 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 виконуватись в алгоритмі щодо як початкових, так і кінцевих наближень. Але це матиме вплив на швидкодію засобів обчислювальної техніки фізичного сервера, а у підсумку опосередковано — і на живучість інформаційної системи, реалізованої на веб-платформі.

Знання про об'єкт як основа адаптивного алгоритму

Сукупність взаємопов'язаних засобів формального визначення інформації та засобів маніпулювання цими визначеннями являє собою те, що складає термін «база знань». Від того, наскільки повно визначено знання про об'єкт, процес та предметну галузь із позиції отримання максимально повної відповіді на запит, залежать функціональні можливості веб-сервісу.

Якщо визначити C_r як реальні (перевірені, формалізовані, структуровані) дані, об'єкт, процес та предметну галузь із позиції функції мети роботи веб-сервісу, а через C_p виразити поточну інформацію про об'єкт, процес та предметну галузь, яку отримано в результаті моніторингу на якийсь момент часу t , то можна отримати залежність

$$C_r - C_p = \Delta m_t, \quad (1)$$

що можна пояснити як залежність повноти бази знань від інформації про стан об'єкта, процесу та предметної галузі в конкретний момент часу. Звичайно, система обробки інформації в Інтернет-середовищі буде живучою у разі $\Delta m_t \rightarrow 0$.

Проте, як впливає з (1), первинна інформація поступатиме неформалізованою і неструктурованою (або слабо систематизованою). Неформалізована інформація N , яка поступатиме до веб-сервісу обробки, повинна бути перетворена на формалізовані дані F , проте з породженням певної неадекватності:

$$C_r - (N + F) = \Delta m_t. \quad (2)$$

І знову виконуватиметься вимога живучості, коли $(N + F) \rightarrow C_r$, а показник $\Delta m_t \rightarrow 0$.

Поєднання формалізованих знань із неформалізованими знаннями у сервісах, реалізованих у Інтернет-середовищі, дозволяє отримати рішення проблемної задачі — забезпечення повної, несуперечної і своєчасної інформації про стан об'єкта чи системи на конкретний момент часу. Таку інформаційну систему можна розглянути в залежності від наявних фактів. Тоді залежність (2) можна навести так:

$$Cr - \left(\sum_{j=1}^n N_j + F \right) = \Delta m_t, \quad (3)$$

де n — кількість вибірок із масиву інформації, який характеризує подію j .

Але формула (3) може бути представлена і так:

$$Cr - \left(\sum_{j=1}^n N_j \right) = \Delta m_t, \quad (4)$$

що означає неможливість формалізації інформації на момент часу t , наприклад, через відсутність такої інформації.

Слід врахувати ще один випадок формалізації інформації на основі (3):

$$\sum_{k=1}^L (C_k - N_k) = \Delta m_t. \quad (5)$$

Вона виникає при надходженні запиту, відповідь на який може представляти собою ієрархію (наприклад, із представленням гіперпосилань на інші варіанти відповіді або додаткові інформаційні ресурси). L — кількість рівнів ієрархії відносно деякої події k . Тоді на вищому рівні ієрархії інформацію буде представлено більш релевантною, ніж на нижчому рівні, де інформація буде обмеженою. Наведену залежність (5) можна назвати частковою формалізацією, а виправляється вона шляхом відсіювання нерелевантної інформації за кожним рівнем ієрархії та встановлення додаткових залежностей.

Відповідно до (1) повна формалізація інформації може бути представлена так:

$$C_r - F = 0. \quad (6)$$

Наведені залежності (1)–(6) дозволяють також зазначити, що для їх досягнення можна в кожному окремому випадку використовувати різноманітні функції (прямі та зворотні тригонометричні, гіперболічні, експоненту, логарифм), які в підсумку дозволять побудувати зв'язки між окремими точками — джерелами інформації, що були задані при виконанні пошуку (ключові слова).

Визначення вимог до адаптивних алгоритмів для побудови веб-сервісу

Для виконання поставленої в роботі задачі з реалізації методу обробки неструктурованої інформації більш підходять наступні математичні інструменти завдяки наявним у них перевагам:

- рекурентний запис розкладань в ряди; для основних елементарних і деяких спеціальних функцій найбільш загальним уявленням є розкладання в ряди нев'язок, при цьому початкове наближення y_0 і відповідне йому x_0 бажано представити у вигляді констант простого виду або легко обчислюваних виразів;

- рекурентний запис розкладань за багаточленами; у деяких випадках є раціональним (наприклад, із застосуванням багаточленів Ерміта), хоча не завжди простим при здійсненні програмування такого алгоритму;

- обчислення ланцюгового дробу, включаючи розкладання за нев'язками, що досить зручно для представлення у вигляді таблиці бази даних;

- цікавий метод для реалізації на веб-сервісі обробки інформації — вираховування значень на основі попередніх членів послідовності для обчислення нескінченних творів та їх приведення до необхідної структури з метою адекватного відображення на сторінці;

— методи Z -апроксимації, що також цікаві для реалізації на веб-сервісі, коли використовується заданий великий набір методів для початкових і завершальних наближень і є можливість вибору співвідношення складності рекурентного відношення та початкового або завершального наближення, що можна використовувати для представлення неявної інформації та мікроданих;

— звичайні ітераційні формули, отримані заздалегідь, наприклад при розкладанні за нев'язками; знову-таки, зручні тим, що існує великий набір методів для отримання ітераційних формул і початкових наближень, що може бути представлено в таблиці бази даних.

Зазначене може бути реалізовано у алгоритмах послідовно-паралельної архітектури.

Для виконання поставлених задач також можна використовувати рішення відповідних диференціальних та інтегральних рівнянь, проте тут не виконується вимога за часом та точністю обчислень. Ітераційні методи, як, наприклад, «цифра за цифрою», не вимагають вираховування занадто складних констант, але дещо програють у точності або у часі, як і динамічні методи рахунку.

Наведені методи мають значну перевагу — можливість скоротити час обчислень за рахунок використання змінної точності обчислень. Із цієї точки зору найцікавішими є методи Z -апроксимації, у яких при кожній наступній ітерації збільшується точність розрахунків.

Варто дещо зазначити про метод вираховування значень на основі попередніх членів послідовності для обчислення нескінченних творів, який дозволяє здійснювати обчислення зі змінною точністю без приведення до заданого інтервалу, що скорочує час обробки інформації.

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

$$\Delta \approx C_n (x / N^m)^n \approx [x / N^{mn+l}],$$

де C_n — константи, які в своїй більшості залежать від параметру n ; N — число, яке визначає величину зменшення інтервалу; m — кількість ітерацій із використанням рекурентної формули; n — порядок члена виразу, що відкидається при апроксимації під час початкового або заключного наближення рекурентного співвідношення $l = -\log_N |C_n|$.

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

Також слід зауважити, 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 апроксимації та сам алгоритм можна вживати з префіксом Z_m . Цей префікс вперше було застосовано у [16], проте без наведення визначення та пояснення причин введення.

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

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

де ΔN , Δm , Δn — приращення відповідних параметрів N , m та n .

Логарифм цього виразу має такий вигляд:

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

Із останнього випливає, що збільшення величин Δm та Δn позначається на зменшенні погрішності Δ . Але при послідовних і паралельних розрахунках алгоритми вибору параметрів будуть розрізнятися. Тому слід зауважити, що збільшення параметра m призводить до збільшення кількості ітерацій за рекурентною формулою та зростання погрішності. Збільшення параметра n призведе до збільшення кількості членів при початковому чи кінцевому наближенні, а також до зменшення C'_n . Тому для кожного фіксованого N необхідно обрати співвідношення між m та n .

Z_m -функція для максимального наближення отриманого значення

Для побудови алгоритму обробки неструктурованої інформації за допомогою Z_m -апроксимації слід ввести математичне представлення Z_m -функції. Під Z_m -функцією розуміється пряме чи зворотне рекурентне співвідношення виду

$$Z_{m+1} = f(Z_m) \quad (7)$$

або

$$Z_{m-1} = f(Z_m). \quad (8)$$

У (7) задається деяке початкове наближення Z_0 , а функція Z_m виступає шуканою функцією. У випадку (8) задане деяке початкове наближення Z_{m_0} , а функція Z_0 виступає шуканою функцією.

Формула (8) може бути отримана з виразу виду

$$Z(x_m / n) = f[Z(x_m)] \quad (9)$$

шляхом заміни у (9)

$$x_m = x / n^m \quad (10)$$

при визначенні Z_m як

$$Z_m = Z(x_m), \quad (11)$$

де n можна представити як базис системи розрахунку.

Формула (11) може бути отримана з виразу

$$Z(nx_m) = f[Z(x_m)] \quad (12)$$

шляхом заміни (10) при прийнятому визначенні (11).

Як початкове наближення для (8) можна взяти декілька членів розкладення функції Z_m в ряд Тейлора.

Для оцінки абсолютної погрішності методу (12) до формули (7) або (8) слід підставити вираз

$$Z_m = \bar{Z}_m + \bar{\Delta}_m, \quad (13)$$

де \bar{Z}_m — точне значення Z_m , $\bar{\Delta}_m$ — абсолютна похибка Z_m .

У підсумку з виразу (13) отримуємо оцінку

$$\bar{\Delta}_{m+1} \leq k_1 \bar{\Delta}_m \quad (14)$$

або

$$\bar{\Delta}_{m-1} \leq k_2 \bar{\Delta}_m, \quad (15)$$

де k_1, k_2 — константи.

Із виразу (14) випливає, що оцінка абсолютної погрішності методу у випадку прямої рекурентної послідовності має такий вигляд:

$$\bar{\Delta}_m \leq k_1^m \bar{\Delta}_0. \quad (16)$$

Із виразу (15) можна зробити висновок, що оцінка абсолютної погрішності методу у випадку зворотної рекурентної послідовності має вигляд

$$\bar{\Delta}_0 \leq k_2^m \bar{\Delta}_m. \quad (17)$$

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

$$\bar{\Delta}_0 \leq p,$$

де p — константа, отримаємо з (16) наступне:

$$\bar{\Delta}_m \leq k_1^m p, \quad (18)$$

а для зворотної рекурентної послідовності з (17) отримаємо

$$\bar{\Delta}_0 \leq k_2^m p. \quad (19)$$

У загальному випадку, враховуючи (18) та (19), Z_m -апроксимацію можна представити у вигляді

$$Z(\varphi(x_m)) = f[Z(\varphi(x_m))],$$

де $x_m = \varphi^{-1}(x)$, $\varphi^{-1}(x)$ — функція, взаємно обернена до функції $\varphi(x)$.

Наприклад, для функції $y = \ln x$ після виконання m разів операції видобування кореня $\sqrt[n]{}$ буде отримано $\lim_{m \rightarrow \infty} x_m \rightarrow 1$, тобто $\ln x_m = 0$ при $m \rightarrow \infty$.

У цьому випадку $\ln x = n^m \ln x_m$.

Тому можна представити запис як $\ln x_{m+1} = n \ln \sqrt[n]{x_m}$, тобто $Z_{m+1} = nZ_m$.

Проте в останньому випадку виникає можливість значного накопичення погрішності. Але у випадку, коли $n = 2$, можна використати формулу $\ln(1+x) = 2 \ln(1+x/1+\sqrt{1+x})$.

Висновок

У роботі представлено один із варіантів реалізації методу обробки неструктурованої інформації за допомогою ряду математичних інструментів у адаптивному алгоритмі. Це стосується алгоритмів, які дозволяють перетворювати великі обсяги неструктурованої неперервної, дискретизованої, регулярної та випадкової інформації на веб-ресурсах у структурований набір даних. Доведено ефективність застосування Z -апроксимації для вирішення задачі забезпечення живучості веб-сервісу шляхом розробки моделі погрішності Z -апроксимації функції з викорис-

танням початкових або заключних наближень при відборі окремих неструктурованих даних у структурований масив інформації. Наведено визначення Z_m -апроксимації як апроксимації з багатократним зменшенням інтервалу, пропорційного значенню величини $1/N^m$, при покроковому переборі неструктурованих даних.

Зазначене може використовуватись при розробці веб-сервісів обробки інформації для довідкових, пошукових, рекомендаційних систем та платформ дистанційного навчання, а також для вдосконалення алгоритмів обробки та відображення інформації для прикладного програмного забезпечення роботи браузерів.

O. Trofymchuk, O. Kryazhych

METHOD FOR PROCESSING UNSTRUCTURED INFORMATION ON WEB RESOURCES

Oleksandr Trofymchuk

Institute of Telecommunications and Global Information Space of the NASU, Kyiv,
itgis@nas.gov.ua

Olga Kryazhych

Institute of Telecommunications and Global Information Space of the NASU, Kyiv,
Ternopil Ivan Puluj National Technical University,
economconsult@gmail.com

Over time, it becomes more difficult to find information on the Internet that users were interested in earlier, even if the user knows on which resource it was posted. The search is performed adequately to the specified parameters, if the information that is on the resource is structured and systematized. Otherwise, the search becomes much more complicated. Therefore, a lot of information that is currently available on the network is not received when displaying results for user requests. The task of this paper is to present to implement a method for processing the volume of unstructured continuous, discretized, regular and random information on web resources into a clearly structured set of distributed data. The paper analyzes the possibility of forming an array of information as the number of points-sources of information. In this case, to implement the process of structuring information, the most universal will be an adaptive algorithm that will add new points-sources of information for its processing by the web service or when searching in arrays of unstructured and poorly systematized information, depending on the distribution of input data. In particular, the Z-transformation is proposed, since Z-approximation methods are initially based on adaptive algorithms that can change their functional features and at the same time provide variable accuracy of calculations. A special feature of solving this task is the distribution of continuous, discretized, regular and random information with its processing in this process to the required format, which can be mathematically described by separate functions that will be used in processing algorithms. This can be used in the development of web-based information processing services for reference, search, recommendation systems and distance learning platforms, as well as used to improve information processing algorithms and display information for browser application software.

Keywords: algorithm, transformation, decomposition, formalization, accuracy, service, survivability, parameter.

REFERENCES

1. «Characterization of Specifications». Characterization of proposed standards. IETF. January 2014. sec. 3. DOI:10.17487/RFC7127. RFC 7127. Retrieved March 11, 2016.
2. Ашманов И., Иванов А. Оптимизация и продвижение сайтов в поисковых системах. СПб. : Питер, 2011. 464 с.
3. Беллман Р., Заде Л. Принятие решений в расплывчатых условиях. *Вопросы анализа и процедуры принятия решений*: сб. пер. М. : Мир, 1976. С. 172–215.
4. Новые слова и значения. Словарь-справочник по материалам прессы и литературы 90-х годов XX века. СПб. : Дмитрий Буланин, 2014. 1360 с.
5. Поспелов Д.А. Ситуационное управление: теория и практика. М. : Наука, 1986. 288 с.
6. Теслер Г.С. Новая кибернетика. Киев : Логос, 2004. 404 с.
7. Шеннон К. Работы по теории информации и кибернетике. М. : Иностранная литература, 1963. 832 с.
8. Энциклопедия кибернетики: в 2 т. / Под. ред. В.М. Глушкова и др. Киев : Главная редакция Украинской Советской энциклопедии, 1974. 1228 с.
9. Трофимчук О.М., Кряжич О.О. Алгоритм опису яружних цільових функцій. *Штучний інтелект*. 2015. № 1–2 (67–68). С. 190–199.
10. Трофимчук О.М., Кряжич О.О. Апроксимація функцій для створення алгоритму опису пересіченої місцевості. *Системні дослідження та інформаційні технології*. 2016. № 1. С. 134–141.
11. Рабинер Л., Гоулд Б. Теория и применение цифровой обработки сигналов. М. : Мир. 1978. 848 с.
12. Zaknich A. Principles of adaptive filters and self-learning systems. New York : *Springer Science & Business Media*. 2005. 408 p.
13. Чумаченко И.В., Косенко В.В. Оптимизация алгоритмического обеспечения в задачах преобразования информации. *Системы обработки информации*: зб. наук. пр. Харків : НАНУ, ПАНМ, ХВУ. 2002. Вип. 1 (17). С. 248–252.
14. Косенко В.В., Можаяв О.О., Гайдаров С.Ю. Методика оптимізації алгоритмів. *Системи обробки інформації*: зб. наук. пр. Харків : НАНУ, ПАНМ, ХВУ. 2004. Вип. 4 (32). С. 114–117.
15. Ахо А., Хопкрофт Дж., Ульман Д. Структуры данных и алгоритмы. М. : Издательский дом «Вильямс», 2000. 384 с.
16. Теслер Г.С. Адаптивные аппроксимации и итеративные процессы. *Математичні машини і системи*. 2004. № 2. С. 22–41.

Отримано 01.11.2022