



МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний технічний університет України

“Київський політехнічний інститут”

«Інформатика та обчислювальна техніка» – ІОТ-2016»

Матеріали наукової конференції студентів, магістрантів та аспірантів

25 – 27 квітня 2016 року

(кафедра «Автоматизовані системи обробки інформації і управління»)

Київ 2016

ЗМІСТ

1	<i>ОЧЕРЕТЯНИЙ О.К.</i>	СИСТЕМА ВЕРИФІКАЦІЇ КОРИСТУВАЧІВ НА ОСНОВІ ФІЗЕОЛОГІЧНИХ ПОКАЗНИКІВ	6
2	<i>ЛУК'ЯНЕНКО В.А., ЖДАНОВА О.Г., СПЕРКАЧ М.О.</i>	МОДЕЛІ ТА МЕТОДИ ОПЕРАТИВНО-КАЛЕНДАРНОГО ПЛАНУВАННЯ ДРІБНОСЕРІЙНОГО ВИРОБНИЦТВА	9
3	<i>КУТОВИЙ А.С., ФІНОГЕНОВ О.Д.</i>	МІЖМОВНЕ ПОРІВНЯННЯ ДОКУМЕНТІВ	13
4	<i>ГЛУЩЕНКО К.С., ГАВРИЛЕНКО О.В.</i>	АНАЛІЗ МЕТОДІВ ПОРІВНЯННЯ ТЕКСТІВ У ВИПУСНИХ РОБОТАХ СТУДЕНТІВ	17
5	<i>КІНДЗЕРСЬКИЙ О.В., ОЛІЙНИК Ю.О.</i>	РЕАЛІЗАЦІЯ МЕТОДУ АВТОРЕГРЕСІЇ НА ОСНОВІ ТЕХНОЛОГІЇ NVIDIA CUDA	20
6	<i>ГАЛІЦЬКА В.В.</i>	ГІБРИДНИЙ АЛГОРИТМ ШТУЧНОЇ БДЖОЛИНОЇ КОЛОНІЇ СКОМБІНОВАНИЙ З АЛГОРИТМОМ ІМІТАЦІЇ ВІДПАЛУ ДЛЯ РОЗВ'ЯЗАННЯ ЗАДАЧІ КОМІВОЯЖЕРА	22
7	<i>ХУССЕЙН С.І.</i>	ДІАГНОСТИКА ЕМОЦІЙНОГО СТАНУ КОРИСТУВАЧА ПЕРСОНАЛЬНОГО КОМП'ЮТЕРА ЗА РУХАМИ МИШКОЮ	26
8	<i>ГОНЧАР В.О.</i>	ЗАСТОСУВАННЯ МУРАШИНОГО АЛГОРИТМУ ДЛЯ РОЗВ'ЯЗАННЯ ЗАДАЧІ СИНТЕЗУ НАДІЙНОЇ МЕРЕЖІ	29
9	<i>ГУЛАК О.С., ОЛІЙНИК Ю.О.</i>	ОПТИМІЗАЦІЯ ПЕРЕДАЧІ АСК-ІВ У РЕЕР-ТО-РЕЕР МЕРЕЖАХ З ВУЗЬКИМ КАНАЛОМ ЗВ'ЯЗКУ	32
10	<i>КОБЕЦЬ Н.М., КОВАЛЮК Т.В.</i>	МОДЕЛЬ КОМПЕТЕНТНОСТІ МАГІСТРА ЗА СПЕЦІАЛЬНІСТЮ «ІНФОРМАЦІЙНІ СИСТЕМИ»	35
11	<i>СЕРДЮК А.І., ГАВРИЛЕНКО О.В.</i>	АНАЛІЗ МЕТОДІВ ІНТЕЛЕКТУАЛЬНОЇ ОБРОБКИ ДАНИХ ДЛЯ РЕКОМЕНДАЦІЙНИХ СИСТЕМ	39
12	<i>ШЕВЧЕНКО Д.С.</i>	НЕСТАЦІОНАРНА ЗАДАЧА КОМІВОЯЖЕРА В УМОВАХ АВІАТРАНСФЕРУ З ДОДАТКОВИМИ ОБМЕЖЕННЯМИ	43
13	<i>БЕРЕЖНОЙ В.Г.</i>	ВИРІШЕННЯ ЗАДАЧІ РОЗПІЗНАВАННЯ ВИДУ РИБИ МЕТОДОМ НЕЧІТКОГО ЛОГІЧНОГО ВИВОДУ	45

14	<i>МАСЛІКОВ В.</i>	ЗАСТОСУВАННЯ ГЕНЕТИЧНОГО АЛГОРИТМУ ДЛЯ РОЗВ'ЯЗАННЯ ЗАДАЧІ ФОЛДИНГУ БЛКІВ НА ГРАФІЧНОМУ ПРОЦЕСОРІ	47
15	<i>ШЛІХТЕНКО Н.М.</i>	МЕТОДИ ПАРАЛЕЛЬНИХ ОБЧИСЛЕНЬ ДЛЯ РОЗВ'ЯЗАННЯ СИСТЕМ ЛІНІЙНИХ АЛГЕБРАЇЧНИХ РІВНЯНЬ	51
16	<i>ЯРМАК Д.О.</i>	МЕТОДИ ПАРАЛЕЛЬНИХ ОБЧИСЛЕНЬ ДЛЯ РОЗВ'ЯЗАННЯ АЛГЕБРАЇЧНОЇ ПРОБЛЕМИ ВЛАСНИХ ЗНАЧЕНЬ	53
17	<i>АРТЕМЕНКО К.В., КОВАЛЮК Т.В.</i>	МОДЕЛЬ ДИСКРИМІНАНТНОГО АНАЛІЗУ В ПРОЦЕСІ НАВЧАННЯ	55
18	<i>СЕБАЛО М.М.</i>	РОЗВ'ЯЗАННЯ ЗАДАЧІ РОЗПІЗНАВАННЯ ПОЗИ ЛЮДИНИ НА ВІДЕО	59
19	<i>КОВАЛЬОВА Д.А., КОВАЛЮК Т.В.</i>	ПЕРЕДАЧА ДАНИХ ЗА ДОПОМОГОЮ СВІТЛОВОЇ ТЕХНОЛОГІЇ LI-FI, РЕАЛІЗОВАНА ДЛЯ МОБІЛЬНОГО ПРИСТРОЮ	61
20	<i>ХАЛИМОН А.Ю, БАКЛАН І.В.</i>	СТВОРЕННЯ ЕФЕКТИВНОЇ ПРОЦЕДУРИ ЛІНГВІСТИЧНОГО МОДЕЛЮВАННЯ ДЛЯ ПРОГНОЗУВАННЯ ПРОЦЕСІВ РІЗНОЇ ПРИРОДИ	65
21	<i>ГРАНЮК О.В., ГОЛОВЧЕНКО М.М.</i>	УНІФІКОВАНА АВТЕНТИФІКАЦІЯ НА WEB-РЕСУРСІВ ЧЕРЕЗ СОЦІАЛЬНІ МЕРЕЖІ	67
22	<i>ШЕВЧЕНКО А.С.</i>	МІНІМАКСНА ТЕОРІЯ СКЛАДАННЯ РОЗКЛАДУ	70
23	<i>РІДКОВЕЦЬ Ю.П., ТОМАХІНА О.С.</i>	ПРОГРАМНОТЕХНІЧНИЙ КОМПЛЕКС АВТОМОБІЛЬНОЇ ДІАГНОСТИКИ	73
24	<i>САРНАЦЬКИЙ В.В., БАКЛАН І.В., ШУЛЬКЕВИЧ Т.В.</i>	РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ КРОКІВ ПОБУДОВИ ГІБРИДНОЇ ЛІНГВІСТИЧНОЇ МОДЕЛІ ТА ПРИХОВАНОЇ МАРКОВСЬКОЇ МОДЕЛІ	77
25	<i>ЗЕМЛЯНСЬКИЙ Д.І.</i>	ПРОБЛЕМА РОБОТИ З ДИНАМІЧНОЮ ЗМІНОЮ СТРУКТУРИ ОБ'ЄКТІВ НА РІВНІ БАЗ ДАНИХ	80
26	<i>ШУЛЬКЕВИЧ Т.В., БАКЛАН І.В., СЕЛІН Ю.М.</i>	РЕАЛІЗАЦІЯ ПОБУДОВИ ЛІНГВІСТИЧНОЇ МОДЕЛІ	81
27	<i>МИХАЙЛОВ О.В.</i>	ПОЗИЦІЙНИЙ БАГАТОКРИТЕРІАЛЬНИЙ АЛГОРИТМ APS/MES-ПЛАНУВАННЯ ВИРОБНИЦТВА	85

28	МЕДВЕДСЬКА К.О., ЖДАНОВА О.Г., СПЕРКАЧ М.О.	ЗАДАЧА ПЛАНУВАННЯ ВИКОНАННЯ ЗАВДАНЬ ЗІ СПІЛЬНИМ ДИРЕКТИВНИМ ТЕРМІНОМ НЕІДЕНТИЧНИМИ ПАРАЛЕЛЬНИМИ ПРИСТРОЯМИ	88
29	ФУНТОВИЙ І.І., СЕЛІН Ю.М.	ТРІВИМІРНЕ МОДЕЛЮВАННЯ В ЗАДАЧАХ РОЗПІЗНАВАННЯ ОБРАЗІВ	92
30	РАМАЗАНОВ Р.В.	ОГЛЯД МЕТОДІВ ІДЕНТИФІКАЦІЇ ЛЮДИНИ ЗА РАЙДУЖНОЮ ОБОЛОНКОЮ ОКА	95
31	ХРАМЧЕНКО М.С., ІГНАТЕНКО С.М., МУХА І.П.	ОРГАНІЗАЦІЯ КОЛЕКТИВНОЇ РОБОТИ З УНІВЕРСАЛЬНИМ ПРОГРАМНО-АПАРАТНИМ КОМПЛЕКСОМ ДЛЯ КОМП'ЮТЕРИЗАЦІЇ НАВЧАЛЬНОГО І НАУКОВОГО ЕКСПЕРИМЕНТУ	98
32	ШОЛОМ І.Д., МУХА І.П.	СПЕЦІАЛІЗОВАНЕ ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ ЗБЕРЕЖЕННЯ ТА ВІДОБРАЖЕННЯ РАДІОЛОКАЦІЙНОЇ ІНФОРМАЦІЇ	100
33	БЕЛОЧУБ Н.М., ГОЛОВЧЕНКО М.Н., НАНОВСКИЙ С.А., НЕЧАЙ Д.А.	КРОССПЛАТФОРМЕННЫЙ ТОНКИЙ КЛИЕНТ CONSOLE В ЕКСПЕРИМЕНТАЛЬНОМ ТЕХНОЛОГИЧЕСКОМ СТЕКЕ METARNA	102
33	ТИМОШИН Ю.А., ОРЛЕНКО А.А., КИРИЧЕК Н.И., РУЖЕВСКИЙ Н.С.	КОНЦЕПЦИЯ И ОСОБЕННОСТИ РАСПРЕДЕЛЕННОЙ СУБД GLOBALSTORAGE	106
34	МУХА І.П. ДЗЮБА В.В., БЕРНАТОВИЧ А.А., ГАЛАСЮК В.И.	ПРОТОКОЛ JSTR В ЕКСПЕРИМЕНТАЛЬНОМ ТЕХНОЛОГИЧЕСКОМ СТЕКЕ METARNA	111
35	ВЫСОЦКИЙ А.С., ИВАНОВА Л.А., ЧЁРНЕНЬКИЙ А.Ю., ШЕМСЕДИНОВ Т.Г.	SINCETV: ПЛАТФОРМА ДЛЯ ИНТЕРАКТИВНЫХ ПРИЛОЖЕНИЙ, РЕАЛИЗУЮЩИХ КОНЦЕПЦИЮ ВТОРОГО ЭКРАНА, ПРИБЛИЖЕННЫХ К РЕАЛЬНОМУ ВРЕМЕНИ	115
36	ОСПЕНКО Д.С., ОЛІЙНИК Ю.О.	МЕТОДИ ВИРШЕННЯ БАГАТОКРИТЕРІАЛЬНИХ ЗАДАЧ	119
37	СЯГАЙЛО Т.А.	ДОСЛІДЖЕННЯ ЗАДАЧИ ПЛАНУВАННЯ ПО КРИТЕРІЮ МАКСИМАЛЬНО ПІЗЬОГО МОМЕНТУ ЗАПУСКУ ТА МІНІМАЛЬНОГО СУМАРНОГО ВИПЕРЕДЖЕННЯ	121
38	ШОЛОМ Д.Л., ТЄЛИШЕВА Т.О.	СПОСІБ АДАПТИВНОГО УПРАВЛІННЯ ПРОЦЕСОМ БУРІННЯ В УМОВАХ НЕВИЗНАЧЕНОСТІ	125

39	<i>МОЖАРОВСЬКИЙ А.С., ТЄЛИШЕВА Т.О</i>	СПОСІБ ОПТИМІЗАЦІЇ ТЕХНОЛОГІЧНОГО ПРОЦЕСУ БУРІННЯ НАФТОВИХ СВЕРДЛОВИН НА ОСНОВІ НЕЧІТКОЇ ЛОГІКИ	130
40	<i>КАТЮЩЕНКО Д.О., ОЛІЙНИК Ю.О.</i>	ОГЛЯД АЛГОРИТМІВ ФОРМУВАННЯ РЕКОМЕНДАЦІЙ З ОБМІНУ ПОДАРУНКАМИ	134
41	<i>ЦУЦАРІН М.М.</i>	ВИКОРИСТАННЯ ТРІЙКОВОЇ ФУНКЦІЇ ВТРАТ У ЗАДАЧІ РОЗПІЗНАВАННЯ ОБЛИЧ	137

СИСТЕМА ВЕРИФІКАЦІЇ КОРИСТУВАЧІВ НА ОСНОВІ ФІЗІОЛОГІЧНИХ ПОКАЗНИКІВ

В даній статті розглянуто практичне використання лінгвістичного програмування на прикладі системи верифікації користувачів. Демонструється використання прихованих марківських моделей для визначення образів. Детально розглянуто прикладне використання алгоритму згідно до останніх тенденцій розробки програмного забезпечення.

The subject of this article is the practical use of linguistic programming on the example of system of user's verification. Was demonstrated the use of Hidden Markov Models to determine the characters. Considered in detail use of the algorithm according to the latest trends in software development.

1. Вступ

З початком передачі повідомлень людство потребувало захисту інформації що передаються. Для вирішення цієї проблеми суспільство винайшло шифрування. Це метод кодування інформації, що передбачає недоступність зашифрованого тексту для того хто не має предмету чи методу що розшифрує. Однак із розвитком шифрування розвивалося мистецтво злому шифрів, на противагу цьому розроблялися нові механізми. В середині другої світової війни для розшифрування німецького шифру британськими математиками був винайдений прообраз першої обчислювальної машини. Таким чином проблема шифрування є причиною появи комп'ютерів як таких.

Усі види шифрування поділяються на три типи, шифрування використовуючи знання, кодове слово чи пароль. Другий тип, це шифрування використовуючи наявність, тобто людина що має отримати послання має володіти ключом, usb-ключ чи магнітна карта. Третій тип це біометрика, тобто частина тіла типу відтиск пальця чи сітківка ока. На даний час набирає популярність використання декількох типів одночасно, для підвищення безпеки збереженої інформації.

2. Новітні розробки у сфері авторизації

Світ безпеки інформації не стоїть на місці, у середині 2015 року Amazon

презентувала новий тип авторизації користувача телефону. Замість звичайного паролю компанія пропонує біометричну авторизацію через малюнок вашої вушної раковини. В той час як ви будете підносити телефон до вуха, телефон просканує фотографію та звірить із заданим малюнком. Так ви зможете зекономити декілька секунд на розблокування просто підносячи телефон до вуха відповідаючи на дзвінок. Так як кожна людина унікальна, кожна її частина тіла має унікальну форму, таким чином авторизація через біометричний тип є однією із найбезпечніших. Однак вона все одно має свої недоліки як пошкодження кінцівки що використовується для авторизації чи використання муляжів чи відмерлих кінцівок.

Однак розвиток біометричних типів авторизації не є єдиним, однією із проблем паролів від банківських карток є обмеженість користувачів у кількості можливих символів для використання та небажання користувачів вигадувати складні паролі. На заміну такому способу розробники з Intelligence Environment винайшли нове рішення для банків, що будується на використанні емотиконів замість цифр. На вибір користувачів пропонується 44 види емотиконів, що є доступними у різних операційних системах. Також розробники вилучили із списку емотикони, що є схожими.

Дослідження було збудовано на початковій ідеї заміни цифр у банківських паролях. Вчені аналізували поведінку покоління мілленіалів у соцмережах та месенджерах та прийшли до висновку що 64% представників покоління використовують емотикони у листуванні. Дані дослідження і наштотували на створення революційного типу паролів. Очевидною перевагою даного методу задання паролів є специфіка запам'ятання людського мозку, малюнки запам'ятовувати набагато простіше за цифри та іншу інформацію. Іншою перевагою є те, що кількість паролів з використанням емотиконів перевищує звичну кількість паролів у 480 разів. Однак цей механізм має свої недоліки, проведені дослідження демонструють, що при виборі паролів емотиконів піддослідні іноді вибирали перші 4 емотикони за пароль. У цьому випадку безпека даних не підвищується, але для вирішення цієї проблеми розробники планують розміщувати емотикони у довільному порядку при кожному використанні системи користувачем.

3. Проблеми систем авторизації

Головною проблемою усіх систем авторизації є можливість підглянути, вкрасти чи підмінити секретний пароль. Однак на сьогодні ми маємо можливість зчитати те що ніхто не зможе підглянути. Одним із найпростіших прикладів є зчитування несвідомих рухів руки людини. Дана особливість проявляється у всіх діях для яких ми використовуємо руки, однак найпростішою перевіркою є перевірка почерку людини. Під час зчитування інформації людина не знатиме по перше що її перевіряють. По друге навіть при ідентичному повторенні порівнюючи з зразком ми будемо знати що це підробка бо людина не зможе ідентично відтворити написане. По третє перевірка підсвідомих рухів надає змогу простої та ефективної діагностики хвороб що проявляються таким чином. Використовуючи даний тип авторизації людина може не переживати про забутий пароль чи відсутність таємного ключа,

все що їй необхідно для авторизації це бути собою.

4. Система верифікації на основі рухів кисті

Дана система збудована на принципі перетворення координат точок на часовому інтервалі у лінгвістичну послідовність. Дане перетворення виконується використовуючи методи лінгвістичного моделювання.

Головним завданням лінгвістичного моделювання є перетворення чисельних рядів, експериментальних даних, багатомірних даних до лінгвістичних послідовностей та відновлення за ними формальної граматики мови. Головна ідея переходу від числового ряду до символічного, від якої залежить результат лінгвістичного моделювання, базується на розбитті попередньо упорядкованої множини вхідних чисельних даних на коректні інтервали.

Для розбиття множини використовується дискретний рівномірний розподіл, що передбачає однакову вірогідність входу точок у інтервал. Даний розподіл допоможе алгоритму у нормалізації даних та апроксимації результатів що значно збільшить коректність вихідних даних. Використанні інших розподілів може мати місце, однак саме даний вид надає найкращу вибірку для вибраного алгоритму.

Множина X розподілена в інтервалі $[\min(x_{s1}); \max(x_{s1})]$ (далі називатимемо a, b). Згідно із завданням розділимо даний відрізок на M інтервалів. Для зручності візьмемо $M = 10$. Якщо щільність розподілу на даному відрізку постійна, тобто всі значення на цьому відрізку рівноймовірні, то маємо:

$$f(x) = \begin{cases} c, & a \leq x \leq b \\ 0, & x < a, x > b \end{cases}$$

Значення постійної c розраховується з умов нормування:

$$1 = \int_{-\infty}^{+\infty} f(x) dx = 0 + \int_a^b c dx + 0 = c(b-a) \Rightarrow c = \frac{1}{b-a}$$

Функція розподілу:

$$F(x) = \begin{cases} 0, & x < a \\ (x-a)/(b-a), & a \leq x \leq b \\ 1, & x > b \end{cases}$$

Якщо випадкова величина може приймати будь-яке з n значень k_1, k_2, \dots, k_n , тоді це є дискретним рівномірним розподілом. Ймовірність випадання k_j дорівнює $1/n$. Простим прикладом дискретного рівномірного розподілу є випадання гральної кості. k

набуває значень 1, 2, 3, 4, 5, 6 і кожен раз k випадає з імовірністю $1/6$. У випадку, коли випадкова величина є дійсним числом, то функція розподілу можна виразити у термінах виродженого розподілу таким чином:

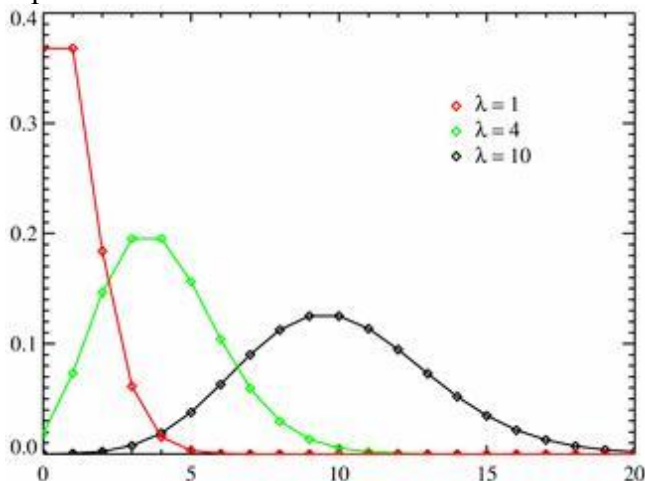


Рисунок 1. Графік дискретного рівномірного розподілу

Після отримання даних вводу алгоритм буде розподіляти отримані точки на інтервали використовуючи дискретний рівномірний розподіл, наступним кроком ми перетворюємо отриману послідовність у алфавіт. При цьому потужність алфавіту має співпадати з кількістю інтервалів. Дані алфавіти будуть зберігатися у системі та використовуватися для перевірки користувачів на реєстрації.

Механізм авторизації за допомогою почерку може мати різноманітні реалізації однак найпростішим є побудова клієнт серверного додатку. Клієнтська частина буде виконувати функцію зчитування даних про почерк користувача : швидкість руху, інтенсивність та інші показники та передавати на сервер отримані дані у модифікованому форматі. Із свого боку сервер буде звіряти отримані дані із збереженими у сховищі та віддавати відповідь про авторизованість користувача.

Дана система має декілька недоліків, що будуються на емоційних станах людини. Існує гіпотеза що під час сильного емоційного збудження користувач буде вести себе нетипово що

призведе до неможливості авторизації його системою. Також існує вірогідність зміни необхідних показників під час довгого періоду часу не користування системою , в цьому випадку система буде вважати що хтось намагається зламати аккаунт і не надасть доступ користувачеві.

Список літератури

1. Г.А Степанкова, І.В Баклан Побудова гібридних моделей на основі прихованих марківських моделей та нейронних мереж // Автоматика. Автоматизація. Електротехнічні комплекси та системи – 2008
2. АВHIMANYU GHOSHAL Amazon wants you to unlock your next phone with your ear// The Next Web -2015
3. Владислав Шаров Биометрические методы компьютерной безопасности // BYTE/Россия. -2005 -№ 4(80)
4. Zeljka Zorz Emojis instead of PIN codes as an alternative for forgetful users // Help Net Security - 2015
5. Дмитрий Выростков Обзор способов и протоколов аутентификации в веб-приложениях// Блог компании DataArt/Хабрахабр - 2015

МОДЕЛІ ТА МЕТОДИ ОПЕРАТИВНО-КАЛЕНДАРНОГО ПЛАНУВАННЯ ДРІБНОСЕРІЙНОГО ВИРОБНИЦТВА

В даній роботі розглянуто вирішення задачі оперативно-календарного планування дрібносерійного виробництва та наведено моделі і методи для вирішення поставленої задачі. Наведено процес розв'язання задачі формування розкладу за узагальненим технологічним графом. Обґрунтовано вибір підходу до визначення вершин узагальненого графу та продемонстровано на конкретних прикладах.

The subject of this article is scheduling for low-volume production. It contains models and methods of operational and production scheduling for solving this issue and provides the process of scheduling according to generic technological graph. The choice of approach to define generalized graph is justified and demonstrated on specific examples.

Ефективність роботи підприємства та його подальший розвиток забезпечується завдяки постійній адаптації до швидкозмінних зовнішніх умов, що викликає необхідність самостійно приймати рішення стосовно обсягів виробництва, що випускається, вибору постачальників, організації збуту, джерел фінансування.

Оперативно-календарне планування різних типів виробництва на підприємствах має ряд особливостей, але які б методи оперативного планування не застосовувалися на підприємстві, їхня мета забезпечити рівномірність перебігу виробничого процесу і випуск якісної продукції у необхідній кількості та у встановлені строки.

Планування роботи дрібносерійного виробництва є ресурсоемним процесом, оскільки необхідно враховувати велику кількість різних факторів виробництва при складанні розкладу його роботи. Саме тому виникає необхідність у створенні ефективних методів та програмних комплексів, які зможуть успішно виконати поставлену задачу за прийнятний час з урахуванням обмежених обчислювальних ресурсів.

Існує безліч методів та алгоритмів, які можуть бути покладені в основу підходу до формування розкладу роботи дрібносерійного виробництва. До них можна віднести методи мережевого планування [Ошибка! Источник ссылки не найден.], генетичні алгоритми [Ошибка! Источник ссылки не найден.], метод

гілок та меж [Ошибка! Источник ссылки не найден.], різні метаевристичні алгоритми [Ошибка! Источник ссылки не найден.], а також поєднання вище зазначених алгоритмів між собою та з іншими допоміжними алгоритмами [Ошибка! Источник ссылки не найден.].

При розробці підходу до складання розкладу роботи дрібносерійного виробництва застосовується різна кількість підходів в залежності від специфіки роботи підприємства. Один з підходів це представлення роботи у вигляді мережевого графу – узагальненого графу проходження завдань, де вершини графу представляють собою виробничі дільниці на яких проходить виготовлення виробів, а ребра – операції, які необхідно виконати для того, щоб отримати готовий виріб.

Узагальнений граф проходження завдань складається з об'єднання графів, які представляють технологічну карту виготовлення певного виробу. За узагальненим графом проходження завдань будується розклад роботи підприємства.

1. Постановка задачі

Є обслуговуюча система, що складається з множини $U, U = u$ виробничих дільниць. На кожній дільниці розміщений один або декілька пристроїв (одиниць обладнання), що можуть працювати паралельно. Всі дільниці є унікальними в сенсі того, що кожна з них виконує один і тільки один вид завдань (технологічних операцій обробки), які не

можуть бути виконані на жодній іншій дільниці.

Дільниця $y(y = \overline{1, u})$ характеризується кількістю пристроїв m_y та вектором коефіцієнтів продуктивності пристроїв $M_y = \{k_{yl}\}, y = \overline{1, u}, l = \overline{1, m_y}$. Системі потрібно виготовити множини виробів $P, |P| = n$. Для кожного виробу $p_i \in P$ відомий директивний термін його виготовлення $d_i, i = \overline{1, n}$. Для виготовлення кожного виробу $p_i \in P, i = \overline{1, n}$, необхідно виконати множини завдань, для яких задано відношення передування. Відношення безпосереднього передування завдань, пов'язаних з виготовленням виробу p_i , задається орієнтованим ациклічним антитранзитивним графом $G_i = \langle V_i, E_i \rangle$, у якому вершини відповідають завданням, а наявність направленої ребра з вершини з індексом j_1 до вершини з індексом j_2 означає, що робота j_2 може бути розпочата лише після завершення роботи $j_1, (j_1, j_2 \in \{1, 2, \dots, g_i\})$, де $g_i = |V_i|$ – кількість завдань, що відповідають виробу i). Вимога щодо властивості антитранзитивності накладається на граф G_i , тому що він представляє відношення безпосереднього передування завдань: наявність дуги між вершинами, що відповідають завданням j_1 та j_3 , коли існує таке завдання j_2 , що $(v_{ij_1}, v_{ij_2}) \in E_i$ і $(v_{ij_2}, v_{ij_3}) \in E_i$, є надлишковою. Для кожного завдання $(ij), i = \overline{1, n}, j = \overline{1, g_i}$, відомі час її виконання t_{ij} на «еталонному» пристрої та індекс $u_{ij} \in \{1, 2, \dots, u\}$ виробничого вузла, на якому це завдання може бути виконане. «Еталонним» будемо називати пристрій з коефіцієнтом продуктивності $k = 1$. Фактичний час виконання завдання (ij) на пристрої l вузла з індексом u_{ij} становить $\frac{t_{ij}}{k_{u_{ij}l}}, l = \overline{1, m_{u_{ij}}}$.

2. Розв'язання задачі

Розглянемо процес розв'язання задачі формування розкладу за узагальненою технологічною картою.

Етап 1. Формування узагальненого графа проходження робіт

1.1. Створення проміжних графів G_i^U на основі графів $G_i, i = \overline{1, n}$.

1.2. Послідовне об'єднання проміжних графів в узагальнений граф G .

Етап 2. Складання розкладу виконання робіт вузлами на основі узагальненого графа.

Описана вище задача може бути представлена як багатоетапна мережева задача календарного планування. Весь процес виготовлення вихідної множини виробів можна представити у вигляді ациклічного графа $G = \langle V, E \rangle$. Граф G є деякою формою об'єднання всіх графів $G_i, i = \overline{1, n}$.

Розглянемо схему алгоритму побудови узагальненої технологічної карти.

КРОК 1. Для кожного з відношень передування робіт, заданого графом $G_i, i = \overline{1, n}$.

сформувати відповідний граф G_i^U , який має всі властивості узагальненого графа G .

Ініціалізувати G будь-яким графом G_i^U .

КРОК 2. ДЛЯ ВСІХ $G_i^U, i = \overline{2, n}$, ВИКОНУВАТИ

2.1. Створити чергу Q , яка складається з усіх вершин графу G_i^U **2.2.** ПОКИ черга не пуста, для ребер $e \in E$ ВИКОНУВАТИ

2.2.1. Отримати вхідне ребро $e = \{v_1, v_2\}$ з вершини отриманої з Q .

2.2.2. Додати дане ребро до графа G при умові збереження його властивості ациклічності, додаючи нові вершини, якщо це необхідно.

2.2.3. Видалити ребро e з графа G_i^U

Якщо з v_1 у графі G_i^U більше не виходить жодне ребро – видалити v_1 .

Якщо в v_2 у графі G_i^U більше не входить жодне ребро – видалити v_2 .

2.2.4. Якщо вершина з черги Q є останньою – отримати вихідне та перейти на крок 2.2.2.

Результатом роботи цього алгоритму G є узагальнений граф передування робіт вихідної множини виробів.

3. Вибір методу побудови графу

Побудова узагальненого графу вимагає об'єднання технологічних карт виробів, які задаються графами меншої розмірності. У

процесі об'єднання графу постає проблема правильного вибору вершин узагальненого графу, в які будуть перенесені ребра графів технологічних карт.

Вибір правильних вершин є дуже важливим, оскільки при «склеюванні» графів можуть виникнути цикли, оскільки графи є направленими, що призведе некоректного розкладу і в подальшому неправильної роботи всього виробництва.

Вибір вершин вимагає великих витрат ресурсів, оскільки для додавання одного ребра потрібно визначити дві вершини в узагальненому графі, тому необхідно ефективно застосувати методи пошуку в графі, які зможуть сформувати коректний узагальнений граф за допустимий час.

Пошук в глибину і в ширину є одними з методів, які дають ефективні результати при вирішенні поставленої задачі [Ошибка! Источник ссылки не найден.].

Для додавання кожного ребра у відповідну позицію в узагальненому графі використовуються методи пошуку в ширину або в глибину. За допомогою їх обираються вхідні та вихідні вершини для нового ребра.

Розглянемо більш детально застосування вище наведених методів.

Вибір відповідних вершин можна провести за чотирма різними методами: пошук в ширину з використанням стратегії створення нової початкової вершини; пошук в глибину з використанням стратегії створення нової початкової вершини; пошук в ширину з використанням стратегії вибору початкової вершини з вже існуючих; пошук в глибину з використанням стратегії вибору початкової вершини з вже існуючих.

Технологічні карти представляються у вигляді направленого ациклічного графу, де кожне ребро є певним типом роботи по виготовленню виробу, а вершини є відображенням виробничих дільниць, на яких здійснюються роботи по виготовленню виробу. На рис. 1,2,3 зображено технологічні карти виробів А, В, С.

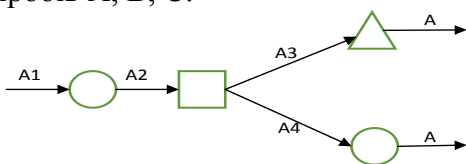


Рис. 1. – Технологічна карта виробу А

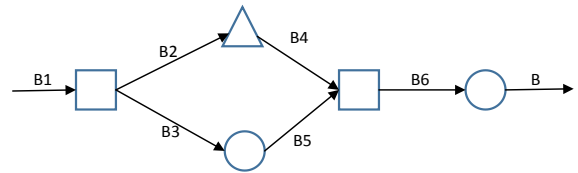


Рис. 2. – Технологічна карта виробу В

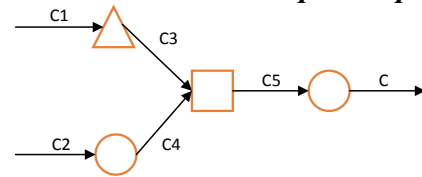


Рис. 3. – Технологічна карта виробу С

Для додавання кожного ребра у відповідну позицію в узагальненому графі використовуються методи пошуку в ширину або в глибину. За допомогою їх обираються вхідні та вихідні вершини для нового ребра. Розглянемо детально кожен з варіантів вибору початкової та кінцевої вершин для ребра, що додається.

Застосування пошуку в ширину з створенням нової вершини. Якщо до узагальненого графу додається ребро, яке не має стартової вершини, то ми завжди створюємо нову, незалежно від того чи в узагальненому графі уже є вершина необхідного типу. Застосувавши алгоритм до початкових графів отримаємо наступний узагальнений граф (рис. 4).

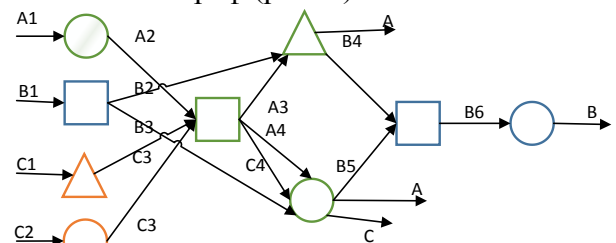


Рис. 4 – Узагальнена технологічна карта за пошуком в ширину та новою вершиною

Застосування пошуку в ширину з вибором вже існуючої вершини (рис. 5). Якщо до узагальненого графу додається ребро, яке не має стартової вершини, то ми шукаємо вже існуючу вершину необхідного типу, і створюємо нову тільки якщо така відсутня.

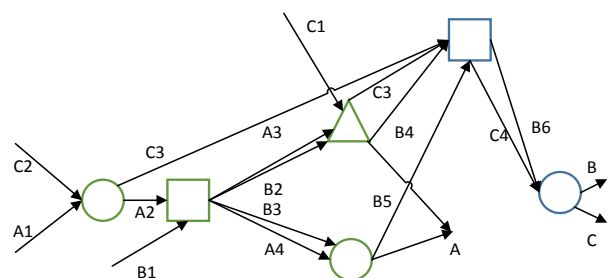


Рис. 5 – Узагальнена технологічна карта за пошуком в ширину

Застосування пошуку в глибину з створенням нової вершини (рис. 6). Якщо до узагальненого графу додається ребро, яке не має стартової вершини, то ми завжди створюємо нову, незалежно від того чи в узагальненому графі уже є вершина необхідного типу.

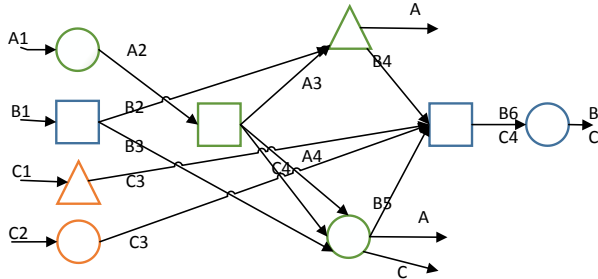


Рис. 6 – Узагальнена технологічна карта за пошуком в глибину та новою вершиною

Застосування пошуку в глибину з вибором вже існуючої вершини (рис. 7). Якщо до узагальненого графу додається ребро, яке не має стартової вершини, то ми шукаємо вже існуючу вершину необхідного типу, і створюємо нову тільки якщо така відсутня.

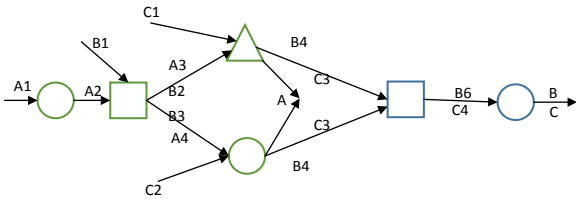


Рис. 7 – Узагальнена технологічна карта за пошуком в глибину

4. Заключення

З наведених вище прикладів можна побачити, що кожен з методів дає різні результати. Застосування методів пошуку в глибину та ширину зі створенням нової вершини дозволяє розвантажити початкові ділянки та почати виготовлення більшої кількості виробів одночасно, але може призвести до вузьких місць у виготовленні виробів в подальшій роботі підприємства.

Пошук в глибину без створення нової вершини є кращим варіантом для використання, але він має суттєвий недолік: якщо узагальнений граф має велику кількість рівнів та вершин на цих рівнях, то алгоритм може дуже сильно заглиблюватись на кожній ітерації, що призведе до значних втрат часу.

Найбільш ефективним з чотирьох методів є пошук в ширину без створення нової початкової вершини. Даний метод дозволяє отримати ефективний розподіл між

вершинами графу на кожному рівні та ребрами графу, що дозволяє отримати найбільш збалансований граф для складання розкладу.

За результатами створеного підходу був розроблений програмний продукт, що надає можливість ефективно планувати роботу дрібносерійного виробництва.

Список літератури

1. Л. В. Ноздріна, В. І. Ящук, О. І. Полотай Управління проектами: Підручник [Текст]/ За заг. ред. Л. В. Ноздріної. — К.: Центр учбової літератури, 2010. — 432 с.
2. Панченко, Т. В. Генетические алгоритмы [Текст] : учебно-методическое пособие / под ред. Ю. Ю. Тарасевича. — Астрахань : Издательский дом «Астраханский университет», 2007. — 87 [3] с.
3. А. А. Лазарев, Е. Р. Рашидович Теория расписаний [Текст]/ Москва, 2011 – 222 с.
4. О. А. Щербина Метаэвристические алгоритмы для задач комбинаторной оптимизации (обзор) [Текст]/ Севастополь, 2014 – 17 с.
5. Т. Кормен Алгоритмы. Построение и анализ [Текст]/ Издательский дом «Вильямс», 2005. – 1296 с.

МІЖМОВНЕ ПОРІВНЯННЯ ДОКУМЕНТІВ

У статті описується метод порівняння текстів на різних мовах. Він заснований на автоматичному перекладі і спроби звести її до однієї синонімічної форми. Алгоритм допомагає ідентифікувати Міжмовний плагіат.

The article describes the method of comparison of texts in different languages. It is based on automatic translation and attempt to reduce it to a single synonymous form. The algorithm helps to identify interlingual plagiarism.

ВСТУП

Проблема пошуку подібних документів тісно пов'язана із задачею пошуку плагіату. У теперішній час, плагіат в українському освітньо-науковому просторі стає дедалі все більш розповсюдженим. Наприклад, за даними аналізу відмов ВАК у присудженні наукових ступенів [1] суттєва доля була відкинута внаслідок знаходження плагіату.

Автоматичне визначення плагіату у наукових роботах є вкрай важким, особливо для технічних напрямів, що в якості основних результатів роботи можуть мати не текстові фрагменти (моделі, методи, алгоритми тощо). Додатковою проблемою для автоматичного пошуку є само визначення плагіату [2]: «плагіат – це оприлюднення (опублікування), повністю або частково, чужого твору під іменем особи, яка не є автором цього твору». Внаслідок такого «нечіткого» формулювання, автоматичне визначення плагіату є майже неможливим: не визначено кількісні показники відсотків збігу, обсяг фрагментів тексту запозичення та таке інше.

Найбільш поширеними формами плагіату є:

– копіювання результатів чужої наукової роботи, чи декількох робіт та оприлюднення результату під своїм ім'ям;

– створення «суміші» власного та запозиченого тексту без належного цитування джерел;

– рерайтинг (перефразування чужої праці без згадування оригінального автора) – цей тип плагіату, якщо він зроблений досконало, на сьогодні важко виявити технічними засобами пошуку плагіату (програмами антиплагіату).

ОСНОВНА ЧАСТИНА

Загалом пошук плагіату (подібності документів) складається з кількох етапів:

- зведення текстів до канонічної форми;
- визначення параметрів тексту для порівняння;
- визначення міри близькості тексту за одним з алгоритмів;
- пост обробка результатів людиною.

Одним з вузьких місць є використання іншомовних джерел, так як це вимагає не тільки роботи з нормалізацією тексту а і необхідність враховувати відмінності перекладу. Розглянемо приклад [3], який демонструє один з найпростіших способів приховування плагіату, за допомогою використання синонімів (табл. 1):

Таблиця 1 – Приклад перекладу і використання синонімів

Оригінальний текст	Текст 1 (переклад)	Текст 2
In practice, monitoring of such a broad range of risk factors is rather difficult, so there is an objective need for the reduction of the entire set of risk factors to several observable and relatively easily evaluated factors.	На практиці моніторинг такого широкого діапазону чинників ризику є досить проблематичною задачею, тому є реальна потреба в скороченні всієї сукупності чинників ризику для небагатьох спостережуваних і відносно просто оцінюваних факторів.	На практиці моніторинг такого широкого спектра факторів ризику є досить складною задачею, тому існує об'єктивна необхідність в скороченні всієї сукупності факторів чинників для декількох спостережуваних і відносно легко оцінюваних чинників.

Даний приклад показує як синонімізація може зменшити відсоток подібності документів при прямому порівнянні майже до 0. Тому при порівнянні документів написаних різними мовами додатково треба використовувати не тільки автоматичний переклад, а й підбір можливих синонімів. Зазвичай якість перекладу, що надають автоматичні засоби, невисока, але їх можна

використати для пошуку плагіату. Про те, для такої задачі додатково необхідно використовувати тезауруси або словники для нормалізації синонімічних форм.

Якщо розглянути порівняння двох текстів, що написані різними мовами, то для оцінки їх подібності можна запропонувати наступні алгоритми (рис. 1):

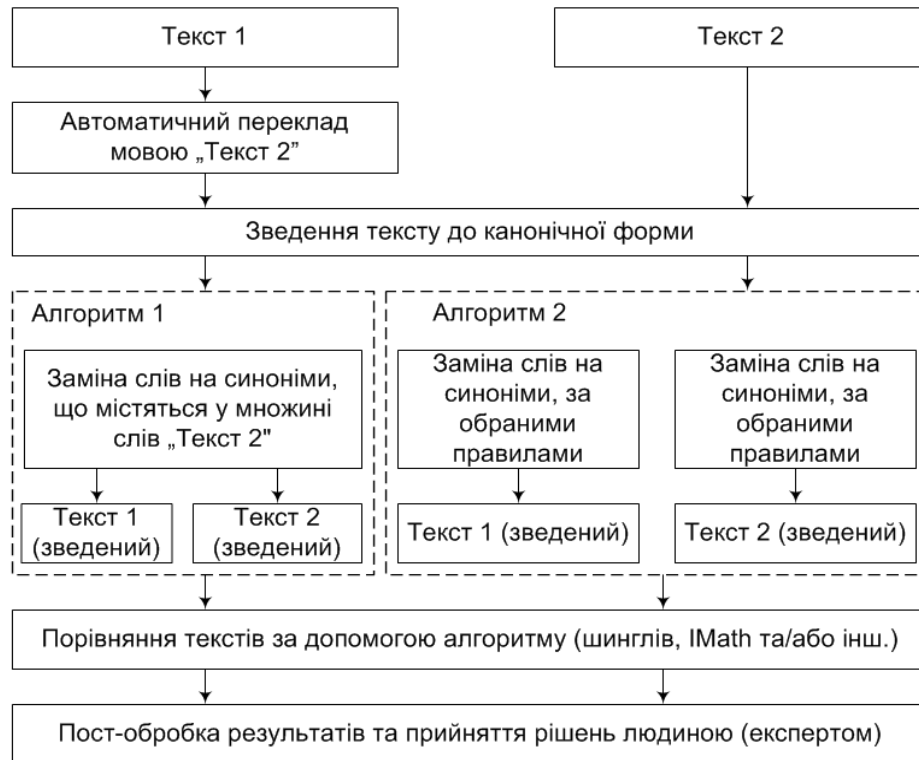


Рисунок 1 – Алгоритми оцінки подібності різномовних документів

Етап 1. Переклад. Один з наявних текстів («Текст 1» на рис. 1) перекладається за допомогою програм автоматичного перекладу до мови «Текст 2».

Етап 2. Зведення тексту до канонічної форми. На цьому етапі текст зводиться до єдиного алфавіту, видаляються всі стоп-символи та стоп-слова, слова зводять до називного відмінку, чоловічого роду однини, інфінітиву. Окремі операції даного етапу можуть передувати перекладу (етап 1), якщо є необхідність, наприклад, виключення стоп-символів у вхідному тексті.

Етап 3. Заміна слів на синоніми. На даному етапі можливі два алгоритми роботи:

1) для кожного слова з канонічної форми «Текст 1» за словником синонімів шукається форма слова, що є в множині слів канонічної форми «Текст 2». Недоліком такого підходу є наявність у «Текст 2» декількох слів, що мають спільні синоніми. В цьому випадку виникає залежність від порядку слідування слів.

2) для кожного слова обох текстів за словником синонімів визначаються форми слів за певними правилами: наприклад, перший з синонімів за алфавітом або за частотою використання. Такий підхід є незалежним від порядку слідування слів в тексті, але потребує наявності перехресних посилань для всіх варіантів синонімів.

Розглянемо дані алгоритми на прикладі:

Переклад «Текст 1»	«Текст 2»
На практиці моніторинг такого широкого діапазону чинників ризику досить проблематичною задачею, тому є реальна потреба в скороченні всієї сукупності чинників ризику для небагатьох спостережуваних і відносно просто оцінюваних факторів.	На практиці моніторинг такого широкого спектра факторів ризику досить складною задачею, тому існує об'єктивна необхідність в скороченні всієї сукупності чинників ризику для декількох спостережуваних і відносно легко оцінюваних чинників.

Після нормалізації:

«Текст 1»	«Текст 2»
практика моніторинг широкий діапазон чинник ризик проблематичний задача є реальний потреба скорочення сукупність чинник ризик небагато спостережуваних відносний простий оцінюваний фактор	практика моніторинг широкий спектр фактор ризик складний задача існує об'єктивний необхідність скорочення сукупність чинник ризик декілька спостережуваних відносний легкий оцінюваний чинник

Нехай у словнику є такі синонімічні групи:

- діапазон, обсяг, спектр;
- фактор, чинник;
- важкий, проблематичний, складний;
- є, існує, наявний;

За алгоритмом 1:

- необхідний, об'єктивний, реальний;
- необхідність, потреба;
- неважкий, легкий, простий.

Після обробки через синонімічні словники тести зводиться до наступних:

«Текст 1»	«Текст 2»
практика моніторинг широкий спектр фактор ризик складний задача існує об'єктивний необхідність скорочення сукупність фактор ризик небагато спостережуваних відносний легкий оцінюваний фактор	практика моніторинг широкий спектр фактор ризик складний задача існує об'єктивний необхідність скорочення сукупність чинник ризик декілька спостережуваних відносний легкий оцінюваний чинник

За алгоритмом 2:

«Текст 1»	«Текст 2»
практика моніторинг широкий діапазон фактор ризик важкий задача є необхідний необхідність скорочення сукупність фактор ризик небагато спостережуваних відносний неважкий оцінюваний фактор	практика моніторинг широкий діапазон фактор ризик важкий задача є необхідний необхідність скорочення сукупність фактор ризик декілька спостережуваних відносний неважкий оцінюваний фактор

Як можна побачити з наведених прикладів, в «Текст 2» є два слова з однієї синонімічної групи, тому «Текст 1» за алгоритмом 1 має меншу схожість з «Текст 2», ніж за 2-м алгоритмом.

Етап 4. Порівняння текстів. Порівняння відбувається одним з алгоритмів, що визначають для кожного з текстів деяку сигнатуру або хеш-функцію. Ступінь подібності документів визначається за окремою метрикою, що визначає відстань (подібність) сигнатур.

ВИСНОВКИ

Запропоновані алгоритми при порівнянні різномовних текстів за допомогою існуючих сервісів автоматичного перекладу (наприклад [4]) та словників синонімів (наприклад [5]), дають змогу автоматизувати пошук плагіату. До недоліків даних алгоритмів слід віднести залежність результатів роботи від якості перекладу та існуючих словників синонімів.

Дані алгоритми не дають можливості визначення близькості тексту при рерайтингу. Але в цьому випадку, алгоритми оцінки подібності одномовних документів теж не нададуть належного результату.

Перспективним виглядає використання даних алгоритмів у поєднанні з алгоритмами, що використовують семантичний аналіз тексту. Але це потребує побудови онтологій предметних областей, розробки

класифікації видів документів та визначення їх приналежності до певної предметної області.

ПЕРЕЛІК ПОСИЛАНЬ

1. Основні причини відмов ВАК України у видачі дипломів кандидата та доктора наук [Електронний ресурс]. – Режим доступу:

http://inmad.vntu.edu.ua/files/AnalyzeVAK_2010.doc. – Дата доступу: 14.04.2016.

2. Закон України «Про авторське право і суміжні права» № 3792-ХІІ від 23.12.1993 р.: редакція від 05.12.2012. – Режим доступу:

<http://zakon4.rada.gov.ua/laws/show/3792-12/page1>. – Дата доступу: 14.04.2016.

3. Зеленков Ю.Г. Сравнительный анализ методов определения нечетких дубликатов для WEB-документов / Зеленков Ю.Г., Сегалович И.В. // Труды 9-ой Всероссийской научной конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» RCDL'2007: Сб. работ участников конкурса. – Переславль-Залесский, Россия: «Университет города Переславля», 2007. – Т. 1. – С. 166–174.

4. Інтернет-сервіс «Google Translate» [Електронний ресурс]. – Режим доступу: <https://translate.google.com.ua/>. – Дата доступу: 14.04.2016.

5. Інтернет-сервіс «Sinonimus.ru» [Електронний ресурс]. – Режим доступу: <http://sinonimus.ru/>. – Дата доступу: 14.04.2016.

АНАЛІЗ МЕТОДІВ ПОРІВНЯННЯ ТЕКСТІВ У ВИПУСНИХ РОБОТАХ СТУДЕНТІВ

У даній статті розглянуто можливість застосування різних методів порівняння текстів у випускних роботах студентів вищих навчальних закладів. Розглянуто загальні принципи методів порівняння текстів, їх класифікацію, проведено їх аналіз. Визначено можливі умови застосування цих методів та обрано найбільш відповідний метод для вирішення поставленої задачі з урахуванням її специфіки.

This article discusses the possibility of using different methods of text comparison in student graduation works. Reviews general ideas of text comparison methods, their classification and demonstrates analysis of certain methods. Determined conditions for usages of each method and selected most sufficient method for resolving the problem with considering its specifics.

1. Вступ

Гостроіснуюча проблема плагіату у сучасній вищій освіті України суттєво впливає на якість знань молодих спеціалістів, які закінчують вищі навчальні заклади України. Це у свою чергу негативно впливає, як на виробничу так і на наукову діяльність. Нездатність молодих спеціалістів вирішувати потенційно нові завдання спонукає їх до запозичення та присвоєння вже існуючих методів, тим самим чином, що і під час навчання у вищому навчальному закладі.

Зазвичай, явище плагіату стає можливим завдяки відсутності ефективних засобів контролю за якістю студентських робіт, тобто перевірки та порівняння на їх на плагіат. Оскільки людина (викладач, науковий керівник) фізично не здатні ретельно перевірити усі можливі студентські роботи постає необхідність у розробці програмного засобу порівняння робіт.

Враховуючи специфіку випускних робіт студентів слід зазначити, що явище плагіату поширюється не лише на їх текстову складову, а і на зображення, рисунки, графіки, плакати, тобто і на супроводжуючий графічний матеріал.

За для більшої детальності, у цій статті будуть розглянуті лише методи, які відносяться до методів порівняння текстів, що можуть бути використані при вирішенні однієї з частин зазначеної проблеми.

2. Загальні принципи методів порівняння текстів

Усі сучасні методи порівняння текстів базуються на побудові певної метрики, яка відображає схожість двох текстів, які порівнюються. Тобто, задається певна шкала, наприклад 0-100, яка у чисельному еквівалентів відображає вірогідність схожості текстів, а при досягненні певної границі метод робить висновок, що скоріш за все мало місце запозичення матеріалів. Слід зазначити, що методи, які будуть наведені нижче не є ідеальними, а тому їх висновки щодо наявності запозичення слід трактувати радше як сигнал до можливої наявності запозичення, а не як остаточне рішення.

В загальному випадку процес порівняння текстів виглядає наступним чином:

1. нормалізація тексту, тобто приведення його до виду позбавленого усіх надлишкових атрибутів;
2. порівняння за певними рисами (певним методом) обраного тексту з наявними текстами у локальному сховищі чи у мережі Інтернет;
3. винесення висновків щодо наявності запозичення обраного тексту;
4. формування звітності придатної для інтерпретації результатів порівняння.

3. Класифікація методів порівняння текстів

Основними типами методів порівняння текстів вважаються:

1. локальні методи, які базуються на синтаксичному аналізу тексту;
2. глобальні методи, які базуються на основі знань щодо всіх документів, що зберігаються у сховищі.

До локальних методів відносяться методи дактилоскопії, які базуються на повних або часткових відбитках текстів.

Глобальні методи базуються на стилometriї – визначенні стилістики написання тексту або на аналізі шаблонів цитування тексту.

В останній час почали набувати популярність методи, які частково поєднують можливості обох типів методів, ці методи зазвичай глобально називають методами аналізу частоти терміну.

Очевидно, що методи локального типу не можуть надавати ефективних результатів, якщо за мету ставиться визначення плагіату, який може бути переказом, перекладом чи плагіатом ідей.

У той самий час за для вирішення цих завдань можуть бути використані глобальні методи, але слід розуміти, що ці методи дуже легко ввести в оману.

Тобто, вибір методу виявлення плагіату повинен залежати від передумов, цілей та завдань, які ставляться перед вирішенням завдань.

4. Аналіз методів порівняння текстів

Найпростіший метод хеш-відбитків полягає в розрахунку хеш-функції для всього тексту та її подальшому порівнянні з хеш-функціями інших текстів. Цей метод може бути використаний лише для порівняння ідентичних текстів, оскільки навіть зміна декількох символів призводить до зміни значення хеш-функції.

Його модифікаціями є методи Long Sentence та Heavy Sentence [1], які замість розрахунку хеш-функції для усього тексту розраховують їх лише для двох найбільших або «найважчих» речень у тексті. Ці методи хоча і є суттєвими модифікаціями, але все ще суттєво не підходять для вирішення реальних задач.

Метод аналізу частоти терміну [2] на відміну від попередніх методів використовує не речення, а слова для побудови хеш-функції. Відбір слів

відбувається наступним чином: для кожного слова розраховується його вага, яка виражається як частота повторення цього слова розділена на загальну кількість слів у тексті. Після чого обирається n слів з найбільшою вагою та з них формують рядок за яким обчислюється хеш-функція.

Існує декілька модифікацій цього методу, наприклад, метод TF-IDF, який враховує тільки ті документи в яких обране слово зустрічалося хоча б один раз. Метод TF-RIDF - базується на моделі розподілення Пуассона, згідно якої слова у розподілені випадково та незалежно, а тому вага слова повинна розраховуватись з урахуванням цього випадкового розподілу. Та I-Match - більш швидка модифікація TF-IDF. Ці методи дають значно кращі результати, але ці результати все ще не є достатньо надійними.

Серед інших методів слід виділити методи Bag-of-words [3], що базується на підходів опорних векторів та Longest Common Subsequence [4] - пошуку найбільшої послідовності, яка є підпослідовністю декількох різних послідовностей. Ці два методи є дуже ефективним для порівняння невеликих текстів, але використанні його на текстах великого розміру дає недостатні результати при великих часових витратах.

Також достатньо ефективним є метод Fingerprints [5], який він полягає у тому, що для кожного тексту у сховищі поступово за певним правилом обирають декілька рядків, які називаються «відбитками», та визначають місце знаходження цього відбитка у тексті. Ці пари «місце-відбиток» зберігаються з посиланням на текст. Потім при порівнянні текстів порівнюються ці пари «місце-відбиток» різних текстів і якщо хоча б одна ж пар співпадає – це може свідчити про запозичення тексту.

Використання нейронних мереж має великий потенціал у розпізнавання текстів, але існуючі методи практично не придатні для розпізнавання та порівняння текстів великих розмірів. Також значними проблемами є налаштування та подальше навчання нейронної мережі.

Методи застосування семантичних мереж базуються на алгоритмах на графах та вирішує завдання порівняння певних моделей, які відображають радше зміст тексту, а не сам текст. Такий підхід до вирішення завдань дозволяє виявляти плагіат на рівні переказу чи навіть на рівні плагіату думок, що є беззаперечною перевагою таких методів. З іншого боку, використання таких методів на текстах великого розміру, що відрізняються структурою практично неможливе.

Найбільш вживаним методом для вирішення проблеми порівняння текстів вважаються так звані метод шинглів [6,7] та його модифікації. З тексту поступово визначаються так звані «шингли» - рядки довжиною N слів, причому кожен наступний шингл зсувається на одне слово вперед. Таким чином перевіряється весь текст та формується множина шинглів для елементів якої розраховують хеш-функції. При порівнянні текстів обираються M випадкових хеш-функції шинглів для обох текстів та порівнюються між собою. Якщо кількість співпалих шинглів перевищує задану границю – можна вважати, що має місце запозичення тексту. Оскільки порівняння текстів саме цим методом

Список літератури

1. Hoad T. Methods for Identifying Versioned and Plagiarised Documents [Текст] / T. Hoad, J. Zobel // Journal of the American Society for Information Science and Technology 54 (3), 2003 – с. 203–215.
2. Ilyinsky S. An efficient method to detect duplicates of Web documents with the use of inverted index [Текст] / S. Ilyinsky, M. Kuzmin, A. Melkov, I. Segalovich. // Proceedings of the Eleventh International World Wide Web Conference. – Honolulu, Hawaii, 2002. – 4 с.
3. Park S. Analysis of Lexical Signatures for Finding Lost or Related Documents [Текст] / S.-T. Park, D. Pennock, C. Lee Giles, R. Krovetz // SIGIR'02. – Tampere, Finland, 2002. – с. 11–18.
4. Maier D. The Complexity of Some Problems on Subsequences and Supersequences

надає дуже якісні результати, основним напрямом модифікації цього методу є зменшення часових витрат при умові незначного погіршення результатів.

До таких них відносять дві основні модифікації: метод мегашинглів та логарифмічних шинглів. Сутність цих методів полягає у скороченні кількості шинглів, для яких будуються хеш-функції.

Висновки

Задля вирішення проблеми плагіату у вищих навчальних закладах необхідний цілий комплекс рішень. Одним з таких рішень може стати саме розробка засобу контролю за якістю студентських робіт. Цей засіб повинен враховувати специфіку випускних робіт студентів та перевіряти не лише текстову частину роботи, а й будь-який супроводжуючий її матеріал. Найбільш перспективним методом, що дозволить вирішити одну з частин завдання, а саме порівняння текстів, виглядає метод шинглів. Оскільки, об'єм випускних робіт студентів зазвичай складає не менше 10000 слів є необхідним розробка модифікації методу, що дозволить пришвидшити його роботу при роботі з текстами таких великих розмірів.

[Текст] / D. Maier // J. ACM (ACM Press) 25 (2) . – New York, NY, USA, 1978. – с. 322–336.

5. Stein B. Fuzzy-Fingerprints for Text-Based Information Retrieval [Текст] / B. Stein // Proceedings of the I-KNOW '05, 5th International Conference on Knowledge Management. – Graz, Austria, 2005. – с.572–579.

6. Broder A. Syntactic clustering of the Web. [Текст] / A. Broder, Glassman, M. Manasse, G. Zweig // Proceedings of the 6th International World Wide Web Conference. – SRC Technical Note 1997-015, 1997. – 13 с.

7. Broder A. On the resemblance and containment of documents. [Текст] / A. Broder // Compression and Complexity of Sequences (SEQUENCES'97). – IEEE Computer Society Washington, DC, USA, 1997. – с. 21–29

РЕАЛІЗАЦІЯ МЕТОДУ АВТОРЕГРЕСІЇ НА ОСНОВІ ТЕХНОЛОГІЇ NVIDIA CUDA

Запропоновано використання технології Nvidia CUDA для прогнозування числових рядів. Описано та реалізовано алгоритм авторегресії за допомогою технології CUDAfy .Net. Проаналізовано результати швидкості розрахунку прогнозу методом авторегресії на графічному(GPU) та центральному(CPU) процесорах.

An application of Nvidia Cuda technology for forecasting numerical series is proposed. An autoregression algorithm using CUDAfy .Net technology is described and implemented. The performance results of forecast calculation using autoregression method are analyzed for graphic (GPU) and central (CPU) processors.

1. Вступ

Задача прогнозування числових рядів є важливою складовою інтелектуального аналізу даних. Проаналізувавши статичні алгоритми, що складають основу оперативної аналітичної обробки даних(OLAP): метод Хольта-Вінтерса, метод авторегресії, метод простого рухомого середнього, стає зрозумілим, що для кожного метода потрібно побудувати певну кількість моделей і вибрати найкращу, по якій і буде робитися прогноз. Чим більше моделей перебрано, тим більша ймовірність кращого прогнозу. Для прискорення обрахунків доцільно розпаралелювати оцінку моделей. Використання розподілених систем є досить складним і вимагає додаткових затрат на устаткування. В той же час користувачі на побутовому рівні мають змогу використовувати графічні процесори Nvidia з підтримкою технології CUDA, яка має високу продуктивність паралельних обчислень.

2. Модель авторегресії

Для прогнозування числових рядів було обрано алгоритм авторегресії. Модель авторегресії(AR(p)) – це модель, в якій значення ряду в даний момент часу лінійно залежить від попередніх значень цього ж ряду. Порядок моделі(p) показує від скількох попередніх значень залежить поточне значення. Тобто авторегресійний процес визначається наступним чином:

$$X_t = c + \sum_{i=1}^p a_i X_{t-1} + \varepsilon_t$$

де a_1, \dots, a_p – параметри моделі(коефіцієнти авторегресії), c – стала, ε_t – білий шум, p – порядок моделі.

Для визначення якості моделі використовується корінь середньоквадратичної похибки:

$$RMSE = \sqrt{\frac{\sum_{t=1}^n (\hat{y}_t - y_t)^2}{n}}$$

3. Опис алгоритму

Крок 1 : Генерація числового ряду чи зчитування даних з файлу.

Крок 2 : Для кожного порядку моделі виконується розпаралелювання, тобто створюється кількість потоків рівна кількості порядків. В кожному потоці відбувається:

- Обчислення коефіцієнтів авторегресії.
- Визначення якості моделі – знаходження кореня середньоквадратичної похибки.

Крок 3 : Прогноз на N періодів вперед, використовуючи авторегресійну модель із найменшою похибкою.

Особливістю програмної реалізації є те, що генерація числового ряду відбувається на CPU, масове розпаралелювання процесів на GPU та обчислення власне прогнозу на CPU. Таким чином досягається максимальна ефективність описаного алгоритму.

4. Результати

Для демонстрації роботи алгоритму було згенеровано послідовність зі 100 випадкових чисел. Найбільш якісною виявилася модель порядку 14.

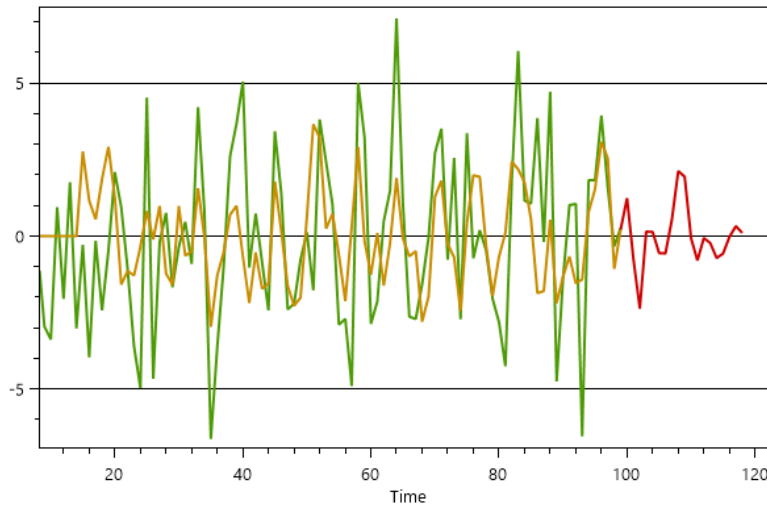


Рис. 1. Прогноз на 20 порядків, AR(14)

На рис.1. вхідні дані зображено зеленим кольором, модель авторегресії – оранжевим та прогноз на 20 порядків вперед – червоним.

В табл.1. наведено порівняльну характеристику часу знаходження моделі із найменшою похибкою та коефіцієнт

прискорення для вхідних даних різної довжини та максимально допустимого порядку авторегресії. Для порівняння швидкодії прогноз виконувався з використанням GPU(NVIDIA GeForce 960) і лише на CPU(Intel Core i5-6500 3.20 GHz).

Табл. 1. Порівняння швидкодії розрахунку на CPU та GPU

Довжина послідовності	Максимальний порядок	Час розрахунку на CPU, ms	Час розрахунку на GPU, ms	Коефіцієнт прискорення
100	16	1	2	0.5
	32	6	4	1.5
500	64	129	43	3
	128	483	87	5.551724138
1000	128	1007	179	5.625698324
	256	3820	442	8.642533937
2000	256	7889	900	8.765555556
	500	28544	2830	10.08621908

5. Висновки

Побудова та аналіз великої кількості моделей авторегресії потребує чималих затрат часу, тому було реалізовано розпаралелювання обчислень за допомогою технології Nvidia CUDA. Практичні результати показують, що зі збільшенням довжини вхідної послідовності та

збільшенням кількості порядків моделей, що аналізуються, збільшується ефективність використання технології CUDA. В даному випадку вдалося досягти зменшення часу обчислень в 10 разів при довжині послідовності 2000 та максимальному порядку моделі 500.

Список використаних джерел

1. Боресков А.В. Параллельные вычисления на GPU. Архитектура и программная модель CUDA: Учеб. пособие / Боресков А.В. и др. – М.: Издательство Московского университета, 2012. – 336 с
2. Васильев К.К. Служивый М.Н Математическое моделирование систем связи : учебное пособие. – Ульяновск : УлГТУ, 2008. – 170 с.
3. Светульников И.С., Светульников С.Г. Методы социально-экономического прогнозирования. – Санкт-Петербург : ГУЭФ, 2010. – 187-197с.
4. Електронний ресурс: <https://cudafy.codeplex.com/>

ГІБРИДНИЙ АЛГОРИТМ ШТУЧНОЇ БДЖОЛИНОЇ КОЛОНІЇ СКОМБІНОВАНИЙ З АЛГОРИТМОМ ІМІТАЦІЇ ВІДПАЛУ ДЛЯ РОЗВ'ЯЗАННЯ ЗАДАЧІ КОМІВОЯЖЕРА

Задача комівояжера представляє собою одну з найважливіших задач оптимізації і полягає у знаходженні Гамільтонова циклу мінімальної вартості. Для розв'язання цієї задачі в цій статті представлений гібридний алгоритм Штучної Бджолиної Колонії скомбінований з алгоритмом Імітації Відпалу. Імітація Відпалу використовується для покращення процесу генерації нового розв'язку в околі існуючого. Також у статті приводиться порівняльна характеристика розглянутого гібридного алгоритму та алгоритму оптимізації бджолиної колонії, що ілюструє ефективність першого.

Traveling Salesman Problem is one of the most important optimization issues and is about finding a Hamiltonian path with minimum cost. In order to solve this problem a hybrid Artificial Bee Colony (ABC) algorithm combined with Simulated Annealing (SA) algorithm is represented in this paper. SA metaheuristic is applied here to increase the quality of food sources generated by onlooker and working bees in phase of neighborhood search around the existing food source. A comparison study between Bee Colony Optimization (BCO) algorithm and proposed algorithm shows that hybrid ABC algorithm outperforms BCO.

1. Вступ

Задача комівояжера (Traveling Salesman Problem, TSP) – одна з найбільш широко досліджених задач комбінаторної оптимізації. Спочатку задача комівояжера була сформульована для сфери маркетингу, а пізніше вона знайшла застосування і в інших галузях управлінської діяльності, особливо там, де мала місце значне територіальне розосередження об'єктів на місцевості. До прикладів практичного використання даного класу задач можна віднести визначення оптимальних маршрутів обслуговування технічних об'єктів, обслуговування територіально розосереджених технічних об'єктів, об'єкту експедицій та скидання відповідних вантажів.

Карабога вперше запропонував Алгоритм Штучної Бджолиної Колонії (Artificial Bee Colony, ABC) для розв'язання задач чисельної оптимізації [1]. Пізніше, комбінаторна версія цього алгоритму була представлена на прикладі

розв'язання узагальненої задачі про призначення [2].

В цій статті представлений новий підхід до моделювання поведінки бджіл-спостерігачів та робочих бджіл під час пошуку їжі, що, враховуючі експериментальні результати, значно покращує базовий алгоритм ABC.

2. Задача комівояжера

Основна ідея задачі комівояжера [3] полягає у наступному: дано множину міст, а також відстань між усіма можливими парами міст. Необхідно знайти шлях, який пролягає через усі міста та повертається у початкове, окрім того сумарна довжина пройденого шляху має бути мінімальною. Тобто необхідно побудувати маршрут таким чином, щоб побувати в кожному місті по одному разу і повернутися у початкове. Математична постановка задачі комівояжера має наступний вигляд:

$$\sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \rightarrow \min \quad (1)$$

$$\sum_{j=1}^n x_{ij} = 1, (i = 1, 2, \dots, n) \quad (2)$$

$$\sum_{i=1}^n x_{ij} = 1, (i = 1, 2, \dots, n) \quad (3)$$

де c_{ij} — матриця відстаней між усіма містами.

Якщо в моделі задачі обмежитися лише умовами (2) і (3), то вона буде еквівалентною задачі про призначення, план якої не обов'язково повинен бути циклічним. Тобто, маршрут комівояжера може розпастися на декілька незв'язних між собою циклів, тоді як насправді він повинен складатися з одного циклу. Щоб забезпечити цю вимогу введемо наступне обмеження:

$$u_i - u_j + nx_{ij} \leq n - 1, \quad (4)$$

$$i, j = 1, 2, \dots, n, i \neq j$$

Тобто, якщо комівояжер пересувається з i -го в j -те місто, то нерівність (4) фіксує порядкові номери цих міст.

Отже математична постановка задачі комівояжера полягає у мінімізації функції (1) при обмеженнях (2), (3) і (4).

3. Алгоритм Штучної Бджолоїної Колонії

Алгоритм штучної бджолоїної колонії (ABC) базується на симуляції поведінки медоносних бджіл під час пошуку їжі [4]. Штучна бджолоїна колонія складається з трьох видів бджіл, що відрізняються одна від одної своєю функцією: робочі бджоли, бджоли-розвідники та незайняті бджоли-спостерігачі. Кожна робоча бджола працює виключно з одним джерелом їжі. Вона зберігає в пам'яті інформацію про це джерело, коли відлітає з вулика, і ділиться цією інформацією з бджолами-спостерігачами в так званій зоні для танців. Після того, як спостерігачі подивляться танці робочих бджіл, вони обирають джерело їжі, яке в подальшому будуть намагатися покращити. Якщо робоча бджола лишає своє джерело їжі, вона перетворюється на розвідника і починає випадковим чином шукати нові джерела.

В алгоритмі ABC кожне джерело їжі відповідає можливому розв'язку задачі, а

кількість нектару, що містить джерело - значенню фітнес функції відповідного розв'язку. Як відомо, на меті оптимізації розв'язку задачі комівояжера є мінімізація довжини повного обходу всіх міст. Якість кожного розв'язку fit_i розраховується наступним чином:

$$fit_i = \frac{1}{f(x_i)} \quad (5)$$

де $f(x_i)$ - це довжина маршруту для розв'язку x_i .

Максимальна кількість розвідників l обчислюється за формулою:

$$l = \frac{cs * D}{3} \quad (6)$$

де cs і D дорівнюють розміру колонії та розмірності задачі відповідно ($D = n$).

Детально комбінаторний варіант алгоритму ABC наведений нижче:

1. Ініціалізація параметрів: розмір колонії cs , максимальна кількість ітерацій $MaxNumber$, максимальна кількість оновлення одного розв'язку $limit$.
2. Ініціалізація початкових розв'язків (позицій джерел їжі) $x_i, i = 1, 2, \dots, cs/2$. Присвоїти кожному $limit_i = 0$.
3. Розрахунок цільових функцій для наявних розв'язків.
4. Запам'ятати найкращий розв'язок.
5. $k = 0$ (лічильник кількості ітерацій).
6. Кількість робочих бджіл $cs/2$ / 2 кількість спостерігачів $cs/2$.
7. ПОВТОРЮВАТИ
 - a. Фаза робочих бджіл: для кожної робочої бджоли:
 - i. Генерація нового розв'язку v_i в околі x_i та розрахунок значення цільової функції для нього. $limit_i = limit_i + 1$.
 - ii. Обрати кращий з v_i та x_i , керуючись жадібним підходом
 - b. Розрахувати ймовірності P_i для знайдених попередньо розв'язків

$$P_i = \frac{0,9 * fit_i}{fit_{best}} + 0,1$$

- с. Фаза спостерігачів: для кожного спостерігача:
- i. Генерація нового розв'язку v_i з x_i , в залежності від P_i та розрахунок значення цільової функції для нього. $limit_i = limit_i + 1$.
 - ii. Обрати кращий з v_i та x_i , керуючись жадібним підходом.
- d. Запам'ятати найкраще зі знайдених розв'язків.
- e. Визначити покинуті розв'язки ($limit_i \geq limit$), і якщо вони існують замінити їх новими.
8. $k = k + 1$.
9. ПОКИ не $k = MaxNumber$.

4. Гібридний алгоритм штучної бджолої колонії скомбінований з алгоритмом імітації відпалу для розв'язання задачі комівояжера

Під час фази робочих бджіл і фази спостерігачів є крок (крок б), під час якого бджоли генерують новий розв'язок із вже існуючого. В класичному варіанті ABC цей етап можна реалізувати, використовуючи перестановки, що є не зовсім ефективним підходом. Тому замість цього, кожен раз при генерації нового розв'язку використовувати метаевристику Симуляції Відпалу.

Для заданої ЗКО (1) позначимо через $N(x) \subset X$ - окіл розв'язку x_i а через $X_{opt} = \{x_{opt}; f(x_{opt}) \leq f(\hat{x}), \hat{x} \in X\}$ - множину всіх оптимальних розв'язків задачі. Враховуючі це, алгоритм імітації відпалу можна записати наступним чином:

1. Ініціалізація початкового розв'язку $x \in X$. Ініціалізація температур $T_0 > T_f > 0$. Ініціалізація лічильника $i = 0$.
2. Генерація (випадковим чином) розв'язку $x^* \in N(x)$. Обчислення $\Delta f = f(x^*) - f(x)$.
3. Якщо задовольняється "критерій Метрополіса", $\min\{1, \min(-\frac{\Delta f}{T_i})\} > \eta \in [0,1]$, тоді $x = x^*$.
4. Якщо для поточної температури T_i виконується умова рівноваги, тоді - крок 5, інакше - крок 2.
5. Якщо не досягнуто критерію зупинки, наприклад $T_i > T_f$ тоді зменшення температури $T_{i+1} = T_i - \Delta T_i, \Delta T_i > 0$. $i = i + 1$. Повернутися на крок 2. Інакше $x_{opt} = x$.
6. Кінець.

5. Результати симуляції

Ефективність гібридного ABC визначалась шляхом розв'язання деяких задач, отриманих з TSPLIB [5]. Результати представлено у таблиці 1, де за *optimum* приймається найкращий з відомих розв'язків, за *best* - найкращий зі знайдених, а за *mean* - середнє арифметичне всіх розв'язків (процес пошуку розв'язку біло виконано 10 разів для кожної задачі). Значення відносної похибки розраховується за наступною формулою:

$$error = \frac{(mean - optimum)}{optimum} * 100\% \quad ($$

Параметри алгоритму: розмір колонії $cs=100$, значення *limit* та *MaxNumber* залежить від розмірності задачі і є змінним.

Табл. 1. Результати симуляції роботи гібридного ABC

Задача	Optimum	Best	Mean	Error(%)	limit	MaxNumber
Eli51	426	426	426	0.00	20	150
St70	638	638	638	0.00	20	250
Kroc100	21282	21282	21296.1	0.57	20	350
Pr1144	26524	26524	26785.8	0.99	35	500
Lin318	42029	42120	42336.5	0.73	35	700
Pcb442	50778	51098	51223.5	0.88	50	800

У таблиці 2 представлені параметри SA в залежності від розмірності задачі.

Табл. 2. Параметри SA

Задача	Початкова температура
Eli51	500
St70	1000
Kroc100	1000
Pr1144	1000
Lin318	2000
Pcb442	5000

У таблиці 3 представлено порівняння гібридного ABC з VCO [6]. Параметри VCO: розмір бджолоїної популяції дорівнює кількості міст, лічильник максимальної кількості ітерацій алгоритму 500. Значення в кожній колонці означає відсоткове відношення між середнім і оптимальним розв'язком.

Табл. 3. Порівняння гібридного ABC з VCO

Задача	Гібридний ABC	VCO
Eli51	0.00	0.84
Kroc100	0.57	2.56
Pr1144	0.99	2.21
Lin318	0.73	7.16

6. Висновки

В основі гібридного алгоритму ABC була покладена поведінка медоносних бджіл під час пошуку їжі. Роботу та ефективність даного алгоритму було протестовано на прикладі розв'язання класичної задачі комівояжера. Згідно з результатами симуляції, гібридизація стандартного алгоритму ABC за рахунок внесення змін у фази генерації нового розв'язку із вже існуючого, позитивно вплинула на ефективність алгоритму, про що свідчать результати порівняння гібридного ABC з VCO.

Література

1. D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Technical Report-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.
2. D. Karaboga and B. Gorkemli, "A combinatorial artificial bee colony algorithm for traveling salesman problem," in Proceedings of INISTA 2011: International Symposium on Innovations in Intelligent Systems and Applications, June 2011, 50-53.
3. Меламед, И., Сергеев, С., Сигал, И. (1989). Задача коммивояжера. Вопросы теории. Автомат. И Телемех., (9), 3-33.
4. Karaboga D, Basturk B (2007a) Artificial bee colony (abc) optimization algorithm for solving constrained optimization problems. In: Proceedings of the 12th international fuzzy systems association world congress on foundations of fuzzy logic and soft computing. Springer, Berlin, IFSA '07, 789–799.
5. G. Reinelt, TSPLIB—A traveling salesman problem library, ORSA Journal on Computing, 3, 1991, 376–384.
6. Wong L. P., Low, M.Y.H. and Chong, C.S., A bee colony optimization algorithm for traveling salesman problem, in Proceedings of Second Asia International Conference on Modeling and Simulation, 2008, 818-823.

ДІАГНОСТИКА ЕМОЦІЙНОГО СТАНУ КОРИСТУВАЧА ПЕРСОНАЛЬНОГО КОМП'ЮТЕРА ЗА РУХАМИ МИШКОЮ

Анотація. У даній статті розглядається метод розпізнавання емоцій людини за рухами мишки за допомогою лінгвістичного моделювання з використанням апарату прихованих марковських моделей. Розглянені підход до аналізу та інтервалізації часових рядів, описані процеси лінгвітизації числових рядів та їх подальшого відтворення формальних граматик.

Annotation. This article considers the method of recognizing human emotions by manipulation of computer mouse with help of linguistic modeling using hidden Markov models. Was reviewed approach to analysis and intervalisation of time series, was described processes of lingvistication of number series and next representation of formal grammars.

Ключові слова: ЛІНГВІСТИЧНЕ МОДЕЛЮВАННЯ, АНАЛІЗ ЧАСОВИХ РЯДІВ, ЕМОЦІЙНЕ НАВАНТАЖЕННЯ

Постановка та актуальність проблеми

На сьогоднішній день життя людини дуже тісно пов'язане із розвитком інформаційних технологій. Результатом цього всезагальне використання персональних комп'ютерів (ПК) у найрізноманітніших сферах життя. Вони являються знаряддям праці або ж об'єктом праці. Швидкими темпами відбувається автоматизація різноманітних виробництв та впроваджуються системи контролю і управління на підприємствах. У зв'язку з цим постає необхідність частоті постійної напруженої роботи персоналу з ПК. Найчастіше це викликає фізичне та емоційне виснаження, що, у свою чергу, призводить до зменшення ефективності та продуктивності роботи, а, у певних випадках, навіть, може спричинити виникнення небезпечних ситуацій на виробництві.

Виходячи з усього вищесказаного можна зробити висновок, що необхідно оцінювати та контролювати емоційний стан користувача ПК для запобігання нервового перенавантаження та стресового стану організму. Одним із найпростіших способів ідентифікації емоцій являється розпізнавання їх за допомогою виявлення безумовних рефлексів як реакції на поточний стан організму, в нашому випадку – безсвідомих мікрорухів руки.

Для виконання даної задачі чудово підходить запропонований метод.

Перевагою такого підходу являється відсутність необхідності використання додаткового спеціалізованого устаткування і можливість реалізації поставленої мети за допомогою звичайного ПК у типовій комплектації.

При розпізнаванні емоційного стану варто оцінювати такі параметри як точність та швидкість реакції, що залежить від просторово-часового компоненту. Тому певний емоція може бути представлена у вигляді часового ряду рухів комп'ютерною мишкою. В результаті аналізу часових рядів були побудовані гібридні моделі на лінгвістичних та прихованих Марковських моделей, кожна з яких відповідає певному емоційному стану людини. У роботі були розглянуті 6 основних емоцій: 1) радість, 2) гнів, 3) подив, 4)страх, 5)смуток, 6)відраза.

Ідентифікація поточного емоційного стану за допомогою лінгвістичних моделей

Головним завданням лінгвістичного моделювання є перетворення чисельних рядів, експериментальних даних, багатомірних даних до лінгвістичних послідовностей та відновлення за ними формальної граматики мови [1]. Головна ідея переходу від числового ряду до символічного, від якої залежить результат лінгвістичного моделювання, базується на розбитті множини вхідних чисельних даних на коректні інтервали.

Для кожної пари станів, наприклад <d,S> підраховуємо $v_{d,S}$ скільки разів вона зустрічається в лінгвістичному ланцюжку

$L(x_{11}), \dots, L(x_{1M})$. На рис.3 наведений приклад сформованої матриці.

TABLE X:		i	h	g	f	e	d	c	b	a	A	B	C	D	E	F	G	H	I
i	39	0	0	0	0	0	0	0	4	0	0	9	0	0	0	0	0	0	0
h	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
g	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
f	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
e	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
d	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
c	4	0	0	0	0	0	0	23	11	0	0	8	0	0	0	0	0	0	0
b	0	0	0	0	0	0	0	15	215	0	0	8	0	0	0	0	0	0	0
a	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	9	0	0	0	0	0	0	8	8	0	0	253	14	0	0	0	0	0	5
C	0	0	0	0	0	0	0	0	0	0	0	19	11	0	0	0	0	0	0
D	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
G	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
H	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0

Рис.3 – Матриці передування

8. На основі отриманої таблиці тепер можна сформулювати правила даної граматики. Поділивши $V_{d,s}$ на загальну кількість входжень літери “d” w_d отримуємо частоту переходів зі стану “d” в стан “S”:

$$v(d \rightarrow S) = \frac{V_{d,s}}{w_d}$$

Правила визначаються лише для тих випадків, коли значення комірки є ненульовим – тобто можливий перехід із одного стану в інший. На рис.4 наведений приклад сформованої формальної граматики.

LINGUA RULES X:	
i	- (0.75) -> i
i	- (0.08) -> b
i	- (0.17) -> B
c	- (0.09) -> i
c	- (0.5) -> c
c	- (0.24) -> b
c	- (0.17) -> B
b	- (0.06) -> c
b	- (0.9) -> b
b	- (0.03) -> B
B	- (0.03) -> i
B	- (0.03) -> c
B	- (0.03) -> b
B	- (0.85) -> B
B	- (0.05) -> C
B	- (0.02) -> I
C	- (0.63) -> B
C	- (0.37) -> C
I	- (1.0) -> C

Рис.4 – Правила формальної граматики

Сформована граматика визначає до якого класу відноситься поточний

емоційний стан людини, що був представлений у вигляді часового ряду.

Список літератури:

1. Баклан І.В. Лінгвістичне моделювання: основи, методи, деякі прикладні аспекти//Системні технології. Регіональний міжвузівський збірник научних робіт. – Випуск3(74) – Дніпропетровськ, 2011. – с.10 – 19.
2. Баклан І.В. Інтервальний підхід до побудови лінгвістичної моделі// Системні технології. Регіональний міжвузівський збірник научних робіт. – Випуск3(86) – Дніпропетровськ, 2013. – с.3 – 8.
3. Канторович, Л.В. О некоторых новых подходах к вычислительным методам и обработке наблюдений. //Sibirskij matematičeskij žurnal – 1962. – Т.3№.5 – с. 701-709.
4. Баклан І. В. Імовірнісні моделі для аналізу та прогнозування часових рядів / І. В. Баклан, Г. А. Степанкова // Искусственный интеллект. - 2008. - № 3. - С. 505-515.

ЗАСТОСУВАННЯ МУРАШИНОГО АЛГОРИТМУ ДЛЯ РОЗВ'ЯЗАННЯ ЗАДАЧІ СИНТЕЗУ НАДІЙНОЇ МЕРЕЖІ

В даній статті розглянуто задачу синтезу надійної мережі. Розглядається задача Штейнера про 2-зв'язний граф, як штейнерову версію задачі проектування 2-зв'язної мережі мінімальної вартості. Наводиться опис мурашиного алгоритму та наведено застосований алгоритм для розв'язання задачі синтезу надійної мережі.

In this article reviewed the reliable network synthesis problem. The problem of Steiner 2-connected graph as Steiner's version of the 2-connected network design problem with minimum cost. Review of the ant algorithm and ant algorithm that used to solve the reliable network synthesis problem.

1. Вступ

Задачі синтезу мереж є математичними моделями чисельних економіко-математичних проблем. Одним з важливих для практики класів задач є задачі оптимального синтезу надійних мереж [1]. На відміну від звичайних задач синтезу мереж в них передбачається, що елементи системи можуть час від часу виходити з ладу, порушуючи нормальне функціонування системи. Припускається, що елементи системи виходять з ладу в певний момент часу, але в той же час не може вийти з ладу не більше одного елемента.

2. Постановка задачі Штейнера про 2-зв'язний граф

Розглянемо неорієнтований граф $G = (V, E)$, де V – множина його вершин, а E – множина ребер. Для кожної пари вершин s, t з множини V шляхом між цими (зв'язаними) вершинами називається послідовність вершин та ребер $s = v_0, e_1, v_1, \dots, v_{l-1}, e_l, v_l = t$, де кожне ребро e_i з'єднує вершин $v_{i-1}, v_i, i = 1, \dots, l$ і жодні вершини чи ребро не зустрічаються в цій послідовності більше, ніж один раз. Коли говорять, що граф $G' = (V', E')$ є підграфом G , це означає, що $V' \subset V$, а множина E' складається з ребер, що належать E , причому їх кінцеві вершини знаходяться в V' .

$N_0 = \frac{N}{\{s\}}$; $\Gamma(i)$ – множина ребер з однією вершиною i на графі, що утворений з графа G шляхом видалення ребер під графа Γ .

Припустимо, що задано множину термінальних вузлів N ($N \subset V$) та невід'ємні вартості c_e для кожного з ребер e множини E . Задано також неорієнтований граф $H = (V_h, E_h)$, де V_h та E_h позначено відповідно множини вершин та ребер графа H . Іноді термінальні вузли з N називають просто вузлами. Вартість будь-якого підграфа $G' = (V', E')$ графа G дорівнює сумі вартостей ребер у множині E' .

Тоді задача Штейнера про 2-зв'язний граф має вигляд [2]:

Цільова функція:

$$\min \sum_{e \in E} c_e x_e, \quad (1)$$

Обмеження:

$$\sum_{j \in \Gamma(i)} x_{ij}^r - \sum_{j \in \Gamma(i)} x_{ji}^r = \begin{cases} 2, & i = s, \\ 0, & i \neq s, r, \\ -2, & i = r, i \in V, r \in N_0 \end{cases} \quad (2)$$

$$0 \leq x_{ij}^r \leq x_{ij} \leq 1, r \in N_0, e = (i, j) \in E, \quad (3)$$

$$x_e = 0 \cup 1, e \in E. \quad (4)$$

3. Опис мурашиного алгоритму

Мурашині алгоритми являють собою імовірнісну жадібну евристику, де ймовірності встановлюються, виходячи з інформації про якість рішення, отриманої з попередніх рішень. Вони можуть використовуватися як для статичних, так і для динамічних комбінаторних оптимізаційних задач. Збіжність гарантована, тобто в будь-якому випадку ми

отримаємо оптимальне рішення, однак швидкість збіжності невідома.

Ідея мурашиного алгоритму - моделювання поведінки мурах, пов'язаного з їх здатністю швидко знаходити найкоротший шлях від мурашника до джерела їжі і адаптуватися до мінливих умов, знаходячи новий найкоротший шлях. При своєму русі мураха мітить шлях феромоном, і ця інформація використовується іншими мурашками для вибору шляху. Це елементарне правило поведінки і визначає здатність мурах знаходити новий шлях, якщо старий виявляється недоступним.

Основні формули мурашиного алгоритму [3]:

- кожне рішення або хід m засноване на двох компонентах: жадібній силі $GF(m)$, яка є короткостроково привабливою для мурахи, і стежки $Tr(m)$, які представляють собою інформацію, отриману від інших мурах. Нехай M безліч всіх можливих ходів. Ймовірність $p_k(m)$, що мураха k вибирає m задається імовірність переходу з вершини у вершину в момент часу:

$$p_k(m) = \frac{GF(m)^\alpha Tr(m)^\beta}{\sum_{m' \in M_k(adm)} GF(m')^\alpha Tr(m')^\beta}, \quad (5)$$

де α і β є параметрами і $M_k(adm)$ є безліч допустимих ходів, які мураха k може виконувати;

- оновлення феромону:
 $Tr(m) = p \times Tr(m) + (1 - p) \times \Delta Tr(m), \forall m \in M \quad (6)$

4. Застосування мурашиного алгоритму для розв'язання задачі синтезу надійної мережі

Нехай N буде кількість мурах в даній популяції. Кількість мурах дорівнює кількості термінальних вершин, кожна з них починає рух зі своєї випадково обраної термінальної вершини. Роль однієї мурахи k є побудова рішення s_k крок за кроком, починаючи з порожнього рішення s_k . На кожен крок, як це було запропоновано в роботі [4], ми виконуємо рух шляхом підключення термінальних вершин $x \notin s_k$ до s_k , використовуючи найкоротший шлях між x і s_k . Таким чином, для визначення жадібної сили термінальної вершини $x \notin s_k$ є:

$$GF(x) = \frac{1}{SP(x, s_k)}, \quad (7)$$

В кінці кожного покоління, феромон глобально оновлюється наступним чином:

$$t(v) = (1 - p_g) \times t(v) + p_g \times \Delta t(v), \forall v \notin R, \quad (8)$$

де $p_g \in [0;1]$ – коефіцієнт випаровування. Підсилення $\Delta t(v)$ оновлюється наступним чином:

$$\Delta t(v) = \sum_k \Delta t_k(v), \quad (9)$$

де підсумовування виконується тільки для N_{best} кращих мурах поточного покоління (N_{best} мурахи з найменшим значенням цільової функції f), де N_{best} є параметром.

$$\Delta t_k(v) \begin{cases} \frac{1}{f(s_k)}, v \in s_k; \\ 0, \text{інакше.} \end{cases} \quad (10)$$

Отже, якщо $t(v)$ є великим, це означає, з одного боку, що більшість з N_{best} кращих мурах поточного покоління вибрали v , і з іншого боку, що ці рішення мають менші значення f в порівнянні для інших рішень, побудованих протягом поточного покоління.

На кожному кроці побудови рішення одною мурашкою, генерується випадкове число r в $[0;1]$. Якщо $r < p$, обраний крок вибирається відповідно до рівняння (5). Також мураха не може повернутись у вершину, яку вона вже пройшла, тому кожна мураха має свій список заборонених вершин.

Крім того, на самому початку процесу, всі значення $t(v)$ задаються 0,5. Таким чином, тільки жадібна сила буде направляти вибір першого мурашки в першому поколінні. Тепер є всі компоненти, які необхідні для розробки евристики для задачі Штейнера про 2-зв'язний підграф.

Знаходження першого маршруту:

Крок 1.1. Ініціалізація $t(v) = 0.5, \forall v \notin R$;

Крок 1.2. Ініціалізація параметрів: N, p_h, α, β ;

Крок 1.3. Встановити $f^* = \infty$;

Крок 1.4. Поки не досягнуто 500 поколінь або 10 поколінь без покращення f^* перейти на крок 5, інакше перейти на крок 1.1.2;

Крок 1.5. Поки $k \leq N$ перейти на крок 1.6, інакше перейти на крок 1.9;

Крок 1.6. Ініціалізація s_k з випадково обраною термінальною вершиною z ;

Крок 1.7. За імовірнісним правилом, мураха випадково обирає вершину j для подальшого руху з i ;

Крок 1.8. З'єднати термінальну вершину x до s_k ;

Крок 1.9. Оновити феромон на ребрі ij ;

Крок 1.10. Повторювати кроки 1.7-1.9 доти, поки всі термінальні вершини не будуть в s_k ;

Крок 1.11. Якщо $f(s_k) < f^*$, $f^* = f(s_k)$ і $s^* = s_k$;

Крок 1.12. Оновлення феромону (глобально);

Крок 1.13. Повернути перший маршрут s^* зі значенням f^* .

Знаходження другого маршруту:

Крок 2.1. Забороняємо мурахам рухатись по першому маршруту;

Крок 2.2. Поки $k \leq N$ перейти на Крок 2.3, інакше перейти на Крок 2.4

Крок 2.3. Виконати Кроки 1.6 – 1.10;

Крок 2.4. Повернути другий маршрут s^* зі значенням f^* .

5. Висновок

Було розглянуто задачу синтезу надійної мережі, а саме штейнерову версію задачі проектування 2-зв'язної мережі мінімальної вартості. Також було запропоновано застосування мурашиного алгоритму для розв'язання вище згаданої задачі, а також наведено сам алгоритм побудови рішення.

Список літератури

1. Михалевич В.С., Трубин В.А., Шор Н.З. Оптимизационные задачи производственно транспортного планирования. Модели, методы, алгоритмы. –М.: Наука, 1986. – 260 с.
2. Шор Н. З., Сергієнко І. В. та інші. Задачі оптимального проектування надійних мереж. — К.: Наук. думка, 2005. — 230 с.
3. Штовба С.Д. Муравьиные алгоритмы // Exponenta Pro. Математика в приложениях, 2003, №4, с.70-75.
4. Takahashi, H., and Matsuyama, A.: An approximate solution for the steiner problem in graphs. Math. Japonica 24(6) (1980) 573–577

ОПТИМІЗАЦІЯ ПЕРЕДАЧІ АСК-ІВ У PEER-TO-PEER МЕРЕЖАХ З ВУЗЬКИМ КАНАЛОМ ЗВ'ЯЗКУ

У статті розглянуто оптимізацію алгоритму передачі АСК-ів, повідомлень-підтверджень про успішну доставку пакетів даних у Peer-to-Peer мережах з вузьким каналом зв'язку. Було запропоновано не надсилати АСК-и миттєво, а накопичувати їх на пірі для надсилання разом з іншими вихідними повідомленнями та групою у окремому повідомленні через певний період часу. Продемонстровано результати тестування системи: досягнуто значне зменшення навантаження на мережу та зменшення відсотку втрачених пакетів при передачі.

The subject of this article is an optimization of the algorithm of transmission of ACKs, the confirmation of successful delivery of data packets in a Peer-to-Peer networks with narrow communication channel. It was suggested not to send ACKs immediately after receiving of message, but to accumulate them in peer and send with other outgoing messages and in a separate message after some period of time. Test results demonstrate that the network load and the percentage of packets lost in transmission were significantly reduced.

1. Вступ

При передачі повідомлень у багатьох ситуаціях необхідно знати, чи дійшло повідомлення його адресату. Для цього у відповідь зі сторони адресата надсилається повідомлення з підтвердженням доставки – АСК (від англ. «acknowledgement») [1]. У військових цілях для передачі повідомлень використовується радіомережа. Її особливістю є те, що вона зазвичай має архітектуру Peer-to-Peer [2], де незалежним вузлом є окремо взята рація. Її особливістю є вузький канал зв'язку та обмін повідомленнями у режимі бродкасту. Це означає, що одночасно лише одна рація може передавати дані, а усі інші їх приймають. Також, перед відправкою кожного повідомлення йде преамбула, під час якої перевіряється чи не зайнятий канал зв'язку і, якщо він вільний, надсилається сповіщення іншим вузлам про передачу нового повідомлення та зайняття каналу. У доволі розповсюдженій у Збройних силах України рації Motorola DP4400 ця процедура займає близько 800 мс [3]. За таких умов, коли постає питання про передачу даних через радіомережу, виникає проблема з відправкою АСК-ів, адже якщо надсилати їх одразу на кожен отриманий пакет, мережа буде весь час під великим навантаженням. Тому, є необхідність у оптимізації передачі АСК-ів у

радіомережах для зменшення відсотку втрачених пакетів.

2. Огляд існуючих рішень

На сьогодні в Україні є два відомих комплекси для військових – це геоінформаційні системи «АРТА» [4] та «Кропива» [5]. Вони можуть використовувати як мережу Wi-Fi, так і радіозв'язок для передачі даних (координати, цільовказівки, накази тощо) з одного планшету на інший. Однак ці системи не мають оптимізований алгоритм передачі АСК-ів. Тому, фактично, розглянута у статті оптимізація є інноваційним рішенням поставленої задачі.

3. Концепція накопичувальних АСК-ів

Для досягнення поставленої мети була продумана концепція накопичувальних АСК-ів. Її основні положення полягають у наступному:

- АСК можуть бути упаковані групою.
- Будь-яке вихідне повідомлення може містити у собі групу АСК-ів, під яку виділяється певне місце.
- До повідомлення можуть бути включені АСК, не призначені його отримувачу. Вони будуть ретрансльовані у контейнерах вихідних повідомлень отримувача. Таким чином, кожен може виступати посередником у доставці АСК.

- Вже відправлений АСК може бути згодом продубльований у контейнерах інших вихідних повідомлень.
 - Найвищий пріоритет мають наші нові (ні разу не відправлені) АСК; далі йдуть чужі (отримані від інших, але призначені не нам) нові; на третьому та четвертому місці – наші дублікати та чужі дублікати відповідно.
 - Якщо у черзі є наші нові АСК, вони формуються у групу та відправляються у окремому повідомленні. Залишене місце заповнюється іншими типами АСК-ів у відповідності до їх пріоритетів. Наступне таке повідомлення може бути лише через певний проміжок часу.
 - Якщо ми отримали підтвердження про доставку повідомлення, у складі якого була група АСК-ів, вони видаляються з черги.
- У кожного типу АСК є свій час життя, зі сплином якого АСК буде видалений з черги.

Описані положення забезпечують мінімальну за кількістю окремих пакетів та водночас надійну відправку АСК-ів.

4. АСК у складі вихідних повідомлень

У кожному пакеті даних, що передаються, можна виділити певне місце під групу вкладених АСК-ів. Наприклад, якщо умовно взяти за максимальний розмір одного пакету `sizeofPart` дорівнює 500 байт, то під цю групу можна було б відвести, скажімо, `maxAcksPerPacketSize` 40 байт. При цьому, якщо тіло самого повідомлення `msgBody` займає від 460 до 500 байт, для АСК-ів доцільно надати тільки залишок місця (рис. 1). Це пояснюється тим, що при виділенні 40 байт, повідомлення було б необхідно розбити на два пакети, що суперечить цілям оптимізації.

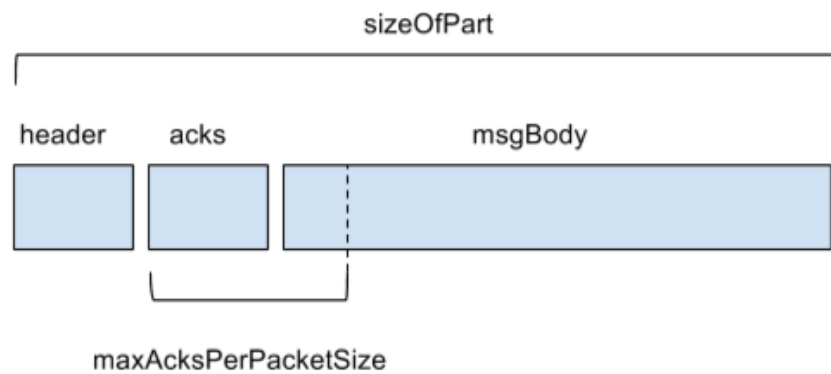


Рис. 1. Виділення місця у пакеті під групу АСК

Враховуючи пріоритети, АСК максимально заповнюють виділене місце. При цьому, вони не мають повторюватися у рамках одного повідомлення. В залежності від особливостей протоколу передачі даних, кількість вкладених АСК-ів, або їх розмір, має бути записана у заголовку пакета.

Якщо у відповідь на відправлене повідомлення ми отримали АСК, усі вкладені у нього АСК вважаються успішно доставленими та мають бути видалені з черги.

5. Відправка окремого повідомлення з групою АСК-ів

З певним інтервалом часу можна формувати групу з наших нових АСК-ів та відправляти адресатам цієї групи окремих пакет. Якщо розмір пакету не перевищує максимальний допустимий, залишене вільне місце можна заповнити АСК інших типів

згідно з їхніми пріоритетами. Цей окремих пакет необхідно обов'язково відправляти у ситуації, коли ми отримали повідомлення і не маємо у черзі на відправку жодних повідомлень. Якщо у черзі немає наших нових АСК-ів, пакет не буде сформований, щоб не забивати радіоефір.

6. Результати випробування системи накопичувальних АСК-ів

Дана концепція була запропонована, розроблена та успішно інтегрована у геоінформаційну систему «Кропива». Тестування її ефективності проводилися на чотирьох раціях: дві портативні рації Motorola DP4400 та дві автомобільні рації «Либідь». Було проведено два десятихвилинні навантажувальні тестування з однаковими налаштуваннями автоматичної відправки повідомлень (розмір пакетів,

інтервал відправки тощо). Результати тестування зображені на рисунку 2.

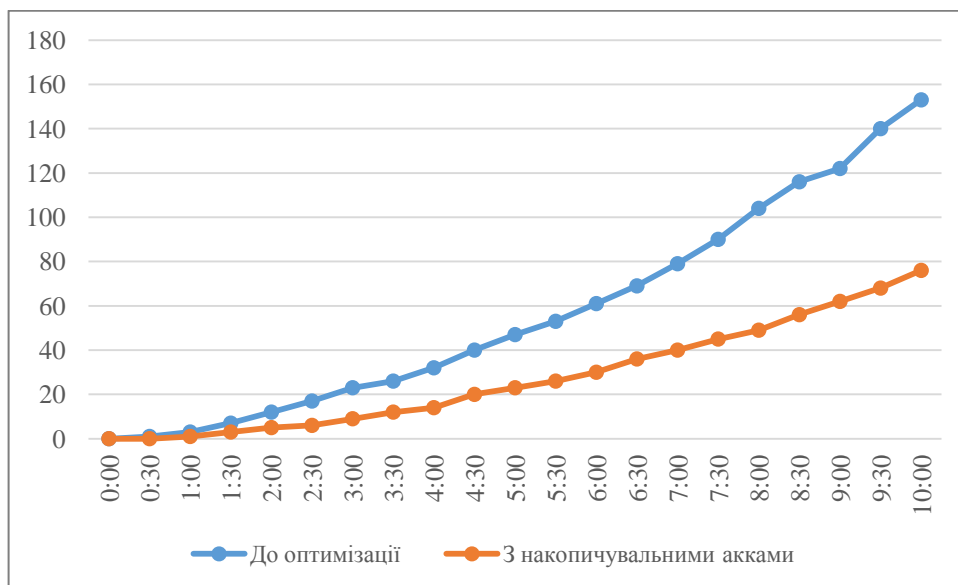


Рис. 2. Графік втрати пакетів

Як видно, втрати пакетів зменшилися вдвічі. Це зумовлено значним зниженням навантаження на мережу. АСК не відправляються, як раніше, одразу у відповідь на повідомлення. Вони відправляються одразу групою у вихідних повідомленнях, що значно зменшує кількість відправок пакетів та час зайнятості радіомережі.

7. Висновки

В даній роботі було запропоновано та описано оптимізацію передачі АСК-ів у Peer-to-Peer мережах з вузьким каналом зв'язку. Було запропоновано формувати їх у групи та надсилати у вихідних повідомленнях, виділивши у пакеті певне місце. Кожен пір у мережі може виступати посередником у

передачі АСК-ів. Також з певним інтервалом часу може надсилатися окремий пакет з групою АСК-ів.

В результаті, розроблений в даній роботі алгоритм дозволяє суттєво зменшити кількість переданих пакетів, що зменшує навантаження на мережу та забезпечує більший відсоток успішно доставлених повідомлень. Це є особливо важливим аспектом при роботі з радіомережами, де рації працюють у режимі бродкасту.

Оптимізація була успішно застосована до геоінформаційної системи «Кропива» та незабаром планується для використання військовими бригадами у зоні АТО.

Список літератури

1. Протокол TCP [Електронний ресурс] // - Режим доступу: <https://uk.wikipedia.org/wiki/TCP> - Дата доступу: 14.04.2016.
2. В. Yang and Н. Garcia-Molina. Improving search in peer-to-peer networks. // In 22nd International Conference on Distributed Computing Systems (ICDCS), July 2002.
3. Портативная радиостанция DP4400/DP4401 [Електронний ресурс] // - Режим доступу: http://www.motorolasolutions.com/ru_ru/products/mototrbo/dp4000-series/dp4400-dp4401.html - Дата доступу: 14.04.2016.
4. Геоінформаційна система «АРТА» [Електронний ресурс] // - Режим доступу: <http://arta.center/#ua> - Дата доступу: 13.04.2016.
5. Інновації для армії. Система "Кропива" та карти Армії SOS [Електронний ресурс] // - Режим доступу: <http://life.pravda.com.ua/technology/2015/08/11/198416/> - Дата доступу: 13.04.2016.

*КОБЕЦЬ Н.М.,
КОВАЛЮК Т.В.*

МОДЕЛЬ КОМПЕТЕНТНОСТЕЙ МАГІСТРА ЗА СПЕЦІАЛЬНІСТЮ «ІНФОРМАЦІЙНІ СИСТЕМИ»

У даній статті розглянута компетентнісна модель магістра з інформаційних систем відповідно до MSIS Curriculum та вимог ІТ індустрії України. Показано, що аналіз вимог стейкхолдерів до випускників за спеціальністю «Інформаційні системи» надають сфері освіти необхідні відомості про області та об'єкти професійної діяльності випускників, її види, завдання та необхідні компетенції майбутніх фахівців.

The article discusses competency model Master of Information Systems accordance with the requirements of MSIS Curriculum and IT industry of Ukraine. It is shown that the analysis requirements of stakeholders to graduates of the specialty "Information systems" provides the necessary information about the area and objects of professional activity of graduates, its types, tasks and competencies needed for future professionals.

1. Вступ

Перехід від освітньої парадигми, що базувалась на системі знань, до парадигми, що будується на системі компетенцій, відповідає сучасному стану розвитку суспільству з його вимогами до високого рівня підготовки кадрів. З позицій компетентнісного підходу рівень освіти в сучасних умовах визначається не стільки обсягом знань, їх енциклопедичністю, скільки здатністю вирішувати професійні проблеми і завдання різної складності на основі наявних знань, умінь і навичок. Освіта має відповідати мінливому стану ІТ-індустрії. Отже, компетентнісна модель магістра з інформаційних систем (ІС), яка формулює вимоги індустрії до ІТ-професії, надаватиме інформацію сфері освіти, необхідну для створення якісних освітніх стандартів і навчальних програм, а також дозволить збудувати систему оцінки якості навчання відповідно до вимог працевлагодів.

2. Постановка проблеми

Підготовка ІТ-спеціалістів у вищих навчальних закладах, що за кількістю та якістю соціально-особистісних та професійних компетенцій відповідатимуть вимогами ІТ-індустрії, є актуальною проблемою.

На сьогодні проблема управління компетентностями випускників ВНЗ

залишається невирішеною, тому актуальними є такі питання:

- роль компетентностей як ключових архітектурних компонентів навчального плану;
- визначення ядра та спеціалізованих компетентностей відповідно до предметних галузей діяльності випускників;
- зміст навчальних програм через опис областей компетенцій, категорій та самих компетентностей.

3. Мета статті

Метою даної статті є аналіз вимог стейкхолдерів, визначення областей, категорій і ядра компетентностей магістрів з інформаційних систем, побудова структури навчального плану в цілому.

4. Компетентності як архітектурні компоненти освітньої парадигми

Навчальна програма для ІТ спеціальностей структурується у вигляді ієрархії [2]: галузь знань – блок (одиниця) знань – змістовий модуль (тема) (рис.1) . Ця структура утворює сукупність знань випускника. Наприклад, остання версія Computer Science Curricula (CS2013) має 18 областей знань і в межах кожного 5-12 одиниць знань. Кожен блок знань, в свою чергу, ділиться на теми. Наприклад, галузь знань «Управління інформацією» має 12 блоків (одиниць) знань, зокрема, блок «Реляційні бази даних» має 11

тем, наприклад, розробка концептуальної схеми, тощо. Потенційна проблема із структурою галузь знань – блок знань – тема є те, що вона орієнтована здебільшого на пізнавальні аспекти вивчення і пропускає емпіричні елементи. Навчальні плани спеціальностей сьогодні представлені в основному переліком дисциплін (як

основних, так і спеціалізованих) з відповідними цілями навчання та темами (рис.1). Такий підхід до розробки навчальних планів не дає уявлення про те, яким чином дисципліни сприяють досягненню цілей, які студенти ставлять перед навчальною програмою.

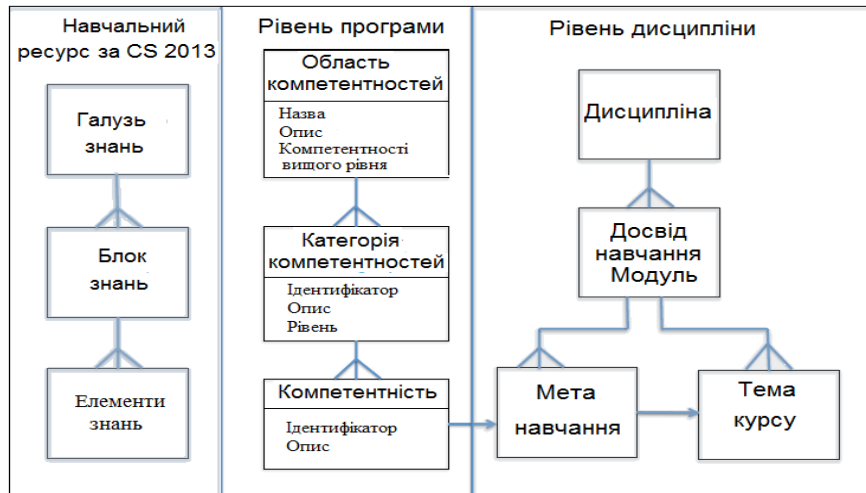


Рис. 1. Компетентності як архітектурні компоненти

Якщо замість сукупності знань та переліку дисциплін навчального плану ідентифікувати набір компетентностей випускників, то можна отримати динамічну комбінацію когнітивних і метакогнітивних навичок, демонстрацію знань і розуміння, міжособистісних, інтелектуальних і практичних навичок, а також етичних цінностей [2].

5. Структура системи компетентностей магістра з ІС

В системі компетентностей магістра з інформаційних систем доцільно виділити такі складові:

- область компетентностей освітнього рівня, що передує магістру;

- ядро компетентностей;
- спеціалізовані компетентності.

Ядро компетентностей магістра ІС включає такі складові: компетентності в галузі Computing та управління ІС, компетентності, що складають індивідуальну фундаментальну підготовку магістрів з ІС, та домен компетентностей [2]. Компетентності в блоці Computing та управління ІС складаються з компетентностей з області управління даними, інформацією, змістом; ІТ інфраструктури, розробки ІС та їх розгортання, управління ІС та експлуатації (рис.2).



Рис. 2. Компетентнісна модель магістра з ІС: область Computing та управління ІС

Домен компетентностей охоплює основні компетенції в області людської діяльності: бізнес, наукові дослідження, урядування, право, охорона здоров'я тощо.

Спеціалізовані компетентності ґрунтуються на ключових компетентностях в області Computing та управління ІС, які дозволятимуть випускникам виконувати більш складні спеціалізовані професійні завданн.

Перелік компетентностей магістра з інформаційних систем має базуватися на переліку дескрипторів європейської рамки ІКТ-компетенцій (e-CF): **планування (plan), реалізація (build), запуск (run), адаптація (enable), управління (manage)**, бізнес-процесах в інформаційних системах згідно з процесами життєвого циклу інформаційних систем (табл.1)

№ п. п	Функціональна область компетентностей	Процес ЖЦ за ISO/IEC 15288:2008, ISO/IEC 12207:2008	Область ІКТ компетенцій згідно з e-CF	ІКТ компетенції згідно з e-CF
1	Передконтрактні роботи	Процес менеджменту моделі ЖЦ	А. Планування Е. Управління	А3. Розробка бізнес-планів Е1. Розробка прогнозів
2	Управління вимогами	Процеси визначення вимог правовласника та аналізу системних вимог	А. Планування	А2. Управлінням рівнем послуг
3	Управління персоналом	Процес менеджменту людських ресурсів	Д. Адаптація	Д3. Організація навчання Д9. Підвищення кваліфікації персоналу
4	Моделювання бізнес-процесів замовника	Процес менеджменту моделі ЖЦ	А. Планування	А1. Узгодження ІС і бізнес-стратегії
5	Управління комунікаціями	Процес менеджменту інформації	Д. Адаптація Е. Управління	Д10. Управління інформацією і знаннями Е4. Управління взаєминами
6	Управління закупівлями	Процеси придбання та поставки	Д. Адаптація	Д4. Забезпечення процесу закупівель
7	Управління договірними стосунками	Процеси менеджменту портфеля проектів	Д. Адаптація	Д8. Управління контрактами
8	Процес розробки	Процеси проектування архітектури системи, реалізації, комплексування системи, кваліфікаційного тестування.	А. Планування В. Впровадження	А5. Проектування архітектури А6. Розробка застосувань В2. Інтеграція систем В3. Тестування
9	Управління якістю	Процес менеджменту якості	Д. Адаптація Е. Управління	Д2. Розробка стратегії забезпечення якості ІС Е6. Управління якістю ІС

10	Розгортання/ Впровадження ІС	Процеси інсталяції програмних засобів, підтримки прийомки ПЗ, функціонування ПЗ, супроводження ПЗ	А. Планування В. Впровадження	А7. Впровадження технологій В1. Проектування і розробка В4. Розгортання рішень
11	Управління змінами	Процес ревізії ПЗ	С. Запуск Е. Управління	С2. Підтримка змін Е5. Поліпшення процесів Е7. Управління змінами
12	Управління документацією	Процес менеджменту документації	В. Впровадження	В5. Розробка документації
13	Управління безпекою	Процес менеджменту ризиків	С. Запуск Д. Адаптація Е. Управління	С4. Управління проблемами D1. Розробка стратегії інформаційної безпеки Е3. Управління ризиками Е8. Управління інформаційною безпекою

Таблиця 1. Процеси ЖЦ, функціональні області компетентностей магістра з інформаційних систем і дескриптори e-CF

Висновки

Визначення моделі компетентностей магістра з інформаційних систем має здійснюватися на основі європейської рамки ІКТ компетенцій, професійних стандартів, що розробляються представниками ІТ індустрії, та стандартів в галузі інформаційних технологій, що описують бізнес-процеси та задачі діяльності в процесі розробки та управління інформаційними системами.

Компетентності магістра з інформаційних систем мають включати перелік компетентностей в галузі Computing та управління ІС, фахові фундаментальні компетентності, які забезпечуватимуть прийняття управлінських рішень, загальні компетентності (домен), спеціалізовані професійні для виконання спеціалізованих професійних ролей

Список літератури

1. Curriculum Guidelines for Undergraduate Degree Programs in Information Systems. 2010. [Електронний ресурс]. Режим доступу: <http://www.acm.org/education/curricula-recommendations>
2. Revising the MSIS Curriculum: Specifying Graduate Competencies. 2016. [Електронний ресурс]. Режим доступу: <http://www.acm.org/education/curricula-recommendations>
3. Ковалюк Т.В., Кобець Н.М. Професійні стандарти в галузі інформаційних технологій та їх гармонізація з ІТ-освітою України. //Інженерія програмного забезпечення. Науковий журнал НАУ. №22 (3), 2015.
4. Професійний стандарт. Фахівець з інформаційних систем. [Електронний ресурс]. Режим доступу: <http://mon.gov.ua/content/Новини/2016/03/15/5-ps-spes-infosystems-13.12.2014.pdf>

ГАВРИЛЕНКО О.В.,
СЕРДЮК А.І.

АНАЛІЗ МЕТОДІВ ІНТЕЛЕКТУАЛЬНОЇ ОБРОБКИ ДАНИХ ДЛЯ РЕКОМЕНДАЦІЙНИХ СИСТЕМ

У даній статті розглянуто можливості застосування різних методів інтелектуальної обробки даних для рекомендаційних систем. Розглянуто етапи попередньої обробки даних, класифікації даних та можливості використання кластерного аналізу у контексті рекомендаційних систем. Визначено можливості використання кожного з методів, а також зроблено висновки щодо можливості ефективного застосування відповідних методів для побудови масштабованої гібридної рекомендаційної системи. На прикладі рекомендації контенту новин визначено методи оптимальні до використання за показниками продуктивності та складності.

This article discusses the possibility of using different methods of data mining for recommendation systems. The stages of pre-processing, data classification and the possibility of using cluster analysis in the context of recommendation systems. Determined possibility of using each method and conclusions about the possibility of effective application of appropriate methods for building scalable hybrid recommender system. In the example of news content recommendations identified best methods to use in terms of performance and complexity.

1. Вступ

Надання персональних рекомендацій, тобто проблема надання матеріалів, які б відповідали інтересам користувача, є важливим питанням в умовах стрімкого розвитку інформаційних технологій та електронних ЗМІ. Щодня створюються мільйони символів нового контенту, а для обробки таких об'ємів інформації людина не здатна фізично.

Рекомендаційні системи (РС), зазвичай, використовують методи та методології з суміжних областей - таких, як людиномашинна взаємодія або інформаційний пошук. Тим не менше, більшість систем мають у своїй основі алгоритм, який є конкретним прикладом технології інтелектуального аналізу даних (ІАД).

2. Попередня обробка даних

Дані визначаються у вигляді колекції об'єктів і їх атрибутів, де атрибут визначається як властивість або характеристика об'єкта. На цьому етапі особливої важливості набувають наступні складові: міра подібності або міра відстані, вибірка та зменшення розмірності даних.

Одним з основних підходів до спільної фільтрації (СФ) є використання

класифікатора k-подібних. Цей метод класифікації – найбільш поширений метод класифікації та кластеризації - сильно залежить від визначення подібності або міри відстані. Існує багато прикладів міри відстані: Евклідова відстань, відстань Мінковського, відстань Махаланобіса, подібність косинуса. Проте подібність між елементами також може бути отримана за їх кореляцією, одним з таких коефіцієнтів є кореляція Пірсона (формула 1):

$$Pearson(x, y) = \frac{\sum(x, y)}{\sigma_x \times \sigma_y}, \quad (1)$$

де x, y - ковариация точок, а σ – їх стандартне відхилення.

Саме такий спосіб визначення міри подібності є найбільш широко використовуваним для РС, через свою гнучкість до використання та інтерпретації. Незважаючи на те, що у роботі [1] наведено, що в загальному випадку точність прогнозування РС не залежить від вибору міри подібності.

Вибірка є основним методом, який використовується в ІАД для вибірки підмножини даних з великого набору даних. Ключовим питанням для вибірки є пошук підмножини даних з початкової

множини, яка має приблизно такі ж самі властивості як і увесь набір. Найпростішим методом вибірки є випадкова вибірка, де існує рівна ймовірність вибору будь-якого елемента, але найбільш загальним підходом до вибірки є вибір без заміни: якщо елемент обрано, то він видаляється з набору. Проте, вибірка може призвести до надмірної спеціалізації при поділі набору даних для навчання та тестування. З цієї причини, процес навчання може повторюватися декілька разів. Загальним підходом для рекомендаційних систем є вибірка за допомогою доступного зворотнього зв'язку від користувачів, наприклад, у формі оцінок.

Зазвичай РС мають не тільки набір даних, який визначений у багатовимірному просторі, але й дуже розсіяну інформацію у просторі. Розрідженість і

проблема розмірності є загальними для всіх РС. Два найбільш відповідних алгоритми зниження розмірності у контексті РС: аналіз головних компонентів (PCA) та сингулярний розклад матриці (SVD).

Аналіз головних компонентів являє собою класичний статистичний метод для знаходження закономірностей у багатовимірному просторі набору даних. PCA дозволяє отримати впорядкований список компонентів, які складають найбільшу кількість дисперсії з даними за найменшою квадратичною похибкою. На рисунку 1 показано аналіз головних компонент для двовимірного набору точок, згенерованих комбінацією функцій Гаусса. Після того, як дані відцентровано, отримуємо головні компоненти, які позначено як u_1 та u_2

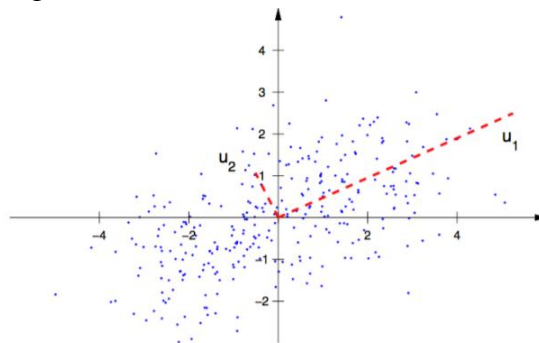


Рис. 1 – PCA для двовимірного простору

PCA потужний засіб, але він має серйозні обмеження, тому інші методи матричної факторизації, такі як SVD та невід'ємна матриця факторизації є більш переважними, ніж використання PCA.

Сингулярний розклад [16] є також потужним засобом для зниження розмірності. Ядро алгоритму SVD полягає у наступній теоремі: завжди можна розкласти задану матрицю A на $A = U\lambda V^T$. Дана $n \times t$ матриця даних A (n об'єктів, t ознак), з якої можна отримати $n \times r$ матрицю U (n об'єктів, r понять), $r \times r$ діагональна матриця λ (сила кожного поняття), і $t \times r$ матрицю V (t ознак, r понять). Однією з найбільших переваг сингулярного розкладу є послідовність алгоритмів використовуваних для обчислення наближеного розкладу. Це дозволяє додавати нові значення без перерахування створеної моделі. Ця ідея була формалізована як онлайн SVD

модель, що стало загальноприйнятим підходом для побудови нових моделей.

На етапі збору даних може з'явитися шум і може вплинути на результат аналізу та інтерпретації даних. У контексті РС розрізняється природний і штучний шум. Очевидно, що штучний шум може впливати на результат роботи рекомендаційної системи. За результатами робіт [2, 3] зроблено висновок, що природні шуми на продуктивність РС не впливають, а створення методів позбавлення від штучних шумів на етапі попередньої обробки є надто витратними, ніж використання алгоритмів оптимізації на подальших етапах.

3. Класифікація даних

Класифікатор є відображенням між простором ознак і простором позначок, де ознаки представляють характеристики елементів класифікації, а позначки представляють класи.

Найбільш популярний класифікатор на основі примірників – класифікатор найближчих сусідів. Так точка може бути класифікована за допомогою знаходження k найближчих точок (найближчих сусідів) з тренувального набору даних.

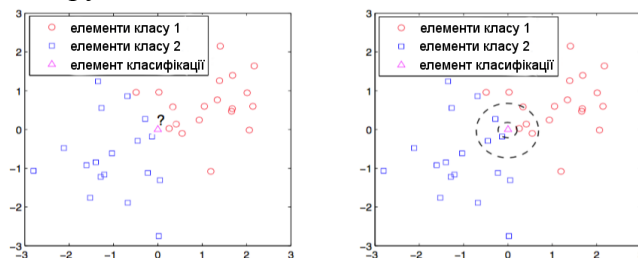


Рис. 2 – Приклад k -найближчих сусідів

Класифікатор k -найближчих є одним з найпростіших алгоритмів машинного навчання.

Найближчі сусіди є одним з найбільш поширених підходів для спільної фільтрації. Однією з переваг даного класифікатора для РС є те, що концептуально вони дуже схожі між собою: пошук схожих елементів, що по суті еквівалентно знаходженню сусідів. Іншою перевагою є те, що такий класифікатор не вимагає постійної підтримки моделі і, принципово, система може адаптуватися до швидких змін матриці оцінок.

Дерево рішень це класифікатор цільового атрибуту (класу) у вигляді деревоподібної структури. Існує багато алгоритмів індукції дерева рішень: алгоритм Ханта, CART, ID3, SLIQ, SPRINT тощо. Основною перевагою використання дерева рішень, у якості класифікатора, це дешева побудова і можливість швидкої класифікації невідомих екземплярів. Дерева рішень також можуть використовуватися для РС. В якості критерія поділу використовують інформацію отриману про кожну ознаку. У своїй роботі [4], Боуза зазначив, що точність таких рекомендацій значно гірша, ніж рекомендація середньої оцінки. Дійсно, дуже складно і недоцільно будувати дерево рішень, яке описує всі змінні, які беруть участь у процесі прийняття рішень.

Класифікатор на основі правил класифікує дані за наступним набором “якщо ... то ...”. Перевагою такого

Припускається, що точка належить до того ж класу, до якого належать найближчі сусіди. На рисунку 2 зображено схематично ідея цієї класифікації.

класифікатора є виразність і символічність, тобто працювати з даними можна без перетворення. Але відповідно до методу дерева рішень, дуже важно побудувати повну модель користувача на основі правил.

Байєсівський класифікатор [5] є ймовірнісною основою для вирішення завдань класифікації. Він заснований на визначенні умовної ймовірності і теоремі Байєса. Підхід дозволяє на основі ймовірності представляти невизначеність зв'язків даних. Частковим випадком байєсівського класифікатора є наївний байєсівський класифікатор. Для оцінки умовної ймовірності $P(A_1, A_2, \dots, A_N | C_k)$, наївний байєсівський класифікатор припускає ймовірність незалежності атрибутів. Тоді маємо

$$P(A_1, A_2, \dots, A_N | C_k) = P(A_1 | C_k) P(A_2 | C_k) \dots P(A_N | C_k).$$

Основною перевагою наївного байєсівського класифікатора є те, що він стійкий до ізольованих точок шуму і невідповідних атрибутів, також він обробляє відсутні значення, ігноруючи екземпляр під час обчислення ймовірності оцінки. Байєсівський класифікатор є досить популярним для РС, заснованих на контенті, але він також може бути використаний для спільної фільтрації [6].

Штучна нейронна мережа [7] являє собою сукупність взаємопов'язаних вузлів і зважених зв'язків. Основною перевагою є залежність від функції активації, оскільки це дає можливість виконувати нелінійні завдання класифікації. Штучні нейронні мережі можуть бути використані

аналогічно до байєсівський мереж довіри, для побудови моделей РС. Проте немає відомих досліджень, які підтверджують доцільність такого використання, а також немає практичного використання нелінійних класифікаторів для завдань, які виконує РС.

Поєднання класифікаторів. Основна ідея полягає у використанні групи класифікаторів, для побудови класифікатора на навчальних даних. Реалізація групи можлива за умови, якщо припустити, що класифікатори є незалежними. У цьому випадку, можна вважати, що об'єднання класифікаторів буде давати результати, які не гірші, ніж гірший класифікатор групи. Таким чином, поєднуючи незалежні класифікатори, похибка лише поліпшиться. Кожен екземпляр має імовірність $(1 - \frac{1}{N})^N$ бути обраним. N – дуже велике число. Використання об'єднання класифікаторів є звичайною практикою для РС. Безумовно, об'єднання класифікаторів дають кращі результати, ніж використання лише одного.

Найбільш загально прийнятою мірою оцінки для РС є середня похибка та середньоквадратична похибка. Ці заходи обчислюють точності без уточнення цілей РС. Якщо розглядати рекомендації як

проблему класифікації, то можна використовувати для оцінки такі методи як точність і повнота. Але алгоритми і класифікатори можуть бути оцінені за багатьма критеріями. Вони включають в себе точність класифікації, обчислювальна складність, чутливість до зашумлених даних, масштабованість тощо.

Висновки

Для створення ефективної рекомендаційної системи, на кожному з етапів необхідно повністю задовольняти її потреби. Для цього використання лише одного методу є недостатнім. Кожен метод має свої переваги і недоліки, але не може бути ефективно застосований для різних типів даних. Тому для цього, на кожному етапі методи будуть об'єднуватися, використовуючи переваги одних, для усунення недоліків інших. Використання такого підходу дає змогу значно розширити спектр атрибутів, а також можливості для обробки даних і побудови відповідних моделей, що значно розширює можливості застосування нових методів у різних сферах. Крім того, такий підхід має значні можливості для масштабування шляхом використання відповідних наборів методів на різних етапах обробки та інтерпретації.

Список літератури

1. Lathia N. The effect of correlation coefficients on communities of recommenders [Текст]: SAC '08: Proceedings of the 2008 ACM symposium on Applied computing / N. Lathia, S. Hailes, L. Capra / – New York: ACM, 2008. – с. 2000-2005.
2. Amatriain X. I like it...i like it not: Evaluating user ratings noise in recommender systems [Текст]: UMAP '09: User Modeling, Adaptation, and Personalization / X. Amatriain, J. M. Pujol, N. Oliver / – Washington: Telefonica, 2009. с. 247–258.
3. Amatriain X. Rate it again: Increasing recommendation accuracy by user rerating [Текст]: Recys '09: Proceedings of the third ACM conference on Recommender systems / X. Amatriain, J. M. Pujol, N. Tintarev, N. Oliver / – Washington: Telefonica, 2009. с. 173-180.
4. Mooney R. J. Content-Based Book Recommending Using Learning for Text Categorization [Текст]: Proceedings of the 5th ACM Conference on Digital Libraries / R. J. Mooney, L. Roy / – New York: ACM, 2000. – с. 195–204.
5. Friedman N. Bayesian network classifiers [Текст] / N. Friedman, D. Geiger, M. Goldszmidt // Machine Learning. – Kluwer Academic Publishers, Netherlands – 1997, с. 131-163.
6. Ghani R. Building recommender systems using a knowledge base of product semantics [Текст]: 2nd International Conference on Adaptive Hypermedia and Adaptive Web Based Systems / R. Ghani, A. Fano / – Chicago: Accenture Technology Labs, 2002. – с. 27-29.
7. Zurada J. Introduction to artificial neural systems [Текст] / J. Zurada // – St. Paul: West Publishing Co., 1992. – 683 с.

НЕСТАЦІОНАРНА ЗАДАЧА КОМІВОЯЖЕРА В УМОВАХ АВІАТРАНСФЕРУ З ДОДАТКОВИМИ ОБМЕЖЕННЯМИ

В статті сформульовано математичну постановку нестационарної задачі комівояжера в умовах авіатрансферу з додатковими обмеженнями та запропоновано алгоритм розв'язання на основі методу імітації відпалу. Серед додаткових умов наявні обмеження на мінімальний відрізок часу між сусідніми вилітами та загальна тривалість подорожі.

The subject of the article is a mathematical formulation of time dependent travelling salesman problem with additional terms for flight connection cases and solution algorithm on the basis of simulated annealing method. Additional terms include restriction of time gap between two flights and restriction of the total duration of the trip.

Розвиток та популяризація інформаційних і комунікаційних технологій дозволяє збирати інформацію і планувати індивідуальні екскурсійні маршрути самостійно. Існуючі сервіси пошуку подорожей орієнтовані на пошук оптимального шляху за певним критерієм лише між двома пунктами. У ряді випадків користувачу бажано поєднати відвідування кількох міст у певний період часу і оптимізувати свій маршрут за вартістю і особистими вподобаннями.

Розглянуто задачу планування авіамаршруту, де відомі бажані міста відвідування і додаткові побажання мандрівника, такі як часовий проміжок подорожі, мінімальна тривалість перебування в місті, максимальна вартість авіатрансферу. Така задача є узагальненим випадком класичної задачі комівояжера і відрізняється від неї наявністю більше ніж одного ребра між двома вершинами, що обумовлено наявністю множини рейсів між двома містами [1,2]. Вартість рейсів різна і залежить від часу відправлення та авіакомпанії.

Розглянемо математичну постановку нестационарної задачі комівояжера. Дано зважений орієнтований мультиграф $G=(V,F)$ де $V=\{v_1, v_2, \dots, v_n\}$ – множина n вершин, які являють собою n міст; F – множина дуг, які відображають наявні авіасполучення. Між двома вершинами може існувати декілька дуг, що відображає наявність декількох рейсів між містами від різних авіакомпаній, які можуть відрізнятися за вартістю і часом. Планування маршруту по містам має вкладатися в часові межі. Нехай t – поточний час, $t \in [t_{min}, t_{max}]$, де

t_{min} – бажаний час початку авіатуру, а t_{max} – бажаний час повернення у початкове місто. Тривалість подорожі можна розрахувати так:

$$T = t_{max} - t_{min}.$$

Позначимо за F множину рейсів $(v_i v_j)^t$ з міста i в місто j в момент часу t , а f_{ij}^t – конкретну дугу, що відповідає рейсу. Нехай ціна рейсу задається функцією $cost(f_{ij}^t)$ – невід'ємна залежна від часу функція, яка визначає загальну вартість перельоту з міста i в місто j у час t . Необхідно знайти оптимальний за вартістю шлях з початкової вершини $s \in V$, що охоплює всі бажані міста та повертається знову у початкову вершину у заданий проміжок часу $[t_{min}, t_{max}]$ чи повідомити що неможливо побудувати авіамаршрут у заданий інтервал. Шляхом x будемо називати упорядковану за часом множину рейсів $(f_{i_1 i_2}^{t_1}, f_{i_2 i_3}^{t_2}, \dots, f_{i_n i_{n+1}}^{t_n})$, для якої $i_1 = s, i_{n+1} = s$. Вартість шляху x визначається як
$$c(x) = \sum_{k=1}^n cost(f_{i_k i_{k+1}}^{t_k}).$$

Метою є знаходження оптимального за вартістю шляху x^* . Додатково може накладатися умова про тривалість перебування комівояжера в місті d . Нехай $x_{ij}^t = \{0,1\}$ – ідентифікатор наявності рейсу в маршруті і приймає значення 1, якщо маршрут містить дугу f_{ij}^t , і 0 в протилежному випадку. Математична постановка нестационарної задачі

комівояжера в умовах авіатрансферу з додатковими умовами задається так:

$$c(x) = \sum_{k=1}^n \text{cost}(f_{kk+1}^{t_k}) \rightarrow \min, \quad (1)$$

$$\sum_{i=t_1}^{t_n} \sum_{j=1}^n x_{ij}^t = 1, \quad \forall j = \overline{1, n}, \quad (2)$$

$$\sum_{i=t_1}^{t_n} \sum_{j=1}^n x_{ij}^t = 1, \quad \forall i = \overline{1, n}, \quad (3)$$

$$t_k - t_{k-1} > d, \quad \forall k = \overline{1, n}, \quad (4)$$

$$t_n - t_1 \leq T, \quad (5)$$

Метою оптимізаційної задачі є мінімізація загальної вартості авіамаршруту, яка подана цільовою функцією (1). Умови (2) та (3) гарантують, що комівояжер прилітає і вилітає з міста лише один раз. Умова (4) гарантує, що часовий проміжок між вилітами не менший ніж заданий, а умова (5) – що тривалість подорожі не перевищує заданий поріг.

Нині не існує прийнятних на практиці алгоритмів розв'язання нестационарної задачі комівояжера. Результатом пошуку ефективних методів розв'язання такої задачі стало використання біонічних алгоритмів, у тому числі, еволюційних та генетичних. Далі пропонується алгоритм, розроблений на основі методу імітації відпалу [3,4].

Для розглядуваної задачі наведемо пояснення ключових понять методу імітації відпалу таких як: стан, сусідній стан, прийняття імовірностей. Станом s будемо називати послідовність вершин графу задачі чи нулів у випадку, коли комівояжер не здійснює переліт. Часовою одиницею вважається один день. Кількість елементів послідовності дорівнює бажаній тривалості подорожі у днях. Номер елемента в послідовності відповідає дню подорожі, а елементи безпосередньо – номеру міста призначення рейсу в поточний день або 0 у випадку відсутності перельоту. Наприклад, можливим станом для планування маршруту тривалістю в 9 днів через 3 міста може бути наступна послідовність: (0, 3, 0, 0, 1, 0, 0, 0, 2).

Сусідні стани утворюються перестановкою двох елементів в послідовності. Імовірність переходу від поточного стану s в новий стан s' визначається ймовірнісною функцією прийняття $P(e, e', T)$, що залежить від енергії двох станів: $e = E(s)$ та $e' = E(s')$; та від глобального параметру T , який змінюється в часі та називається “температура”: $P = e^{-\frac{e'-e}{T}}$.

До основних кроків методу імітації відпалу додано додатковий крок, так званий крок покращення. Він полягає у пошуці дешевшого рейсу в найближчому околі. Якщо такий рейс знайдено, то розглядуваний сусідній стан буде модифікований заміною рейсу. Наприклад, за певних параметрів алгоритму для міста номер 3 будуть проаналізовані рейси з міста 2 у місто 3 у перший та третій день:

$$(\overline{0, 3, 0, 0, 1, 0, 0, 0, 2}) \rightarrow \left[\begin{array}{l} (\overline{3, 0, 0, 0, 1, 0, 0, 0, 2}) \\ (\overline{0, 0, 3, 0, 1, 0, 0, 0, 2}) \end{array} \right].$$

Загальну схему алгоритм розв'язання нестационарної задачі комівояжера можна подати так.

1. Початковий стан генерується випадковим чином. Поточний стан дорівнює початковому. Вибір початкової температури.
2. Оцінка поточного стану.
3. Основний крок алгоритму.
 - 3.1. Генерування сусіднього стану.
 - 3.2. Покращення сусіднього стану.
 - 3.3. Оцінка сусіднього стану.
 - 3.4. Критерій зміни поточного стану.
4. Зменшення температури. Якщо температура більша за визначений поріг – перехід до кроку 3, інакше - до кроку 5.
5. Завершення алгоритму. Поточний стан є розв'язком задачі.

Отже, сформульовано математичну постановку нестационарної задачі комівояжера в умовах авіатрансферу з додатковими обмеженнями та запропоновано алгоритм розв'язання на основі методу імітації відпалу.

Список літератури

1. Fox, K., Gavish, B., Graves, S.: An n-constraint formulation of the (time dependent) traveling salesman problem. *Oper. Res.* 28, 101–102 (1980).
2. Bigras, L.-Ph., Gamache, M., Savard, G.: The time-dependent traveling salesman problem and single machine scheduling problems with sequence dependent setup time. *Discrete Optim.* 5, 685–699 (2008)
3. Kirkpatrick S., Gelatt C. D., Vecchi M. P. Optimization by simulated annealing. *Science.* v220 (1983), pp 671–680.
4. Aarts E. H. L., Korst J. H. M. Simulated annealing and Boltzmann machines. Chichester: Wiley, 1989.

ВИРІШЕННЯ ЗАДАЧІ РОЗПІЗНАВАННЯ ВИДУ РИБИ МЕТОДОМ НЕЧІТКОГО ЛОГІЧНОГО ВИВОДУ

В даній статті описується розроблений евристичний алгоритм вирішення задачі розпізнавання виду риби, що базується на нечіткому логічному виводі, також досліджується ефективність отриманих результатів, порівнюючи з відомими результатами вирішення задачі іншими методами.

Developed heuristic algorithm solving fish species recognition problem by using fuzzy logic method is described in this article, also the effectiveness of the obtained results is explored comparing with known results of problem solving by using other methods.

Задача розпізнавання виду риби стала важливою в прикладному середовищі, і є новою, цікавою задачею розпізнавання зображення. Дослідження даної задачі можуть бути використані для побудови автоматизованої системи, що стане в нагоді людям для різних цілей – детального вивчення підводного світу, покращення екологічної ситуації, дослідження видів морської та річної фауни, що знаходиться під загрозою вимирання.

Тому актуальним завданням була розробка евристичного алгоритму вирішення даної задачі та дослідження результатів роботи розробленого алгоритму, що базується на методі нечіткого логічного виведення.

Був розроблений наступний алгоритм розпізнавання виду риби наступний:

1. Виділити мінімальну прямокутну область з зображенням риби та бінаризувати цю частину.
2. Виділити простір входів вхідної області:
 - 2.1. Виділити хвостову частину риби
 - 2.2. Поділити хвостову частину на області – 4 вертикальних та 4 горизонтальних області у вигляді прямокутників для подальшого розпізнавання.
3. Визначити вхідні лінгвістичні змінні – відсоток частки зображення у відповідних областях хвостової частини зображення.
4. Визначити характеристичну функцію для вхідних лінгвістичних змінних.
5. Порівняти значення характеристичних функцій з правилами бази правил, що

була побудована раніше на основі навчальних даних.

6. На основі порівняння на попередньому пункті отримати вихідну лінгвістичну змінну – назву виду риби.
7. Якщо вихідна лінгвістична змінна не є суперечливою, тобто за вихідну лінгвістичну змінну був вибраний тільки один вид риби, то кінець алгоритму, інакше якщо було обрано декілька видів риб, продовжити алгоритм з пункту №8.
8. Прийняти за вхідну область область плавників риби.
9. Виділити область плавників риби на зображенні.
10. Поділити область плавників на 4 вертикальних та 4 горизонтальних прямокутника для подальшого розпізнавання.
11. Обчислити вхідні лінгвістичні змінні як відсотки частки зображень у виділених прямокутниках областях плавників.
12. Визначити характеристичну функцію для отриманих вхідних лінгвістичних змінних.
13. Порівняти отримані значення характеристичних функцій з правилами тих декількох видів риб (що були отримані у пункті №7) бази правил, що була побудована раніше.
14. Отримати єдине значення вихідної лінгвістичної змінної – одну назву виду риби.

Як вибірку для основного навчання та тестування системи розпізнавання виду риби, побудованої за алгоритмом описаним

вище, було взято 30 видів риб загальної популярної вибірки, використаної в різних наукових працях, і були отримані наступні результати (Таблиця 1). Всього для навчання та тестування було використано 159 зображень риб. Зокрема, отримані

результати успішності розпізнавання виду риби нечітким логічним виводом були порівняні з результатами розпізнавання за допомогою нейронних мереж та методу евклідових відстаней в статті [2].

Табл. 1. Результати дослідження

Наукова назва виду риби	Q	M	N
Rastrelliger brachysoma	5	4	1
Pangasius pangasius	5	5	0
Terapon Jarbua	6	5	1
Nemipterus hexodon	5	5	0
Cynoglossus macrolepidotus	2	2	0
Dermogeny pusilla	5	4	1
Liza subviridis	6	5	1
Barbonymus gonionotus	6	4	2
Oreochromis niloticus	6	6	0
Katsuwonus pelamis Linnaeus	5	5	0
Oreochromis niloticus niloticus	5	5	0
Prisipomoides typus	6	5	1
Epinephelus areolatus	6	5	1
Paratromateus niger	6	6	0
Salaroides leptolepis Bleeker	5	5	0
Lactarius lactarius	5	5	0
Caesis cuning	4	4	0
Sillago maculata	6	4	2
Arius truncatus Curier & Valennennes	3	2	1
Pampus argenteus	6	6	0
Hypophthalmichthys molitrix	6	5	1
Labeo rohita	5	4	1
Psettodes erumei	6	6	0
Auriglobus Modestus	5	5	0
Zonichthys Nigrofascita	5	5	0
Chitala Chitala	5	3	2
Drepane punctata	6	5	1
Eleutheronema tetradactylum	6	5	1
Carrassius auratus	6	6	0
Pterophyllum altum	6	6	0

де Q – кількість зображень для тестування, M – кількість успішно розпізнаних зображень відповідного виду, N – кількість некоректно або взагалі нерозпізнаних зображень

Успішне розпізнавання методом нечіткого логічного виводу мало місце у 89.5% випадків і є вищим за відповідний показник розпізнавання методом евклідових відстаней у 81.67%, але меншим ніж методом нейронних мереж у 99%. Але середній час розпізнавання виду однієї риби методом нечіткого логічного виводу є меншим ніж методом нейронних мереж – 48.42 сек проти

154.4 сек, але більшим ніж методом евклідових відстаней – 24.4 сек.

Дана система може бути покращена збільшенням кількості видів риб та кількості зображень, доданням можливості розпізнавання виду риби в будь-якому середовищі та можливо в динамічному середовищі.

Список літератури

1. Shape- and Texture-Based Fish Image Recognition System [Електронний ресурс] // Режим доступу: http://kasetartjournal.ku.ac.th/kuj_files/2013/A1310041317311406.pdf

ЗАСТОСУВАННЯ ГЕНЕТИЧНОГО АЛГОРИТМУ ДЛЯ РОЗВ'ЯЗАННЯ ЗАДАЧІ ФОЛДИНГУ БІЛКІВ НА ГРАФІЧНОМУ ПРОЦЕСОРІ

В даній статті розглядається генетичний алгоритм для розв'язання задачі фолдингу білків на графічних процесорах. Розповідається про математичну модель що може бути застосована для опису процесу фолдингу, описано генетичні оператори для відповідної задачі, а також специфіка застосування генетичного алгоритму на графічному процесорі з використанням технології CUDA. Запропонована архітектура програмного забезпечення для проведення відповідних обчислень, їх збереження та візуалізації.

The subject of the article is an application of genetic algorithm for solving the protein folding problem on graphics processing unit. Mathematical model for protein folding problem is described with corresponding genetic operators. Specifics of genetic algorithm implementation on GPU using CUDA technology is described. Architecture of software for performing computations, saving and visualizing their results, is proposed.

1. Математична модель

Білки є одним з найважливіших класів біологічних макромолекул - вони виконують структурні функції у м'язах та кістках, каталітичні функції у всіх біохімічних реакціях, які формують метаболізм, і вони координують рух та трансдукцію сигналів [1].

Після синтезу білка у вигляді лінійної послідовності амінокислот, він швидко згортається у стабільну тривимірну структуру – у цьому стані він може виконувати свої функції. Знання тривимірної структури білка має багато застосувань у медицині та генній інженерії.

В «AB off-lattice» моделі, послідовність амінокислот білка представлена лише двома видами мономерів: «А» - гідрофобними та «В» - гідрофільними [2]. У цій моделі відстань між мономерами (σ_i) прирівнюється одиниці, і кожен мономер σ_i приєднаний до наступного мономера σ_{i+1} через зв'язок який утворює кути θ та α відносно свого попередника. Функція енергії (цільова функція) може бути записана наступним чином:

$$E = -k_1 \sum_{i=1}^{N-2} \mathbf{e}_i \cdot \mathbf{e}_{i+1} - k_2 \sum_{i=1}^{N-3} \mathbf{e}_i \cdot \mathbf{e}_{i+2} + \sum_{i=1}^{N-2} \sum_{j=i+2}^N E_{LJ}(r_{ij}, \sigma_i, \sigma_j), \quad (1)$$

де \mathbf{e}_i – одиничний вектор зв'язку між мономерами σ_i та σ_{i+1} . Оскільки $\mathbf{e}_i \cdot \mathbf{e}_{i+1} = \cos \theta_i$, $\mathbf{e}_i \cdot \mathbf{e}_{i+2} = \cos \alpha_i$, білок довжини N може бути представлений $N-2$ кутами зв'язку та $N-3$ двограними кутами. r_{ij} – відстань між мономерами σ_i та σ_j . Енергія взаємодії між двома мономерами (що залежить від їх типу), визначається через потенціал Ленард-Джонса (Lennard-Jones potential):

$$E_{LJ}(r_{ij}, \sigma_i, \sigma_j) = 4C(\sigma_i, \sigma_j) \left(\frac{1}{r_{ij}^{12}} - \frac{1}{r_{ij}^6} \right). \quad (2)$$

Значення $C(\sigma_i, \sigma_j)$ обирається таким чином щоб сприяти формуванню гідрофобного ядра:

$$C(\sigma_i, \sigma_j) = \begin{cases} 1, & \sigma_i = \sigma_j = A \\ 0.5, & \text{в інших випадках.} \end{cases} \quad (3)$$

Два параметри цільової функції k_1 та k_2 визначають силу локальної взаємодії мономерів.

2. Генетичні оператори

Для прискорення обчислень багатьох складних задач, у тому числі і для задачі фолдингу білків, використовуються графічні процесори (GPU) [3].

Генетичний алгоритм добре підходить для паралельних обчислень, оскільки операції

схрещування, мутації та обчислення цільової функції можуть виконуватись паралельно для всіх елементів популяції.

Схрещування двох елементів популяції $h_i = \{\theta_2^i, \theta_3^i, \dots, \theta_{n-1}^i, \alpha_2^i, \alpha_3^i, \dots, \alpha_{n-1}^i\}$ та $h_j = \{\theta_2^j, \theta_3^j, \dots, \theta_{n-1}^j, \alpha_2^j, \alpha_3^j, \dots, \alpha_{n-1}^j\}$ відбувається за формулами:

$$\theta_m' = r\theta_m^i + (1-r)\theta_m^j, \quad m \in \overline{2, n-1}, \quad (4)$$

$$\alpha_m' = r\alpha_m^i + (1-r)\alpha_m^j, \quad m \in \overline{3, n-1}. \quad (5)$$

В процесі мутації всі кути θ_m' та α_m' змінюються на невелике значення r_m , тобто:

$$\theta_m'' = r_m \theta_m', \quad m \in \overline{2, n-1}, \quad (6)$$

$$\alpha_m'' = r_m \alpha_m', \quad m \in \overline{3, n-1}. \quad (7)$$

Після цього розраховується значення $\Delta E = E - E_{\min}$. Якщо $\Delta E < 0$, то конформація приймається, у іншому випадку конформація приймається з ймовірністю $e^{-(\Delta E/T)}$. Значення T змінюється від максимального до $T_{terminal}$ з певним інтервалом за формулою:

$$T_{i+1} = \lambda T_i (0 \leq \lambda \leq 1). \quad (8)$$

Ітерації алгоритму припиняються коли значення T знизиться до $T_{terminal}$, або якщо протягом певної кількості генерацій не знаходиться розв'язок кращий за рекордний. Отже кроки алгоритму є наступними:

Крок 1. Ініціалізація параметрів: N , $N_{non-contested}$, $max_no_improve_gens$, r_{\min} , r_{\max} , T_{start} , $T_{terminal}$, $R_{mutation}$, λ ;

Крок 2. Ініціалізація початкової популяції та копіювання даних у пам'ять графічного процесору;

Крок 3. Ініціалізація значення глобального рекорду $f_{rec} = \infty$ та рекорду для кожної з підпопуляцій $f_{rec}^i = \infty$;

Крок 4. Поки $T > T_{terminal}$ та f_{rec} було покращено протягом останніх $max_no_improve_gens$ генерацій перейти на Крок 5., інакше на Крок 10;

Крок 5. Відсортувати популяцію за спаданням фітнес функції і вибрати перші $N_{non-contested}$ елементи до нової популяції; якщо номер ітерації кратний 50, записати 5

кращих елементів популяції до глобальної пам'яті, і замінити 5 найгірших елементів своєї підпопуляції на 5 елементів з глобальної пам'яті записаних іншим блоком;

Крок 6. Якщо кількість елементів у новій популяції менше N , перейти на крок 7, інакше на Крок 4.

Крок 7. Вибрати 2 елементи методом рулетки (елементи з кращим значенням фітнес-функції мають вищу ймовірність бути відібраними) та виконати схрещування методом описаним вище;

Крок 8. З ймовірністю $R_{mutation}$ виконати операцію мутації методом описаним вище;

Крок 9. Визначити фітнес функцію отриманої конформації. Якщо значення енергії нижче ніж в рекорду (у цьому випадку також оновити значення f_{rec}^i та f_{rec})

або якщо $rand < e^{-(\Delta E/T)}$ - прийняти рішення; перейти на Крок 6;

Крок 10. Повернути розв'язок зі значенням f_{rec} .

3. Опис програми

Програма складається з серверної та клієнтської частин, які представляють собою цілісну систему. Програмне забезпечення можна використовувати наступним чином:

- клієнт – серверну систему, де сервер розташований на окремому потужному комп'ютері, а клієнти можуть до нього під'єднуватись за допомогою будь-якого сучасного веб-браузера;
- серверна частина може отримувати запити і надавати розв'язки іншим застосункам, які використовують її API (передають і отримують дані у JSON форматі);
- як окрему програму, що встановлюється на персональний комп'ютер користувача і може бути запущена з консолі, шляхом передавання їй вхідних параметрів, в тому числі й шляху до файлу з послідовністю амінокислот.

Інтерфейс програмного продукту представляє собою WEB-сторінку. Обчислення конформації білка на основі отриманих вхідних даних відбувається на сервері. Отримані розв'язки зберігаються в базі даних, для того щоб можна було швидко отримати доступ до вже розв'язаних задач,

або порівняти швидкість розв'язання різними методами. Програмне забезпечення виконано за технологією обміну даними між клієнтом і сервером, яка передає ключові дані у загальному вигляді у форматі JSON.

Частина серверного коду написана на мові Ruby з використанням фреймворку Ruby on Rails, який відповідає за обробку запитів з клієнтської частини, формування вихідних HTML сторінок та роботу з базою даних. Ця частина спроектована за шаблоном MVC (Model-View-Controller), що дозволяє розподілити серверну логіку та спосіб її відображення.

Для виконання власне обчислень конформації створений окремий обчислювальний модуль (підпрограма). Для мови розробки цього модулю було визначено наступні вимоги:

- мова повинна підтримувати високорівневі конструкції, за допомогою яких зручно описувати математичні моделі;
- мова має компілюватися в машинний код, для високої продуктивності виконання, оскільки основною ціллю даної роботи є пришвидшення розв'язання задачі фолдингу білків;
- програма має запускатися на останніх версіях ОС Windows (починаючи з Windows 7 – це обмеження накладається засобом розробки «CUDA Toolkit 7.5»).

Спираючись на ці вимоги, було обрано мову програмування C++ (для програмування частин програми що виконуються на графічному процесорі використана мова C та версія набору розробника для CUDA - «CUDA Toolkit 7.5»).

Для роботи серверної частини необхідна відеокарта nVidia що підтримує технологію CUDA, як мінімум версію 3.0. У випадку, якщо комп'ютер сервера використовується для перегляду і клієнтської частини, важливо знати, що на час обчислень, у випадку, якщо встановлений лише один графічний процесор, зображення на моніторі може «підвисати», що пов'язано з завантаженням процесору.

Мовою програмування клієнтської частини є JavaScript. Для створення зовнішнього вигляду інтерфейсу користувача

використана мова розмітки гіпертексту HTML5.

Використання додаткових засобів розробки дозволяє скоротити час розробки, поліпшити якість коду і його структуру. Для спрощення доступу до елементів інтерфейсу, розподілення коду і його представлення, використані такі технології і бібліотеки, як jQuery та Bootstrap 4.

Для перегляду отриманих структур білків використана бібліотека PV, що є JavaScript бібліотекою що використовує технологію WebGL для швидкого відображення тривимірної моделі білків.

Для початку роботи над задачею необхідно ввести вхідні дані. Вхідні дані можуть подаватися відразу у вигляді послідовності лише з двох типів амінокислот (гідрофобні або полярні), або у вигляді звичайних амінокислот (кожна кислота кодується однією літерою латинського алфавіту) – в такому випадку послідовність буде перетворена на послідовність з двома видами амінокислот на стороні сервера. Також можна задати в якості вхідних даних файл у форматі FASTA, що є найбільш розповсюдженим форматом для зберігання даних про послідовності амінокислот в білках. Файли в такому форматі можна завантажити, наприклад, з сайту PDB (Protein Data Bank).

Також необхідно вибрати алгоритм пошуку. Відповідно реалізованими є послідовна та паралельна версія генетичного алгоритму в рамках «AB off-lattice» моделі – таким чином можна оцінювати ефективність пришвидшення роботи алгоритму за рахунок обчислень на графічному процесорі та їх якість.

Набір вхідних параметрів і для послідовної і для паралельної версії є однаковим. Для кожного параметру є значення за замовчанням, яке, при бажанні, користувач може змінити у формі відправки задачі на розв'язання.

До вхідних параметрів належать:

- *max_no_improve_gens* – максимальна кількість популяцій без покращення розв'язку (при перевищенні – обчислення припиняються);
- *N* – розмір популяції;

- $N_{non-contested}$ – кількість елементів старої популяції що переходять у нову без конкурсу;
- r_{min} , r_{max} – мінімальне та максимальне значення коефіцієнту інтерполяції що використовується при схрещуванні для отримання лінійної комбінації батьків;
- T_{start} – початкова температура (використовується при прийнятті рішення прийняти чи відкинути розв'язок, що гірший за поточний оптимум);
- $T_{terminal}$ – термінальна температура;
- λ – коефіцієнт спаду температури впродовж поколінь;
- $R_{mutation}$ – ймовірність мутації, після проходження процедури схрещування.

На виході алгоритму формується вектор пар кутів (зв'язку та торсійних) разом із відповідним значенням цільової функції. Окрім цього користувач може переглянути у браузері тривимірну модель отриманої конформації та зберегти дані про розв'язок у базі даних.

Список літератури

1. Czibula G., Vocicor M.I., Czibula I.G. A Reinforcement Learning Model for Solving the Folding Problem // International Journal of Computer Technology and Applications, 2011 - №2 - pp. 171-182.
2. Benitez C.M.V. Molecular Dynamics for Simulating the Protein Folding Process Using the 3D AB Off-Lattice Model // In: Advances in Bioinformatics and Computational Biology. // Campo Grande. 2012. - pp. 61-72.
3. Scalabrin M.H., Parpinelli R.S., Benitez C.M.V., Lopes H.S. Population-based harmony search using GPU applied to protein structure prediction // International Journal of Computational Science and Engineering, 2014 - pp. 106-118.

МЕТОДИ ПАРАЛЕЛЬНИХ ОБЧИСЛЕНЬ ДЛЯ РОЗВ'ЯЗАННЯ СИСТЕМ ЛІНІЙНИХ АЛГЕБРАЇЧНИХ РІВНЯНЬ

У даній статті розглянуто сучасні потреби у вирішенні якісно нових науково-технічних завдань, що призводять до необхідності обробки великих об'ємів інформації за прийнятний час, зокрема – до розв'язання систем лінійних алгебраїчних рівнянь з великими матрицями коефіцієнтів. Наведено огляд основних алгоритмів розв'язання СЛАР із симетричними та несиметричними матрицями коефіцієнтів та проведено оцінку їх ефективності. Також описано особливості розробки паралельних алгоритмів засобами мови програмування JavaScript.

The subject of the article is the current needs in solving qualitatively new scientific and technical problems that lead to the need of handling large volumes of information in a reasonable time, in particular - solving systems of linear equations with high matrix coefficients. There is a review of basic algorithms for solving SLE with asymmetric and symmetric coefficient matrix and evaluation of their effectiveness. Also the features of the development of parallel algorithms for programming language JavaScript are provided.

Ключові слова: СИСТЕМА ЛІНІЙНИХ АЛГЕБРАЇЧНИХ РІВНЯНЬ, НЕСИМЕТРИЧНА МАТРИЦЯ, СИМЕТРИЧНА МАТРИЦЯ, РОЗРІДЖЕНА МАТРИЦЯ, ПАРАЛЕЛЬНІ ОБЧИСЛЕННЯ, JAVASCRIPT, МЕТОД ГАУСА, МЕТОД ХОЛЕЦЬКОГО

Сучасні потреби у вирішенні якісно нових науково-технічних завдань призводять до необхідності обробки великих об'ємів інформації за прийнятний час, зокрема – розв'язання систем лінійних алгебраїчних рівнянь (СЛАР) з великими матрицями коефіцієнтів.

Системи лінійних алгебраїчних рівнянь виникають при вирішенні низки прикладних задач, що описуються диференціальними, інтегральними або системами нелінійних (трансцендентних) рівнянь. Значна частина чисельних методів вирішення різних (особливо - нелінійних) завдань включає в себе вирішення систем лінійних рівнянь як елементарний крок відповідного алгоритму [1]. СЛАР можуть з'являтися у задачах математичного програмування, апроксимації функцій, статистичної обробки даних, при дискретизації крайових диференціальних задач методом кінцевих різниць та в багатьох інших випадках.

Іншою важливою особливістю таких задач є високий порядок матриць СЛАР - до десятків мільйонів [2]. Це зумовлюється бажанням використовувати більш точні дискретні моделі, що дає можливість отримувати наближені розв'язки більш близькі до розв'язків вихідних

задач, краще враховувати локальні особливості даного процесу чи явища.

Зокрема, системи рівнянь з розрідженими матрицями великої розмірності виникають у задачах аналізу міцності конструкцій у цивільному та промисловому будівництві, в авіабудуванні, суднобудуванні, ракетобудуванні, при моделюванні матеріальних та інформаційних потоків у великих технологічних мережах (системи газопостачання і газорозподілу, каналізаційні та тепlopостачальні системи, електромережі та комп'ютерні мережі тощо) [3].

Область застосування методів розв'язування СЛАР з розрідженими матрицями постійно розширюється.

Через високі порядки, незважаючи на розріджену структуру матриць, розв'язування таких задач вимагає значних обчислювальних ресурсів, які можуть бути надано сучасними комп'ютерами з паралельною організацією обчислень. Тому на сьогодні є актуальною проблема створення і використання ефективних паралельних алгоритмів розв'язування задач з розрідженими матрицями. Ефективними вважаються алгоритми, що дозволяють отримати достовірний розв'язок задачі з мінімальним

використанням ресурсів комп'ютера – процесорів, пам'яті, часу.

Метою даного дослідження є зменшення часу, необхідного для розв'язання СЛАР із симетричними і несиметричними матрицями коефіцієнтів засобами мови програмування JavaScript.

Для досягнення мети необхідно вирішити такі задачі:

- дослідити особливості паралельних обчислень засобами JavaScript;
- провести аналіз існуючих алгоритмів розв'язання СЛАР з несиметричними і симетричними матрицями коефіцієнтів: метод Гауса та метод Холецкого;
- синтезувати паралельний алгоритм розв'язання СЛАР з урахуванням особливостей паралельного обчислення засобами JavaScript;
- розробити програмну реалізацію алгоритмів у вигляді бібліотеки;
- розробити інтерфейс для зручного вводу задач і виведення результату розв'язання.

Однією з найперспективніших технологій для паралельних обчислень засобами JavaScript є Web Worker. Специфікація цієї технології визначає API для створення фонових скриптів у веб-застосунках. Такі об'єкти дозволяють запускати довготривалі скрипти для виконання задач, що вимагають великого обсягу

обчислень. При цьому інтерфейс користувача не блокується і керування системою продовжує бути доступним для користувача.

Об'єкти Web Worker використовують потокову передачу повідомлень для реалізації паралельності [4]. Вони ідеально підходять для поновлення інтерфейсу, забезпечення його ефективності та реалізації оперативної взаємодії з користувачами.

Хоча кожний скрипт кожного Web Worker виконується в окремому потоці операційної системи, але точки зв'язку з іншими потоками ретельно контролювані. У всіх об'єктів Web Worker немає доступу до непотокобезпечних компонентів і дані між потоками передаються за допомогою відправки повідомлень з даними в серіалізованому вигляді. Все це дозволяє уникати більшості проблем паралельного програмування при роботі з мовою JavaScript [5].

Розроблений таким чином веб-ресурс є зручним інструментом для паралельного розв'язання систем лінійних алгебраїчних рівнянь з несиметричними та симетричними матрицями коефіцієнтів. Всі розрахунки здійснюватимуться на стороні клієнта, прямо у веб-браузері, що мініфікує час, необхідний для розв'язання задач, що зводяться до розв'язання СЛАР.

Перелік посилань:

1. Ильин В. А. Линейная алгебра: Учебник для вузов — 6-е изд., стер. [Текст] / Ильин В. А., Позняк Э. Г. — М.: ФИЗМАТЛИТ, 2004. — 280 с.
2. Джордж А. Численное решение больших разреженных систем уравнений = Computer Solution of Large Sparse Positive Definite Systems. [Текст] / Джордж А., Лю Дж. — М.: Мир, 1984. — 333 с.
3. Гантмахер Ф. Р. Теория матриц. — 2 изд. [Текст] / Гантмахер Ф. Р. — Москва : Наука, 1967. — 576 с.
4. Using Web Workers [Електронний ресурс] // Режим доступу: https://developer.mozilla.org/en-US/docs/Web/API/Web_Workers_API/Using_web_workers
5. The Basics of Web Workers [Електронний ресурс] // Режим доступу: <http://www.html5rocks.com/en/tutorials/workers/basics/>

МЕТОДИ ПАРАЛЕЛЬНИХ ОБЧИСЛЕНЬ ДЛЯ РОЗВ'ЯЗАННЯ АЛГЕБРАІЧНОЇ ПРОБЛЕМИ ВЛАСНИХ ЗНАЧЕНЬ

У статті розглядаються задачі, що зводяться до розв'язання часткової алгебраїчної проблеми власних значень з симетричними додано визначеними матрицями розрідженої структури великої розмірності. Проведено дослідження паралельних алгоритмів розв'язання АПВЗ на ефективність. Також розглянуто можливості реалізації цих паралельних алгоритмів на мові програмування JavaScript.

The subject of the article is the problems related to a partial solution of algebraic problem of eigenvalues of symmetric positively defined matrices with tenuous structure of large dimensions. Also there is a review of parallel algorithms for solving algebraic eigenvalues problem and their efficiency using JavaScript programming language.

Ключові слова: ПАРАЛЕЛЬНІ АЛГОРИТМИ, JAVASCRIPT, ВЛАСНІ ЗНАЧЕННЯ, ВЛАСНИЙ ВЕКТОР, АЛГЕБРАІЧНА ЗАДАЧА ВЛАСНИХ ЗНАЧЕНЬ, МЕТОД ІТЕРАЦІЙ НА ПІДПРОСТОРИ

Велика кількість наукових та практичних задач, зокрема, при дослідженні стійкості конструкцій, розрахунку динаміки напружено-деформованого стану об'єктів різної природи та ін., зводяться до розв'язання часткової алгебраїчної проблеми власних значень з симетричними додано визначеними матрицями розрідженої структури великої розмірності [1].

Необхідність знаходження заданої кількості найбільших за абсолютною величиною власних чисел та відповідних векторів розріджених матриць великої розмірності виникає, наприклад, при дослідженні коливальних у фізиці [2]. Коливання пружних конструкцій, коливання в аеродинаміці, коливання в електричних колах, коливання молекул та атомів у фізиці елементарних часток – всі ці задачі є різноманітними прикладними аспектами однієї і тієї самої математичної задачі.

Іншою прикладною задачею, що потребує знаходження власних векторів великих розріджених матриць є задача ранжування результатів веб-пошуку, одним із способів вирішення якої є визначення міри центральності вузлів графу веб-сторінок [3]. Якщо задати граф G , вузлами якого є веб-сторінки, а ребрами якого є гіперпосилання, то міра центральності кожного вузла буде характеризувати імовірність того, що людина,

читаючи веб-сторінки та переходячи по гіперпосиланням, опиниться на відповідній веб-сторінці. Так, алгоритм PageRank, що використовується пошуковою системою Google, базується саме на визначенні міри центральності кожної веб-сторінки (яку розробники назвали PageRank цієї веб-сторінки). Звичайно, кількість веб-сторінок в мережі Інтернет є дуже великою (за оцінками Google [4], в 2008 році кількість веб-сторінок в мережі Інтернет досягла 10^{12}), але кожна веб-сторінка має порівняно невелику кількість гіперпосилань (до сотень), тому матриця суміжності графу G буде розрідженою.

Такі задачі можуть бути ефективно розв'язані за допомогою однокрокових та двокрокових ітераційних методів. Проте їх розв'язування на комп'ютерах потребує великих об'ємів обчислювальних ресурсів та часу їх виконання.

Алгебраїчна проблема власних значень (АПВЗ, алгебраїчна задача на власні значення) полягає в знаходженні таких значень λ , за яких існує відмінні від нульового розв'язки системи лінійних алгебраїчних рівнянь:

$$Ax = \lambda Bx, \quad A, B \in M^{n \times n}, \quad x \in C^n, \quad (1)$$

$$\lambda \in C,$$

де $M^{n \times n}$ – множина квадратних матриць порядку n . Числа λ називають власними

значеннями задачі (1.1), а вектори x – власними векторами цієї задачі. Задача (1) називається узагальненою проблемою власних значень.

При дослідженні стандартної проблеми власних значень необхідно відрізнити наступні випадки [5]:

- збурення простого власного значення матриці, що має лінійні елементарні дільники;
- збурення кратного власного значення матриці, що має лінійні елементарні дільники;
- збурення простого власного значення матриці, що має один або більше нелінійних елементарних дільників;
- збурення кратного власного значення відповідного нелінійному елементарному дільнику повної матриці;
- збурення кратних власних значень λ_i , коли є більш ніж один елементарний дільник із множником $(\lambda_i - \lambda)$ і принаймні один із них нелінійний.

У загальному випадку дослідження властивостей алгебраїчної проблеми власних значень з наближеними вихідними даними може включати в себе побудову інваріантних підпросторів (наприклад, власних, кореневих), визначення канонічних форм (наприклад, Жордана, Фробеніуса), оцінювання обумовленості власних значень і власних векторів, дослідження збурення розв'язку залежно від похибки вихідних даних, оцінювання достовірності отриманих комп'ютерних розв'язків, тобто обчислення оцінки близькості математичного та комп'ютерного розв'язку та оцінки спадкової похибки отриманого розв'язку.

Оскільки збільшення тактової частоти існуючих процесорів досягає граничного значення і більшість сучасних комп'ютерів мають багато процесорів, які можуть бути

використані для обчислення результатів, то актуальною є проблема розробки і використання паралельних алгоритмів, які значно скорочують час розв'язання АПВЗ.

Паралельне програмування стає домінуючою парадигмою комп'ютерній архітектурі в основному саме у формі багатоядерних процесорів. І хоч використання можливостей паралелізму стало уже звичайною практикою в програмуванні, є мови програмування, в яких паралелізм застосовується не так активно, як у інших. До таких мов можна віднести Javascript – найпопулярнішу мову веб-програмування. Проте, враховуючи неймовірно стрімкий розвиток веб-технологій, саме у цієї мови дуже великі перспективи у застосуванні паралельних обчислень.

На сьогодні найзручніший спосіб реалізації паралельного виконання коду у браузері – технологія Web Worker.

Специфікація об'єктів Web Worker визначає API для створення фонових скриптів у веб-додатках. Ці об'єкти дозволяють запускати у «фоновому режимі» довготривалі скрипти для виконання завдань, що вимагають великого обсягу обчислень, не блокуючи інтерфейс користувача або інших скриптів, що відповідають за взаємодію користувача з системою.

Об'єкти Web Worker запускаються в окремому потоці і використовують потокову передачу повідомлень для реалізації паралельності. Вони ідеально підходять для оновлення інтерфейсу, забезпечення його ефективності та реалізації оперативної взаємодії з користувачами. Саме ця особливість дозволяє використовувати даний підхід для реалізації паралельних алгоритмів на JavaScript.

Перелік посилань:

1. Приказчиков В.Г. Прототипы итерационных процессов в задаче на собственные значения [Текст] / Приказчиков В.Г. // Дифференциальные уравнения. – 1980. – Т. 16, № 9. – С. 1688
2. Sagan Hans. Boundary and Eigenvalue Problems in Mathematical Physics [Текст] / Sagan Hans – Dover Publications, 1989. – ISBN 0-486-66132-6.
3. Tore Opsahl. Node centrality in weighted networks: Generalizing degree and shortest paths [Текст] / Tore Opsahl, Filip Agneessens, John Skvoretz – 2010 – Vol. 32. – Pp. 245–251
4. [Електронний ресурс] // We knew the Web was big Режим доступу : <http://googleblog.blogspot.com/2008/07/we-knew-web-was-big.html>
5. SLEPC (Scalable Library for Eigenvalue Problem Computations) [Електронний ресурс] / Universitat Politècnica de València (Spain). – Режим доступу: <http://slepc.upv.es>

МОДЕЛЬ ДИСКРИМІНАНТНОГО АНАЛІЗУ В ПРОЦЕСІ НАВЧАННЯ ПОСТАНОВЦІ МЕДИЧНИХ ДІАГНОЗІВ

В даній роботі розглядається математична модель навчання постановці медичних діагнозів у формі «серйозної гри». Аналізуються основні вимоги до «серйозної гри» у сфері постановки медичних діагнозів. Пропонується модель класифікації захворювань за складністю діагностування за допомогою дискримінантного статистичного аналізу.

The article considers the mathematical model of medical diagnosis establishment training in the form of Serious game. The authors analyze the requirements for Serious game in a field of medical diagnosis establishment. The article offers a diagnosis classification model by its establishment difficulty using discriminant analysis.

Ключові слова: СЕРЙОЗНІ ІГРИ, МЕДИЧНИЙ ДІАГНОЗ, ДИСКРИМІНАНТНИЙ АНАЛІЗ, РОЗПОДІЛ ЗА СКЛАДНІСТЮ

1. Вступ

В наші дні сучасні відео- та комп'ютерні ігри все частіше використовують у класній кімнаті. Ідея полягає у тому, щоб зробити навчання інтерактивним та цікавим. Чимало країн не лише почали використовувати комп'ютерні ігри у школах, вони також спонукають учнів розробляти власні ігри у дуже ранньому віці.

Навчання завдяки комп'ютерним іграм допомагає багатьом: студентам, людям з особливими потребами, фахівцям у специфічних галузях. Наприклад, у Массачусетському технологічному інституті, одному з найпрестижніших технологічних навчальних закладів світу, день починається з фішок, гральної колоди карток. В ігровій лабораторії професори та студенти розробляють «серйозні ігри» майбутнього.

У даній роботі розглядається розробка «серйозної гри» для студентів-медиків, яка допоможе самостійно перевіряти і поповнювати свої знання, щодо встановлення правильних медичних діагнозів. На відміну від інших сфер освіти, наприклад інформаційних технологій, у студентів медичних напрямків існують складності у самостійному відточуванні отриманих знань на практиці. Тому актуальною

задачею є розробка ефективного програмного продукту, який дозволить студенту покращувати практичні навички при постановці діагнозів за рахунок організації навчального процесу у вигляді «серйозної гри».

2. Вимоги до «Серйозної гри»

Одне з визначень «серйозної гри», наступне: "Серйозна гра представляє собою гру, в якій головною метою є освіта (в її різних формах), а не розваги". Звертаючись до цього визначення, для «серйозної гри» у сфері постановки діагнозів, можна виділити наступні вимоги з боку навчального процесу:

- **доступність джерел знань** – для забезпечення цілісного процесу навчання доцільним є зберігання допоміжної інформації безпосередньо у грі, що дозволяє користувачу не вдаватись до пошуку в інших джерелах. Прикладом може бути зберігання інформації про симптоми хвороби, її патогенез або ж епідеміологію;
- **наявність списку альтернативних відповідей** – предметна область, передбачає можливість неточностей, зумовлених сутністю захворювань, можливої схожості їх протікання та симптомів. Тому, не зважаючи на правильність

постановки діагнозу користувачем, доречним є формувати список альтернативних діагнозів, які можуть бути поставлені при відповідних вихідних даних;

- **наявність кривої складності** - під кривою складності розуміється функція, відповідно до якої доцільно розташовувати згенеровані задачі за складністю. Це дозволяє структурувати навчальний процес та зробити його більш керованим.

Аналізуючи вищенаведені вимоги, можна відмітити, що найбільше на навчальний процес впливає саме наявність кривої складності та її правильне формування. Простота доступу до джерел знань, наявність пояснень та можливість побачити альтернативні відповіді є важливими факторами, але їх, у певному сенсі, можна назвати супровідними або допоміжними. Якщо відкинути їх, то користувач отримає не дуже зручну, але функціонуючу гру-вікторину. Якщо ж не реалізовано оцінку складності задачі, то не можливо ні вибудувати саму траєкторію складності задач, ні обрати задачі, які відповідають рівню знань гравця. Тому надалі, у даній роботі, основна увага буде приділятися саме оцінці складності діагнозів та побудові кривої складності задач.

Для переходу до поняття кривої складності задач, звернемося до поняття

«фокусу» і «поток», які відіграють велику роль у навчанні, «серйозних іграх» та звичайних іграх. Одним із найважливіших факторів якісного засвоєння матеріалу при навчанні є концентрація учня, тобто його сфокусованість. Це твердження має пряму аналогію з іграми і, відповідно, «серйозними іграми». Джесі Шел пише про це так: «Коли ми створюємо ігри, наша мета створити досвід, який буде утримувати увагу гравця так довго і інтенсивно, наскільки це можливо. Коли, щось повністю захоплює нашу увагу і уяву на довгий час, ми входимо у цікавий психічний стан. Здається, що решта світу відпадає і у нас немає ніяких нав'язливих думок. Все, про що ми думаємо, це те, що ми робимо, і ми повністю втрачаємо відчуття часу.» Такий стан стійкого фокусу називається «поток». «Потік» був предметом глибокого дослідження американського психолога Міхая Чіксентміхая та інших вчених. «Потік» - це почуття повної концентрації і сфокусованості на предметі діяльності, успіху та великого задоволення від виконаного. Вся діяльність у стані потоку повинна утримуватись у вузькому просторі «виклику» (або складності), якій розміщено між двома екстремальними станами нудьги і розчарування у власних силах. Досягнення одного з цих станів веде до зміни, фокусу і втрати потоку. Шел пропонує наступний варіант проходження каналом потоку (Рис. 1):

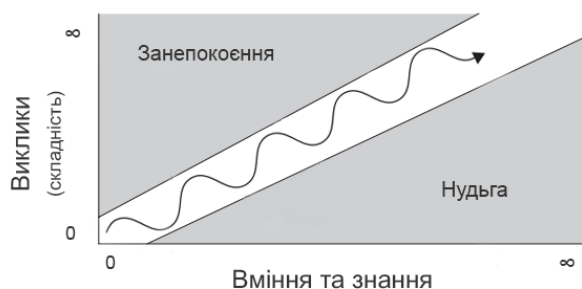


Рис. 1 – Приклад проходження «каналу потоку» по кривій

Описане вище поняття «каналу потоку» дозволяє природно перейти до кривої складності та побудови керованого навчального процесу. Він доволі добре відображає процеси збільшення складності задач і отримання нових знань.

3. Математична модель дискримінантного аналізу

Дискримінантний аналіз є статистичним методом, який дозволяє вивчати відмінності між двома і більше групами об'єктів за декількома змінними одночасно. Дискримінантний аналіз допомагає виявляти відмінності між

групами та дає змогу ідентифікувати (класифікувати) об'єкти за принципом максимальної схожості. Класифікація пов'язана з отриманням однієї або декількох дискримінантних функцій, що забезпечують можливість віднести (ідентифікувати) даний об'єкт до однієї з груп.

Характеристики, використовувані для того, щоб відрізнити один клас від іншого, називаються дискримінантними змінними. Ці змінні мають вимірюватися або за інтервальною шкалою, або за шкалою відносин.

На рис. 2 зображені об'єкти, що належать двом різним множинам M_1 і M_2 . Кожен об'єкт характеризується в даному випадку двома змінними x_1 і x_2 . Якщо розглядати проєкції об'єктів (точок) на кожну вісь, то ці множини перетинаються, тобто по кожній змінній окремо деякі об'єкти обох множин мають подібні характеристики. Щоб найкращим чином розділити дві множини, потрібно побудувати відповідну лінійну комбінацію змінних x_1 і x_2 . Для двовимірного простору ця задача зводиться до визначення нової системи координат. Причому нові осі L і C повинні бути розташовані таким чином, щоб проєкції об'єктів, що належать різним множинам, на вісь L , були максимально розділені. Вісь C перпендикулярна осі L і розділяє дві «хмари» точок найкращим чином, тобто таким, що множини опинилися по різні сторони від цієї прямої.

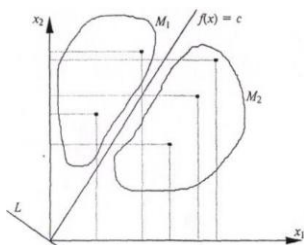


Рисунок 2. – Геометрична інтерпретація дискримінантної функції та змінних

При цьому ймовірність помилки класифікації повинна бути мінімальною. Сформульовані умови повинні бути враховані при визначенні коефіцієнтів a_1 і a_2 наступної функції:

$$f(x) = a_1x_1 + a_2x_2 \quad (1)$$

Функція $f(x)$ називається канонічною дискримінантною функцією, а величини x_1 і x_2 – дискримінантними змінними.

Нехай x_{ij} – середнє значення j -ого признаку у об'єктів i -ї множини (класу). Тоді, для множини M_1 середнє значення функції $f_1(x)$ буде дорівнювати:

$$\bar{f}_1(x) = a_1\bar{x}_{11} + a_2\bar{x}_{12} \quad (1.1)$$

для множини M_2 середнє значення функції $f_2(x)$ буде дорівнювати:

$$\bar{f}_2(x) = a_1\bar{x}_{21} + a_2\bar{x}_{22} \quad (1.2)$$

Геометрична інтерпретація цих функцій – дві паралельні прямі, що проходять через центри класів (множин) (Рис. 3).

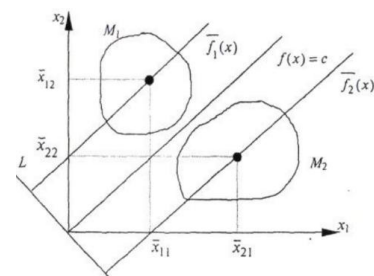


Рисунок 3. – Центри розділених множин та константа дискримінації

Коефіцієнти дискримінантних функцій a_i визначаються таким чином, щоб $\bar{f}_1(x)$ та $\bar{f}_2(x)$ якомога більше відрізнялися, тобто, щоб для двох класів (множин) був максимальним вираз:

$$\bar{f}_1(x) - \bar{f}_2(x) = \sum_{i=1}^{n_1} a_i x_{1i} - \sum_{i=1}^{n_2} a_i x_{2i} \quad (1.3)$$

4. Застосування дискримінантного аналізу для класифікації задач діагностування захворювань за складністю

Класифікувати захворювання за складністю діагностування пропонується спираючись на симптоми, що характеризують захворювання. При цьому слід враховувати, що деякі симптоми проявляються не одразу, тому використовується поділ процесу протікання захворювання на три етапи, що є адекватним поділом для більшості захворювань. В загальному випадку можна сказати що, якщо є дві пари захворювань і

у першій парі симптоми переважно співпадають, а у другій – майже немає однакових, тоді легше діагностувати захворювання з другої пари.

Нехай S – це множина симптомів. В такому випадку кожне захворювання можна представити наступною матрицею (таблиця 1):

Табл. 1. – представлення моделі захворювання

	Перший етап	Другий етап	Третій етап
S_1	S_{11}	S_{12}	S_{13}
S_2	S_{21}	S_{22}	S_{23}
...
S_n	S_{n1}	S_{n2}	S_{n3}

У такій матриці стовпці це періоди розвитку хвороби, а рядки це симптоми. Тоді S_{ij} – ймовірність появи конкретного симптому на конкретному періоді. Канонічна дискримінантна функція, в такому випадку матиме наступне математичне представлення:

$$F_{km} = a_0 + a_1 X_{1km} + \dots + a_p X_{pkm} \quad (2)$$

де F_{km} – значення канонічної дискримінантної функції для m -го захворювання в групі k ;

$X_{ikm} = \begin{pmatrix} X_{i1km} \\ X_{i2km} \\ X_{i3km} \end{pmatrix}$ – значення дискримінантної змінної X_i для m -го захворювання в групі k ; a_i – коефіцієнти, забезпечуючі виконання необхідних вимог.

Коефіцієнти a_i для першої функції обираються таким чином, щоб середнє

значення двох різних класів, якомога більше відрізнялися один від одного. Коефіцієнти другої функції обираються так само, тобто відповідні середні значення повинні максимально відрізнятися по класам, при цьому накладається додаткова умова на те, щоб значення другої функції були некорельовані зі значеннями першої, а третя функція – з першими двома і так далі. Максимальне число дискримінантних функцій, яке можна отримати таким способом дорівнює числу класів без одиниці або числу дискримінантних змінних, в залежності від того, яка з цих величин менша.

5. Висновок

Для оцінки складності діагностування захворювання розроблено модель, засновану на їх розділенні на класи. Класифікація відбувається на основі схожості прояву симптомів захворювань, які представлені векторами ймовірностей прояву симптому на різних періодах розвитку захворювання. Чим більше число входжень схожих векторів симптому у інші захворювання і чим більше таких симптомів, тим складнішим стає діагностування захворювання.

Для дослідження розробленої моделі, використовується дискримінантний статистичний аналіз, оскільки він надає можливість класифікувати об'єкти за принципом максимальної схожості і опрацювати об'єкти які з деяких причин не вдалося ідентифікувати та віднести до одного із класів.

Список літератури

1. Jesse Schell. Carnegie Mellon University The Art of Game Design A Book of Lenses
2. David Michael and Sande Chen. Serious Games: Games That Educate, Train, and Inform
3. Буреєва Н.Н. Многомерный статистический анализ с использованием ППП "STATISTICA". Учебно-методический материал по программе повышения квалификации «Применение программных средств в научных исследованиях и преподавании математики и механики». Нижний Новгород, 2007, 112 с.
4. Барковский С.С., Захаров В.М., Лукашов А.М., Нурутдинова А.Р., Шалагин С.В. Многомерный анализ данных методами прикладной статистики: Учебное пособие – Казань: Изд. КГТУ, 2010. – 126 с. Табл. 5 . Ил. 105. Библиогр.: 12 наим.
5. А. Ю. Яковлева. Пропедевтика внутренних заболеваний. Конспект лекций: Эксмо; Москва; 2007

РОЗВ'ЯЗАННЯ ЗАДАЧІ РОЗПІЗНАВАННЯ ПОЗИ ЛЮДИНИ НА ВІДЕО

В даній статті розглянуто задачу розпізнавання пози людини на відео. Розповідається про роботу, що ведеться у цьому напрямку. Наводиться визначення пози в контексті даного підходу. Описується алгоритм вирішення цієї задачі, що включає наступні послідовні кроки: вибір рамок-кандидатів з кадрів, виявлення ознак у них, застосування класифікатора, вибір рамок-кандидатів з найкращим результатом, застосування згладжувального алгоритму до всієї послідовності кадрів.

The subject of the article is human pose estimation in videos. Scientific work in given field is described. Pose definition in terms of proposed approach is given. Pose estimation algorithm consists of the next sequential steps: candidate boxes are chosen from the individual frames, features are extracted from these proposal boxes, individual body part detectors are applied to the extracted features and the likeness of the boxes to certain body parts is received, smoothing algorithm is applied to all video sequence.

Зазвичай, задачі розпізнавання пози людини дається наступне визначення: пошук розташування частин тіла людини (або пози) на зображенні [1]. Існує широкий спектр програмного забезпечення, що безпосередньо використовує інформацію про позу людини. Наприклад, взаємодія людини з машиною за допомогою камери, або діагностика стану людини через візуальне спостереження за нею. Цю задачу можна віднести до однієї з найцікавіших у комп'ютерному баченні. Проте вона досі залишається нерозв'язаною при довільних умовах.

В останні роки зроблені значні успіхи у вирішенні задачі розпізнаванні образів та, в тому числі, пози людини. Велику роль у цьому зіграло застосування і модифікація нейронних мереж. Одною з найбільш поширених моделей нейронних мереж є перцептрон. Проте вона не дуже підходить для вирішення задач розпізнавання образів. Велика розмірність вхідних зображень призводить до різкого збільшення числа нейронів і синаптичних зв'язків у мережі. В результаті сильно збільшується час і обчислювальна складність процесу навчання. Крім того, звичайний перцептрон ігнорує топологію вхідних даних, не враховуючи чітку

двомірну структуру зображень. Тому перевага надається згортковій нейронній мережі, яка дозволяє уникнути цих недоліків.

Проте поточні рішення є достатньо складними для застосування покадрово для розпізнавання на відео, оскільки вимагають занадто великих обчислювальних можливостей. Тому зараз активно ведуться роботи над розробкою спрощених алгоритмів, що матимуть достатньо точні результати та адекватну складність у вирішенні задачі розпізнавання пози людини на відео. Розробці такого рішення і присвячена дана робота.

Тож вирішується задача розпізнавання пози людини на відео. Її ціллю є знайти позу у кожному кадрі відео при допущенні, що вона змінюється плавно між кадрами вздовж тимчасових осей. Тож, ми розраховуємо знайти позу у всій послідовності кадрів. Дається наступне визначення пози: розташування рамок, що обмежують людське тіло та її кінцівки. Тобто, щоб знайти позу, необхідно розмістити рамки навколо людини та частин її тіла у кожному кадрі відео.

Загалом, даний алгоритм розпізнавання пози людини на відео можна описати наступним чином. На першому кроці

обираються рамки-кандидати з окремих кадрів, які можуть містити один із елементів пози – всю людину або її кінцівку. Після цього здійснюється виявлення ознак у рамках-кандидатах, які дозволять класифікувати зображення у рамці. Далі до виявлених ознак застосовується класифікатор, що показує ймовірність знаходження частини тіла у рамці-кандидаті. Обираються рамки з найкращим результатом. До них застосовується згладжувальний алгоритм, що дає фінальний результат – послідовність поз у кадрах, згладжених за тимчасовими осями.

В першій частині алгоритму обираються рамки-кандидати з кадрів. Для даної задачі підхід із застосуванням рухомого вікна не є можливим, через розмір простору пошуку та обчислювальну складність виявлення ознак у кожній рамці-кандидаті. Для вибору можливого розташування об'єктів було обрано селективний алгоритм пошуку [5]. Його особливістю є об'єднання областей зображення з підходом від низу до верху і використанням різноманітних стратегій, що дозволяє швидко отримати рамки-кандидати для незалежних класів. В нашому випадку, це можливі розташування людини та її кінцівок в кожному кадрі.

Другою частиною алгоритму є виявлення ознак. З можливих способів отримання ознак було обрано підхід з глибинними ознаками. Для цього використовується згортова нейронна мережа з сімома шарами, з яких 5 є згортковими та 2 повноз'єднаними. Ця конфігурація мережі показала високу ефективність для задач класифікації [4][3]. Для цього зображення у рамці-кандидаті деформується, щоб відповідати вхідним розмірам згорткової нейронної мережі. Вихідним результатом є 4096-розмірний вектор ознак.

Наступним кроком алгоритму є класифікація рамок за отриманими ознаками. Висока розмірність ознак дозволяє застосувати простий лінійний класифікатор - метод опорних векторів [2]. Практика застосування МОВ для

глибинних ознак мала значний успіх [3]. В даному випадку використовується бінарний метод опорних векторів, один для кожного класу, де класами є людина та частини тіла. Метод опорних векторів тренується на окремому наборі даних, щоб давати оцінку рамці-кандидату. Ця оцінка показує ймовірність приналежності об'єкту до певного класу.

Після того, як оцінки для рамок-кандидатів обчислені, у кожному кадрі обираються ті, що отримали найвищу оцінку. Після цього до них застосовується згладжувальний алгоритм. Цей алгоритм використовує просторові зв'язки між рамками у сусідніх кадрах. Для цього застосовуються методи динамічного програмування до рамок у кожному кадрі. В результаті обирається найкращий шлях зміни рамок протягом усього відео.

Список літератури

1. Thomas B. Moeslund, Adrian Hilton, and Volker Krüger. A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, 104(2-3): 90–126, November 2006. URL: <http://linkinghub.elsevier.com/retrieve/pii/S1077314206001263>.
2. Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.
3. Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *CoRR*, abs/1311.2524, 2013.
4. Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.
5. J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders. Selective search for object recognition. *International Journal of Computer Vision*, 104(2):154–171, 2013.

ПЕРЕДАЧА ДАНИХ ЗА ДОПОМОГОЮ СВІТЛОВОЇ ТЕХНОЛОГІЇ LI-FI, РЕАЛІЗОВАНА ДЛЯ МОБІЛЬНОГО ПРИСТРОЮ

В даній статті розглянуто власну реалізацію роботи технології Li-Fi, проаналізовано її основні переваги та недоліки в порівнянні із технологією Wi-Fi, описано алгоритм роботи технології, описано розроблений алгоритм для дешифровки потокового відео, що є однією із необхідних складових роботи Li-Fi. Описано реалізацію роботи на мобільному пристрої.

This article deals with its own implementation of technology Li-Fi, analyzed its main advantages and disadvantages compared to the technology of Wi-Fi, described algorithm technology described algorithm for decoding streaming video, which is one of the necessary components of Li-Fi. We describe the implementation of a mobile device.

Ключові слова: LI-FI, WI-FI, БЕЗДРОТОВА ПЕРЕДАЧА ДАНИХ, ПЕРЕДАЧА ДАНИХ ЗА ДОПОМОГОЮ СВІТЛА, АЛГОРИТМ ДЕШИФРОВКИ ПОТОКОВИХ ДАНИХ, АЗБУКА МОРЗЕ, МОБІЛЬНЕ ЗАСТОСУВАННЯ, VLC

Вступ

В даний час відбувається швидкий розвиток в області використання світла. Побойовання з приводу надмірного споживання енергії веде до поетапної ліквідації джерел розжарювання, і приводить до розвитку і зростання у світлодіодів [1].

Li-Fi технологія бездротова і використовує VLC або інфрачервоний і ультрафіолетовий спектр (замість радіочастотних хвиль), є OWC технології, яка передає набагато більше інформації, і була запропонований як рішення проблеми обмеженої пропускної спроможності радіохвильових технологій. Термін Li-Fi аналогічний до Wi-Fi, за винятком того, що для передачі використовується світла, а не радіохвилі.

Технологія VLC – це інформаційне середовище, що використовує видиме світло з оптичним спектром 400-800 ТГц. Ця технологія може використовувати флуоресцентні лампи для передачі сигналів зі швидкістю 10 Кбіт/с або світлодіоди зі швидкістю до 500 Мбіт/с. Уявіть спалах, який ми використовуємо для передачі сигналів азбуки Морзе. Якщо передавати сигнали вручну, світло посилає повідомлення,

використовуючи світловий сигнал, але так як він передається переривчасто, його не можна розглядати як повноцінне джерело освітлення. Тепер уявіть, що спалахи світла передаються дуже швидко через комп'ютер, в результаті чого спалах перетворюється у випромінювання постійного світла, тобто у нас з'являється освітлення та комунікація, які задовольняють визначенню видимій світловій комунікації. Звичайно, нам знадобиться ще приймач, здатний отримувати інформацію або дані, але це не так вже й складно[2]. Можливість передавати дані таким способом виникла внаслідок широкого використання світлодіодних ламп. Пропорційно до збільшення ефективності цих пристроїв їх вартість зменшується. За деякими прогнозами, в недалекому майбутньому вони стануть домінуючим джерелом для загального освітлення.

1. Технічні складові технології

Впровадження технології VLC матиме наступні переваги комунікації [3]:

- розширення бездротової інфраструктури;
- уникання дефіциту радіочастотного спектру (в 10000 разів більше потужності);

- уможливлення дуже високої пікової швидкості передачі даних (10 Гбіт/с);
- уможливлення Інтернету речей (у 100 разів більше пристроїв);
- значно підвищений безпечний бездротовий зв'язок (зниження перехоплення сигналів);
- підвищення енергоефективності шляхом об'єднання передачі даних і освітлення (скорочення споживання енергії в 100 разів);

- повне усунення проблем зі здоров'ям.

2. Порівняння Li-Fi із Wi-Fi

Наведемо порівняння деяких технічних деталей: максимальна частота Wi-Fi – 5 ГГц, у той час як Li-Fi взагалі не має частоти. Li-Fi вже має стандарт IEEE 802.15, а швидкість передачі даних дорівнює 1 Гбіт/с. Діапазон дії Wi-Fi становить близько 100 м, тоді як Li-Fi обмежений лише потужністю світлодіодного світла. Більше технічних деталей наведено у таблиці №1.

Таблиця 1.1 Порівняння технічних характеристик Li-Fi та Wi-Fi

Характеристика	Wi-Fi	Li-Fi
Частота	5 ГГц	У світла немає частоти
Стандарт	IEEE 802.11	IEEE 802.15
Діапазон	100 м	В залежності від світлодіодного світла
Швидкість передачі даних	800 Кб/с – 11 Мбіт/с	>1 Гбіт/с
Енергоспоживання	Середнє	Низьке
Затрати	Середні	Високі
Безпечність	Середня	Висока

Li-Fi має великий потенціал та значні переваги у порівнянні із Wi-Fi:

- швидкість нової технології перевищує швидкість Wi-Fi;
- не будуть траплятися збої у зєднанні, а також так звані «мертві зони» радіохвиль;
- у порівнянні із Wi-Fi, Li-Fi більш безпечний завдяки межах прямої видимості. Це означає, що ніхто ззовні не зможе втрутитися у систему. Проте, можуть виникнути проблеми, якщо залучити телеоб'єктив та оптимально налаштований фотодатчик;
- нова Інтернет система не така енерговитратна як Wi-Fi. У той час як Wi-Fi тяжіє до великого енергоспоживання, Li-Fi може працювати на сонячних батареях. Перевага цього у тому, що можна мати доступ до бездротового Інтернету та бездротової зарядки, від якої живитиметься і сам Li-Fi, і сам акумулятор.

3. Алгоритм дешифровки даних

Алгоритми роботи технології Li-Fi є комерційною таємницею, тому щоб відтворити роботу технології, необхідно створити власний унікальний алгоритм, що буде реалізовувати роботу технології. Для вирішення поставленого завдання необхідно оброблювати кадри потокового відео (що отримуємо з камери мобільного пристрою) на предмет розпізнання необхідного джерела світла, яке передає повідомлення. Для цього необхідно використати певні алгоритми з розпізнання образів. Оскільки дуже складно знайти універсальний алгоритм з розпізнання образів, що підходить для конкретного випадку, необхідно створити свій власний. Зважаючи на постановлену задачу та рамки характеристик пристрою, на якому буде реалізовано алгоритм, маємо наступний план. Для початку необхідно «захопити» потокове відео з камери пристрою. Із буферу потоку беремо перший кадр у черзі. Робимо первинну обробку та зменшуємо вхідне зображення (кадр) до розмірів 320 на 240 пікселів. Дані параметри розраховуються в

залежності від швидкодії процесору. Над зменшеним отриманим зображенням робимо серію перетворень за допомогою фільтрів. Для роботи будемо використовувати бібліотеку алгоритмів комп'ютерного зору OpenCV.

Першим фільтром буде перетворення кольорового зображення на чорно-біле. Таке перетворення значно скоротить кількість обчислень у майбутньому.

Наступний фільтр – інтерполяція для зменшення шумів. Для цього використовуємо метод найближчого сусіда. Для найкращого результату можна використовувати бікубічну інтерполяцію, але це значно підвищує кількість необхідних розрахунків. Третій фільтр, що використовується, це звичайний контрастний фільтр. За допомогою його на зображенні позбудемось зайвих об'єктів та виділимо необхідні об'єкти. Отримуємо чорно-біле зображення без шумів та з виділеними областями об'єктів.

Наступний етап – отримане зображення зберігаємо в оперативній пам'яті пристрою. Беремо наступний кадр із черги, повторяємо з ним всі попередні дії та порівнюємо поточний кадр із попереднім. При порівнянні ми проходимо всі пікселі почергово і якщо різниця в кольорі більша за 30%, то записуємо координати пікселя та поточний час в масив вхідних закодованих даних. Після того, як порівняли усі пікселі зображення, об'єднуємо сусідні пікселі у групи в нашому масиві закодованих даних.

На кожній ітерації обробки зображення може утворюватися декілька таких груп. Щоб їх розділити, при занесенні нової зони до масиву вхідних закодованих даних ми порівнюємо центр зони із деякою ΔX . У результаті утворюється двовимірний масив $M[X, Y]$, де X - індекс збережених зон, а Y – індекс черги, елементами якої є дані про час та зміну освітлення у цих зонах. Таким чином, ми виявляємо зони, де було змінене освітлення. Повторюємо ці дії для всіх кадрів із буферу. Друга частина алгоритму займається фільтрацією та розпізнаванням тексту із масиву закодованих даних.

Результатом першої частини роботи алгоритму є отриманий масив вхідних закодованих даних. Елементи двовимірного масиву $M[X, Y]$ зберігають зони освітлення (індекс X) та об'єкти із даними про час, місце та стан освітлення зони (індекс Y). Стан освітлення визначає, чи змінилося освітлення із яскравого на темне, чи навпаки. Для того, щоб пришвидшити обробку даних, можна здійснити розгалуження обрахунків на потоки. Кожний потік буде оброблювати зону $M[X]$ розмірністю $0..N$, де N – кількість станів освітлення, що обробила перша частина алгоритму для цієї зони. Далі алгоритм отримує перший елемент із черги, порівнює його стан із наступним, обраховує час, що пройшов між цими сигналами, та порівнює із еталонним варіантом "паузи" з азбуки Морзе. У нашому випадку пауза між символами становитиме 0.2 секунди, крапка – 0.2 секунди, тире 0.6 секунди, пауза між словами – 1.4 секунди. (За правилами азбуки Морзе, крапка становить $1 \times n$ тривалості паузи між символами, тире становить $3 \times n$ тривалості крапки, пауза між символами $7 \times n$). Якщо сигнал довший або менший за паузу, крапка чи тире довші, ніж на 0.1 секунди, вважаємо такий сигнал шумом. Якщо сигнал задовольняє вимогам, то результат заноситься до багатовимірного масиву декодованих символів $D[X, Y]$, де X – номер зони із масиву $M[X, Y]$, а Y – поточний декодований символ. Результатом роботи алгоритму є двовимірний масив $D[X, Y]$, що містить декодовані дані із камери. Результат виводимо у рядки на екран.

На основі даного алгоритму створено мобільний додаток, який розпізнає повідомлення, що передані за допомогою світла, тобто, є наглядною демонстрацією роботи технології Li-Fi.

4. Висновки

Бездротові комунікації стали таким самим ресурсом, як і електрика, продукти харчування або повітря, яке ми використовуємо щодня в нашому звичайному житті. Li-Fi вже у найближчому майбутньому стане використовуватись у побуті, адже:

- має великий потенціал. Зазвичай ми використовуємо дорогі і обмежені радіо-електромагнітні хвилі. Ці обмеження не співвідносяться з вимогами до станів бездротової передачі даних, і ми легко виходимо за рамки спектру;
- має велику ефективність; ефективність більшості базових станцій становить лише близько 5%. Останні 95% енергії використовуються для зниження температури базових станцій. Якщо ми говоримо про потужність: вже існує готова інфраструктура, тому що у нас є
 - в 10000 разів більше спектра у видимому світлі і у нас є дуже величезна кількість лампочок всюди у буденному житті;
 - є доступним; наприклад, всі пасажирів літака повинні вимкнути свої мобільні телефони під час польоту, але у салоні літака є світло, яке може стати джерелом Li-Fi;
 - є безпечним; радіохвилі можуть проникати крізь стіни, небезпека в тому що ваш сусід може проникнути в мережу і отримати ваші дані і ви не будете знати про це.

Список літератури

1. High-Speed Wireless Indoor Communication via Visible Light J. Grubor¹ , O.C. Gaete Jamett² , J.W. Walewski² , S. Randel² , K.-D. Langer¹ ¹ Fraunhofer Institute for Telecommunications, Heinrich-Hertz-Institut, Einsteinufer 37, 10587 Berlin ² Siemens Corporate Technology, Information and Communications, Networks and Multimedia Communications, Otto-Hahn-Ring 6, 81730 Munich.
2. Sustainable Energy-Efficient Wireless Applications Using Light Mohsen Kavehrad, Pennsylvania State University, IEEE Communications Magazine, December 2010.
3. Visible Light Communications: challenges and possibilities Dominic C. O'Brien¹ , Lubin Zeng¹ , Hoa Le-Minh¹ , Grahame Faulkner¹ , Joachim W. Walewski,² Sebastian Randel,² ¹ University of Oxford (UK); ² Siemens AG, Corporate Technology, Information and Communications, Munich (Germany).

СТВОРЕННЯ ЕФЕКТИВНОЇ ПРОЦЕДУРИ ЛІНГВІСТИЧНОГО МОДЕЛЮВАННЯ ДЛЯ ПРОГНОЗУВАННЯ ПРОЦЕСІВ РІЗНОЇ ПРИРОДИ

В даній статті розглянуто основні проблеми, підходи та етапи створення та застосування процедури лінгвістичного моделювання для задачі прогнозування процесів, що описуються часовими рядами. Оглянуто новизну підходу, потенційні недоліки та переваги методу, вирішення проблем, що виникають від час його застосування.

The subject of the article is the most important problems, approaches and phases of creating and application of linguistic modeling approach for solve prediction problems for process, which can be described as time-based series. Overview of innovations, potential drawbacks and advantages of approach, solution of problems, that arise during applications of method is done.

І.В.Бакланом в співпраці з своїми учнями розробляється концепція моделювання часових процесів з допомогою прийомів і методів лінгвістичного моделювання та синтаксичного розпізнавання образів [1,3].

Новизна даного підходу - Спроба трактувати часовий ряд як образ для розпізнавання із складною дискретною внутрішньою структурою на протигагу існуючим методам дослідження часових рядів добре відпрацьовані на неперервних функціях відомої форми і невідомих параметрів (наприклад регресійному аналізу). Цей підхід надає стійкості процесу моделювання – воно не залежить від співпадіння кількісних показників.

В рамках лінгвістичного підходу припускається, що внутрішня структура процесу може бути дискретною і описуваною деякою мовою з відповідною граматиною. Гіпотези можуть бути отримані за допомогою використання інтервальної математики [2] (автоматичного розбиття перебігу процесу в часі на однорідні проміжки) або надаватися самим дослідником з використанням знань предметної області.

Стадії процесу лінгвістичного моделювання:

- Збір даних, що описують схожі процеси
- Валідація даних та їх очистка від

недостовірних або неефективних з точки зору моделювання (шум, впливи інших, домінуючих факторів). В цілісному процесі аналізу даних на цей етап може припадати до 80% від всіх зусиль, витрачених на моделювання.

- Синтаксичне розпізнавання даних – розбивка навчальних даних на інтервали і присвоєння їм символів. Присвоєння символів довільне, вони служать лише для подальших етапів моделювання. Основною проблемою бачиться якраз цей етап, оскільки формальна процедура для нього не побудована, і скоріше за все не може бути побудована взагалі.

- Розпізнавання синтаксису – в моделі процесу інтервали даних замінюються символами. Отриману послідовність аналізують на наявність граматичних конструкцій. На виході отримуємо список граматичних конструкцій з імовірностями їх наявності в процесі, а також матрицю імовірностей переходу з символу в символ. Цей етап тісно перекликається з моделюванням (прихованих) марківських процесів, а також з методом правдоподібних траекторій.

- Верифікація моделі – перевірка моделі на тренувальній і тестовій вибірці, порівняння результатів, в разі необхідності повторне проходження процедури

розпізнавання символів і синтаксису.

- Моделювання цільового процесу – використовуючи побудовану модель, будується множина синтаксичних конструкцій з ймовірностями їх настання. Ця множина і використовується для рішень, для допомоги прийняття яких проводилося моделювання.

Одною з проблем побудови хорошої лінгвістичної моделі є розпізнавання форм, символів процесу, що розпізнається. Існує декілька ідей можливого вирішення даної проблеми:

- Підхід білого ящика, коли існує людина-експерт, що може надати правильну інформацію про розбивку конкретного процесу на фази. Очікуваними перевагами є висока достовірність в ситуаціях, коли відома внутрішня структура процесу, недоліками – недостовірність в протилежному випадку.

- Автоматичні або напівавтоматичні підходи: коли варіант розбивки вибирається на основі аналізу вхідних даних, без використання сторонніх даних. Такими підходами можуть бути:

- Розбивка на основі похідної процесу – як першої (перша похідна як напрям розвитку процесу – зростання або спад), так другої (прискорення зміни), і можливо похідних інших порядків.

- Розбивка на основі значень – розбивка точок даних на класи (кластеризація) беручи до уваги значення числового ряду в даній точці.

- Розбивка на основі повторюваних кривих – розбивка здійснюється на основі співставлення інтервалів процесу і зарані доступних типових кривих, що описують частини процесу.

Іншою проблемою є верифікація отриманої моделі. Хороша лінгвістична модель має володіти наступними характеристиками:

- Включати в себе символи для всіх інтервалів, що об'єктивно існують – тобто бути не простішою, ніж процес, що описується.

- Не включати в себе символи, що означають інтервали, не притаманні даному процесу. Імовірними джерелами таких інтервалів і символів, що їм відповідають, можуть бути шуми даних, а також вплив інших пов'язаних процесів.

- Побудований набір символів має давати на виході такі послідовності, для яких можливо скласти граматику, придатну для прогнозування. Якщо ж це правило не виконується, то можливе або порушення попередньої характеристики, або ж обраний часовий ряд є випадковим за своєю структурою, і описати його з допомогою лінгвістичного моделювання неможливо.

Для перевірки гіпотези необхідно провести порівняльне дослідження пропонованого методу з існуючими методами аналізу часових рядів: регресія ковзного середнього, метод найменших квадратів, авторегресія.

Описаний метод найкраще підходить для процесів аперіодичних процесів із складною внутрішньою структурою, які мають виражені внутрішні стани – тобто такі, що можуть бути описані прихованими марківськими моделі. Для моделей із структурою, що описуватиметься простими математичними моделями, скоріше за все, лінгвістичне моделювання не матиме подібних переваг.

Література

1. Баклан І.В. Лингвистическое моделирование: основы, методы, некоторые прикладные аспекты // Системные технологии. Региональный межвузовский сборник научных работ. – Выпуск 3 (74). – Днепропетровск, 2011. – с.10-19
2. Кантарович Л.В. О некоторых новых подходах к вычислительным методам и обработке наблюдений // Сибирский Математический Журнал. – 1962. Т.3, No. 5. – С.701-709
3. Fu K.S. Syntactic Pattern Recognition and Application. – N.J.: Prentice-Hall, Inc. Englewood Cliffs., 1982. – 596 p.

УНІФІКОВАНА АВТЕНТИФІКАЦІЯ НА WEB-РЕСУРСАХ ЧЕРЕЗ СОЦІАЛЬНІ МЕРЕЖІ

В даній статті розглянуто потенціал можливості автентифікації на web-ресурсах через соціальні мережі, розглянуто процес отримання доступу до даних користувачів на сторонніх ресурсах через протокол авторизації OAuth і запропоновано уніфікований застосунок для спрощення налаштування автентифікації через соціальні мережі.

In this article discussed the potential of authentication to web-resources through social networks, the process of accessing user data to third-party resources through authorization protocol OAuth and proposed unified application to simplify the configuration of authentication through social networks.

Ключові слова: АВТЕНТИФІКАЦІЯ, СОЦІАЛЬНІ МЕРЕЖІ, УНІФІКАЦІЯ, OAUTH.

1. Введення

Автентифікація, поряд з ідентифікацією і авторизацією, є важливою частиною процесу отримання доступу для роботи в інформаційній системі або на веб-ресурсі. Цей процес включає в себе перевірку належності вказаного ідентифікатора користувачеві, який взаємодіє із системою.

На найбільш вживаних сайтах неавторизовані користувачі мають лише право на читання деякої інформації, наприклад статті, коментарі інших користувачів та ін.; для інших дій – залишення власного коментаря чи завантаження користувацьких файлів – необхідно пройти процедуру авторизації. Для цього користувачі мають мати логін та пароль. За їх відсутності доведеться пройти, так звану, процедуру реєстрації – отримати логін і пароль для доступу в систему. Для спрощення даної процедури вже давно використовується авторизація через соціальні мережі, проте впровадження даного механізму потребує певних знань.

2. Виділення спільних рис соціальних мереж

Аккаунт в соціальних мережах як правило містить прізвище, ім'я, дату народження, «аватар» власника. Використання даної інформації значно спрощує процедуру

реєстрації користувача в системі, а також дозволяє легко оновлювати дані користувачів при їх зміні. В свою чергу, даний механізм дозволяє користувачам робити зміни в персональній інформації на веб-ресурсі «А», а далі просто оновлювати дані на бажаному ресурсі «Б» по даним з ресурсу «А». Оскільки вплив соціальних мереж постійно зростає, то і зростає важливість автентифікації через соціальні мережі.

Кожна мережа містить різний набір інформації про своїх користувачів, тому різні соціальні мережі можуть надавати різну кількість даних. Розглянемо, які дані можна отримати із соціальних мереж Вконтакте, Facebook, Google, Mail.ru та Однокласники (табл. 1).

Необхідно зауважити, що різні соціальні мережі по-різному структурують інформацію про освіту та роботу; також використовуються різні формати дат. В даній таблиці вказані не всі дані, які можна отримати із вказаних соціальних мереж. Як бачимо, всі вказані соціальні мережі надають ім'я, прізвище, фото та стать користувача; більшість надає дату народження, місто, країну, пошту.

3. Концепція уніфікованого застосунку для автентифікації через соціальні мережі

При проектуванні бази даних програміст має знати кількість стовпців для кожної з

Табл. 1. Порівняння соціальних мереж

	Вконтакте	Facebook	Google	Mail.ru	Однокласники
Ім'я	+	+	+	+	+
Прізвище	+	+	+	+	+
Дата народження	+	+		+	+
Фото	+	+	+	+	+
Стать	+	+	+	+	+
Місто	+	+			+
Місцезнаходження	+	+			
Країна	+	+			+
Пошта		+	+	+	
Мобільний	+				
Освіта	+	+			
Робота	+	+			
Військова служба	+				

таблиць, і цей процес ускладнюється тим, що різні соціальні мережі надають неоднакову кількість інформації про своїх користувачів. Тому пропонується створення уніфікованого застосунку, який буде надавати певний, наперед визначений набір даних про користувача і надаватиме можливість взаємодії із декількома соціальними мережами. Маючи такий застосунок програмісту не доведеться розбиратися із кожною соціальною мережею окремо, достатньо буде розібратися із специфікацією застосунку. Загальна схема роботи уніфікованого застосунку зображена на рис 1.

З отриманої інформації можна зробити висновок, що уніфікований застосунок має надавати наступний набір інформації: ім'я, прізвище, стать, дата народження, фото, місто, країна, пошта, а також надавати програмісту дати в певному форматі.

Отримати дані про користувача соціальної мережі можна через авторизацію по протоколу Open Authentication (OAuth) [1]. Для роботи із цим протоколом необхідно створити застосунок у відповідній соціальній мережі. Саме через цей застосунок відбуватиметься доступ до даних користувача. Для роботи із застосунком соціальної мережі необхідно знати ідентифікатор застосунку, захищений ключ і посилання для перенаправлення. Пропонується ці дані зберігати в окремій таблиці в БД для спрощення і захисту доступу до них. На додаток, захищений ключ застосунку не підлягає розголошенню. В разі спроби авторизації через соціальну мережу користувач буде перенаправлений на сторінку відповідного застосунку. На цій сторінці користувач має підтвердити надавання прав

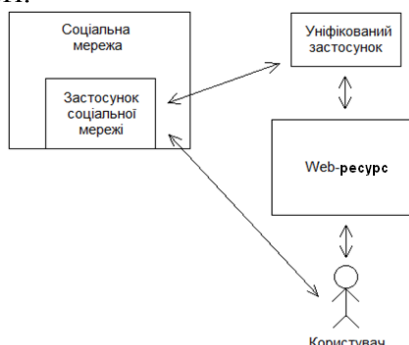


Рис. 1. Схема роботи уніфікованого застосунку

доступу до особистої інформації. В разі підтвердження, застосунок отримає ключ авторизації (authorization code) і відправить його на сервер web-ресурсу. В свою чергу web-ресурс може використати цей ключ авторизації для отримання токenu доступу (access token) від серверу соціальної мережі

[2]. Власне сам токен доступу може бути використаний для отримання доступу до захищеної інформації. Приклад схеми зображено на рис. 2. Даний протокол можна застосувати до всіх вище вказаних соціальних мереж.

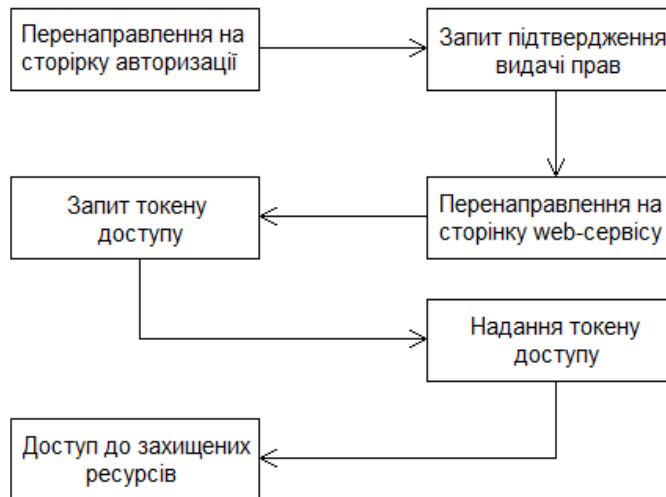


Рис. 2. Схема отримання доступу по

4. Висновок

В даній статті розглянуто схему отримання доступу по протоколу OAuth, що підтримується соціальними мережами, і може бути використаний для отримання доступу до інформації про користувача соціальної мережі. Також виконано порівняння декількох соціальних мереж за інформацією, що вони надають про своїх користувачів. На основі зібраних даних було запропоновано уніфікований додаток, що надаватиме можливість отримання фіксованої кількості даних про користувача соціальної мережі. В свою чергу ці дані можна використовувати на сторонніх web-ресурсах для автентифікації, спрощення реєстрації, оновлення особистих даних і т.д

Список літератури

1. J. Richer User Authentication with OAuth 2.0 [Електронний ресурс]. – Режим доступу: <http://oauth.net/articles/authentication/>
2. E. Hammer-Lahav, D. Recordon, D. Hart The OAuth 2.0 Authorization Protocol draft-ietf-oauth-v2-13 [Електронний ресурс]. – Режим доступу: <https://tools.ietf.org/html/draft-ietf-oauth-v2-13#section-4.1>

ШЕВЧЕНКО А. С.,
ЖДАНОВА О. Г.,
СПЕРКАЧ М. О.

Мінімаксна задача складання розкладу

Робота присвячена задачі складання розкладу без переривань в багатопроекторній обчислювальній системі. Побудовано математичну модель такої системи та розроблений евристичний алгоритм вирішення задачі на основі табу-пошуку. Проведені експериментальні дослідження, що підтверджують ефективність запропонованого алгоритму.

The scheduling task without interruption in multiprocessor systems is considered in the article. The mathematical model of the system is built and heuristic algorithm for solving the problem based on Tabu search is developed. Also conducted experimental studies that confirm the effectiveness of the proposed algorithm.

Задачі складання розкладу виникають при проектуванні та експлуатації автоматизованих систем обробки інформації та управління. Такі автоматизовані системи функціонують при безпосередній взаємодії із зовнішнім середовищем, і повинні за найкоротший час повернути середовищу результати обробки. В загальному формулюванні задача складання розкладу полягає в тому, щоб при заданих властивостях робіт та ресурсів і накладених на них обмеженнях знайти ефективний алгоритм впорядкування робіт, що оптимізує бажану міру ефективності.

Постановка задачі. Загальна система робіт для заданого набору ресурсів може бути визначена як система $S = \{T, \|\tau_{ij}\|\}$ таким чином: $T = \{T_1, \dots, T_n\}$ є набір робіт, що підлягають виконанню; $\|\tau_{ij}\|$ являє собою цілочисельну матрицю розміру $n \times m$, елемент якої $\tau_{ij} > 0$ є час виконання роботи $T_i (1 \leq i \leq n)$ на процесорі $P_j (1 \leq j \leq m)$. Показником ефективності розкладу σ є його довжина – максимальний час завершення $F(\sigma) = \max\{f_j(\sigma)\}$, де $f_j(\sigma)$ – сума тривалостей робіт, призначених на j -й процесор.

Існує вже досить велика кількість алгоритмів, що можуть розв'язувати таку задачу, такі як динамічне програмування, метод гілок та меж, випадковий пошук, евристичні та генетичні алгоритми тощо. Але точні алгоритми надто трудомісткі і складні для задач великих розмірностей, а наближені не завжди дають бажану точність розв'язку. В роботі пропонується модифікація алгоритму табу-пошуку для вирішення мінімаксної задачі складання розкладу.

Основними питанням, що виникають при застосуванні алгоритму Табу-пошуку до задачі складання розкладу в багатопроекторних системах, є питання збереження стану розкладу в пам'яті для уникнення потрапляння знову в той самий окіл локальних розв'язків та питання стратегії переходів до інших локальних мінімумів. Перше питання було вирішено за допомогою програмних можливостей об'єктно-орієнтованої мови програмування Java, що використовувалася для реалізації алгоритму, а саме можливості створення хеш-коду об'єктів, що є об'єктами-сутностями розкладу. Кожен такий стан, в якому система вже побувала, зберігається у так званому списку заборон. Щодо другого питання, була запропонована стратегія попарних

перестановок найдовшої роботи процесору, що найбільше завантажений, з будь-якими іншими роботами інших процесорів. Таким чином розроблений алгоритм табу-пошуку має вигляд:

Крок 1. Будуємо початковий розв'язок x за допомогою алгоритму ПРЗП (Процесор з раннім закінченням першим). Розв'язок може бути представлений деякою матрицею $\|r_{ij}\|$ розміру $n \times m$, де $r_{ij} = 1$ якщо i -робота виконується на j -процесорі, і $r_{ij} = 0$ в іншому випадку. Вважаємо $F_{pz} := F(x)$. Формуємо порожній список заборон (список станів, в яких була система).

Крок 2. Знаходимо процесор із «найдовшим» розкладом і попарно переставляємо з роботами іншого, наступного по довжині процесору. Знаходимо новий розв'язок z такий, що

а) $z \in N'(x)$, де $N'(x)$ частина околиці $N(x)$, взята випадковим чином.

б) $F(z) = \min\{F(y) \mid y \in N'(x)\}$;

в) перехід $x \rightarrow z$ не є забороненим або $F(z) < F_{pz}$.

Крок 3. Переходимо в нову точку $x := z$. Розраховуємо хеш-код розв'язку. Додаємо хеш-код до списку заборон.

Крок 4. Якщо $F(x) < F_{pz}$, то змінюємо рекорд $F_{pz} := F(x)$.

Крок 5. Якщо виконано критерій зупинки, то STOP, інакше - перейти на крок.2.

За рахунок того, що алгоритм на кожній новій ітерації обирає новий стан, в якому система ще не була, алгоритм переміщується по локальним мінімумам задачі, зберігаючи найкращі розв'язки задачі.

Для кожної системи робіт можна визначити нижню границю довжини розкладу, в основу розрахунку якої покладено підхід, при якому

передбачається, що кожна робота паралельно виконується усіма процесорами до її повного закінчення, тобто m паралельних процесорів розглядаються, як один більш продуктивний. За цієї умови величина

$$T^* = \sum_{i=1}^n \frac{1}{\sum_{j=1}^m \tau_{ij}}$$

дає найменш можливу довжину розкладу для системи робіт, яка задається матрицею $\|\tau_{ij}\|$.

Ефективність алгоритму залежить від набору вхідних даних задачі. Інтуїтивно зрозуміло, що швидкість і ефективність алгоритму для системи «на кожному з процесорів виконується досить велика кількість робіт, тривалості яких не дуже сильно відрізняються одна від іншої» і «на кожному з процесорів виконується невелика кількість робіт, тривалості яких сильно відрізняються одна від іншої» будуть різними. Отже, для аналізу ефективності запропонованого алгоритму були проведені серії випробувань на задачах із різною кількістю робіт та процесорів, різними характеристиками тривалостей робіт, та з різним обмеженням на кількість операцій табу-пошуку. В якості міри ефективності отриманих розкладів розглядалась величина відхилення їх довжини від «ідеального» значення (нижньої границі T^*). В таблиці 1 наведені результати експериментів над індивідуальними задачами, з конкретно заданими кількістю робіт (n), кількістю процесорів (m), математичним сподіванням часу виконання роботи E та розкидом цієї величини D . Для чистоти експерименту було вирішено зафіксувати математичне сподівання ($E = 1000$) і варіювати лише розкидом часу виконання робіт.

Таблиця 1. Результати табу пошуку для задач малих розмірностей

n	m	E	D	T _{max} (час процесора, що найбільш завантажений)		CPU Time(ms)		T*	Похибка, %
				Ітерації табу алгоритму		Ітерації табу алгоритму			
				50	200	50	200		
6	2	1000	100	2700.00075	2700.00075	143	502	2700.00075	0.00023
10	2	1000	100	4500.00123	4500.00123	145	503	4500.00123	0.000142
14	2	1000	100	6300.01434	6300.00004	145	502	6300.00734	0.000111
18	3	1000	100	5400.01185	5399.01185	147	514	5399.01185	0.018518
22	3	1000	100	7200.01109	7200.0001	150	507	7200.001	0.00014
26	3	1000	100	8100.01964	8100.01964	151	527	8099.01964	0.012346
30	3	1000	100	9000.02017	9000.02017	163	534	9000.02014	3.33E-07
34	3	1000	100	10800.02582	10700.02582	164	560	10303.0302	4.6018
38	4	1000	500	5000.00463	5000.00463	164	614	4750	5.000088
42	4	1000	500	5500.005058	5490.005058	175	669	5250	4.545542
46	4	1000	500	6000.006051	5999.006051	182	675	5750	4.166763
50	4	1000	500	6500.004903	6500.004903	185	684	6250	3.846226
54	4	1000	500	7000.00736	7000.00036	190	687	6750	3.57153
58	5	1000	500	7500.007464	7500.007464	191	690	7250	3.33343
62	5	1000	500	8000.008232	79980.00823	190	708	7750	3.1251

Таблиця 2. Результати табу пошуку для задач великих розмірностей

n	m	E	D	T _{max} (час процесора, що найбільш завантажений)		CPU Time(ms)		T*	Похибка, %
				Ітерації табу алгоритму		Ітерації табу алгоритму			
				100	500	100	500		
100	5	1000	100	18000.05331	18000.00611	217	753	18000.05331	1.56E-05
200	5	1000	100	36000.11118	36000.10053	261	911	36000.11118	9.47E-07
300	5	1000	100	54000.16269	54000.12269	312	874	54000.1578	9.06E-06
400	10	1000	100	36000.11001	36000.01001	513	891	36000.10071	2.58E-05
500	10	1000	100	45000.14971	45000.02971	544	990	45000.14572	8.86E-06
600	10	1000	100	54000.17448	53998.17448	546	993	54000.00048	0.000322
700	10	1000	500	63000.19595	63000.19445	602	994	63000.10495	0.000144
800	10	1000	500	72000.22292	72000.22292	572	998	72000.18292	5.56E-05
900	15	1000	500	81000.24333	81000.24333	753	1100	80999.24323	0.001235
1000	15	1000	500	33500.03466	33500.03466	911	1201	33333.3	0.497715
1100	15	1000	500	37000.03491	36998.03491	874	1313	36666.63	0.901094
1200	15	1000	500	40000.04526	40000.04506	891	1489	39999.96	0.000213
1300	20	1000	500	43500.0446	43489.0446	990	1500	43333.29	0.383344
1400	20	1000	500	35000.03961	35000.00261	993	1508	35000	0.000113
1500	20	1000	500	37500.04143	37497.04122	994	1514	37500	0.00011

Як бачимо, із результатів таблиці, порівнюючи з нижньою оцінкою розкладу, алгоритм показав себе дуже добре. Слід зауважити, що розв'язки Табу пошуку близькі до нижньої границі на задачах з невеликим розкидом, оскільки на таких задачах досить легко знаходиться розв'язок близький до оптимального. Інша ситуація спостерігається на задачах із великим розкидом. Краще алгоритм працює на задачах великих розмірностей, аніж на малих, що пояснюється більшою

можливістю перетасування задач.

Література.

1. Красовский Д.В. Алгоритмы решения задачи составления оптимального расписания без прерываний: Дис.... канд. физ.-мат. наук. М., 2007
2. Красовский Д.В., Фуругян М.Г. Некоторые алгоритмы составления многопроцессорных расписаний с использованием параллельных вычислений / ВЦ РАН. - М., 2006. - 27 с.
3. <http://www2.informatik.uni-osnabrueck.de/knust/class>

ПРОГРАММНО-ТЕХНОЛОГИЧЕСКИЙ КОМПЛЕКС ДЛЯ АВТОМОБИЛЬНОЙ ДИАГНОСТИКИ

В данной статье будет рассмотрена теория передачи информации в автомобиле и методы его диагностики, что такое электронный блок управления автомобиля, зачем он нужен и принцип его работы. Рассказывается про диагностическое оборудование, плюсы и минусы дилерских устройств. Краткое описание предложенного нами диагностического устройства, его технические характеристики и принцип действия.

This article will consider the theory of information transmission in the vehicle and methods of diagnosis, which is an electronic control unit of the car, why is it needed and how it works. It is told about the diagnostic equipment, the pros and cons of dealer devices. Brief description of our proposed diagnostic device, its technical characteristics and function.

Ключевые слова: АВТОМОБИЛЬНАЯ ДИАГНОСТИКА, CAN-ШИНА, STM32

1. Введение

Сейчас тяжело представить мир без автомобилей, темп жизни вынуждает двигаться все быстрее и быстрее. На ряду с этим значительно повысились требования комфорта, экономичности и эко стандарты, в связи с этим современные автомобили радикально отличаются от "Модели - Т" Генри Форда. Для удовлетворения запросов потребителей и экологических стандартов в транспортные средства были добавлены электронные узлы и агрегаты. Сегодня автомобиль, впрочем, как и любое транспортное средство, представляет собой совокупность механических узлов, контролируемых электронными блоками управления, параметры которых снимаются средствами датчиков. Такая сложность конструкции обусловлена необходимостью качественных и своевременных диагностик. Дилерские диагностики достаточно тяжелые в освоении и имеют не маленькие габариты, иногда некоторые операции выполняются часами, что заставляет тратить время в пустую, а некоторых полезных операций вовсе нет (таких как чтение закрытой области памяти, в

которой хранятся настоящий пробег авто и её VIN номер). Задача стоит в том чтобы создать компактное диагностическое устройство, а так же сократить время на выполнение операций, добавить возможности чтения памяти посредством диагностического порта автомобиля.

2. Теория автомобиля

Автомобиль – это сложный механизм, в котором каждый агрегат выполняет определенную функцию. Тормозная система, трансмиссия, подсветка, климатическая установка, управление приводом сидений – перечислять системы можно довольно долго, однако самым главным устройством в каждой такой системе является электронный блок управления. В некоторых современных автомобилях насчитывается до 80 электронный блок управления. [1]

Электронный блок управления (ЭБУ) принимает информацию от многочисленных датчиков, обрабатывает ее и, отталкиваясь от полученных данных, отдает команды исполнительным устройствам системы. ЭБУ является составным звеном бортовой сети автомобиля, он ведет постоянный обмен данными с другими компонентами

системы: антиблокировочной системой, автоматической коробкой передач, системами стабилизации и безопасности автомобиля, круиз-контролем, климат-контролем. Обмен информацией ведется посредством CAN-шины, которая объединяет все электронные и цифровые системы современного автомобиля в одну сеть. [2]

3. Теория CAN-шины

В середине 80-х годов прошлого столетия компанией BOSCH была предложена новая концепция сетевого интерфейса CAN (Controller Area Network). CAN-шина обеспечивает подключение любых устройств, которые

могут одновременно принимать и передавать цифровую информацию (дуплексная система). Собственно, шины представляет собой витую пару. Данная реализация шина позволила снизить влияние внешних электромагнитных полей, возникающих при работе двигателя и других систем автомобиля. Скорость передачи данных по CAN-шине может достигать до 1 Мбит/с. Большим плюсом в проведении диагностических работ является наличие единого унифицированного диагностического разъема (колодка OBD) увидеть его можно на рисунке 1. [3]

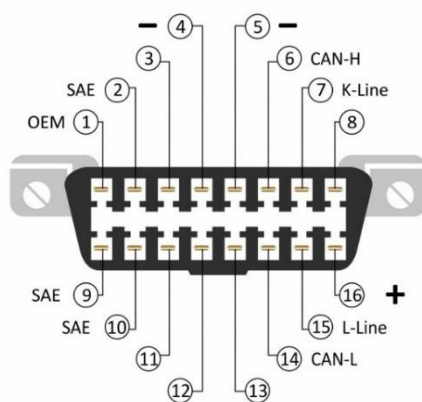


Рис. 1. Диагностический разъем автомобиля

В качестве примера CAN сообщений (пакетов) чтение ошибок ЭБУ на машине Range Rover рисунок 2. Мы видим список идентификаторов блока, который принимает команды по одному идентификатору, а отвечает по другому, на примере выделены галочками номер

отправки (7E0) и номер приема (7E8). На запрос диагностического оборудования ЭБУ начинает отвечать пакетом CAN сообщений. Данный пакет интерпретируется диагностической программой и выводит на экран код ошибки.

ID:	Count:	Data:	Delta:	No:	Time:	ID:	Data:	Ascii:
<input checked="" type="checkbox"/> 7E0	2	00 00 00 00 00 00 00 00	0	7	2.816174	7E8	25 23 02 51 13 27 0A 1A	%#.Q.'..
<input checked="" type="checkbox"/> 7E8	18	21 23 25 0A 17 23 00 00	0	8	2.858071	7E8	26 87 23 00 90 13 27 01	&##.E.'.
				9	3.025884	7E8	27 93 17 27 04 86 17 27	'..'.t.'
				10	3.068411	7E8	28 02 38 00 27 05 32 00	{.8.'.'2.
				11	3.109765	7E8	29 23 01 16 17 27 01 98	}#...'.'0
				12	3.152292	7E8	2A 00 23 01 05 17 23 01	*.###.#.
				13	3.193697	7E8	2B 08 00 27 24 55 00 27	+..'qU.'
				14	5.375263	7E8	2C 01 10 17 27 C1 21 00	...'E'!
				15	5.442150	7E8	2D 23 C1 26 00 23 C1 01	-#B&.#B.
				16	5.458596	7E8	2E 00 23 17 12 85 23 17	..#...#.
				17	5.500549	7E8	2F 12 84 23 17 12 57 23	/..#..U#
				18	5.542502	7E8	20 92 06 68 23 C1 51 08	'.h#BQ.
				19	5.710807	7E8	21 23 25 0A 17 23 00 00	!#%.'..#..

Рис. 2. Пакеты CAN-шины чтения ошибок с ЭБУ

4. Диагностика автомобиля

Компьютерная диагностика автомобиля – это процесс, при котором происходит чтение неисправностей и коррекция работы автомобиля. Для этого применяются дилерские сканеры и прочие системы. Современное

диагностирующее оборудование и программное обеспечение позволяют считывать и засекать малейшие изменения в работе систем управления двигателем, трансмиссией, панели приборов и прочих.

Благодаря ей удастся получить правдивую информацию о текущем состоянии блоков управления, деталей и узлов машины. [4] Диагностика автомобиля, является неотъемлемой частью его эксплуатации. Она позволяет преждевременно предотвратить серьезные поломки авто, повысить безопасность движения на дорогах,

снизить топливные затраты и сэкономить затраты на ремонт.

5. Предлагаемое решение

Нами была предложена диагностическая система, построенная на базе процессора STM32F107 компании STMicroelectronics, подробно про характеристики процессора можно ознакомиться в таблице 1.

Табл. 1. Характеристики STM32F107

Название	Описание
Процессор	72 МГц Cortex-M3
Flash память	256 Кбайт
ОЗУ	64Кбайта
АЦП	Два 12-разрядных (16 внешних каналов)
ЦАП	Два 12-разрядных
USB OTG	Имеется
CAN	Два порта

Данное диагностическое оборудование позволяет использовать нестандартный функционал, такой как считывание и запись конфигурации (комплектации) автомобиля, считывание и запись программного обеспечение ЭБУ, программирование электронных ключей, проводить полный комплекс диагностических процедур. Помимо всего прочего данное диагностическое оборудование поможет правоохранительным органам искать "Двойников" и угнанные автомобили, благодаря возможности считывания закрытой области памяти, в которой

хранится такая информация как: VIN номер, пробег, идентификационные номера блоков управления.

Чтение конфигурации происходит отправкой нескольких команд, а ее запись средствами команд не поддерживается, так как у блока нет команд записи. Для решения этой проблемы можно расширить возможности блока, на рисунке 3 показано часть кода этого алгоритма, а именно отправку команд через наше устройство, где оно получает команды через виртуальный COM Port, и передает их в CAN-шину автомобиля.

```
PackList pl; // Объект для хранения пакетов с устройства
ARM::SetPackList(pl); // Указываем куда нужно записывать пакеты

// Отправляем команды ЭБУ для идентификации автомобиля
ARM::SendStringToCan(IDevice::GetDeviceParam()->uCanId, "01 02 03 04 00 00 00 00", TRUE);
ARM::SendStringToCan(IDevice::GetDeviceParam()->uCanId, "05 06 07 08 00 00 00 00", TRUE);
while(ARM::IsWorking()); // Ожидаем окончания работы

ARM::SetPackList(NULL); // Отключаем чтение ответов

// Входим в режим программирования
ARM::SendStringToCan(IDevice::GetDeviceParam()->uCanId, "10 10 10", TRUE);

/* Расширяем возможности блока, а именно добавляем возможность записи конфигурации */
ARM::SendStringToCan(IDevice::GetDeviceParam()->uCanId, "2F C0 00 16 80 C0 00 9F");
ARM::SendStringToCan(IDevice::GetDeviceParam()->uCanId, "20 8C 00 03 EC 64 14 C0");
ARM::SendStringToCan(IDevice::GetDeviceParam()->uCanId, "21 9F 8C 00 03 C3 EA 97");
ARM::SendStringToCan(IDevice::GetDeviceParam()->uCanId, "22 1C 9F 8C 00 03 CA 0A");
ARM::SendStringToCan(IDevice::GetDeviceParam()->uCanId, "23 97 1C 8B 4A A8 1A E3");
ARM::SendStringToCan(IDevice::GetDeviceParam()->uCanId, "24 23 04 B0 A8 40 E2 06");
/* ..... */
ARM::SendStringToCan(IDevice::GetDeviceParam()->uCanId, "26 03 EC 8C 16 C0 9F 8C");
ARM::SendStringToCan(IDevice::GetDeviceParam()->uCanId, "27 00 03 EC 70 04 C0 A8");
ARM::SendStringToCan(IDevice::GetDeviceParam()->uCanId, "28 00 E2 14 C0 0A C0 10");
ARM::SendStringToCan(IDevice::GetDeviceParam()->uCanId, "29 02 80 C3 E1 AA 10 E3");
ARM::SendStringToCan(IDevice::GetDeviceParam()->uCanId, "2A 0E 06 80 CF 81 82 10");
ARM::SendStringToCan(IDevice::GetDeviceParam()->uCanId, "2B A8 00 E3 09 06 80 A8");
ARM::SendStringToCan(IDevice::GetDeviceParam()->uCanId, "2C 00 E2 06 9B 00 13 88");
ARM::SendStringToCan(IDevice::GetDeviceParam()->uCanId, "2D 9F 8C 00 03 EC 86 15");
ARM::SendStringToCan(IDevice::GetDeviceParam()->uCanId, "2E C0 04 B0 A8 40 E3 06");
/* ..... */
```

Рис. 3. Пример команды записи конфигурации

Так, для примера, диагностическое устройство компании "Rotunda" для автомобилей марки Land Rover и Jaguar, имеет большие габариты, а наше решение поместится в спичечный коробок. Разница в весе составит в несколько раз меньше. На рисунке 4 представлены: диагностическое устройство компании "Rotunda" (номер 1) и наши устройства (номер 2), слева представлена первая

версия устройства, справа две модернизированные копии, габаритные размеры которых уменьшились дважды, в сравнении с предшественницей. Такая разница обусловлена использованием другой элементной базы и сокращенным количеством портов ввода-вывода информации. Основную площадь платы занимает процессор.



Рис. 4. Диагностические устройства

6. Заключение

На сегодняшний день наша разработка является самым маленьким диагностическим устройством, которое имеет низкую себестоимость, высокое быстродействие, простоту в эксплуатации и расширенный функционал. Данное устройство является полноценным диагностическим

оборудованием, которое позволяет проводить весь спектр дилерских процедур, таких как (сброс адаптаций смесеобразования, активация регенерации противосажевого фильтра и другие) и производить процедуры, которые не возможно произвести иными устройствами.

Список литературы

1. Электронный источник: <http://autopulse.net/media/ebu-v-avtomobile.html> Дата посещения 13.04.2016
2. Электронный источник: <http://vodi.su/ebu-ustroystvo-printsip-raboty/> Дата посещения 11.04.2016
3. Электронный источник: http://www.radioradar.net/radiofan/motorcar_enthusiast/can_auto.html Дата посещения 10.04.2016
4. Электронный источник: http://amastercar.ru/automaster/komp_diagnostika.shtml Дата посещения 13.04.201

РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ КРОКІВ ПОБУДОВИ ГІБРИДНОЇ ЛІНГВІСТИЧНОЇ МОДЕЛІ ТА ПРИХОВАНОЇ МАРКОВСЬКОЇ МОДЕЛІ

Стаття присвячена програмній реалізації етапу побудови гібридної моделі “Лінгвістична модель — прихована марковська модель” динамічного процесу - отримання стохастичної матриці передування елементів алфавіту формальної граматики.

The subject of the article is a software implementation of building a hybrid model “Linguistic model – hidden Markov model” of a dynamic process - getting stochastic matrix of precedence of formal grammar alphabet elements.

1. Вступ

Процес побудови лінгвістичної моделі складається з чотирьох основних етапів [1]. Саме реалізацією цих етапів є програма LingModel, виконана на мові Java в проектному середовищі IntelliJ IDEA.

2. Основна частина

Лінгвістичної моделлю для заданого часового ряду $X^0 = \{X_1^0, X_2^0, \dots, X_n^0\}$, розподілу ймовірностей визначеного функцією розподілу $F(x)$ і алфавіту $A = \{a_1, a_2, \dots, a_n\}$ називається множина прихованих Марковських моделей (далі ПММ) $M = \{M_1, M_2, \dots, M_k\}$, яка може бути отримана в результаті наступних побудов [1,2].

Нехай дані часові ряди вираховань $X^i = \{X_1^i, X_2^i, \dots, X_{n-i}^i\}$, $i \in \overline{1..n-1}$ такі, що $\forall i \in \overline{1..n-1}, \forall j \in \overline{1..n-1}, X_j^i = X_{j+1}^{i-1} - X_j^{i-1}$. Для кожного X^i визначимо інтервали $[p_1^i, p_2^i], [p_2^i, p_3^i], \dots, [p_m^i, p_{m+1}^i]$, де p_j^i виберемо так, що якщо N_j^i - кількість елементів ряду X^i , що належать інтервалу $[p_j^i, p_{j+1}^i]$, то $N_j^i = F(p_{j+1}^i) - F(p_j^i)$. Слід зауважити, що кількість інтервалів для кожного часового ряду вираховань постійно і дорівнює кількості символів в заданому алфавіті.

Для кожного X^i побудуємо лінгвістичний ланцюжок $a_1^i a_2^i \dots a_n^i$ із символів алфавіту A , асоціювавши кожен елемент X_j^i з символом a_t , якщо X_j^i належить інтервалу $[p_t^i, p_{t+1}^i]$.

Нехай $P^i(a)$ - частота виникнення символу a в ланцюжку $a_1^i a_2^i \dots a_n^i$, а $P^i(a|b)$ - частота виникнення пари символів ab в ланцюжку $a_1^i a_2^i \dots a_n^i$. Тоді, станам ПММ відповідають символи алфавіту, а ймовірність переходу зі стану a_p^j в a_q^j дорівнюватиме:

$$\frac{P^i(a_p^j | a_q^j)}{P^i(a_p^j)} \quad (1)$$

Було створено програмний засіб, що дозволяє будувати лінгвістичну модель для заданого часового ряду на основі двох алфавітів: 0..9 и z..a, A..Z і трьох імовірнісних розподілів: рівно ймовірного, нормального (Гауса) і Лапласа.

Для побудови наступної лінгвістичної моделі були обрані дані, середнє значення яких постійно в часі, алфавіт $A = \{0, 1, \dots, 9\}$, а розподіл F - рівно ймовірний.

LingModel											
		1-й вычет		2-й вычет		3-й вычет		4-й вычет		5-й вычет	
		0	1	2	3	4	5	6	7	8	9
0	0.24	0.12	0.06	0.02	0.00	0.00	0.02	0.05	0.11	0.37	
1	0.12	0.14	0.13	0.04	0.01	0.00	0.07	0.09	0.18	0.21	
2	0.08	0.14	0.13	0.11	0.04	0.00	0.10	0.12	0.15	0.13	
3	0.02	0.07	0.10	0.17	0.10	0.00	0.30	0.10	0.09	0.04	
4	0.01	0.02	0.06	0.12	0.19	0.00	0.48	0.10	0.03	0.01	
5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
6	0.02	0.04	0.04	0.12	0.14	0.00	0.45	0.11	0.05	0.03	
7	0.03	0.06	0.10	0.14	0.06	0.00	0.22	0.18	0.12	0.08	
8	0.09	0.14	0.15	0.10	0.03	0.00	0.09	0.11	0.17	0.14	
9	0.21	0.14	0.10	0.03	0.01	0.00	0.05	0.06	0.11	0.28	

Рис. 1. Матрица ПММ для першого вирахування

LingModel											
		1-й вычет		2-й вычет		3-й вычет		4-й вычет		5-й вычет	
		0	1	2	3	4	5	6	7	8	9
0	0.14	0.08	0.06	0.01	0.00	0.01	0.00	0.02	0.06	0.62	
1	0.08	0.10	0.09	0.04	0.00	0.04	0.01	0.08	0.16	0.40	
2	0.05	0.09	0.15	0.07	0.00	0.07	0.06	0.17	0.17	0.17	
3	0.01	0.03	0.07	0.20	0.00	0.16	0.27	0.15	0.06	0.04	
4	0.00	0.00	0.00	0.25	0.00	0.25	0.12	0.25	0.12	0.00	
5	0.01	0.05	0.13	0.24	0.00	0.20	0.20	0.08	0.04	0.04	
6	0.00	0.03	0.09	0.41	0.00	0.20	0.18	0.07	0.01	0.01	
7	0.03	0.09	0.25	0.20	0.00	0.10	0.07	0.08	0.10	0.08	
8	0.09	0.20	0.25	0.07	0.00	0.03	0.03	0.07	0.08	0.17	
9	0.37	0.25	0.09	0.02	0.00	0.02	0.01	0.02	0.06	0.17	

Рис. 2. Матрица ПММ для другого вирахування

LingModel											
		1-й вычет		2-й вычет		3-й вычет		4-й вычет		5-й вычет	
		0	1	2	3	4	5	6	7	8	9
0	0.06	0.05	0.05	0.02	0.01	0.01	0.01	0.05	0.19	0.55	
1	0.05	0.06	0.05	0.03	0.03	0.01	0.04	0.19	0.34	0.18	
2	0.03	0.07	0.09	0.07	0.03	0.04	0.10	0.28	0.22	0.07	
3	0.02	0.03	0.05	0.09	0.07	0.11	0.30	0.21	0.09	0.03	
4	0.01	0.01	0.04	0.05	0.18	0.34	0.29	0.05	0.02	0.00	
5	0.01	0.01	0.04	0.10	0.36	0.30	0.12	0.04	0.02	0.01	
6	0.01	0.03	0.10	0.23	0.32	0.11	0.08	0.07	0.04	0.01	
7	0.06	0.15	0.27	0.16	0.05	0.05	0.07	0.08	0.08	0.03	
8	0.17	0.32	0.18	0.06	0.02	0.02	0.04	0.07	0.07	0.05	
9	0.57	0.19	0.07	0.02	0.01	0.01	0.01	0.04	0.05	0.04	

Рис. 3. Матрица ПММ для третьего вирахування

LingModel											
		1-й вычет		2-й вычет		3-й вычет		4-й вычет		5-й вычет	
		0	1	2	3	4	5	6	7	8	9
0	0.03	0.03	0.02	0.02	0.00	0.01	0.02	0.06	0.28	0.53	
1	0.03	0.06	0.05	0.04	0.02	0.00	0.04	0.22	0.42	0.11	
2	0.02	0.05	0.06	0.05	0.06	0.06	0.12	0.35	0.20	0.03	
3	0.01	0.03	0.05	0.08	0.07	0.12	0.34	0.22	0.06	0.02	
4	0.00	0.02	0.02	0.04	0.19	0.39	0.27	0.05	0.02	0.00	
5	0.00	0.01	0.03	0.11	0.45	0.28	0.08	0.03	0.01	0.00	
6	0.02	0.05	0.11	0.27	0.31	0.08	0.08	0.06	0.03	0.00	
7	0.04	0.18	0.31	0.18	0.05	0.03	0.06	0.08	0.06	0.03	
8	0.22	0.33	0.19	0.05	0.02	0.01	0.03	0.04	0.07	0.03	
9	0.71	0.13	0.04	0.01	0.00	0.00	0.01	0.04	0.02	0.03	

Рис. 4. Матрица ПММ для четвертого вирахування

LingModel											
		1-й вычет		2-й вычет		3-й вычет		4-й вычет		5-й вычет	
		0	1	2	3	4	5	6	7	8	9
0	0.02	0.02	0.02	0.01	0.01	0.00	0.01	0.04	0.12	0.74	
1	0.03	0.04	0.05	0.02	0.01	0.01	0.04	0.18	0.38	0.24	
2	0.02	0.05	0.07	0.04	0.02	0.04	0.14	0.34	0.22	0.06	
3	0.01	0.03	0.05	0.05	0.05	0.22	0.42	0.14	0.02	0.02	
4	0.00	0.02	0.05	0.11	0.17	0.38	0.20	0.04	0.02	0.01	
5	0.01	0.03	0.06	0.22	0.31	0.25	0.10	0.02	0.01	0.01	
6	0.01	0.03	0.15	0.42	0.17	0.08	0.04	0.06	0.02	0.01	
7	0.03	0.17	0.37	0.15	0.04	0.03	0.04	0.07	0.05	0.05	
8	0.16	0.41	0.20	0.04	0.01	0.02	0.03	0.04	0.05	0.04	
9	0.60	0.21	0.05	0.02	0.01	0.01	0.01	0.02	0.04	0.04	

Рис. 5. Матрица ПММ для пятого вирахування

LingModel											
		1-й вычет		2-й вычет		3-й вычет		4-й вычет		5-й вычет	
		0	1	2	3	4	5	6	7	8	9
0	0.02	0.02	0.01	0.00	0.00	0.00	0.01	0.04	0.11	0.78	
1	0.04	0.05	0.04	0.02	0.02	0.02	0.03	0.15	0.40	0.23	
2	0.03	0.05	0.08	0.04	0.02	0.03	0.16	0.36	0.18	0.05	
3	0.01	0.02	0.06	0.06	0.05	0.24	0.41	0.11	0.03	0.01	
4	0.01	0.02	0.04	0.08	0.23	0.40	0.16	0.04	0.02	0.01	
5	0.01	0.02	0.05	0.27	0.36	0.20	0.05	0.03	0.01	0.01	
6	0.01	0.04	0.17	0.47	0.12	0.07	0.04	0.04	0.03	0.01	
7	0.04	0.16	0.41	0.14	0.04	0.05	0.04	0.06	0.04	0.02	
8	0.13	0.47	0.20	0.03	0.01	0.02	0.02	0.05	0.05	0.03	
9	0.63	0.22	0.05	0.01	0.01	0.00	0.02	0.02	0.02	0.03	

Рис. 6. Матрица ПММ для шостого вирахування

Для спрощення сприйняття, більші значення матриць зафарбовані темнішим кольором. Значення в наведених матрицях слід тлумачити так: в матриці для п'ятого вирахування значення 0.74 в комірці на

перетині рядка 0 і стовця 9 означає, що ймовірність переходу в ПММ для п'ятого вирахування між станами, відповідними символам вхідного алфавіту «0» і «9» (в напрямку 0 → 9) дорівнює 0.74.

Можна помітити, що в міру збільшення номера вирахування, найбільші значення ймовірності прагнуть розташуватися біля головної діагоналі матриці. Це прагнення можна пояснити тим, що після збільшення значення на деяку величину найбільш імовірним буде його зменшення на таку ж саму величину (міркування можна

застосувати тільки в разі постійності середнього значення в часі).

3. Основні результати і висновки

Були проведені чисельні експерименти щодо побудови лінгвістичної моделі варіантів часового ряду - різниць 1-го – 6-го вирахування, та наступний аналіз отриманих результатів.

Список літератури

1. Баклан І.В. Інтервальний підхід до побудови лінгвістичної моделі // Системні технології. Регіональний міжвузівський збірник наукових праць. – Випуск 3(86). – Дніпропетровськ, 2013. – С.3-8.
2. Баклан И.В. Лингвистическое моделирование: основы, методы, некоторые прикладные аспекты // Системные технологии. Региональный межвузовский сборник научных работ. – Випуск 3 (74). – Днепропетровск, 2011. – с.10 – 19.
3. Баклан І. В. Класифікація моделей марковського типу: Наукова монографія // Баклан І. В., Степанкова Г.А. — К. : НАУ, 2012. — 84 с.

ПРОБЛЕМА РОБОТИ З ДИНАМІЧНОЮ ЗМІННОЮ СТРУКТУРИ ОБ'ЄКТІВ НА РІВНІ БАЗИ ДАНИХ.

В даній роботі розглядаються архітектурні рішення для побудови та роботи з моделями баз даних типу сутність-атрибут-значення. Здійснюється аналіз архітектурних рішень та можливостей використання моделі в рамках стандартного реляційно-орієнтованого підходу до проектування баз даних для спрощення роботи користувачів.

In this paper we consider architectural solutions for building models and working with databases with entity-attribute-value pattern. The analysis of architectural solutions and the possibilities of using models under the standard relational-oriented approach to database design to simplify the user experience.

Зазвичай в більшості існуючого програмного забезпечення при розробці нового функціоналу або додавання нових функцій до вже існуючих модулів, виникає проблема динамічної зміни існуючих об'єктів, додавання нових типів, нових сутностей, нових властивостей, тощо без зупинки роботи існуючої системи. При стандартному, реляційно-орієнтованому підході, необхідно зробити наступні кроки: 1) зупинити роботу користувачів в програмному забезпеченні; 2) оновити структуру таблиць.[2] Однак, для більшості промислових систем, котрі працюють в online-режимі 24/7, це не завжди є можливим, так як повна зупинка роботи користувачів в системі може привести до значних збитків підприємств або компаній.

В якості одного з можливих варіантів вирішення подібної проблеми пропонується використовувати модель сутність-атрибут-значення. Однак, застосування моделі сутність-атрибут-значення в чистому вигляді має безліч проблем, такі як: складність підтримки, необхідність оптимізації при роботі.[1]

Незважаючи на це, при правильному конфігурування функції кешування та індексів, можна добитись хоча б близької по швидкості роботи до звичайних реляційно-орієнтованих моделей.

Для реалізації цієї ідеї, спроектовано та розроблена архітектура програмного забезпечення, в котрому реалізовані наступні функції: 1) створення та конфігурування моделі; 2) перегляд поточної моделі в режимі реляційної бази даних; 3) міграція даних між різними системами управління базами даних[3]. Використання розробленого програмного забезпечення спростить конфігурування моделей типу сутність-атрибут-значення для спеціалістів, які не мають достатнього рівня кваліфікації для проектування моделей баз даних.

В роботі наведено архітектура програмного забезпечення, котре дозволяє збільшити продуктивність моделей типу сутність-атрибут-значення, та їх використання для реляційно-орієнтованих моделей. Програмне забезпечення використовує кілька інструментів оптимізації, які пришвидшують роботу.

Список літератури

1. Подход к реализации бд со статической структурой на основе модели данных eav. [Електронний ресурс] – Режим доступу: <http://cyberleninka.ru/article/n/podhod-k-realizatsii-bd-so-staticheskoy-strukturoy-na-osnove-modeli-dannyh-eav>
2. The Anti-Pattern – EAV(il) Database Design? EAV [Електронний ресурс] – Режим доступу: <https://mikesmithers.wordpress.com/2013/12/22/the-anti-pattern-eavil-database-design/>
3. Крэнке Д. Теория и практика построения баз данных. – СПб: Питер, 2003 – 800 с.

*ШУЛЬКЕВИЧ Т.В.,
СЕЛІН Ю.М.,
БАКЛАН І.В.*

РЕАЛІЗАЦІЯ ПОБУДОВИ ЛІНГВІСТИЧНОЇ МОДЕЛІ

В докладе рассмотрены вопросы построения лингвистической модели для анализа и прогнозирования динамических процессов. Приведен математический аппарат реализации лингвистической модели.

The report addressed issues of building linguistic models for the analysis and prediction of dynamic processes. An implementation of the mathematical apparatus of the linguistic model.

Однією з головних складових задачі прогнозування певних процесів є проблема аналізу цього процесу. Під терміном «аналіз процесу» будемо розуміти наступне. По-перше, необхідно розпізнати сам факт, що відбувся певний процес. По-друге, визначити інтенсивність і час його протікання, а також, по можливості, розпізнати його природу (причини та сутність).

У наявних задачах розпізнавання образів високу ступінь важливості має структурна інформація, яка описує кожний об'єкт. При цьому процес розпізнавання повинен не тільки віднести об'єкт до визначеного класу (тобто класифікувати його), але й описати ті сторони об'єкта, які роблять небажаним віднесення його до іншого класу. Типовим прикладом такої задачі є розпізнавання образів, або ж, у більш широкому смислі, аналіз сцен. Для цього класу задач розпізнавання розглянуті об'єкти зазвичай складні, а кількість необхідних ознак іноді занадто велика. Це робить дуже привабливою ідею описання складного об'єкту розпізнавання у вигляді ієрархічної композиції більш простих образів. Саме у випадку аналізу складних об'єктів та наявності великої кількості можливих описів непрактично вважати, що кожний опис визначає клас.

Внаслідок цього для розв'язання задачі розпізнавання стає необхідним створення докладного описання кожного об'єкта, а не просто його віднесення до деякого класу.

Для подання ієрархічної структурної інформації, яка міститься в кожному образі, тобто для описання образу за допомогою

більш простих підобразів, кожний з яких в свою чергу описаний за допомогою ще більш простих підобразів і так далі, був запропонований лінгвістичний (синтаксичний), або структурний, підхід [1]. Цей підхід виявляє аналогію між структурою образів та синтаксисом мов. В даному випадку образи визначаються як конструкції, поєднані різними способами з підобразів аналогічно тому, як фрази та речення будуються за допомогою з'єднувальних слів, а слова будуються за допомогою поєднання букв. Безперечно, що для забезпечення корисності цього підходу, відібрані найпростіші підобрази, які називають примітивами образу, і які повинні розпізнаватися значно простіше, ніж самі образи. „Мову”, яка забезпечує структурний опис образів в термінах набору примітивів образу та операцій їх об'єднання, іноді називають також „мовою опису образів”. Правила, які керують об'єднанням примітивів до образів, зазвичай визначаються так званою „граматикою” мови опису образів. Після того як кожний примітив образу ідентифікований, процес розпізнавання завершується виконанням синтаксичного аналізу, тобто граматичним розбором „речення”, яке описує об'єкт. Ця процедура визначає – правильне воно чи ні синтаксично (або граматично) відносно заданої граматики. Одночасно синтаксичний аналіз дає також опис структури речення, яке подає цей об'єкт (зазвичай у формі деревоподібної структури).

Лінгвістичний підхід до розпізнавання образів дозволяє описувати велику сукупність складних об'єктів,

використовуючи невеликі сукупності простих примітивів об'єкта та правил граматики. Різні відношення, а також операції об'єднання, визначені на підоб'єктах, зазвичай можуть бути виражені логічними або математичними операціями.

Лінгвістична система розпізнавання образів складається з трьох основних частин – блока попередньої обробки даних, блока описання об'єкта та блока синтаксичного аналізу.

Проста блок-схема зображена на рис. 1. Блок попередньої обробки здійснює, поперше, кодування та апроксимацію об'єкта, а по-друге, фільтрацію, відновлення та посилення сигналів (вимірів). Вихідний об'єкт спочатку кодується або ж апроксимується таким чином, щоб було зручно для подальшого опрацювання.

Інше кажучи, кожний попередньо опрацьований об'єкт сегментується на складові частини та примітиви об'єкта на підставі наперед визначених синтаксичних операцій та операцій об'єднання; в свою чергу кожна складова частина об'єкту ідентифікується за допомогою заданого набору примітивів. Тепер кожний об'єкт подається набором примітивів із визначеними синтаксичними операціями. Наприклад, в термінах операції „конкатенації” кожний об'єкт подається рядком примітивів, які утворюють ланцюжок. Більш досконалі системи повинні мати можливість визначати різні синтаксичні відношення всередині об'єкту.

Рішення щодо правильної синтаксичної побудови об'єкту, тобто належність його до класу об'єктів, описаних цим синтаксисом, приймається „Аналізатором синтаксису” або „блоком граматичного розбору”. Відпрацьовуючи синтаксичний аналіз або

граматичний розбір, аналізатор зазвичай може створити повне синтаксичне описання об'єкту в термінах граматичного розбору або дерева граматичного розбору, якщо тільки подання об'єкту синтаксично правильне.

Найпростішим способом розпізнавання є метод „співпадання з еталоном”. Рядок примітивів, який описує вихідний об'єкт, порівнюється із рядками примітивів, що представляють еталонні об'єкти. У відповідності до обраного „співпадання” або чинника подібності, вихідний об'єкт зараховується до того ж класу, до якого відноситься об'єкт-прототип, що найкращим чином відповідає вихідному об'єкту. Інформація про ієрархічну структуру в цьому випадку ігнорується. З іншого боку, повний граматичний розбір рядку, відповідного до вхідного об'єкту, використовує повний структурний опис об'єкту. Між цими способами вирішення задачі розпізнавання образу розташований цілий ряд проміжних результатів.

Щоб отримати граматику, яка описує інформацію про структуру класу об'єктів, є необхідним спосіб виведення граматики, який може виводити граматику на основі заданого набору об'єктів у мовноподібному поданні для навчання. Структурний опис розглянутого класу об'єктів – результат навчання на реальних зразках об'єктів з цього класу. Цей опис надалі використовується у формі граматики для опису об'єктів та синтаксичного аналізу. Навчання в більш загальному вигляді повинно мати в складі здібність до навчання на найкращій множині примітивів та відповідному структурному опису для розглянутих класів об'єктів.

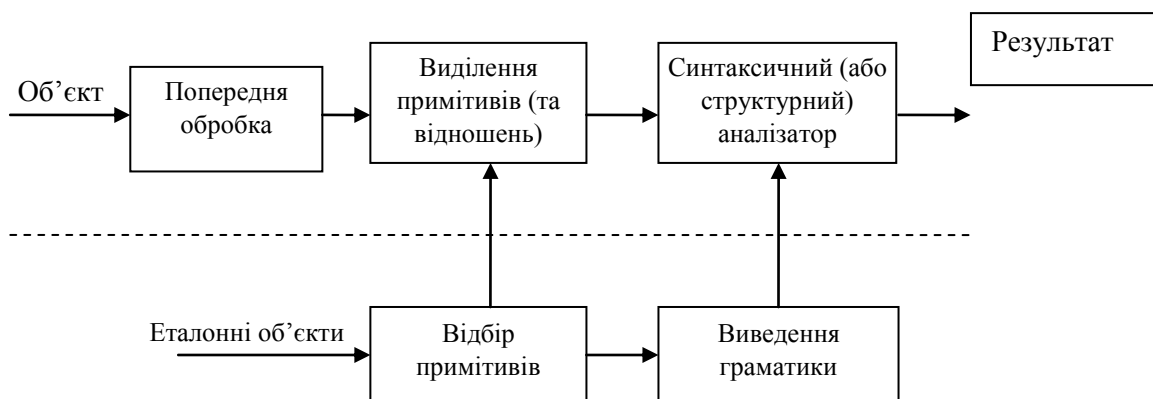


Рис. 1. Функціональна схема лінгвістичної системи розпізнавання образів

Математично ідея методу полягає в наступному. Маємо ряд спостережень деякого процесу, що їх зроблено за якийсь час $\{y(1); y(2); \dots; y(n)\}$,

Змінна $y(i)$, $i = \overline{1, N}$ тут представлена фізичними значеннями відповідного процесу. За обраним критерієм обирається ділянка траєкторії “найближча” до ділянки, яка передуює прогнозованій точці. Надалі оцінюється прогноз за формулою $y(n+1) = y(i+p)$ (1), де

$$I = \min \left\{ \sum_{i=1}^p |y(j+i-1) - y(n-p+i)| \right\} \quad J = 1, 2, \dots, n-p;$$

$$J = \min_i |y(i+j-1) - y(n)| \quad i = I, I+1, \dots, I+p-1.$$

Формалізувати метод можна наступним чином. Нехай ми маємо наступні вектори спостережень $Y_1 = (y_1, y_2, \dots, y_p)^T$; $Y_2 = (y_2, y_3, \dots, y_{p+1})^T$; \dots ; $Y_k = (y_k, y_{k+1}, \dots, y_{k+p-1})^T$; \dots ; $Y_N = (y_{n-p+1}, y_{n-p+2}, \dots, y_n)^T$;

Знаходимо найближчу точку із умови мінімальної відстані

$$Y_k = \arg \min_j d(Y_n, Y_j). \quad (2)$$

Є й інші способи пошуку найближчої точки, наприклад, найбільш поширена метрика – квадрат евклідової відстані

$$d(Y_k, Y_n) = (Y_k - Y_n)^T (Y_k - Y_n).$$

Наведемо модифікацію методу „подібних траєкторій”. Нехай існує ряд спостережень $\{y(1); y(2); \dots; y(n)\}$, а вектори, які ми маємо відшукати, мають вигляд:

$$Y_k = (y_k, y_{k+1}, \dots, y_{k+p-1})^T; \dots; Y_N = (y_{n-p+1}, y_{n-p+2}, \dots, y_n)^T,$$

при цьому

$$y_k = y_{n-p+1} + A; y_{k+1} = y_{n-p+2} + A; y_{k+p-1} = y_n + A,$$

тобто $\overline{Y}_k = \overline{Y}_N + A$, де $A = const$, яка означає що процеси мають різні інтенсивності. Тобто маємо однакові процеси, які відбуваються на різних рівнях значень. Константа A задалегідь невідома.

В цьому випадку ми маємо наступне. Для того щоб не визначати рівень траєкторії, тобто не визначати ці константи, будемо шукати різниці між сусідніми значеннями.

Наприклад, розглянемо дві частини одного ряду спостережень

$\overline{Y}_N =$	8,3	10,1	8,7	7,4	6,7	7,6	8,5	7,4	6,8	7,7	8,6	7,5
$\overline{Y}_k =$	5,9	7,7	6,3	5	4,3	5,2	6,1	5	4,4	5,3	6,2	5,1

Знайдемо різниці між сусідніми значеннями обох векторів:

Як бачимо, обидва вектори мають одну і ту саму траєкторію, але на різних рівнях. Таким чином проблему прогнозування можна розбити на дві частини. Перша – відшукати проміжки траєкторій на яких

$Y_n - Y_{n+1} =$	-1,8	1,4	1,3	0,7	-0,9	-0,9	1,1	0,6	-0,9	-0,9	1,1
$Y_k - Y_{k+1} =$	-1,8	1,4	1,3	0,7	-0,9	-0,9	1,1	0,6	-0,9	-0,9	1,1

повторюються процеси. Друга – за допомогою набраної статистики зробити прогноз розвитку відповідного процесу.

Для пошуку траєкторій де є повторювання на різних рівнях розробимо алгоритм.

Нехай існує ряд спостережень $\{y(1); y(2); \dots; y(n)\}$, де $y(i)$; $i = \overline{1, N}$ – це деякі значення, які отримано в ході спостереження з деяким часовим кроком $\Delta t_i = const \quad i = \overline{1, N}$. Обчислимо перші різниці

$$\Delta y(i) = y(i) - y(i+1), \quad i = \overline{1, N-1}$$

між сусідніми значеннями ряду.

Далі застосуємо структурний підхід [2], який раніше застосовувався в методах розпізнавання образів. Сутність методу полягає в наступному: ранжуємо обчислені різниці, наприклад, за зростанням. Визначаємо певну кількість символів алфавіту $B = \{b_k\}$, де k – довжина алфавіту. Відсортований ряд розбивається на інтервали відповідно до кількості символів алфавіту і кожному з інтервалів призначається свій символ. Тобто замість числового ряду $\Delta y(i)$ отримаємо

символьний аналог ряду спостережень $\langle e_i \rangle$

Далі ми підраховуємо кількість однакових ланцюжків $\langle e_i e_{i+1} \rangle$;

$\langle e_i e_{i+1} e_{i+2} \rangle$; $\langle e_i e_{i+1} e_{i+2} e_{i+3} \rangle_i$ т.д.

символів. Потім ми підраховуємо всі ланцюжки по 2, 3, 4 і т.д. символів. Частота

появлені того чи іншого ланцюжка показує ймовірність повторення тих чи інших послідовностей. Таким чином ми перевіряємо чи є в числовій послідовності подібні траєкторії.

В послідовності e_i проаналізуємо частоту існування пар символів $\langle e_i e_{i+1} \rangle$ і побудуємо таблицю ймовірностей виникнення символу e_{i+1} : $P(e_{i+1} | e_i)$, $i = \overline{1, N-1}$.

Далі в послідовності e_i аналізуємо частоту появи трійок $\langle e_{i-1} e_i e_{i+1} \rangle$ будуюмо таблицю ймовірностей $P(e_{i+1} | e_i, e_{i-1})$, $i = \overline{1, N-2}$. В загальному випадку аналізуємо частоту існування ланцюжків $P(e_{i+1} | e_i \dots e_{i+n})$. Тобто обчислюємо ймовірність появи символу e_{i+1} за умови що попередніми символами є $e_i \dots e_{i+n}$.

Моделювання за допомогою символів і є лінгвістичне моделювання.

Лінгвістичне моделювання – комплекс методів, методик і алгоритмів, які використовують процес перетворення числових масивів інформації до

лінгвістичних послідовностей, на основі яких відновлюється формальна граматики.

Лінгвістична модель динамічного процесу складається з таких елементів:

$$F = (D, I, L, G), \quad (3)$$

де D – сукупність часових рядів динамічного процесу та рядів, похідних від вхідних даних;

I – спосіб та правила інтервалізації;

L – морфізм відображення інтервального представлення ряду на певний алфавіт;

G – відновлена граматики динамічного процесу [3].

За визначеними вище правилами числові значення послідовності замінюються символами. Тобто замість числового ряду $y(i)$; $i = \overline{1, N}$ отримаємо символний аналог ряду спостережень $\langle e_i \rangle$. За допомогою обчислених ймовірностей можемо зробити ймовірнісний прогноз виникнення символу e_i за умови що відомі ланцюжки попередніх символів $e_i, e_{i-1}, \dots, e_{i-k}$. Прогнозування відбувається шляхом пошуку ланцюжків символів і співставлення їх із ланцюжками з бази даних наявних спостережень. Прогноз подається у вигляді множини значень ймовірностей виникнення того чи іншого стану процесу.

Список літератури

1. Фу К. Структурные методы в распознавании образов. – М.: Мир, 1977. - 319с
2. Баклан І.В. Структурний підхід до аналізу та прогнозування поведінки часових рядів / І.В. Баклан, Ю.М. Селін // Вісник Херсонського національного технічного університету. — Херсон: ХНТУ, 2005. — № 2. — С.27–31.
3. Баклан И.В. Лингвистическое моделирование: основы, методы, некоторые прикладные аспекты // Системные технологии. Региональный межвузовский сборник научных работ. – Выпуск 3 (74). – Днепропетровск, 2011. – с.10–19.

ПОЗИЦІЙНИЙ БАГАТОКРИТЕРІАЛЬНИЙ АЛГОРИТМ APS/MES-ПЛАНУВАННЯ ВИРОБНИЦТВА

В статті розглянуто задачу побудови розкладу виробництва з урахуванням багатьох критеріїв. Побудовано математичну модель такої системи і запропоновано евристичний алгоритм вирішення задачі, який дає задовільний результат в умовах промислової експлуатації, що підтверджує ефективність запропонованого рішення.

The article considers the problem of constructing a production schedule based on many criteria. A mathematical model of the system and propose a heuristic algorithm for solving the problem which gives a satisfactory result in terms of industrial exploitation, confirms the effectiveness of the proposed solutions.

Ключові слова: APS/MES, ПЛАНУВАННЯ ВИРОБНИЦТВА, УПРАВЛІННЯ ВИРОБНИЦТВОМ, ОПЕРАТИВНЕ ПЛАНУВАННЯ, РОЗКЛАД, РОБОЧІ ЦЕНТРИ, ТЕХНОЛОГІЧНІ ОПЕРАЦІЇ, ОБМЕЖЕННЯ, ЕВРИСТИЧНІ АЛГОРИТМИ, БАГАТОКРИТЕРІАЛЬНА ОПТИМІЗАЦІЯ

Вступ

Автоматизація системи управління виробництвом дозволяє значно підвищити ефективність підприємства. Система управління в АСУ являє собою замкнутий контур оперативного управління: в якості основного інструменту управління виступає оперативний план (змінне завдання), званий іноді розкладом, а роль зворотного зв'язку грає диспетчеризація виконання оперативного плану.

Постановка задачі APS/MES-планування

APS/MES - це сучасна концепція оперативного планування виконання MPS, MRP планів виробництва оптимальним чином за допомогою оптимізації роботи обладнання. APS/MES-планування полягає у формуванні розкладу виконання технологічних операцій на робочих центрах з точністю до хвилин. Розкладом вважаємо призначення виконання операцій для виготовлення продукції на робочих центрах. Кількість операцій та порядок їх виконання для виготовлення продукції описується технологічним маршрутом виробу. У загальному випадку, горизонтом APS/MES-

планування є поточний тиждень, кожен день план перераховується для підтримки актуальності та гнучкості.

Основною метою APS/MES-планування виробництва на замовлення є побудова такого плану, за яким у першу чергу будуть задовільнені договірні умови у вигляді директивних строків виробництва і поставки продукції замовнику, а у другу – мінімізовані затрати на виробництво. Таким чином, планування має ураховувати множини критеріїв, серед яких найуживанішими є наступні:

- мінімізація зривів директивних строків заказів;
- мінімізація переналагоджень робочих центрів;
- мінімізація пролежування сировини і незавершеного виробництва;
- мінімізація тривалості виконання робіт.

Основною вхідною інформацією для планування є:

- MPS-план - перелік продукції для виконання з директивними строками виконання і пріоритетами;
- MRP-план - перелік операцій і технологічних зв'язків між ними;

- нормативи часу на виконання операцій MRP-плану на допустимих робочих центрах;
- перелік доступних робочих центрів і властивості їх функціонування;
- графіки робіт робочих центрів.

На підставі вхідної інформації проводиться розрахунок ефективної послідовності виконання операцій з урахуванням директивного терміну і пріоритетів виготовлення продукції. Така задача відноситься до задач теорії розкладів (ТР).

Зазвичай пошук кращого розкладу виконання робіт здійснюється за допомогою одного з чотирьох підходів:

- математичне програмування;
- комбінаторний підхід;
- евристичний підхід;
- імовірнісний підхід.

При застосуванні методів математичного програмування для вирішення задач ТР неминуча експоненціальна залежність часу виконання завдання від кількості робочих центрів і операцій.

Зазвичай, поняття «оптимальне рішення задачі», математики, із застереженням на ідеальну модель, формулюють так: не існує інших варіантів розв'язання задачі, які приводять до кращого результату. У свою чергу, економісти стверджують, що рішення задачі завжди обмежено часом і ресурсами, і поняття «оптимальне рішення» замінюють поняттям «краще рішення». Краще рішення - рішення, яке найкраще підходить за обраним критерієм оцінки за весь час роботи алгоритму.

Незадовільний стан розвитку точних методів вирішення задач ТР зумовив розробку наближених методів, що дозволяють отримувати прийнятні рішення при порівняно невеликих витратах часу і коштів. Евристичні алгоритми засновані на прийомі, який називається прийомом зниження вимог. Він полягає у відмові від пошуку оптимального рішення при плануванні в реальному часі. Евристичні алгоритми використовують різні ефективні

прийоми, підтвержені на практиці. Одним з напрямів евристичних методів вирішення задач ТР полягає в формуванні правил або функцій переваги (пріоритетів). Для кожної операції з множини операцій розраховується значення функції переваги і на поточному кроці обирається та операція, для якої ця функція досягає екстремуму.

Математична постановка задачі

Розглянемо математичну постановку задачі побудови розкладу з урахуванням багатьох критеріїв. Нехай $G = (O, M, E)$ – деякий допустимий розклад, де O – це множина технологічних операцій (робіт), M – множина робочих центрів, E – множина обмежень на порядок виконання операцій та допустимість виконання операцій на робочих центрах.

Тоді цільова функція має наступний вигляд:

$$f(G) \rightarrow \min,$$

де функція f є векторною функцією.

У якості часткових критеріїв f_i в залежності від умов виробництва може використовуватися:

- сумарний час запізнення заказів;
- сумарний час переналагоджень робочих центрів;
- сумарний час пролежування сировини і незавершеного виробництва;
- сумарна тривалість розкладу та інші.

Кожному критерію на окремому підприємстві експертним шляхом ставиться у відповідність вага.

В даному алгоритмі використовується поняття *фронту операцій*: перелік операцій, забезпечених вхідними операціями.

Даний алгоритм є жадібним у тому сенсі, що на кожному кроці обирається наступна краща операція з фронту операцій для призначення на робочий центр.

Даний алгоритм називається позиційним, бо розрахунок значень часткових критеріїв і оцінка операцій виконується по кожному критерію окремо у порядку ваги (пріоритету). Такий підхід дозволяє не виконувати для кожної операції розрахунок значень всіх

критеріїв, а отже значно підвищити швидкодію розрахунку.

Алгоритм формування розкладу з урахуванням багатьох критеріїв

Крок 1. Початок

Крок 2. Формування масиву операцій.

Крок 3. З усього масиву отримати фронт операцій.

Крок 4. Обрати операцію для призначення на робочий центр. Для кожної операції з фронту виконати наступне:

а. Розрахувати значення найвагомшого критерію.

б. Отримати множину операцій, у яких значення критерію максимальне.

с. Якщо у множині одна операція – перейти до кроку 5.

д. Інакше виконувати для операцій з цієї множини розрахунок наступних по вазі критеріїв до тих пір, поки не буде обрана одна краща операція.

Крок 5. Призначення виконання операції на РЦ.

Крок 6. Якщо залишилися операції на призначення - перейти до кроку 3.

Крок 7. Зберегти призначення операцій.

Крок 8. Кінець.

Алгоритм призначення виконання операцій на робочі центри

Крок 1. Початок.

Крок 2. Отримати операцію до призначення.

Крок 3. Отримати ранній час початку операції.

Крок 4. Отримати перелік допустимих робочих центрів.

Крок 5. Якщо є хоча б один допустимий робочий центр, перейти до кроку 7, інакше перейти до кроку 6.

Крок 6. Призначити операцію на виконання на віртуальному

робочому центрі «Завод», тривалість виконання вважати 1 хвилиною.

Крок 7. Вважати перший допустимий робочий центр рекордом.

Крок 8. Для кожного робочого центру:

Крок 9. Отримати тривалість виконання операції на робочому центрі.

Крок 10. Отримати вікно робочого центру, в яке можна виконувати операцію.

Крок 11. Виходячи з тривалості вікна і тривалості виконання операції отримати час закінчення операції.

Крок 12. Якщо час закінчення операції менше рекорду - вважати поточний робочий центр рекордом.

Крок 13. Призначити виконання операції на робочий центр-рекорд.

Крок 14. Кінець.

Висновок

Впровадження APS/MES-системи планування виробництва дозволяє значно збільшити оперативність і гнучкість планування в цілому, логічним наслідком якого є підвищення ефективності підприємства в цілому.

У завданнях оперативного планування виробництва дуже важлива швидкість розрахунку допустимого розкладу. У даній статті було розглянуто один з алгоритмів складання розкладу виконання операцій на робочих центрах з урахуванням багатьох критеріїв, який дає досить хороший результат за прийнятний час.

Література

1. Гаврилов Д.А. Управление производством на базе стандарта MRP, 2-е изд. — СПб.: Питер, 2005, 416 с.
2. Уоллас Т., Сталь Р. Планирование продаж и операций: Практическое руководство — СПб.: Питер, 2010, 272 с.

МЕДВЕДСЬКА К.О.,
ЖДАНОВА О.Г.,
СПЕРКАЧ М.О.

ЗАДАЧА ПЛАНУВАННЯ ВИКОНАННЯ ЗАВДАНЬ ЗІ СПІЛЬНИМ ДИРЕКТИВНИМ ТЕРМІНОМ НЕІДЕНТИЧНИМИ ПАРАЛЕЛЬНИМИ ПРИСТРОЯМИ

В даній роботі розглянута задача планування виконання завдань зі спільним директивним терміном неідентичними паралельними пристроями з максимально пізнім моментом запуску. Базуючись на методі пошуку з заборонами (Tabu search), розроблено адаптований алгоритм вирішення задачі. Описаний евристичний метод вибору пристроїв для обміну завданнями, що дозволяє алгоритму поступово виходити з локальних оптимумів та наближатись до глобального або близького до нього.

The article considers the tasks scheduling problem with common deadlines, non-identical parallel machines and as late as possible start. The article offers an adaptive, Tabu search-based algorithm, which solves the problem. The authors describe the heuristic method of choosing machines to exchange tasks, which allows the algorithm to progressively escape from local optima and approach to global optimum or close to it.

Ключові слова: ОПЕРАТИВНО-КАЛЕНДАРНЕ ПЛАНУВАННЯ, ЦЕХОВЕ ПЛАНУВАННЯ, СКЛАДАННЯ РОЗКЛАДУ, ПОШУК З ЗАБОРОНАМИ

1. Вступ

Задача, що розглядається у роботі належить до задач теорії розкладів, методи якої використовуються для оптимізації оперативно-календарних планів. Одною з важливих складових оперативно-календарного планування (ОКП) є цехове планування, в якому вирішуються завдання, де в систему з неідентичними приладами одночасно надходить множина робіт і можуть бути встановлені обмеження на терміни запуску і випуску продукції. Для приладів необхідно скласти розклад виконання завдань, що дозволяє досягнути ефективного використання ресурсів. Отже, є потреба в розробці таких алгоритмів побудови розкладів, які можуть забезпечити високу якість одержуваних результатів і не вимагають значних обчислювальних ресурсів.

2. Постановка задачі

Задано множину завдань $J = \{1, 2, \dots, n\}$ та кількість пристроїв m . Пристрої працюють паралельно і є взаємозамінними у тому сенсі, що кожний з пристроїв може виконувати будь-яке завдання з множини J . Пристрої відрізняються один від одного продуктивністю виконання завдань. При

цьому пристрої можна впорядкувати за швидкістю виконання завдання і цей порядок однаковий для всіх завдань: для кожного пристрою i існує коефіцієнт k_i такий, що тривалість виконання завдання j на пристрої i дорівнює $k_i p_j$.

«Еталонним» будемо називати пристрій з коефіцієнтом продуктивності $k=1$. В цьому сенсі величина p_j є тривалістю виконання завдання j на еталонному пристрої. Передбачається, що всі завдання множини J надходять одночасно та мають спільний жорсткий директивний термін d , процес обслуговування кожного завдання протікає без переривань до завершення обслуговування завдань. Всі пристрої працюють без переривань. Необхідно знайти максимальний момент запуску пристроїв r_{\max} , що дозволяє отримати допустимий розв'язок (розклад, у якому усі завдання не запізнюються).

Ця задача є узагальненням задачі, що розглядається в [1]. Властивості задачі, алгоритм побудови початкового розкладу, ознаки оптимальності детально описані у роботі [2].

Задача зводиться до задачі мінімізації загального часу виконання усіх робіт.

3. Загальні принципи методу пошуку з заборонами

Родоначальником алгоритму пошуку з заборонами (tabu search - TS) (1986 год) являється Фред Гловер, який запропонував принципово нову схему локального пошуку [3, 4, 5]. Вона дозволяє алгоритму не зупинятися в точці локального оптимуму, як це робить стандартний алгоритм локального спуску, а переміщатись від одного локального оптимуму до іншого в надії знайти серед них глобальний.

Ідея алгоритму «Пошуку з заборонами» полягає в тому, щоб не зупинятися в локальному оптимумі, як це роблять алгоритми локального спуску, а продовжувати пошук, керуючись тими ж правилами, забороняючи відвідування вже пройдених точок [3,4,5].

Загальна схема алгоритму пошуку з заборонами:

1. Побудувати допустимий початковий розв'язок $x_0 \in D$ (D – множина допустимих розв'язків), покласти $F^* = F(x_0)$, Сформувати пустий список заборон $Tabu_l(x_0) = \emptyset$.

2. $k = 0$, $x_k = x_0$.

3. **ДОКИ** не виконаний критерій зупинки робити наступне:

3.1 Сформувати окіл $N_p(x_k)$.

3.2 **ЯКЩО** $N_p(x_k) = \emptyset$,

ТО $x_{k+1} = x_k$,

ІНАКШЕ знайти x_{k+1}

такий, що

$F(x_{k+1}) = \min\{F(i) | i \in$

$N_p(x_k) \setminus Tabu_l(x_k)\}$.

3.3 **ЯКЩО** $F^* > F(x_{k+1})$,

ТО $F^* = F(x_{k+1})$.

3.4 Покласти $k = k + 1$ і оновити список заборон $Tabu_l(x_k)$.

КІНЕЦЬ АЛГОРИТМУ

В якості критерію зупинки використовуються або зупинка по числу ітерацій, або необхідна точність по відношенню до заданої нижньої границі. Допустимий початковий розв'язок визначається за допомогою якогось простого алгоритму.

4. Розробка табу-алгоритму розв'язання задачі складання розкладу

для неідентичних паралельних пристроїв

Узагальнимо описаний вище принцип у вигляді алгоритму для вирішення поставленої задачі. Для цього введемо наступні позначення:

σ^0 – початковий розклад;

σ^* – найкращий відшуканий розклад (рекордний розв'язок);

σ – поточний розклад;

k – номер поточної ітерації;

K – максимальна кількість ітерацій;

J – множина всіх завдань;

m – кількість пристроїв;

p_j – тривалість завдання $j \in J$.

Розглянемо деякий допустимий розклад σ .

Позначимо в цьому розкладі:

$Tabu_l(\sigma)$ – табу-список – список станів (розв'язків), перехід до яких заборонений.

Список має довжину l – це максимальна кількість розв'язків, що зберігаються в один момент у списку.

Далі використовуються такі позначення:

$J_i(\sigma)$ – множина завдань, що виконується пристроєм i , $J = \{J_i \cup \dots \cup J_m\}$;

$T_i(\sigma) = \sum_{j \in J_i(\sigma)} k_i p_j$ – тривалість зайнятості пристрою i ;

$\Delta_i(\sigma) = \max\{0; T_i(\sigma) - C^*\} = \max\left\{0; \sum_{j \in J_i(\sigma)} k_i p_j - C^*\right\}$ –

виступ пристрою i ;

$R_i(\sigma) = \max\{0; C^* - T_i(\sigma)\} = \max\left\{0; C^* - \sum_{j \in J_i(\sigma)} k_i p_j\right\}$ –

резерв пристрою i ($i = \overline{1, m}$).

$I_\Delta(\sigma)$ – множина таких пристроїв, для яких $\Delta_i(\sigma) > 0$;

$I_R(\sigma)$ – множина таких пристроїв, для яких $R_i(\sigma) > 0$;

$Z_i(\sigma) = \max\{0; \blacklozenge_i(\sigma) - T_i^*\} = \max\left\{0; \sum_{j \in J_i(\sigma)} k_i p_j - T_i^*\right\}$

(аналог величини $\Delta_i(\sigma)$, але в даному випадку розраховується відхилення від «ідеальної» тривалості зайнятості пристрою T_i^* в оптимальному контурі розкладу);

$E_i(\sigma) = \max\{0; T_i^* - T_i(\sigma)\} = \max\left\{0; T_i^* - \sum_{j \in J_i(\sigma)} k_i p_j\right\}$

(аналог величини $R_i(\sigma)$);

$I_Z(\sigma)$ – множина таких пристроїв, для яких $Z_i(\sigma) > 0$;

$I_E(\sigma)$ – множина таких пристроїв, для яких $E_i(\sigma) > 0$.

$I_0(\sigma)$ – множина таких пристроїв, для яких $Z_i(\sigma) = E_i(\sigma) = 0$.

Для покращення поточного розкладу необхідно направити зусилля на мінімізацію величини

$\max_i\{Z_i(\sigma) + E_i(\sigma)\}$, $i = \overline{1, m}$ – тобто, максимального відхилення від «ідеальної» тривалості зайнятості пристрою T_i^* в оптимальному контурі розкладу (якщо $\delta > 0$, то T_i^* – результат вирішення допоміжної задачі [2]; якщо $\delta = 0$, то $T_i^* = C^*$). Для цього необхідно виконувати обмін завданнями між пристроями.

Вибір пристроїв для обміну завданнями

Пропонується виконувати обмін завданнями між пристроями, що належать множинам $I_Z(\sigma)$, $I_E(\sigma)$ та $I_0(\sigma)$. Для спрощення будемо виконувати обмін одним завданням між донором і реципієнтом (перестановка типу 1-0, 0-1). В загальному випадку більш складний обмін множинами завдань буде реалізований множиною обмінів типу 1-0, 0-1 деякої послідовності ітерацій алгоритму.

Введемо додатково: $f_i(\sigma) = Z_i(\sigma) + E_i(\sigma)$ – значення часткової функції відхилення, характеризує величину відхилення значення критерію по пристрою i від «ідеальної» тривалості зайнятості пристрою T_i^* . Введемо величину $f(\sigma) = \max_i\{f_i(\sigma)\}$, $i = \overline{1, m}$ – максимальне значення функції відхилення для поточного розкладу σ .

Розробимо евристики, за допомогою яких будемо визначати пристрої h та s , між якими потрібно виконати обмін завданням.

1. *Пристроєм донором (h)* будемо називати пристрій, що має максимальне значення функції відхилення $f(\sigma) = \max_i\{f_i(\sigma)\}$, $i = \overline{1, m}$.

2. *Пристроєм реципієнтом (s)* будемо називати будь-який пристрій, окрім h обраний за наступними правилами.

Опишемо правило вибору пристрою-реципієнту s . Нехай:

$T_i(\sigma)$ – час зайнятості пристрою i ;

$\frac{T_i(\sigma)}{k_i}$ – час зайнятості пристрою i в еталонних величинах.

Для усіх пристроїв $i = \overline{1, m}$ розрахуємо значення:

$$\beta_i = \frac{T_h}{k_h} - \frac{T_i}{k_i}. \quad (1)$$

Додамо в алгоритм таку ймовірнісну складову: чим більше значення β_i , тим більше ймовірність обрати i -тий пристрій.

$P_i = \frac{\beta_i}{\sum_{i=1}^m \beta_i}$ – ймовірність вибору пристрою i .

Примітка. Якщо $i = h$, то $\beta_i = 0$ і пристрій h не буде обраний у якості пристрою реципієнту.

Для кожного розкладу σ та визначених пристроях h та s , ми можемо визначити множину сусідніх станів $N_{hs}(\sigma)$. Вона сформована зі станів, до яких ми можемо перейти за один крок перестановками типу (1-0) та (0-1) між пристроями h та s . Для алгоритму пошуку з заборонами значення $N_{hs}(\sigma)$ є околom розкладу σ .

На кожній ітерації алгоритму буде знаходитись найкращий стан $\sigma' \in N_{hs}(\sigma) \setminus Tabu_l(\sigma)$ з множини сусідніх станів (околу), що не належать табу-списку. Найкращим станом будемо вважати такий, для якого виконується:

$$F(\sigma') = \min_{v \in N_{hs}(\sigma) \setminus Tabu_l(\sigma)} \{f(v)\}.$$

Умови завершення

Для заданої задачі маємо декілька умов завершення, при досягненні хоча б однієї з них робота алгоритму завершується:

- перевищено встановлений ліміт кількості ітерацій $k = K$, де $K = const$ значення якої задається на початку роботи алгоритму;
- знайдено розв'язок, що задовольняє достатній умові оптимальності 1 $\sum_{i=0}^m \Delta_i(\sigma) = \sum_{i=0}^m R_i(\sigma) = 0$;
- знайдено розв'язок, що задовольняє достатній умові оптимальності 2 $T'_i(\sigma) = [c_i^*] + x_i^*$, $i = \overline{1, m}$ або $T_i(\sigma) = T_i^*$, $i = \overline{1, m}$, де $T_i^* = C^* + k_i(x_i^* - e_i)$.

На основі розроблених евристик вибору пристроїв та завдань для обміну, описаних правил формування околу поточного розв'язку побудований адаптований алгоритм пошуку з заборонами для розв'язання задачі складання розкладу для неідентичних паралельних машин.

Адаптований алгоритм пошуку з заборонами:

КРОК 0. Визначити оптимальний контур (достатню умову оптимальності).

Задати значення K та l .

КРОК 1. Побудувати початковий розклад σ^0 .

КРОК 2. Розрахувати:

- $\delta = \sum_{j=0}^n p_j - \sum_{i=0}^m |c_i^*|$;
- $f(\sigma^0) = \max_{i=1, \dots, m} \{Z_i(\sigma^0) + E_i \sigma^0\}$.

Покласти $k = 0$, $\sigma_k = \sigma^0$, $F^* = f(\sigma^0)$,
 $\sigma^* = \sigma_k$, $Tabu_l(\sigma_k) = \emptyset$

КРОК 3. Перевірити виконання умов завершення.

КРОК 4. Визначити пристрій-донор h та пристрій-реципієнт s .

КРОК 5. Для поточного розкладу σ_k сформувавати окіл $N_{hs}(\sigma_k)$, як усі можливі розклади, що можуть бути отримані перестановкою типу 1-0 чи 0-1 між пристроями h та s .

КРОК 6. ЯКЩО $N_{hs}(\sigma_k) \setminus Tabu_l(\sigma_k) = \emptyset$
ТО перейти на **КРОК 4**.

КРОК 7. Знайти σ_{k+1} таке, що
 $F(\sigma_{k+1}) = \min_{v \in N_{hs}(\sigma_k) \setminus Tabu_l(\sigma_k)} \{f(v)\}$.

КРОК 8. ЯКЩО $F^* > F(\sigma_{k+1})$
ТО оновити найкращий поточний розв'язок: $F^* = F(\sigma_{k+1})$, $\sigma^* = \sigma_{k+1}$.

КРОК 9. Додати σ_k до списку заборон:
 $Tabu_l(\sigma_{k+1}) = Tabu_l(\sigma_k) \cup \{\sigma_k\}$.

КРОК 10. Збільшити лічильник ітерацій
 $k = k + 1$. Перейти на **КРОК 3**.

КРОК 11. Визначити максимально пізній момент запуску завдань на виконання:
 $r_{max} = r(\sigma_k) = d - \max_{i=1, \dots, m} T_i(\sigma_k)$.

КІНЕЦЬ АЛГОРИТМУ

5. Висновок

Розглянута задача планування виконання завдань зі спільним директивним терміном неідентичними паралельними пристроями з максимально пізнім моментом запуску. Базуючись на методі пошуку з заборонами (Tabu search), розроблено адаптований алгоритм вирішення задачі. Описаний евристичний метод вибору пристроїв для обміну завданнями, що дозволяє алгоритму поступово виходити з локальних оптимумів та наблизитись до глобального або близького до нього.

Для оцінки ефективності розробленого алгоритму була проведена серія експериментів, на підставі яких можна зробити наступні висновки:

- найкращі результати демонструє алгоритм на задачах великої розмірності, що пояснюється більшою можливістю перетасування задач;
- збільшення розсіювання тривалостей завдань скорочує час, необхідний для пошуку розв'язку; це особливо помітно на задачах з короткими завданнями;
- для задач з короткими завданнями усі алгоритм дає результати, близькі до оптимуму.

Список літератури

1. Павлов О.А., Жданова О.Г., Місюра О.Б., Сперкач М.О. Поліноміальна складова ПДС-алгоритму розв'язання однієї задачі теорії розкладів // Технологічний аудит та резерви виробництва.-Х.: 2013.-№6/3 (14). – С.47–51.
2. Сперкач М.О., Жданова О.Г. Задача визначення максимального пізнього моменту початку виконання завдань із спільним жорстким директивним терміном паралельними приладами різної продуктивності / М.О. Сперкач, О.Г. Жданова // Матеріали дев'ятої міжнародної науково-практичної конференції «Математичне та імітаційне моделювання систем МОДС 2014». – м. Чернігів.: ЧДІЕУ, Київ-Жукин, 23-27 червня 2014 р. – с. 108 – 112.
3. Glover F. Tabu search - part I, ORSA Journal on Computing 1(3),pp.190-206, 1989.
4. Glover F. Tabu search - part II, ORSA Journal on Computing 2(1),pp.4-32, 1990.
5. Glover F., Laguna M. Tabu search in modern heuristic techniques for combinatorial problems, Blackwell Publishing, 1992.

ТРИВИМІРНЕ МОДЕЛЮВАННЯ В ЗАДАЧАХ РОЗПІЗНАВАННЯ ОБРАЗІВ

У цій статті розглянуто практичне застосування тривимірного моделювання з метою розпізнавання образів, використовуючи програмний продукт 3ds max. Описується процес створення 3д моделі і основні чинники, що впливають на її якість. Демонструються основні положення, для створення повноцінної моделі.

This article deals with the practical application of three-dimensional modeling for the purpose of pattern recognition using 3ds max software. It describes the process of creating a 3D model and the main factors affecting its quality. Demonstrates basic provisions for the creation of a full-fledged model.

3D-моделювання - це процес створення тривимірної моделі об'єкта. 3D-Завдання моделювання - розробити візуальний об'ємний образ бажаного об'єкта. За допомогою тривимірної графіки можна і створити точну копію конкретного предмета, і розробити нове, навіть нереальне зображення об'єкту.

Тривимірна графіка активно застосовується для створення зображень на площинах екранів або аркушах друкованої продукції в науці і

промисловості, наприклад, в системах автоматизації проектних робіт (САПР, для створення твердотільних елементів: будівель, деталей машин, механізмів), архітектурної візуалізації (сюди відноситься і так звана «віртуальна археологія»), в сучасних системах медичної візуалізації.(див. рис.)

Найширше застосування - в багатьох сучасних комп'ютерних іграх, а також як елемент кінематографа, телебачення, друкованої продукції тощо. [1]

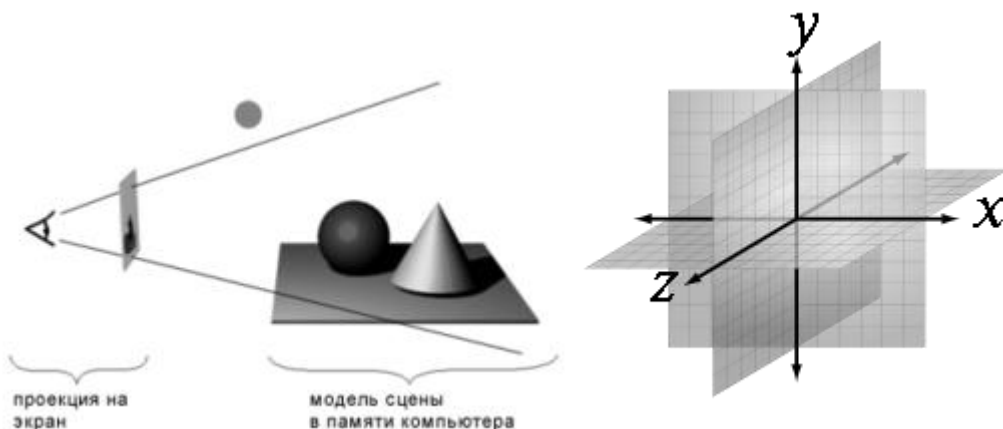


Рис. 1 Загальна схема створення тривимірного зображення

В доповіді пропонується застосовувати 3D-моделювання в задачах розпізнавання образів. Для цього треба мати наявну базу даних вже готових зображень (обличчя людей, зображення тварин, листків рослин, комах тощо). Для максимально точного розпізнавання образу потрібно створити

якісну 3D модель. В першу чергу, це залежить від кількості фотографій або креслень створюваного об'єкта. Чим їх більше, тим детальніше і реалістичніше вийде модель. Основні положення знімків для створення точної моделі це - вид зверху, з боків, спереду і ззаду. Вони

дозволяють створити модель з усіх боків. Використання знімків з різними кутами не має сенсу.

Для такого моделювання пропонується застосовувати пакет 3ds Max. [2]. Autodesk 3ds Max (раніше 3D Studio MAX) - повнофункціональна професійна програмна система для створення і редагування тривимірної графіки і анімації, розроблена компанією Autodesk.

Містить найсучасніші засоби для художників і фахівців в області мультимедіа. Працює в операційних системах Windows, і Windows NT (як в 32, бітних так і в 64) бітних. У березні 2014 випущена версія 17.0 цього продукту під назвою Autodesk 3ds Max 2015. Написана на C # (WPF), також використовує бібліотеку Developer Express (DevExpress).

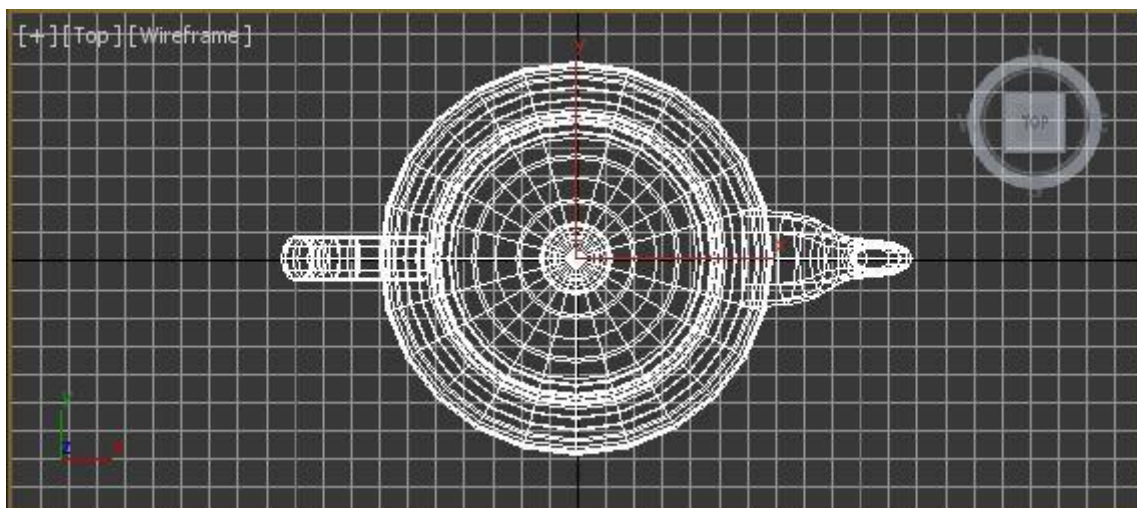


Рис. 2 Вид моделі зверху у пакеті 3ds Max

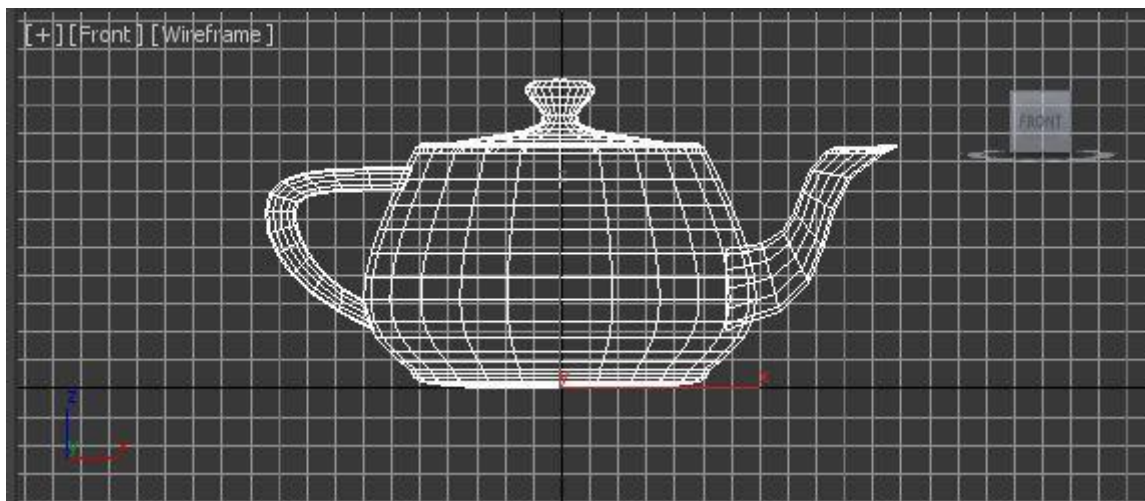


Рис. 3 Вид моделі збоку у пакеті 3ds Max

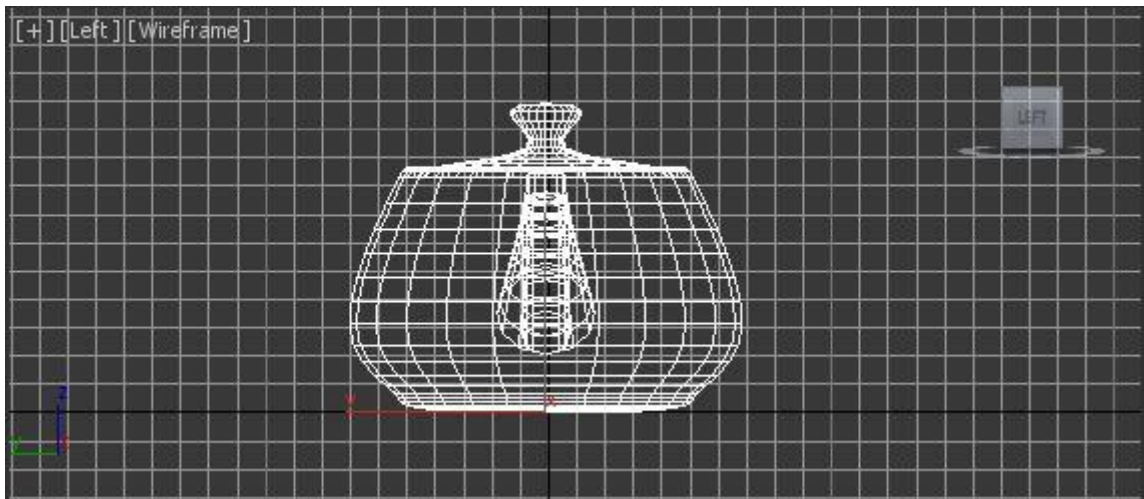


Рис. 4 Вид моделі спереду у пакеті 3ds Max

Список літератури

1. Херн Д. Компьютерная графика и стандарт OpenGL [Текст] /, Дональд Херн, М. Паулин Бейкер; Пер. с англ. 3-е издание - М.: Издательский дом "Вильямс", 2005. - 1168 с. - 5000 экз - ISBN 5-8459-0772-1, 0-13-015390-7
2. Рис С. Анимация персонажей в 3D Studio MAX. [Текст] / Стефани Рис; Пер. с англ. – М.: Издательство BOOKS, 2009. — 450 стр. - 5000 экз. - ISBN 5-88782-268-6, 1-57610-054-5

ОГЛЯД МЕТОДІВ ІДЕНТИФІКАЦІЇ ЛЮДИНИ ЗА РАЙДУЖНОЮ ОБОЛОНКОЮ ОКА Анотація.

У статті представлений огляд сучасних методів ідентифікації і верифікації людини за райдужною оболонкою ока. Спочатку викладаються добре відомі факти про будову ока та його райдужної оболонки. Далі описуються основні принципи роботи біометричних систем і показники якості їх роботи. Наступний розділ описує відомі методи: виділення зіниці, виділення райдужної оболонки, перетворення до ідеалу, визначення меж райдужної оболонки. Потім описуються алгоритми визначення ключових точок на основі вейвлет-перетворень. Зокрема, описується застосування фільтра Габора. Описано методи порівняння з еталоном на основі відстані Хеммінга і методу проєкційної фазової кореляції. Обговорюються переваги і недоліки описаних підходів.

Annotation

An overview of modern methods of identification and verification of human iris. Initially taught by well-known facts about the structure of the eye and its iris. The following describes the basic principles of biometric systems and the quality of their work. The following sections describe the known methods, the selection of the pupil, iris allocation, conversion to the ideal, defining the boundaries of the iris. Then described algorithms identify key points based on wavelet transformation. In particular, the application describes Gabor filter. Methods for comparison with a standard based on the Hamming distance and projection phase correlation method. We discuss the advantages and disadvantages of this approach

Вступ

У сучасному світі надзвичайно гостро стоїть проблема захисту інформації. На сьогоднішній день використання пральних системи ідентифікації вже не задовольняє вимогам безпеки. Щоб забезпечити достатній рівень безпеки, пароль повинен бути складним. Складність пароля забезпечується спільним використанням літер (як малих, так і великих), цифр і знаків і його довжиною. Причому для кожного інформаційного ресурсу рекомендується створювати свій пароль. Дуже часто це призводить до ому, що користувач просто забуває такий пароль, і для його відновлення необхідно ідентифікувати людину, яка звернулася в службу підтримки. Також паролі системи ідентифікації ніяк не захищені від використання паролів третіми особами для несанкціонованого доступу. Щоб задовольнити всі зростаючі потреби в підвищенні рівня безпеки інформації, все частіше для ідентифікації особистості

використовуються методи біометрії. При біометричній аутентифікації використовуються унікальні характеристики окремо взятої людини. Це можуть бути як вроджені ознаки (відбитки пальців, райдужна оболонка ока), так і придбані характеристики (почерк, голос або хода). У даній роботі розглядаються методи ідентифікації людини за райдужною оболонкою ока. Такий вибір не випадковий. Ідентифікація по райдужній оболонці ока є одним з найбільш точних і надійних способів біометричної ідентифікації. Це пов'язано з тим, що райдужна оболонка ока має особливу структуру, яка унікальна для кожної людини. При цьому методи ідентифікації по райдужній оболонці ока є безконтактними. Ще однією перевагою подібних методів є те, що носіння контактних лінз, навіть кольорових, не є проблемою. У процесі ідентифікації ніяк не враховується інформація про колір очей. Це робить застосування подібних систем

ідентифікації та верифікації ще більш привабливим.

Біометрія райдужної оболонки ока

Біометричний контроль доступу - автоматизований метод, за допомогою якого шляхом перевірки (дослідження) унікальних фізіологічних особливостей або поведінкових характеристик людини здійснюється ідентифікація особи. Фізіологічні особливості, наприклад, такі як папілярний візерунок пальця, геометрія долоні або малюнок (модель) веселкової оболонки ока, є постійними фізичними характеристиками людини. Даний тип вимірювань (перевірки) практично незмінний також як і самі фізіологічні характеристики. Поведінкові ж характеристики, такі як підпис, голос або клавіатурний почерк, знаходяться під впливом, як керованих дій, так і менш керованих психологічних факторів. Оскільки поведінкові характеристики можуть змінюватися з плином часу зареєстрований біометричний зразок повинен оновлюватися при кожному його використанні. Хоча біометрія, заснована на поведінкових характеристиках, менш дорога і представляє меншу загрозу для користувачів, фізіологічні риси дозволяють здійснити велику точність ідентифікації особистості і її безпеку. У будь-якому випадку, обидва методи забезпечують значно вищий рівень ідентифікації, ніж самі по собі паролі або карти. На відміну від пароля або персонального ідентифікаційного номера (ПІН), біометрична характеристика не може бути забута, втрачена, або вкрадена. Оскільки біометричні характеристики кожної окремої особистості унікальні, вони можуть використовуватися для запобігання крадіжки або шахрайства. Сьогодні існує більш ніж 10,000 комп'ютеризованих місць, сховищ, дослідницьких лабораторій, банків крові, банкоматів, військових споруд, доступ до яких контролюється пристроями, які канують унікальні фізіологічні або поведінкові характеристики індивідуума. Біометричні системи ідентифікації, доступні в даний час або знаходяться в стадії розробки

включають в себе системи доступу по відбитку пальця, аромату, ДНК, формі вуха, геометрії особи, температурі шкіри обличчя, клавіатурного почерку, відбитком долоні, сітківці ока, малюнку райдужної оболонки ока, підписи і голосу. У цій статті зупинимося на технології розпізнавання по райдужній оболонки ока. Визнання унікальності малюнка райдужної оболонки ока кожної людини дозволяє реалізувати ще одну, досить точну, технологію ідентифікації. Алгоритми, що використовуються в цій технології настільки точні і компактні, що дозволили б зареєструвати жителів всієї планети в одній базі даних з дуже пристойними значеннями FAR і FRR.

Райдужна оболонка

Дана технологія розпізнавання заснована на відмінностях в малюнку райдужної оболонки ока у видимому (або інфрачервоному) світлі. Основа для ідентифікації - так звана trabecular meshwork, це тканина (остаточно формується к 8-ому місяця вагітності), яка робить видимими поділу райдужної оболонки на радіальні сектора. Інші видимі характеристики включають кільця, борозни, веснянки, і область "корони". Попросту кажучи, в процесі розпізнавання відбувається сканування та перерахунок видимих характеристик райдужної оболонки в 512 байтовий код (IrisCode), вираз для якого потім використовується в процесі ідентифікації користувача. 512 байт - це не найкомпактніший шаблон серед біометричних методів ідентифікації, але кількість інформації, що отримується в цьому методі справді величезне. З 11 міліметрового діаметра райдужної оболонки алгоритми Даугмана дозволяють отримати в середньому 3,4 біта інформації на 1 квадратний міліметр площі. Щільність витягується інформації така, що можна говорити, що райдужна оболонка має 266 унікальних "точок" ідентифікації в порівнянні з 10-60 точками для інших біометричних методів. Після обліку кореляційних функцій алгоритму і реальних характеристик, властивих більшості

людських очей, доктор Даугман робить висновок, що 173 "незалежні двійкові ступеня свободи" (точки ідентифікації) можуть бути вилучені за допомогою його

алгоритму, що становить виключно велику кількість інформації для біометричних методів.

Список літератури:

1. Баклан І.В. Лінгвістичне моделювання: основи, методи, деякі прикладні аспекти//Системні технології. Регіональний міжвузівський збірник научних робіт. – Випуск3(74) – Дніпропетровськ, 2011. – с.10 – 19.
2. Баклан І.В. Інтервальний підхід до побудови лінгвістичної моделі// Системні технології. Регіональний міжвузівський збірник научних робіт. – Випуск3(86) – Дніпропетровськ, 2013. – с.3 – 8.
3. Ушмаев О. С. Адаптация биометрической системы к искажающим факторам на примере дактилоскопической идентификации // Информатика и её применение, 2009. Т. 3. Вып. 2. С. 25–33.
4. Ushmaev O., Novikov S. Biometric fusion: Robust approach // 2nd Workshop on Multimodal User Authentication — MMUA'06 Proceedings. Toulouse, France, 11–12 May 2006. <http://mmua.cs.uchb.edu/MMUA2006/Papers/127.pdf>.

ОРГАНІЗАЦІЯ КОЛЕКТИВНОЇ РОБОТИ З УНІВЕРСАЛЬНИМ ПРОГРАМНО-АПАРАТНИМ КОМПЛЕКСОМ ДЛЯ КОМП'ЮТЕРИЗАЦІЇ НАВЧАЛЬНОГО І НАУКОВОГО ЕКСПЕРИМЕНТУ

Розглянутий підхід до організації колективної роботи з універсальним програмно-апаратним комплексом для комп'ютеризації навчального і наукового експерименту в режимі реального часу.

The considered approach to teamwork with universal hardware and software system for computerizing the educational and scientific experiment in real time.

Постановка проблеми.

Рівень фундаментальних досліджень та прикладних розробок у галузі фізики суттєво залежить від забезпечення наукових установ сучасним обладнанням і спеціалізованими комплексами, зокрема, для проведення наукових експериментів. Суттєве значення при цьому має рівень застосування в процесі досліджень інформаційно-комунікаційних технологій (ІКТ) [1].

Використання передових технологій при проведенні фізичних експериментів має широке розповсюдження і в навчальному процесі. Зокрема, засобом комп'ютерно-орієнтованого процесу навчання є віртуальні фізичні лабораторії, які надають можливість для створення віртуальних моделей різних фізичних систем [2].

Тому актуальною є задача розробки сучасних ІКТ для комп'ютеризації навчального і наукового фізичного експерименту.

Універсальний програмно-апаратний комплекс для комп'ютеризації навчального і наукового експерименту.

Навчальний експеримент у закладі освіти є основою вивчення багатьох природничих дисциплін. Зокрема, якість знань і практична підготовка учнів з фізики перебувають у прямій залежності від можливості якісного проведення фізичного експерименту.

Спеціалізовані системи, призначені для використання в навчальному експерименті,

розробляються багатьма іноземними фірмами. Всі вони будуються за принципом *сенсор – перетворювач - комп'ютер - користувач* і передбачають індивідуальну роботу студента або учня з конкретною експериментальною установкою. Основні їх недоліки [3, 4, 5]:

- пристрої є персональними і не підтримують розподілену клієнт-серверну технологію;
- не мають методико-інформаційної складової;
- вимагають досить дорогу апаратну частину, особливо Собга4;
- існуючі системи практично не масштабуються;
- не можливо оновити дорогу апаратну частину;
- програмне забезпечення не мультиплатформене;
- мають закриті протоколи передачі даних.

Інститутом прикладної фізики НАН України запропоновано новий підхід в побудові таких систем, який передбачає орієнтацію не на одного користувача, а на колективну роботу з експериментальною установкою. З використанням цього підходу розроблений вітчизняний універсальний програмно-апаратний комплекс (УПАК), який покриває велику кількість недоліків існуючих аналогів [3].

УПАК складається з набору датчиків, модуля реєстрації і передачі даних, мережевого комутатора (опціонально),

пристроїв візуалізації та аналізу даних (у загальному випадку – у вигляді комп'ютера і спеціального програмного забезпечення). Він є програмно-апаратним комплексом широкого призначення, що може входити до складу будь-якого наукового і навчального обладнання.

Але залишається один недолік - відсутність мобільного додатку, який дозволить кожному користувачу оброблювати дані одного і того ж досліджу з використанням власного мобільного пристрою. Тому було вирішено створити клієнт-серверний мобільний додаток, за допомогою якого можна підключитися до УПАК, проводити індивідуальні дослідження, їх аналіз та зберігати потрібні результати експерименту.

Підхід до організації колективної роботи з УПАК.

Принципова схема використання розробленого мобільного додатку з УПАК представлена на рис.1

Мобільний пристрій (телефон або планшет) підключається до комутатора за допомогою WiFi та запитує дані, що були отримані в результаті експерименту. Далі пропонуються завдання, результати яких необхідно внести користувачеві до таблиці, за якою побудуються певні графіки, на основі яких потрібно буде зробити висновки.

Зараз додаток знаходиться в стадії тестування на платформі iOS. Планується розробка під інші мобільні операційні системи.

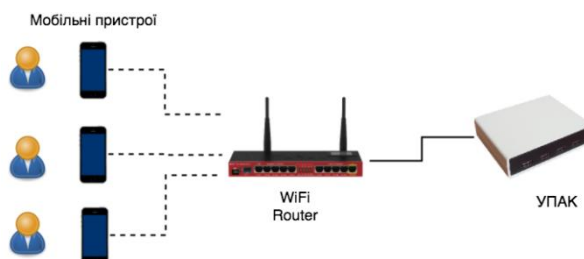


Рис. 1 Принципова схема використання УПАК

Заключення

Запропоновано підхід до організації колективної роботи з універсальним програмно-апаратним комплексом для комп'ютеризації навчального і наукового експерименту за рахунок розробки мобільного додатку, який дозволяє декільком користувачам оброблювати дані

одного і того ж експерименту в режимі реального часу.

Розроблено мобільний додаток, який надає користувачу можливість виконувати збір даних і керувати пристроями дистанційно, що забезпечується доступом до експериментальної установки через мережу Internet.

Список літератури

1. Програмно-апаратний комплекс для комп'ютеризації навчального і наукового експерименту Лопаткін Р.Ю., Купрієнко В.В., Пелепей Р.Л., Іващенко В.А., Ігнатенко С.М. // Патент на корисну модель № 58345 від 11 квітня 2011 року.
2. Комплексное программно-аппаратное решение для общеобразовательной школы [Електронний ресурс] / Ю. В. Федорова. – Режим доступу : \WWW/ URL: <http://www.int-edu.ru/page.php?id=886> – 10.12.2015 р. – Назва з титул. екрана..
3. PHYWE - Cobra4 Wireless. [Електронний ресурс] – Режим доступу : \WWW/ URL: <https://www.phywe.de/en/12605-89> – 10.12.2015 р. – Назва з титул. екрана..
4. FourierSystemsInc. [Електронний ресурс] – Режим доступу : \WWW/ URL: <http://fourier-sys.com/> – 10.12.2015 р. – Назва з титул. екрана..
5. PASCO : Home. [Електронний ресурс] – Режим доступу : \WWW/ URL: <http://pasco.com/> – 10.12.2015 р. – Назва з титул. екрана

СПЕЦІАЛІЗОВАНЕ ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ ЗБЕРЕЖЕННЯ ТА ВІДОБРАЖЕННЯ РАДІОЛОКАЦІЙНОЇ ІНФОРМАЦІЇ

Розглянута реалізація спеціалізованого програмного забезпечення для обробки радіолокаційної інформації.

Considered the implementation of specialized software for processing radar information

Постановка проблеми.

Аналіз конфлікту на сході України показує зростаючу роль безпілотних літальних апаратів (БПЛА) у вирішенні завдань повітряної розвідки та підтримання ведення бойових дій. Радіолокаційні станції (РЛС) розвідки повітряних рухомих цілей, які є на озброєнні ЗСУ, за своїми тактико-технічними характеристиками не повною мірою відповідають сучасним викликам в основному через обмежені можливості, обумовлені недосконалою системою спостереження та ідентифікації цілей. Основними недоліками таких систем є те, що обробка даних у них здійснюється на апаратному рівні, і в автоматизованому, а не автоматичному режимі.

У цих умовах актуальною є задача удосконалення системи спостереження та ідентифікації цілей РЛС, яка повинна забезпечити автоматичне отримання достовірної інформації про ціль у реальному масштабі часу.

Апаратно-програмний комплекс для обробки радіолокаційної інформації.

Апаратно-програмний комплекс (АПК) для обробки радіолокаційної інформації призначений для автоматичного знімання інформації від РЛС і її обробки в режимі реального часу.

До його складу входять РЛС, пункти прийому та обробки, один з яких управляє роботою комплексу та пеленгаторною мережею передачі даних.

Спеціалізоване програмне забезпечення (СПЗ) розробленого АПК

реалізоване у вигляді сукупності наступних підсистем:

- підсистеми управління та обробки даних РЛС;
- підсистеми виявлення дії сторонніх радіосигналів і їх обробки;
- підсистеми відображення отриманої інформації на екран;
- підсистеми збереження даних у базі даних.

Підсистема управління та обробки даних РЛС передбачає застосування імпульсно-доплеровського метода виявлення місця положення повітряної рухомої цілі, величини та напрямку її швидкості.

Підсистема виявлення дії сторонніх радіосигналів і їх обробки визначає також протокол передачі даних, аналізує самі дані та ін.

Підсистема відображення отриманої інформації - клієнтський додаток, що взаємодіє із першими двома підсистемами по захищеному каналу зв'язку та відображає отриману інформацію на екрані.

Інформація підсистеми відображення представляється наступним чином: дані РЛС відображаються у вигляді точок із позначками швидкості у полярній координатній площі, дані пеленгатора - відображаються на круговій діаграмі у вигляді пелюстків, довжина та напрям яких залежить від потужності та напрямку отриманих сигналів.

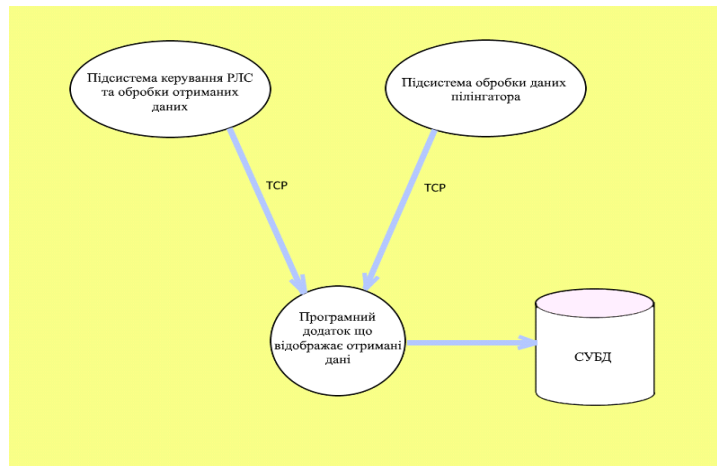


Рис. 1

Клієнтський додаток підключається по TCP-протоколу до підсистем управління РЛС та пеленгатора, які перетворюють отримані аналогові сигнали у цифрові, здійснюють їх обробку та по існуючим вже підключенням відправляють оброблені дані у клієнтський додаток. Архітектура розробленого СПЗ представлена на рис. 1.

Заключення

Розроблене СПЗ забезпечує програмну обробку радіолокаційної інформації, що підвищує точність та швидкість отримання результатів.

Запропоноване СПЗ дає змогу точно визначати місцезнаходження БПЛА та інших повітряних об'єктів, а також локацію станції, що керує ними.

Список літератури

1. Радиолокационный комплекс РЛС 1Л220У «Зоопарк-2» [Электронный ресурс]. – Режим доступа: <http://eizvestia.com/armiya/full/746-radiolokacionnyjkompleks-rls-1l220u-zoopark-2>.
2. Подразделения БПЛА. Электронный ресурс: <http://www.waronline.org/IDF/Articles/armament/uavs/units/>

*БЕЛОЧУБ Н.М.
ГОЛОВЧЕНКО М.Н.
НАНОВСКИЙ С.А.
НЕЧАЙ Д.А.*

КРОССПЛАТФОРМЕННЫЙ ТОНКИЙ КЛИЕНТ CONSOLE В ЭКСПЕРИМЕНТАЛЬНОМ ТЕХНОЛОГИЧЕСКОМ СТЕКЕ METARHIA

Тонкий клиент Console предназначен для повышения унификации и повторного использования графических интерфейсов пользователя в мобильных приложениях, веб-приложениях и клиент-серверных приложениях для экспериментального стека технологий Metarhia. Console построен на формальном синтаксисе, который отображается на каждой из платформ в компоненты пользовательского интерфейса, реализованные различными способами, но обернутые в унифицированную реализацию библиотеки графических компонентов и предоставляющую одинаковые программные интерфейсы, свойства и события для разных сред запуска. Компонентный подход, совмещенный с реактивным принципом связывания данных и компонентов, широким применением метаданных и декларативных структур для формального описания интерфейсов позволяет существенно упростить разработку UI и унифицировать работу над кроссплатформенными приложениями, снизить влияние человеческого фактора и сгладить разницу между интерфейсами на разных платформах.

Console UI renderer is designed to increase unification and reuse of graphical user interfaces in mobile applications, web-applications and client-server applications for experimental technological stack Metarhia. Console is based on formal syntax, which is natively rendered on each platform into graphical user interface components, with all kinds of realization but wrapped in the same realization of graphical user interface kit providing the same program interfaces, properties and events for different runtime environment. Component based approach combined with reactive principle of linking data and components, wide usage of metadata and declaratory structures for formal description of interfaces, allows to significantly simplify graphical user interfaces development and unify work on cross-platform applications, to decrease influence of human factor and to smooth difference between graphical user interfaces on different runtime environment.

Ключові слова: ТОНКИЙ КЛИЕНТ, ГРАФИЧЕСКИЙ ИНТЕРФЕЙС ПОЛЬЗОВАТЕЛЯ, КРОССПЛАТФОРМЕННОСТЬ, МЕТАДААННЫЕ, МЕТАПРОГРАММИРОВАНИЕ

1. Проблема

Существующий подход к разработке графических интерфейсов для клиент-серверных приложений имеет достаточно много недостатков, таких, как большое количество альтернативных и конкурирующих технологий, которые необходимо интегрировать и увязать в одной информационной системе. Для создания интерфейса приложения, которое будет работать на большинстве пользовательских платформ, понадобятся, как минимум, 4 специалиста (рис. 1). Но это всё равно не решает проблему поддержки нескольких приложений, поскольку мы получим не одно приложение с четырьмя

интерфейсами, а четыре разных приложения, и в будущем придется тратить ресурсы на их обновление. Также из-за особенностей существующих технологий нет гарантии, что эти приложения будут иметь приемственность, и пользователь сможет их использовать без изменения своих навыков. Еще одним недостатком является то, что сервер БД, сервер приложений и клиент разрабатываются отдельно, и их интеграция нередко становится более сложной задачей, чем их разработка. При необходимости внесения изменений, приходится модифицировать структуру БД, API сервера приложений и клиентские приложения. А при

эксплуатации и поддержке, делать синхронные модификации во всех компонентах системы, на разных языках и платформах, тестировать их и повторно внедрять, мигрировать данные, что приводит паузам в предоставлении сервиса.

Применение HTML в современном вебе далеко от его первоначальной задачи. Язык гипертекстовой разметки был создан в 1990 году в CERN. Он предусматривал удобное

создание текстовых документов (заголовки, параграфы, списки, таблицы, цитаты и т.д.) без их привязки к средствам отображения (текст с разметкой HTML должен был без проблем воспроизводиться на разном оборудовании). Позже эта идея была принесена в жертву в связи с необходимостью графического оформления веб-страниц.

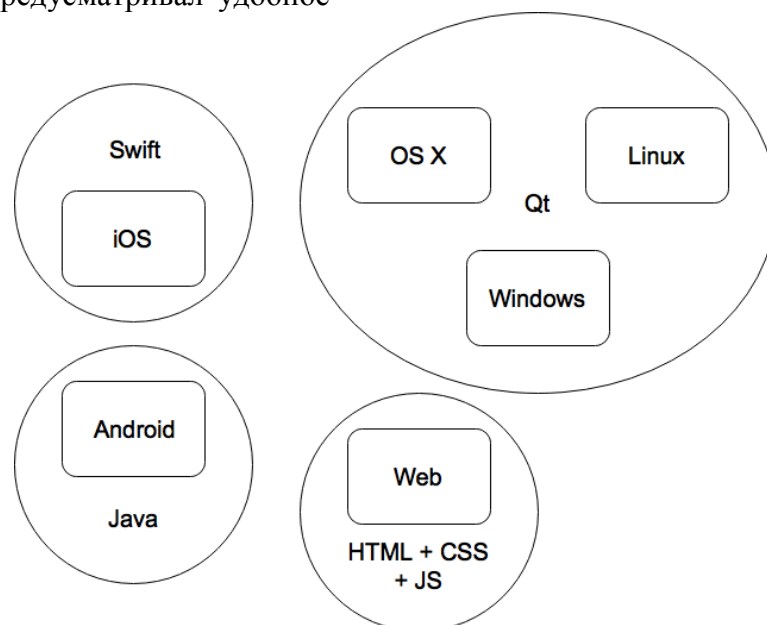


Рис. 1 - Традиционный стек технологий для построения UI

2. Поиск решения

Тонкий клиент Console [1] создается как экспериментальная платформа для решения перечисленных проблем в сфере кроссплатформенных пользовательских интерфейсов. Набор экранов разрабатывается один раз и, с помощью существующих рендереров, может быть динамически собрано для разных платформ. В результате, для создания пользовательской части приложения необходим лишь специалист по разработке на платформе Console (рис. 2). Благодаря поддержке стилей, графические интерфейсы пользователя, созданные с помощью Console, будут выглядеть одинаково на всех платформах, и пользователю будет удобно использовать их на разных устройствах. Описание приложения в виде метамodelей

позволяет создать не только графический интерфейс пользователя, но и серверную часть со структурой БД и API интерфейсами. В результате внесения изменений требует лишь изменение декларативного описания в специализированном формальном синтаксисе [2], который является подмножеством JavaScript, самого распространенного из современных языков. До 80-90% интерфейса можно получить из описания метамодели при помощи транслятора, после чего можно внести изменения и добавить недостающие параметры, события и методы, быстро получить формы и CRUD, навигацию по пользовательским экранам.

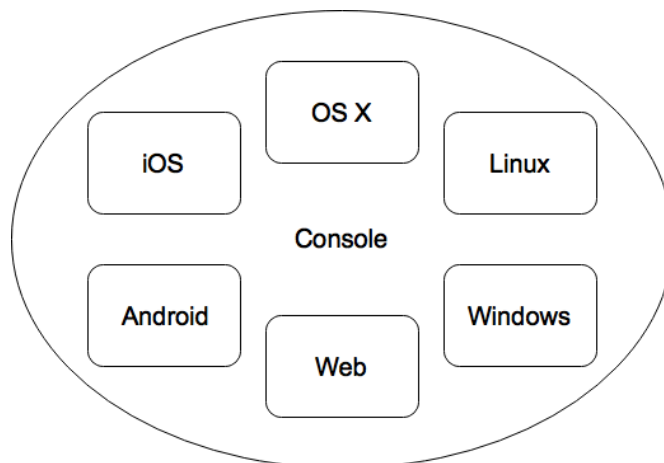


Рис.2 - Цель проекта Console

На данный момент разрабатывается реализация тонкого клиента Console для рендеринга интерфейсов в стеке технологий Metarhia на следующих ОС и платформах:

- Qt (C++) с поддержкой Linux, OS X, Windows и др.
- Node.js (C++, Qt, JavaScript) с поддержкой Linux, OS X, Windows
- iOS (Swift)
- Android (Java)
- архаичного веба, гипертекстовых браузеров (HTML, CSS, JavaScript)

3. Преимущества компонентного подхода

Современный веб прошел эволюцию от статического представления текста до сложных интерактивных пользовательских интерфейсов. За этот период были опробованы разные подходы к созданию веб-приложений. Самым перспективным оказался компонентный подход, в сторону которого движется HTML5.

Используя компонентный подход мы “инкапсулируем” реализацию элемента пользовательского интерфейса и получаем возможность повторного использования созданных ранее элементов экранов и форм. В тонком клиенте Console, мы получаем возможность унифицированно согласовать реализации компонентов на разных платформах.

4. Основные принципы построения UI в Console

Для создания пользовательских интерфейсов используются метаданные и декларативные структуры, основанные на формальном синтаксисе. Существует 2 специализированных подмножества языка описания предметной области: Metamodel

Definition Language, для метамodelей и GUI Definition Language, используемый для интерфейса, в котором задаётся список компонентов и их стили оформления. Между этими языками мы можем осуществлять автоматическую трансляцию, в том числе динамическую, без остановки функционирования системы. На любом этапе, автоматическая трансляция может быть дополнена или заменена кодом в декларативном стиле, включающем отдельные фрагменты в функциональном, реактивном, событийном и объектно-ориентированном стилях. Синтаксисы полностью совместимы с языком JavaScript и являются его подмножеством, а так же являются подмножеством синтаксисов сериализации объектов (Object Serialization) протокола JSTP (JavaScript Transfer Protocol) и могут быть порождены из структур данных JSTP Record Data при наличии JSTP Record Metadata.

В тонком клиенте Console используется компонентный подход и реактивный принцип связывания данных и компонентов. Это подразумевает, что изменение данных в одном из клиентов влечет за собой реакцию на это во всех остальных, при условии, что они являются взаимосвязанными (подписаны на события).

Стандарты оформления пользовательского интерфейса могут значительно отличаться для разных платформ. Так, например, Android-приложения должны соответствовать стандарту Material Design от Google [3], а приложения для iOS – стандартам от Apple [4]. Поэтому в Console подразумевается, что, описывая один раз метамодель

приложения, разработчик получает интерфейс, который незначительно отличается на разных платформах, и при этом соответствует всем рекомендациям по проектированию.

5. Заключение

Предложенный метод внедряется в платформу интерактивного телевидения SinceTV и применяется в новом Кампусе КПИ, где уже показал свою высокую эффективность как в быстром действии пользовательских интерфейсов, так и в скорости разработки. Применение формальных синтаксисов позволяет более точно ставить технические задания, проводить тестирование UI/UX и динамически изменять внешний вид приложения без замены версии, что

критически важно в интерактивных системах и системах приближенных к реальному времени, таких, как телевизионные. Вместе с сопутствующими технологиями стека Metarhia и использованием метапрограммирования, тонкий клиент Console позволяет унифицированный процесс разработки программного обеспечения с высокой степенью автоматизации, повысить повторное использование кода, снизить монотонность труда разработчиков, а благодаря повышению уровня абстракции, мы существенно расширяем сферу применения информационных разрабатываемых систем.

Список литературы

1. Шемсединов. Т.Г. Console [Электронный ресурс]. – Режим доступа: <https://github.com/metarhia/Console> (дата обращения 15.04.2016).
2. Шемсединов. Т.Г. JSTP / JavaScript Transfer Protocol [Электронный ресурс]. – Режим доступа: <https://github.com/metarhia/JSTP> (дата обращения 15.04.2016).
3. Material Design for Android [Электронный ресурс]. – Режим доступа: <http://developer.android.com/design/material/index.html> (дата обращения 15.04.2016).
4. iOS Human Interface Guidelines: Designing for iOS [Электронный ресурс]. – Режим доступа: <https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/MobileHIG/index.html> (дата обращения 15.04.2016).

ТИМОШИН Ю. А.,
 ОРЛЕНКО А. А.,
 КИРИЧЕК Н. И.,
 РУЖЕВСКИЙ Н. С.

КОНЦЕПЦИЯ И ОСОБЕННОСТИ РАСПРЕДЕЛЕННОЙ СУБД GLOBALSTORAGE

Экспериментальный стек технологий Metarhia, включающий протокол JSTP, тонкий клиент Console и сервер приложений Impress Application Server, имеет свою гибридную СУБД GlobalStorage, построенную на принципах распределенного хранения (распределенного чтения и вставки), минимизации межпроцессового взаимодействия, глубокой совместимости с сетевым протоколом JSTP, хранения данных на диске и памяти в формате, близком протоколу, максимального использования памяти, широкого использования метаданных и специального декларативного формата описания предметной области, включая сущности, связи и атрибуты, из которого можно автоматически построить схему хранения. Вместе с сопутствующими технологиями GlobalStorage образует полную экосистему для разработки информационных систем и прикладного программного обеспечения широкой сферы применения, позволяющих заменить как веб-технологии, так и более старые трехзвенные архитектуры.

Experimental Metarhia technological stack, which includes JSTP protocol, Console thin client and Impress Application Server, has its own hybrid database management system called GlobalStorage built on principles of distributed storage (distributed reading and inserting), minimizing inter-process communication, deep compatibility with the network protocol JSTP, storing data on disk and memory in format that is similar to the protocol, maximizing memory usage, wide usage of metadata and specialized declarative format of domain specification, including entities, links and attributes, that can be used for automatic schema generation. Together with adjacent technologies, it forms a full ecosystem for information systems and general-purpose applied software development capable of replacing both web technologies and older three-tier architectures.

Ключевые слова: СУБД, МЕТАДААННЫЕ, МЕТАМОДЕЛЬ, METARHIA, GLOBALSTORAGE, DBMS, RDBMS, SQL, NOSQL, NEWSQL, JAVASCRIPT, NODEJS, JSTP, METADATA, METAMODEL

1. Введение и постановка задачи

Гибридная распределённая СУБД GlobalStorage необходима для реализации экспериментального стека технологий Metarhia, как специализированная СУБД, обладающая рядом ключевых характеристик, позволяющих обслуживать такие характеристики Metarhia, как интерактивность, версионность, гетерогенность, распределенность и динамическую модификацию структур данных во время жизненного цикла приложений в соответствии с метамоделью предметной области на базе интерпретации метаданных. Средой запуска для стека технологий Metarhia является Node.js, который использует асинхронный неблокирующий ввод-вывод для максимально эффективной работы сервера приложений и отсутствия простоя из-за ожидания ввода-вывода, что увеличивает его

производительность и, соответственно, скорость работы при диспетчирезации большого количества запросов, однако отсутствие ввода-вывода ещё быстрее, чем асинхронный ввод-вывод. Поэтому GlobalStorage максимально использует оперативную память, упреждающее чтение, отложенную запись и минимизирует преобразования данных. Также СУБД встраивается в процесс сервера приложений Impress Application Server для того, чтобы избежать лишнего межпроцессового взаимодействия (аналогично Mnesia в экосистеме Erlang).

Ключевыми требованиями к GlobalStorage стали: сквозной формат представления структур данных в разных компонентах системы, минимизация преобразования данных

и межпроцессового обмена, бессхемное хранение, скаффолдинг и построение применение метаданных, принцип бездисковых СУБД для работы с данными в оперативной памяти, дополненное ленивым сохранением данных для постоянного хранения. Само постоянное хранение возможно как на базе файловых систем, так и в реляционных, объектных, бессхемных и документных СУБД. В GlobalStorage синхронизация данных между клиентом и сервером проводится как в реальном режиме времени, так и в отложенном режиме, т.е. имеется возможность работать в онлайн (интерактивно вместе с другими пользователями), так и в оффлайне с данными, сохраненными в локальном хранилище, и иметь двухстороннюю синхронизацию данных с версионностью и ветвлением. GlobalStorage предоставляет абстракцию слоя доступа к данным, т.е. заменяет ORM системы, давая на клиенте и на сервере одинаковый API доступа к данным.

Метаданные — специальный декларативный формат описания предметной области, включающий сущности, связи и атрибуты, из которого можно автоматически построить схему хранения в реляционной СУБД, структуры памяти и структуры для бессхемных СУБД, различные GUI, серверное API и т.д. Для упрощения интеграции между системами и глобальной унификации структур данных в рамках всех систем, работающих на стеке технологий Metarhia, т.е. GlobalStorage, Impress, JSTP и Console, часто используемые сущности стандартизированы и их метаданные хранятся в распределённых модерируемых репозиториях. Однако имеется и возможность работать с неунифицированными данными, специфическими для конкретных систем.

2. Обзор существующих аналогов

Riak — это распределённая NoSQL СУБД типа «ключ-значение», которая отличается высокой надёжностью и доступностью и широкими средствами масштабирования на большое количество узлов и датацентры, географически отдалённые друг от друга. Riak, среди прочих, используется такими компаниями, как AT&T, Comcast и GitHub.

К основным возможностям Riak относятся репликация данных в кластере из нескольких узлов и обмен данными между ними в режиме peer-to-peer, возможность масштабирования не только в рамках датацентра, но и на

географически отдалённые друг от друга сервера и использование механизма векторных часов для причинно-следственного упорядочивания изменений и разрешения конфликтов, два режима работы (с отложенной синхронизацией и приближенный к реальному времени), API, основанное на протоколах HTTP и Protocol Buffers, составные запросы, полнотекстовый поиск, нативная поддержка MapReduce для JavaScript и Erlang хранение данных в оперативной памяти или на диске, использование Bitcask либо LevelDB в качестве бекенда для персистентного хранения.

К недостаткам Riak можно отнести то, что не все перечисленные возможности доступны в open-source версии данной СУБД (в частности, отсутствуют средства масштабирования на несколько датацентров).

CouchDB — это бессхемная распределённая документо-ориентированная СУБД с открытым исходным кодом, использующая JSON для хранения данных и JavaScript для реализации MapReduce-запросов.

Основные особенности: поддержка репликации с несколькими ведущими узлами, использование механизма MVCC (Multi-Version Concurrency Control), позволяющего одновременно хранить несколько версий документа до их слияния прикладным кодом при возникновении конфликта, что позволяет избежать необходимости блокировки файла базы данных во время записи, API для доступа к СУБД на основе архитектуры REST и протокола HTTP, наличие веб-сервера и возможность использовать СУБД как сервер приложений, возможность репликации данных на устройствах (таких как смартфоны) и работы приложения оффлайн.

Недостатки: HTTP, будучи довольно низкопроизводительным протоколом, является единственным способом взаимодействия с СУБД, даже в рамках одного физического сервера; CouchDB предназначена для прямого общения клиента с сервером СУБД, однако пользователи имеют доступ ко всем документам в базе данных, если же необходима возможность установки прав на отдельные документы, то единственным решением с использованием только средств CouchDB является создание отдельных баз данных для каждого пользователя и репликация документов между ними.

RethinkDB также является распределённой безсхемной NoSQL СУБД с открытым исходным кодом, хранящей данные в формате JSON. Её отличительной особенностью является возможность клиентскому приложению подписываться на изменения данных и ориентация на разработку реактивных приложений.

Язык запросов ReQL реализован не на уровне сервера, а встраивается в клиентское приложение и любой запрос в конечном итоге транслируется в цепочку вызовов функций.

Недостатком RethinkDB является более низкая по сравнению с другими СУБД скорость работы [5].

SQLite является простой и компактной встраиваемой реляционной СУБД, которая используется во многих приложениях (в частности, Mozilla Firefox, Skype, Android). Сама SQLite представляет собой библиотеку без сторонних зависимостей, с которой компонуется приложение, а вся база данных хранится в единственном файле. Также есть возможность создавать базы данных в оперативной памяти. Достоинствами данной СУБД являются простота использования, компактность и отсутствие накладных расходов на межпроцессное взаимодействие между приложением и СУБД, а недостатками — невозможность масштабирования и непригодность для использования в многопользовательских клиент-серверных системах из-за блокирования базы данных при обращении к ней.

VoltDB — это реляционная NewSQL СУБД, хранящая данные в оперативной памяти.

Под NewSQL понимают класс современных реляционных СУБД, которые совмещают в себе преимущества NoSQL (гибкость, распределённость, масштабируемость) и транзакционные требования классических СУБД.

VoltDB поддерживает выполнение запросов на языке SQL из-под скомпилированных хранимых процедур Java (SQL/JRT) и именно хранимые процедуры являются единицами транзакций. Постоянность хранения данных обеспечивается регулярными сохранениями снимков базы данных на диск. Начиная с версии 5.1, поддерживается репликация баз данных.

К недостаткам данной СУБД относится неполная поддержка SQL, отсутствие

ограничителей CHECK и FOREIGN KEY, отсутствие поддержки AUTO_INCREMENT и т.д.

Redis — это нереляционная высокопроизводительная СУБД типа «ключ-значение» с открытым исходным кодом.

Особенности: хранение базы данных в оперативной памяти, за счёт чего достигается высокая скорость работы, и наличие механизмов снимков и журналирования для постоянного хранения на диске; механизм обмена сообщениями (publisher/subscriber), при помощи которого приложения могут создавать каналы, подписываться на них и отправлять сообщения всем подписчикам; поддержка репликации данных с основного узла на подчинённые для оптимизации чтения; поддержка транзакций и пакетной обработки команд; поддержка языка Lua для выполнения запросов на сервере; Redis Sentinel как средство мониторинга узлов, уведомления о сбоях, поставки конфигурации и автоматического переключения ведущего узла при сбое.

К недостаткам Redis можно отнести большое потребление памяти (помимо самих данных, которые всегда целиком хранятся в оперативной памяти, также дополнительная память расходуется при каждом сохранении снимка базы данных), отсутствие возможности отката транзакций, немасштабируемый однопоточный режим работы Redis и необходимость реализовывать развёртывание кластера серверов СУБД и шардинг данных на стороне приложения. Также для кого-то может быть недостатком отсутствие официальной поддержки Windows.

MongoDB — документоориентированная NoSQL СУБД с открытым исходным кодом.

Особенности: JavaScript как язык запросов, профилирование запросов, поиск и обновление как одна атомарная операция, эффективное хранение двоичных данных больших объёмов, таких как фото и видео, журналирование операций, отказоустойчивость и масштабируемость, асинхронная репликация в режиме «ведущий — ведомый», основанная на передаче журнала изменений с ведущего узла на ведомые, полнотекстовый поиск с учётом морфологии на многих языках.

Недостатком MongoDB является отсутствие поддержки транзакций.

Mnesia — это распределённая СУБД реального времени с открытым исходным

кодом, написанная на языке Erlang и являющаяся частью Open Telecom Platform. Данная СУБД является гибридной, поддерживающей как реляционный, так и нереляционный подходы, но её реляционная модель значительно отличается от других реляционных СУБД. Записи в таблицах состоят из ключа и значения, которое может быть структурой данных любой сложности. Отношения между таблицами описываются при помощи других таблиц.

Mnesia тесно интегрируется в инфраструктуру Erlang/OTP, использует Erlang как язык запросов и при использовании для построения приложения других Erlang-решений (таких, как веб-сервер Yaws) всё приложение может целиком работать в рамках адресного пространства одной виртуальной машины Erlang.

Mnesia хорошо подходит, когда нужен крайне быстрый доступ к данным в Erlang-приложении, однако не имеет достаточного количества преимуществ для других платформ и потребляет больше ресурсов [8], чем классические реляционные СУБД, при работе с реляционными данными.

3. Особенности СУБД GlobalStorage

Гибридная распределённая СУБД GlobalStorage совмещает многие идеи из рассмотренных выше существующих СУБД, добавляя также и свои уникальные возможности. GlobalStorage характеризуется тесной интеграцией и глубокой совместимостью с сетевым протоколом JSTP и его форматами данных. JSTP используется для синхронизации данных между клиентом и сервером, между узлами на серверной стороне, для получения метаданных из глобального репозитория. Данные на диске и в памяти также хранятся в формате, близком протоколу. В отличие от документо-ориентированных NoSQL СУБД, таких как MongoDB, формат передачи данных по протоколу и формат хранения могут не содержать идентификаторов полей, что существенно уменьшает их размер; значения полей хранятся в массивах, которые можно поставить в соответствие полям модели (схеме/метаданным) позиционным сопоставлением.

Одной из основных особенностей GlobalStorage является предоставление одинакового, унифицированного API при

использовании разных движков для хранения данных. На сервере может использоваться как собственный движок, построенный на основе файловой системы, так и различные реляционные и нереляционные СУБД в качестве хранилища данных. Для клиентских приложений также реализованы различные движки хранения для разных платформ. GlobalStorage также предоставляет абстракцию слоя доступа к данным, т.е. заменяет ORM системы, но не обязательно делает маппинг на реляционную модель, хоть такая возможность тоже встроена.

GlobalStorage позволяет нескольким серверам объединяться для распределенного хранения, а нескольким клиентам обмениваться данными в режиме peer-to-peer. Данные имеют сквозную идентификацию во всей системе, т.е. вставка данных может быть распределенной и не приведет к конфликтам идентификаторов. Достоверность данных обеспечивается их распределенным хранением, версионностью и подписыванием версий хеш-значениями.

Синхронизация данных между клиентом и сервером происходит как в реальном режиме времени, так и в отложенном режиме, т.е. имеется возможность работать как в онлайн-режиме (интерактивно вместе с другими пользователями), так и в оффлайн-режиме с данными, сохраненными в локальном хранилище, и иметь двухстороннюю синхронизацию данных с версионностью и ветвлением (аналогично CouchDB).

4. Заключение

СУБД GlobalStorage предназначена для унификации доступа к данным из разных платформ и языков через единый API как на сервере, так и на клиент, объединяя в себе множество разных характеристик, которые встречаются в различных СУБД, но не доступны в комплексе. Кроме того, GlobalStorage применяется в разработке нового Кампуса КПИ. Вместе с родственными технологиями стека Metarhia (протоколом JavaScript Transfer Protocol и тонким клиентом Console) и схемами описания структуры БД, GlobalStorage позволяет существенно упростить процесс разработки прикладного программного обеспечения и решать задачи, ранее не доступные пользователям универсальных СУБД и требующие создания специализированных СУБД под задачу.

Список литературы

1. Mithun Satheesh; Bruno Joseph D'mello; Jason Krol. Web Development with MongoDB and NodeJS - Second Edition. — Packt Publishing, 2015. — 300 с. — ISBN 978-1-78528-745-9.
2. Anderson, J. Chris; Slater, Noah & Lehnardt, Jan (November 15, 2009), CouchDB: The Definitive Guide (1st ed.), O'Reilly Media, сс. 300, ISBN 0596158165
3. Lennon, Joe (December 15, 2009), Beginning CouchDB (1st ed.), Apress, сс. 300, ISBN 1430272376
4. Redis DB Documentation [Электронный ресурс]. Режим доступа: <http://redis.io/documentation>
5. A Comparison of MongoDB and RethinkDB with Patent Data // Juristat [Электронный ресурс]. Режим доступа: <https://juristat.com/a-comparison-of-mongodb-and-rethinkdb-with-patent-data/>
6. Представлена новая открытая СУБД VoltDB [Электронный ресурс]. Режим доступа: <https://habrahabr.ru/post/94922/>
7. Чезарини Ф., Томпсон С. 13. Начала работы с Mnesia // Программирование в Erlang = Erlang Programming. — М.: «ДМК Пресс», 2012. — 488 с. — ISBN 978-5-94074-617-1.
8. Performance Mnesia vs. MySQL and even row numbers after export/import to MySQL // ejabberd [Электронный ресурс]. Режим доступа: <https://www.ejabberd.im/node/3332>

*МУХА И. П.
ДЗЮБА В. В.
БЕРНАТОВИЧ А. А.
ГАЛАСЮК В. И.*

ПРОТОКОЛ JSTP В ЭКСПЕРИМЕНТАЛЬНОМ ТЕХНОЛОГИЧЕСКОМ СТЕКЕ METARHIA

Протоколы ветхого веба не могут быть эффективно использованы для построения современных распределенных информационных систем с интерактивной составляющей и двусторонним взаимодействием. Изменить их кардинально нельзя из-за того, что они были изначально плохо спроектированы, их назначение существенно отличалось от требуемого сейчас: обратной совместимости и необходимости поддерживать большую базу наследуемого кода. В таких условиях лучшим решением может быть кардинальный пересмотр технологического стека и создание альтернативных экспериментальных прототипов, включая реализацию на основных операционных системах и языках программирования. Первым шагом к новому стеку технологий Metarhia может стать протокол взаимодействия JSTP, который основан на широко распространенном синтаксисе структур данных JavaScript, понимаем любым разработчиком, требует минимальных усилий в реализации парсера, сериализации и десериализации на любых платформах, минимизирует преобразования данных при хранении, обработке и передаче между системами, поддерживает несколько видов взаимодействия (вызовы, события, потоки данных, синхронизация состояния) и может быть легко использован в качестве альтернативного для уже существующих приложений, т.е. имеет простую и понятную схему миграции с других транспортов.

Decrepit web protocols can not be used effectively for building modern distribution information systems with interactive component and two-way interaction. They can not be changed cardinally because they have been developed initially bad, their old purposes were definitely different from nowadays purposes: backward compatibility and necessity of supporting big amount of legacy code. In these conditions the best solution can be cardinal reviewing of technical stack and creating alternative experimental prototypes, including realization on general operation systems and programming languages. First step to new technology stack Metarhia can be interaction protocol JSTP which is based on widely-known data structure syntax of JavaScript which is understandable by any developer, requires minimal efforts for parser realization, serialization and deserialization on any platforms, minimizes data transformation on storage, processing and transferring between systems, supporting few interaction kinds (calls, events, data streams, state synchronizing) and can be easily used as alternative to existing applications, has simple and clear migration structure from other transports.

Ключевые слова: JSTP, JAVASCRIPT, NODE.JS, NETWORK PROTOCOLS, DATA FORMATS, METADATA, СЕТЕВЫЕ ПРОТОКОЛЫ, ФОРМАТЫ ДАННЫХ, МЕТАДАННЫЕ, METAMODEL,

1. Причины

Изначально протокол HTTP создавался во времена web-приложений, которые во основном были красивой оболочкой над базой данных либо просто набором приятных статических web-страниц, связанных гиперссылками. В таких условиях: когда запросы от одного клиента производятся относительно редко — технология REST кажется весьма полезной. Но для разработки более интересной логики RESTа не хватает настолько же, насколько HTTP не подходит для разработки высоконагруженного

соединения из-за избыточности информации в запросе.

С развитием JavaScript стали популярны такие форматы запросов, как JSONP, JSON. Их преимуществом является близость к синтаксису самого языка программирования. В JavaScript JSON является валидным объектом, а JSONP по сути возвращает исполняемый код, что обеспечивает невероятную гибкость. К тому же теперь с появлением Node.js мы имеем возможность во всем web-приложении работать с одним языком программирования.

Протокол JSTP объединяет эти преимущества, добавляет возможность использования метаданных и реактивного программирования.

2. Концепция

Это семейство форматов данных и соответствующих библиотек для работы с ними, которые основаны на нескольких простых допущениях:

- передавать данные в виде JavaScript кода можно красивее и удобнее, чем в формате сериализации JavaScript объектов JSON;
- передавать структуру вместе с данными каждый раз — это избыточно и нужно оптимизировать формат сериализации и протокол, выделив метамодель и передавать ее только если получающая система еще не имеет закешированной версии метамодели;
- протокол взаимодействия между двумя JavaScript приложениями должен обладать такими характеристиками:
- необходимо минимизировать преобразование данных при передаче между системами, хранении и обработке, минимизировать переключивание из одних структур в другие, экономить память и канал связи;
- количество структур данных, необходимых для работы большинства систем является конечным, а сами структуры должны стать фактическими стандартами в результате консенсуса специалистов с возможностью их версионного изменения;
- нестандартизированные структуры данных могут передаваться между системами, снабженные метаданными, которые позволяют их интерпретировать и до известной степени обеспечить универсальную обработку, если удаленные стороны доверяют друг другу, а формализация данных не имеет смысла;

3. Структура семейства форматов

Record Serialization:

```
{
  name: 'Marcus Aurelius',
  passport: 'AE127095'
}
```

Object Serialization:

```
{
  name: ['Marcus', 'Aurelius'].join(' '),
  passport: 'AE' + '127095'
}
```

Record Metadata:

```
{
  name: 'string',
  passport: 'string?'
}
```

Record Data:

```
[ 'Marcus Aurelius', 'AE127095' ]
```

JavaScript Transfer Protocol:

```
{
  event: [17, 'accounts'],
  insert: ['Marcus Aurelius', 'AE127095']
}
```

4. Record Serialization

Это просто JavaScript описывающий структуру данных. В отличие от JSON в нем не нужно помещать ключи в двойные кавычки, можно вставлять комментарии, гибко форматировать и все остальное, что можно в обычном JavaScript. Например:

```
{
  name: 'Marcus Aurelius',
  birth: {
    date: '1990-02-15',
    place: 'Rome'
  },
  contacts: {
    email: 'marcus@aurelius.it',
    phone: '+380505551234',
    address: {
      country: 'Ukraine',
      city: 'Kiev',
      zip: '03056',
      street: 'Pobedy',
      building: '37',
      floor: '1',
      room: '158'
    }
  }
}
```

5. Object Serialization

Если немного усложнить парсер, переключивая все ключи, которые он экспортирует к нему же в песочницу, то можно будет использовать выражения, вызовы функции и определения функций:

```
api.jstp.parse = function(s) {
  var sandbox = vm.createContext({});
  var js = vm.createScript('(' + s + ')');
  var exported = js.runInNewContext(sandbox);
  for (var key in exported) {
    sandbox[key] = exported[key];
  }
  return exported;
};
```

Пример данных:


```

{
  name: ['Marcus', 'Aurelius'].join(' '),
  passport: 'AE' + '127095',
  birth: {
    date: new Date('1990-02-15'),
    place: 'Rome'
  },
  age: function() {
    var difference = new Date() - birth.date;
    return Math.floor(difference /
31536000000);
  }
}

```

Из примера видно, в функциях можно использовать ссылки на поля структуры, например: `birth.date`.

6. Record Metadata

Это метаданные, т.е. данные о структуре и типах данных, описанные в том же формате JavaScript объектов. Определения полей описываются при помощи специального синтаксиса. Например: `number(4)` это число, имеющее не более 4 разрядов и поле не может принимать `undefined`, а `number(2,4)?` это число от 2 до 4 разрядов или `undefined`.

Имена типов начинаются с маленькой буквы: `string`, `number`, `boolean`, а ссылки на другие записи начинаются с большой: `Birth`, `Address`. Все описания записей хранятся в специальном хранилище структур и могут кешироваться на серверах и пользовательских устройствах.

7. Record Data

Это чистые данные, из которых удалены все имена полей, а хеши заменены на массивы. Если поле не имеет значения, т.е. `undefined`, то значение в массиве просто пропущено. Например: `[1,,4]` - это 4 поля, первое и последнее имеют значения 1 и 4 соответственно, а второе и третье равны `undefined`.

Пример экземпляра записи `Person`:
`['Marcus Aurelius','AE127095'],['1990-02-15','Rome'],['Ukraine','Kiev','03056','Pobedy','37','158']`

Если мы имеем данные и соответствующие метаданные, то можем развернуть из них полный документ. Например:

```

var data = ['Marcus Aurelius','AE127095'];
var metadata = { name: 'string', passport:
'[string(8)]' };
var person = api.jstp.decode(data, metadata);

```

```

console.dir(person);
{ name: 'Marcus Aurelius', passport:
'AE127095' }

```

8. JavaScript Transfer Protocol

JSTP это протокол передачи данных, использующий в качестве формата кодирования данных синтаксис JavaScript объектов и поддерживающий метаданные. Протокол имеет 6 типов пакетов:

- `call` - вызов метода удаленного API
- `callback` - ответ удаленного API
- `event` - событие с прикрепленными к нему данными
- `state` - синхронизация данных
- `stream` - передача потока данных
- `handshake` - рукопожатие
- `health` - служебные данные о состоянии и использовании ресурсов
- предполагается расширение типов пакетов

```

// Номер пакета 17, вызов, имя интерфейса
auth, метод newAccount
{call:[17,'auth'],newAccount:['Payload data']}

```

```

// Ответ на пакет 17, результат done,
идентификатор записи 15703
{callback:[17],ok:[15703]}

```

```

// Событие в пакете 18, интерфейс auth, имя
события insert
{event:[18,'auth'],insert:['Marcus
Aurelius','AE127095']}

```

Структура пакета:

- пакет это объект, с несколькими ключами;
- первый ключ - это заголовок, имя ключа - тип пакета;
- второй ключ - идентификатор:

9. Заключение

Предложенный протокол предназначен для принципиально новой концепции интерактивного интернета, например, внедряется в платформу интерактивного телевидения `SinceTV`, и в другие классы систем: корпоративные ИС, системы высокочастотной торговли и социальной торговли, многопользовательские игры, интерактивные АРМы для групповой работы, например, новый Кампус КПИ. Вместе с сопутствующими технологиями стека `Metarhia` и методикой моделирования предметной области, а так же сервером приложений `Impress Application Server`, тонким клиентом `Console`, позволяет

унифицировать межпроцессовые взаимодействия, трансляцию событий, клиент-серверные и пиринговые взаимодействия. JSTP позволяет оптимизировать использование сетевых ресурсов, повысить связанность клиента и сервера таким образом, что коммуникация становится прозрачной для прикладного

кода, т.е. вызовы и события ретранслируются, а прикладные и программные компоненты не нужно модифицировать при масштабировании. JSTP специально спроектирован для масштабирования интерактивности с линейным наращиванием серверных мощностей.

Список литературы

1. The JSON Data Interchange Format [Электронный ресурс]. – Режим доступа: <http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf> (дата обращения 15.04.2016).
2. Node.js v5.10.1 Documentation. Executing JavaScript [Электронный ресурс]. – Режим доступа: <https://nodejs.org/api/vm.html> (дата обращения 16.04.2016).
3. EcmaScript 2015 language specification [Электронный ресурс]. – Режим доступа: <http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-262.pdf> (дата обращения 16.04.2016).
4. Hypertext Transfer Protocol -- Http/1.1 [Электронный ресурс]. – Режим доступа: <https://www.w3.org/Protocols/rfc2616/rfc2616.html> (дата обращения 18.04.2016).

*ВЫСОЦКИЙ А.С.
ИВАНОВА Л.А.
ЧЁРНЕНЬКИЙ А.Ю.
ШЕМСЕДИНОВ Т.Г.*

SINCETV: ПЛАТФОРМА ДЛЯ ИНТЕРАКТИВНЫХ ПРИЛОЖЕНИЙ, РЕАЛИЗУЮЩИХ КОНЦЕПЦИЮ ВТОРОГО ЭКРАНА, ПРИБЛИЖЕННЫХ К РЕАЛЬНОМУ ВРЕМЕНИ

В данной статье рассмотрена платформа SinceTV для интерактивных приложений, реализующих концепцию второго экрана, приближенных к реальному времени. Описана концепция второго экрана, средства, с помощью которых работает указанная платформа, особенности её работы, а также преимущества, которые она предоставляет.

The subject of the article is the SinceTV platform developed for creation of interactive applications which implements the second screen concept. The second screen concept, facilities, features and advantages of the platform are described.

Ключевые слова: ИНТЕРАКТИВНОЕ ТЕЛЕВИДЕНИЕ, ВТОРОЙ ЭКРАН, SINCETV, ПРЯМОЕ TCP-СОЕДИНЕНИЕ, JSTP, WEBSOCKET, SSE, INTERACTIVE TELEVISION, SECOND SCREEN CONCEPT, DIRECT TCP-CONNECTION

1. Введение

В данной статье рассмотрена платформа SinceTV для интерактивных приложений реализующих концепцию второго экрана, приближенных к реальному времени. Описана концепция второго экрана, средства, с помощью которых работает указанная платформа, особенности её работы, а также преимущества, которые она предоставляет.

2. Интерактивное взаимодействие с телезрителем (переименовать)

В момент своего широкого распространения телевидение представляло собой средство массовой информации с информативной и развлекательной составляющими. В процессе развития последней, в эфире стремительно увеличивалось количество телешоу, в которых приглашали гостей в студию, а телеведущие взаимодействовали со зрителями. На протяжении своего существования СМИ коммерциализировались, и сейчас телевидение представляет большой интерес как средство популяризации для различных компаний, в частности с помощью телепередач. Несмотря на то, что данный вид СМИ нес информативную и развлекательную

функции ещё более 50 лет назад, концепция построения телепередач практически не изменилась, хотя возможности коммуникационных сетей общего назначения и мобильных устройств связи уже давно позволяют сделать каждого телезрителя активным участником эфира. Такой подход значительно поднял бы рейтинги существующих телепередач и был бы выгодным для их инвесторов, но все еще не существует программных платформ, позволяющих объединить многомиллионную аудиторию в консолидированном интерактивном взаимодействии.

3. Концепция “второго экрана”

Данная концепция подразумевает интерактивное взаимодействие с происходящим на экране телевизора, которое реализуется с помощью приложений для различных устройств зрителя. В зависимости от происходящего на экране, пользователю таких программ приходят разного рода оповещения. Посредством реакции на такие оповещения и осуществляется общение телезрителя со студией: он может получать и сохранять различную информацию, участвовать в голосованиях, опросах,

взаимодействовать с телеведущими и гостями студии, к примеру, соревноваться наравне с ними в какой-либо телевизионной игре.

4. Платформа SinceTV

На сегодняшний день, платформа SinceTV является одной из наиболее серьезных разработок для приложений второго экрана, приближенных к реальному времени, которая поддерживает одновременное взаимодействие более чем 10 миллионов зрителей с синхронизацией событий порядка 30-90 миллисекунд и имеет возможность линейно масштабироваться при росте нагрузки. Платформа также предоставляет конструктор сценариев, с помощью которого телевизионные шоу различного характера могут быть реализованы за достаточно небольшое время.

Тестирования высоконагруженного кластера SinceTV показывает высокую надежность, эффективность использования вычислительных ресурсов и множество других, точно измеримых преимуществ перед немногими существующими на рынке аналогами.

5. Особенности реализации платформы SinceTV

В отличие от существующих аналогов, платформа SinceTV имеет следующие особенности:

1. **Интерактивность,** приближенная к реальному времени. Приложения на платформе SinceTV имеют скорость доставки уведомлений около 30-90 миллисекунд. Это достигается с помощью постоянно установленного во время прямого эфира TCP-соединения. Такой подход принципиально отличается от реализации через AJAX или отправки push-уведомлений. Push-уведомления могут приходиться с точностью от 5 до 15 минут, задерживаться или даже теряться. Уведомления, получаемые через AJAX предполагают задержки 2-3 секунды, а через SSE или WebSocket примерно от 60 миллисекунд до 2 секунд. Такое сообщение энергетически затратно для мобильных устройств, но используя протокол JSTP, имеется возможность

оптимизировать эти затраты. Для браузеров же используется SSE и WebSocket,

2. **Конструктор форматов и язык GUI и сценариев.** Платформа предоставляет собственный синтаксис, который позволяет полностью описать все активности и правила без низкоуровневого программирования, что повышает скорость создания новых форматов и экранов. Все программные компоненты строятся динамически, например, есть возможность заменить игровые экраны за несколько минут до шоу без переустановки приложения.

3. **Высокие нагрузки с линейным масштабированием.** Это достигается благодаря балансировке на стороне клиента. Список серверов и портов мы отдаем клиентам с CDN, что позволяет делать это распределенно, отдавать разные списки для разных стран, не нагружать свой сервера.

4. **Большая экономия на серверах и возможный демпинг в гонке интерактивности.** Благодаря асинхронной обработке и высокой производительности Node.js и сервера приложений Impress, платформа задействует для каждого 2 млн пользователей сервер класса MG-128 (Xeon E5 на OVH).

5. **Движок игровой динамики,** который берет на вход правила игры, поток событий игры, пересчитывает баллы игроков, рейтинг и статусы. Это предоставляет возможность создавать сложные интерактивные игровые шоу с большим количеством участников и большой интенсивностью игровых событий. Пересчет игровой динамики может происходить распределенно, т.е. дублироваться на серверах и на клиентах, чтобы сократить пересылку данных и задержку, связанную с этим.

6. **Автоматизация развертывания.** В платформе используется автоматическое тестирование и сборка приложений, автообновление ПО на серверах кластера SinceTV на базе drone.io, docker, npm и git.

7. Логирование HTTP и JSTP запросов, подключения и отключения пользователей, системных событий сервера приложений, действий игрока/зрителя, команды администратора эфира, платежей.

8. Интеграция с другими системами. Возможна интеграция при помощи эмбединга, интеграция при помощи API (HTTP, JSTP), интеграция с TV через XML API, генерация хромакей экранов.

6. Преимущества, которые предоставляет платформа

Платформа SinceTV предоставляет следующие преимущества:

1. Поддержка прямого эфира и программ в записи.

2. Интеграция с социальными сетями. На данном этапе реализована функциональность:

а. Вход через соцсети с возможностью одновременного подключения нескольких аккаунтов разных социальных сетей.

б. Получение информации о пользователе, которую он соглашается предоставить.

Публикации на страницах социальных сетей пользователя

1. Интеграция с системами распознавания аудио-контента ACR, позволяющая распознать канал и программу, которую смотрит зритель в данное время и синхронизировать клиент с сервером.

2. Интеграция с платежными системами. На данном этапе реализована работа с платёжными системами PayPal и LiqPay.

3. Интеграция с рекламными сетями. Предоставляется возможность брать рекламу из наиболее крупных сетей (Google, Apple) и отображать ее в мобильных приложениях и веб-интерфейсе. Владельцем рекламного места может быть как канал, так платформа или отдельное шоу.

4. Интеграция с системами сбора статистики с помощью Google Analytics.

5. Управление экраном пользователя.

6. Подсистема приватного облака. Платформа предоставляет решение всех вопросов нагрузки и построения приватного облака для хостинга. Платформа использует балансировку на стороне клиента, что минимизирует задержки фиксации ответов и сокращает необходимость межпроцессового взаимодействия.

7. Поддержка одновременно нескольких эфиров. Нагрузка распределяется между серверами равномерно с перетеканием сессий, т.е. предоставлена возможность даже портировать состояние пользователя с сервера на сервер.

8. Подсистема безопасности: программный firewall, ограничения по IP и геолокации, защита от DOS/DDOS, программная балансировка нагрузки, предотвращение двойного голосования.

9. Клиентское соединение и протокол JSTP. Клиент спроектирован как тонкий клиент с состоянием, существует механизм восстановления состояния приложения зрителя при сбоях, происходит восстановление соединений при разрыве.

10. Кроссплатформенность мобильных приложений. SinceTV поддерживает мобильные устройства на платформах iOS, Android и веб-браузеры Chrome, Firefox, Safari для MacOS, Linux, Windows.

11. Привязка к номеру телефона пользователя.

12. Универсальная административная панель форматов, а также конструктор форматов.

13. Возможность трансформации полученных в приложениях баллов в средства.

7. Заключение

В данной статье была рассмотрена платформа SinceTV для интерактивных приложений, реализующих концепцию второго экрана, приближенных к

реальному времени. Рассмотрена проблема устаревшей концепции телевидения. Также описана концепция второго экрана, продемонстрирована необходимость платформы для упрощённой работы согласно данной концепции. Описана платформа SinceTV, средства, с помощью которых она работает, особенности её реализации, а также преимущества, которые она предоставляет.

Платформа является простым решением для желающих обеспечить интерактивность своих телепередач. Она имеет особенности реализации, значительно отличающие её от немногочисленных платформ-аналогов, и выдвигающих её по сравнению с ними на совершенно другой уровень. Помимо скорости работы и гибкости создания приложений, платформа также предоставляет множество различных

преимуществ, таких как интеграция с социальными сетями, платёжными системами, системами сбора статистики и т.п.

Список литературы

1. Губский А.И. Платформа Since - задачи и решения [Электронный ресурс]. – Режим доступа: <https://andrew.gubskiy.com/publication/item/patform-since> (дата обращения 15.04.2016).
2. SinceTV [Электронный ресурс]. – Режим доступа: <http://sincetv.com/about-us/> (дата обращения 15.04.2016).
3. Шемсединов. Т.Г. JSTP / JavaScript Transfer Protocol [Электронный ресурс]. – Режим доступа: <https://github.com/metarhia/JSTP> (дата обращения 15.04.2016).

*ОСПЕНКО Д.С.,
ОЛІЙНИК Ю.О.*

ОГЛЯД МЕТОДІВ ВИРІШЕННЯ БАГАТОКРИТЕРІАЛЬНИХ ЗАДАЧ ПРИ ФОРМУВАННІ РАЦІОНУ ДНЯ

В даній роботі розглядаються деякі методи вирішення багатокритеріальних задач для вибору найбільш оптимального рішення із множини допустимих альтернатив.

In this paper we consider some methods for solving task with many criteria to select the best solution from the set of acceptable alternatives

Задача формування раціону дня полягає у виборі продуктів харчування за різними критеріями, такими як: вік людини, вага, період доби, кількість фізичних навантажень та ін. Усі ці чинники впливають на рішення вибору раціону дня.

Прийняття рішень, як вже досліджено, представляє собою вибір найбільш преференційного варіанту рішення із множини допустимих альтернатив або упорядкування множини рішень. Задача вибору варіантів, та як більш складна і охоплююча задача – задача розподілу ресурсів, постають споконвічними проблемами, з якими стикається людство у всіх сферах свого буття.[1] Нехай СПР обрано певну ціль щодо розвитку економічної системи, множину альтернативних стратегій (рішень), реалізація яких дає змогу досягти поставленої цілі, а також низку критеріїв оцінювання цих стратегій (в якості критеріїв можуть використовуватися спеціалізовані економічні показники). Науковий аналіз проблем прийняття рішень в економічній діяльності та підприємстві починається з моменту, коли хоча б частина альтернатив і (чи) критеріїв відома. У сучасній науці стосовно прийняття рішень центральне місце посідають багатокритеріальні задачі вибору. Вважається, що врахування багатьох критеріїв наближує постановку задачі до реального життя.[2]

Сутність методу послідовних поступок полягає в тому, що вихідна багатокритеріальна задача замінюється послідовністю однокритеріальних задач, область допустимих рішень яких звужується від задачі до задачі за допомогою додаткових обмежень, які враховують вимоги до критеріїв. При формуванні кожної задачі, по відношенню до важливішого критерію робиться поступка, величина якої залежить від вимог задачі і оптимального рішення за цим критерієм.

Головна ідея методу головного критерію полягає в тому, що вихідна багатокритеріальна задача оптимізації замінюється задачею оптимізації за одним критерієм з додатковими обмеженнями, які дозволяють в певному сенсі врахувати вимоги, які описувалися іншими критеріями. При формуванні раціону дня, цей метод використовувати не дуже вдало, адже не можна виділити головний критерій.

Розглянемо методи рішення, що полягають в зведенні початкової багатокритеріальної задачі до скалярної шляхом введення деякого узагальненого критерію. Всі ці методи мають схему: 1. Всі критерії нормують, тобто приводять до порівнянного безрозмірного вигляду; 2. Їх «згортають» в одну цільову функцію, так званий узагальнений критерій, враховуючи їх відносну важливість за допомогою вагових коефіцієнтів. І серед них обирають найкращу альтернативу. Цей метод є

дуже зручним при вирішенні задачі формування раціону дня, адже враховується відносна важливість критеріїв, що є дуже важливим.

Методи жорсткого пріоритету засновані на тому, що критерії розташовані за важливістю в ряд пріоритету $y_1 > y_2 \dots > y_m$ на основі якого проводиться послідовна оптимізація критеріїв. Принцип послідовної оптимізації на основі жорсткого пріоритету полягає в тому, що не допускається підвищення рівня менш важливих критеріїв, якщо це викликає, хоч би незначне, зниження рівня важливішого критерію. Практично це приводить до того, що спочатку відшуковується локальний оптимум для найбільш важливого критерію на всій множині допустимих альтернатив X , який фіксується у вигляді додаткового обмеження. Потім виконується пошук локального оптимуму другого за важливістю

критерію, але вже для нової допустимої множини X^01 і так далі. Таким чином, відбувається поступове звуження допустимої множини до єдиного оптимального рішення або оптимальної підмножини.[3]

Зауважимо, що більшість методів багатокритеріальної оптимізації передбачають виділення оптимального рішення безпосередньо із множини всіх рішень. У зв'язку з цим корисно проаналізувати отримані результати, щоб з'ясувати чи завжди вони приводять до отримання ефективних рішень, і якщо ні, то спеціально передбачити можливість поліпшення рішення, що виділяється, до ефективного.

В роботі наведено огляд методів вирішення багатокритеріальних задач для вибору найбільш оптимального раціону дня із множини допустимих альтернатив.

Список літератури

1. ПОБУДОВА СППР НА ОСНОВІ МЕТОДІВ ЕКСПЕРТНОГО ОЦІНЮВАННЯ [Електронний ресурс] – Режим доступу: <http://lib.chdu.edu.ua/pdf/posibnuku/313/11.pdf>
2. Економічний ризик : ігрові моделі [Електронний ресурс] – Режим доступу: <http://library.if.ua/book/132/8842.html>
3. Методи розв'язання багатокритеріальних задач [Електронний ресурс] – Режим Доступу: <http://studopedia.org/>

ДОСЛІДЖЕННЯ ЗАДАЧІ ПЛАНУВАННЯ ПО КРИТЕРІЮ МАКСИМАЛЬНО ПІЗЬНОГО МОМЕНТУ ЗАПУСКУ ТА МІНІМАЛЬНОГО СУМАРНОГО ВИПЕРЕДЖЕННЯ

В даній статті розглянуто задачу складання оптимального розкладу для одного приладу з різними директивними термінами з максимально пізнім моментом запуску та мінімальним сумарним випередженням. Було знайдено та розглянуто алгоритми, які шукають оптимальне рішення за одним критерієм: мінімальне сумарне випередження. На прикладі зображено використання алгоритму та знайдено оптимальний розв'язок для двох критеріїв окремо та перевірено чи є отриманий розклад оптимальний за двома критеріями.

In this article was considered the problem of optimal schedule for a device with a different due time in maximum late moment of start and minimum total tardiness. Similar algorithms was founded and seen, but they only founded minimum total tardiness. In the example shown as using the algorithm and found the optimal solution for the two criteria apart and checked the schedule as optimal obtained by two criteria.

. Вступ

В даній статті розглянуто задачу складання оптимального розкладу для одного приладу з різними директивними термінами в максимально пізній момент запуску та мінімальним сумарним випередженням. Даний розв'язок є актуальним у сучасному світі та не було знайдено аналогів рішення даної постановки задачі. Були знайдені аналогічні розв'язки для пошуку розкладу, але лише для одного критерію а саме мінімізація сумарного випередження. Було розглянуто на прикладі використання алгоритму та знайдено оптимальний розв'язок по двом критеріям окремо та перевірено чи є даний розклад оптимальний по двом критеріям.

2. Постановка задачі

Задана множина незалежних завдань $J = \{1, 2, \dots, n\}$ кожне з яких складається з однієї операції. Для завдань j відома тривалість виконання p_j та директивний термін виконання d_j . Завдання поступають в систему одночасно. Переривання завдань під час виконання не допускаються, тобто після виконання першого, починається виконання другого і т.д, поки не будуть виконані всі завдання. Необхідно знайти допустимий розклад, в

якому момент початку виконання завдань є найбільш пізнім та сумарне випередження моментів закінчення завдань відносно директивних термінів приймає мінімальне значення.

3. Пошук аналогів

В роботі було оглянуто алгоритм складання оптимального розкладу. Не було знайдено алгоритмів, які складають одночасно оптимальний розклад за двома критеріями, але були знайдені алгоритми, які шукають оптимальні розклади по одному з критеріїв. А саме по другому критерію - мінімізація сумарного випередження відносно директивних термінів.

Алгоритми, які були знайдені по другому критерію, в загальному опираються на те, що сортується директивні терміни по зростанню значень $D_1 < D_2 < \dots < D_m$, які розміщуються на часовій прямій в $m+1$ інтервалів I_1, I_2, \dots, I_{m+1} , де $I_1 = [0, D_1], I_2 = [D_1, D_2] \dots I_{m+1} = [D_m, \infty]$.

Робота називається вчасно зробленою, якщо вона є виконаною в директивний термін, та робота називається запізненою, якщо вона виконана після директивного терміну. Складність алгоритмів є

$$O(m!(\binom{n+m}{m})(P/m)^{3m}n) \text{ та}$$

$$O(m!(\binom{P_{\max}+m}{m})(m!(\binom{n+m}{m}))(P/m)^{O(m^2)}n) [1]$$

4. Опис алгоритму

Алгоритм, який було розглянуто по двом критеріям будується на тому, що розклад вважається оптимальним по двом критеріям, якщо виконується наступна умова [2]:

$$E(r) + nr = E(0), \quad (1)$$

де r - момент запуску виконання завдань, $E(r)$ -сумарне випередження при моменті запуску r , $E(0)$ - сумарне випередження при нульовому моменті запуску.

Для алгоритму спочатку будується розклад максимізації моменту запуску, потім на основі побудованого розкладу, будується розклад по критерію мінімізації сумарного випередження, після чого перевіряється чи є отриманий розклад оптимальним використовуючи формулу (1). У випадку, коли умова один не

виконується знаходиться значення інтервалу, на який потрібно зменшити момент запуску, після чого знову відбувається перевірка по формулі (1) та у випадку не виконання умови процедура повторюється, поки не буде знайдено оптимальний розклад з моментом запуску при якому буде задовільнена умова (1).

Складність поліноміальної складової алгоритму є $kO(n^2)$, де k - є заданою кількістю зміни структур розкладів. Під час використання алгоритму, можна побачити, що оптимальність розкладу по двом критеріям, без зміни моменту запуску задач розкладу та мінімізації сумарного випередження, частіше за все виконується. Тобто, у більшості випадків розклад, який є оптимальним за двома критеріями разом, є також оптимальним за двома критеріями окремо.[2]

5. Приклад алгоритму

Розглянемо приклад, з кількістю задач, що надходять $n = 10$.(табл.1)

Табл. 1. Послідовність σ для першого прикладу

№	j	p_j	d_j
1	3	8	10
2	1	10	22
3	9	14	37
4	2	13	50
5	4	14	65
6	5	12	77
7	6	23	100
8	8	18	120
9	7	22	150
10	10	15	170

Використовуючи таблицю 1 визначаємо момент запуску виконання завдань, який дорівнює різниці максимального директивного терміну та сумі тривалості

усіх завдань. Отже, $r = d_{\max} - \sum_{i=1}^n p_i = 21$.

Перевіримо розклад з моментом запуску 21 на виконання усіх директивних термінів.(табл.2)

Табл.2. Послідовність при моменті запуску 21

№	j	p_j	d_j	C_j	T_j
1	3	8	10	29	19
2	1	10	22	39	17
3	9	14	37	53	16
4	2	13	50	66	16
5	4	14	65	80	15
6	5	12	77	92	15
7	6	23	100	115	15
8	8	18	120	133	13
9	7	22	150	155	5
10	10	15	170	170	0

В таблиці 2 можна побачити, що маємо запізнення в розкладі. Максимальне запізнення дорівнює 19, отже новий момент початку запуску $r = 21 - 19 = 2$.

В таблиці 2 можна побачити, що маємо запізнення в розкладі. Максимальне

запізнення дорівнює 19, отже новий момент початку запуску $r = 21 - 19 = 2$. Розклад з новим моментом запуску зображений в таблиці 3.

Табл.3. Послідовність при моменті запуску 2

№	j	p_j	d_j	C_j	T_j
1	3	8	10	10	0
2	1	10	22	20	0
3	9	14	37	34	0
4	2	13	50	47	0
5	4	14	65	61	0
6	5	12	77	73	0
7	6	23	100	96	0
8	8	18	120	114	0
9	7	22	150	136	0
10	10	15	170	151	0

В таблиці 3 не маєя запізнення виконання завдань, отже отримано допустимий розклад з моментом запуску 2.

Далі використаємо алгоритм Алгоритм A1[2] для створення оптимального

розкладу по критерію мінімізації сумарного випередження.

Після декількох ітерацій отримаємо наступний результат.

Табл.4. Розклад при моменті запуску 0

№	j	p_j	d_j	C_j	T_j
1	3	8	10	8	2
2	1	10	22	18	4
3	9	14	37	32	5
4	2	13	50	45	5
5	4	14	65	59	6
6	5	12	77	71	6
7	6	23	100	94	6
8	8	18	120	112	8
9	7	22	150	134	16
10	10	15	170	149	21
				$E(\sigma) = 79$	

Тепер перевіримо оптимальність розкладу за допомогою формули (1).

$$E(r_{\max}) + nr_{\max} = E(0) \Rightarrow 59 + 10 * 2 = 79.$$

Отже, розклад отриманий по двом критеріям є оптимальним.

6.Висновок

На прикладі описаному в пункті 5 було зображено роботу алгоритму пошуку оптимального розкладу по двом критеріям - максимально пізнього моменту запуску та мінімального сумарного випередження.

Список літератури

1. Stavrov G.Kolliopoulos, George Steiner. Approximation algorithms for minimitation the total weighted tardiness on single machine – 2006 – p.1-10
2. Згуровский М . З . Задача построения допустимого расписания с максимально поздним моментом запуска и минимальным суммарным опережением / М . З . Згуровский , А . А . Павлов , Е . А . Халус // Системні дослідження та інформаційні технології – 2015. – №2. – С.7-

СПОСІБ АДАПТИВНОГО УПРАВЛІННЯ ПРОЦЕСОМ БУРІННЯ В УМОВАХ НЕВИЗНАЧЕНОСТІ

В поточній статті розглянуто застосування сучасних моделей машинного навчання та використання принципів методу міркування на основі прецедентів для вдосконалення та підвищення ефективності адаптивного управління процесом буріння свердловин. Наведено основні висновки з обширного дослідження особливостей процесу буріння в умовах невизначеності. Загальний підхід та пропонувані спосіб прогнозування параметрів буріння з елементами самовдосконалення на основі існуючих знань демонструється на прикладі визначення оптимального показника осьового навантаження на долото.

The subject of the article is an application of the modern methods of machine learning and case-based recognition to a task of improvement and efficiency raise of adaptive well drilling control. The main conclusions from research of drilling process and existing approaches are presented. The proposed method to prediction of main parameters of well drilling process with machine leaning approach is demonstrated using determination of optimal bit on weight parameter as example.

Актуальність проблеми

Проблема підвищення ефективності та зниження вартості буріння нафтових та газових свердловин вже тривалий час є предметом наукових та прикладних досліджень, проте залишається актуальною й на сьогодні. Незважаючи на те, що є чимало теоретичних розробок та певних практичних реалізацій систем, що дозволяють автоматизувати процес буріння майже повністю та дозволяють обійтися без втручання людини в процес, на практиці все ж поширеніший є ручне або напівавтоматичне управління процесом. Але оскільки, по перше, процес буріння є складним і динамічним, повна автономність його інтелектуальних складових в автоматизованих системах процесу буріння не є доцільною. По друге, процес поглиблення завжди проходить в умовах неповноти інформації, що дозволяє віднести задачу оптимального управління поглибленням до проблеми управління динамічними об'єктами в умовах невизначеності. А отже, як і показує досвід, роль фахівця – бурового майстра в управлінні процесом є ключовою [1,2].

Основна задача бурового майстра полягає в моніторингу за зміною параметрів буріння з метою забезпечення оптимального протікання процесу, а також

відслідковуванню непередбачуваних та виключних ситуацій. Виходячи з отриманої інформації і здійснюється адаптивне управління буровою установкою. В випадку більшості існуючих варіантів управління можна виділити декілька пов'язаних з таким підходом проблем.

Спостереження за процесом на об'єкті буріння виконується через моніторинг об'ємного графу параметрів в режимі реального часу, що означає необхідність постійної концентрації уваги за поточним станом. Проте при великому об'ємі даних з різних джерел це спричиняє велике навантаження на інженера і досконало володіти ситуацією є практично неможливим. Певні важливі симптоми можуть бути пропущені. По друге, в випадку певних позаштатних ситуацій чи неефективності процесу, процес може зупинятися на тривалий час, оскільки оцінка ситуації та пошук рішень може вимагати затрат значного часу для аналізу раніше отриманого масиву параметрів з залученням експертів [3].

Предмет та мета дослідження

Основна мета даного дослідження – знаходження способу підвищення ефективності адаптивного управління

процесом буріння з використанням новітніх інформаційних технологій.

Для більш глибокого вивчення даного питання було проведено дослідження ряду робіт в галузі оптимізації управління процесом буріння [1,2,3]. Розглянуто залежність різних показників ефективності процесу буріння, включаючи контрольовані, такі як навантаження на долото, частота обертання ротора; та впливу факторів середовища, таких як температура, тиск, густина та природа матеріалів. Визначено, що в сучасній науці так само гостро постає тема забезпечення надійності, коректності, стійкості та відновлюваності програмного забезпечення, що використовується в процесі буріння для підвищення для ефективності [4].

В результаті систематизації матеріалів дослідження було встановлено, що значна кількість розробок, які фокусуються на оптимізації параметрів буріння шляхом аналізу даних в режимі реального часу[5]. Проте існуючі рішення часто базуються на вдосконаленні та адаптації давно розроблених моделей, як наприклад оптимізація параметрів буріння методом множинної регресії на основі проведення експериментів[6]. Швидкі темпи й глобалізація процесів розвитку інформаційних технологій дозволяє

застосовувати й випробовувати зовсім інші методи для отримання оптимальних результатів.

Розгляд інноваційних наукових підходів до показав велике практичне значення та значні успіхи у застосуванні наукових знань про великі масиви даних та штучний інтелект у різних галузях та сферах економічної діяльності. На основі цього пропонується спосіб адаптивного управління бурінням свердловини з використанням методів машинного навчання(machine learning, ML) та міркування на основі прецедентів (case-based reasoning, CBR). Такий спосіб дозволить зняти перераховані вище обмеження та використати досвід для прогнозування параметрів ефективного процесу в залежності від спостережуваних факторів, а також додатково надає якісну оцінку розрахованому прогнозу.

Міркування на основі прецедентів

Суть підходу полягає в базуванні на накопиченні досвіду та подальшої адаптації рішення відомої задачі до нової. Прецедентний підхід дозволяє спростити процес прийняття рішення в умовах часових обмежень та наявності різного роду невизначеностей в процесі, вихідних даних та експертних знаннях, а також в випадку виникнення різного роду нештатних та аномальних ситуацій [7].

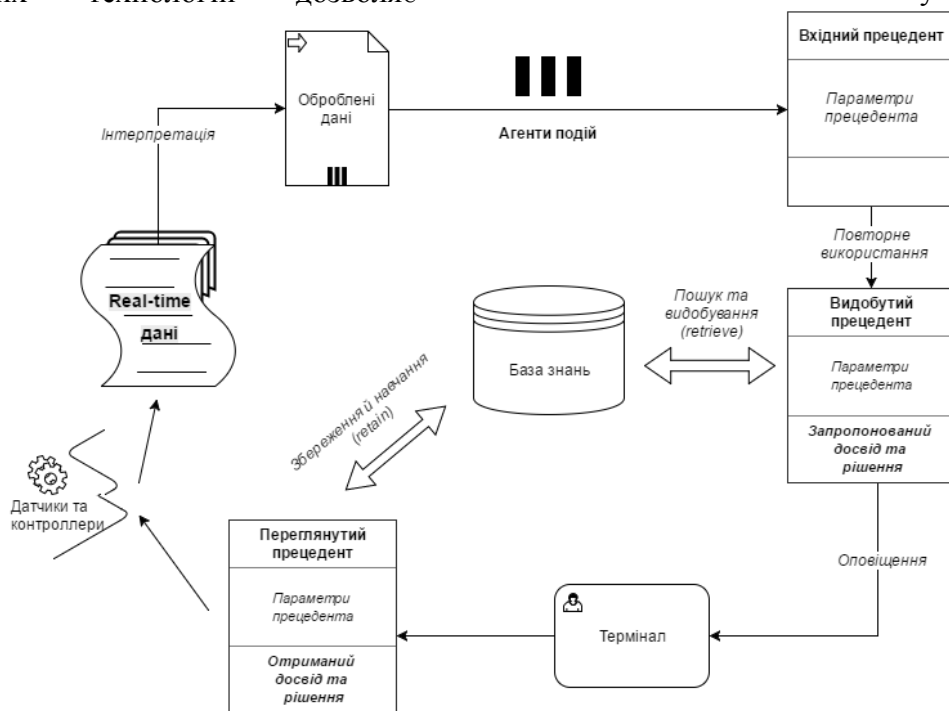


Рис 1. Структурна схема циклу міркування на основі прецедентів

Подібні ситуації часто виникають в процесі буріння скважини протяжністю в декілька кілометрів через різні геологічні формування та пласти. І для кожної свердловини характерним є виникнення як нових проблем, так і таких що вже мали місце в минулому. Підхід міркування на основі прецедентів дає можливість перевикористання попередніх знань про конкретні ситуації для запобігання та ефективного вирішення проблем [8]. На рисунку 1 подана структурна схема загального випадку CBR-циклу, адаптованої для використання в прогнозуванні ситуацій та підтримці прийняття рішення в процесі моніторингу буріння.

Таким чином, використання методу міркування на основі прецедентів та аналогій забезпечує спрощення та поліпшення ефективності адаптивного управління бурінням в умовах невизначеності. Забезпечується це шляхом аналізу потоку даних та параметрів в режимі реального часу, розпізнавання шаблонів, систематизація та візуалізація даних; передбачення проблем, пошук рішень та надання порад інженеру як адаптувати процес для підвищення ефективності та уникнення позаштатних ситуацій.

Постановка задачі прогнозування параметрів буріння (в загальному вигляді) за допомогою ML

Множина вхідних даних (навчальна вибірка) задається у вигляді вектора $x = \{X_1, X_2, \dots, X_i, \dots, X_M\}$ та відповідних їм значень вихідної змінної $y = \{y_1, y_2, \dots, y_i, \dots, y_M\}$, де i - порядковий номер точки спостереження.

Спрощене рівняння швидкості проходки свердловини можна представити у вигляді

$$ROP = C_1 \cdot DR \cdot WOB^{c_2} \cdot RPM^{c_3},$$

де ROP (rate of penetration) – швидкість проходки;

DR (drilling resistance) – буримість породи;

WOB (weight on bit) – навантаження на долото;

RPM (rotations per minute) – частота обертання ротора;

$C1, C2, C3$ – сталі.

Рівняння не описує процес буріння максимально адекватно, про те було вибрано, щоб абстрагуватися від надмірних деталей та складності та показати, що швидкість проходки залежить від певних контрольованих параметрів буріння. Одна з задач бурового майстра й полягає в контролюванні параметрів, щоб швидкість проходки залишалася оптимальною. В свою чергу, для кожного з параметрів можна встановити залежність від інших спостережуваних показників буріння. На основі цього й необхідно, маючи достатньо велику навчальну вибірку співвідношень спостережуваних показників та конкретного оптимального значення відповідного параметру, побудувати таку модель $Y_i = Y(X_i)$, яка зможе достатньо точно прогнозувати оптимальне значення параметру для будь якого набору спостережуваних показників.

Застосування ML для визначення оптимального показника на прикладі навантаження на долото

Для достатньо великого масиву даних з навчальної вибірки, та отриманих показників сформуємо гіпотезу (в загальному вигляді):

$$h_{\theta}(x) = \theta^T x$$

де x – вектор параметрів, θ – вектор відповідних коефіцієнтів.

Для оцінки точності моделі на базі навчальної вибірки сформуємо функцію втрат $E(\theta)$:

$$E(\theta) = \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2 \quad (1)$$

Таким чином задача побудови оптимальної моделі зводиться до пошуку таких коефіцієнтів θ , за яких $E(\theta)$ прийматиме мінімальне значення.

Оскільки $E(\theta)$ було підібрана таким чином, що являє собою опуклу функцію з одним глобальним мінімумом, доцільним для визначення оптимальних коефіцієнтів θ є використання методу градієнтного

спуску. Метод градієного спуску є ітеративним методом, який можна ефективно реалізувати в програмному вигляді, добре працює для великого значення n (кількості параметрів, що розглядаються) та є ефективним при правильному виборі коефіцієнту α .

Крок методу градієнтного спуску можна задати у вигляді функції:

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} E(\theta), \text{ для } j=0, \dots, n \quad (2)$$

Підставивши рівняння (1) в рівняння (2), отримуємо

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

Для запобігання перенаванчання виконаємо нормалізацію моделі методом статистичної регуляризації:

$$\theta_j := \theta_j - \alpha \frac{1}{m} \left[\sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) x_j^{(i)} + \frac{\beta}{m} \theta_j \right] \Rightarrow$$

$$\theta_j := \left(1 - \alpha \frac{\beta}{m} \right) \theta_j - \alpha \frac{1}{m} \times \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

Для застосування моделі, необхідно визначити, які ж саме параметри можна використати для визначення навантаження на долото. Для кращої демонстрації принципів даного способу без надмірного ускладнення прикладу було вирішено обмежитися випадком роторного буріння для твердих порід. При бурінні твердих крихких порід відбувається структурне руйнування породи по всьому об'єму (ΔV) на дрібні шматки різномірної форми.

Таким чином, як відомо, роботу долота можна подати як

$$A = Fv\Delta t,$$

де Δt – час роботи долота, а енергію, як

$$\Delta W = \Delta V \sigma_{\text{пов}} k = v\Delta t \cdot d \cdot l \cdot \sigma_{\text{пов}} k_n,$$

де $\sigma_{\text{пов}}$ – коефіцієнт поверхневого натягу матеріалу породи. Він характеризує сили міжмолекулярного тяжіння та визначає силу, необхідну для розриву міжмолекулярних зв'язків;

а k_n – коефіцієнт об'ємного руйнування, що характеризує крихкість породи, її здатність до розколювання на шматки аж до порохоподібного стану. Він залежить як від властивостей самої породи, так і від параметрів буріння та властивостей рідини для промивання – густини ρ , пластичної в'язкості n , динамічного напруження зсуву τ , тощо.

Таким чином можна показати, що сила руйнування породи, а відповідно й навантаження на долото залежить від зазначених вище параметрів, які можна враховувати при побудові моделі прогнозування оптимального навантаження на долото (для оптимізації швидкості проходки).

$$F = \frac{v\Delta t \cdot d \cdot l \cdot \sigma_{\text{пов}} k_n}{v\Delta t} = d \cdot l \cdot \sigma_{\text{пов}} k_n$$

Висновки

На основі аналізу предметної області та різноманітних способів, було показано на практичному прикладі можливість й доцільність застосування інтелектуальних технологій для оптимізації управління бурінням. В подальшому є доцільним проведення дослідження інших параметрів та формалізація їх у вигляді функцій (залежності контрольованих параметрів, що впливають на ефективність поглиблення свердловини) для формування аналогічних моделей їх адаптивного налаштування, а також дослідження способів вибору оптимальнішої моделі для гіпотези для здійснення ще точнішого прогнозування.

Список літератури

1. Семенцов Г. Н., Горбійчук М. И., Тельшева Т. А. Способ и алгоритмы оперативного управления процессом углубления скважин. // Нефтяная промышленность. Серия: Автоматизация и телемеханизация в нефтяной промышленности. – 1984.- №1. – с. 32-35.
2. Горбійчук М. И. Оптимізація процесу буріння глибоких свердловин : монографія / М. И. Горбійчук, Г. Н. Семенцов. – ІваноФранківськ : Факел, 2006. – 493 с
3. Crichton, M. T.; Lauche, K.; and Flin, R. 2005. Incident Command Skills in the Management of an Oil Industry Drilling Incident: A Case Study. *Journal of Contingencies and Crisis Management* 13(3): 116–128.
4. Martha Belem, Saldivar Márquez, Islam Boussaada, Hugues Mouni. *Analysis and Control of Oilwell Drilling Vibrations: A Time-Delay Systems*. – Switzerland 2015
5. Booth, J. 2011. Real-Time Drilling Operations Centers: A History of Functionality and Organizational Purpose — The Second Generation. *SPE Drilling & Completion* 26(2): 295–302.
6. Bourgoyne A.T. Jr., Young F.S., "A Multiple Regression Approach to Optimal Drilling and Abnormal Pressure Detection", *SPE* 4238, August 1974
7. Aamodt, A., Plaza, E., Case-Based Reasoning: Fundamental Issues, Methodological Variations, and System Approaches. *Artificial Intelligence Communications*, Vol. 7, No. 1, pp. 39-59, 1994.
8. Marling, C., Rissland, E., Aamodt, A., Integrations with case-based reasoning. *Knowledge Engineering Review*, Cambridge University Press. , 2005. 20(03): p. 241-245.

СПОСІБ ОПТИМІЗАЦІЇ ТЕХНОЛОГІЧНОГО ПРОЦЕСУ БУРІННЯ НАФТОВИХ СВЕРДЛОВИН НА ОСНОВІ НЕЧІТКОЇ ЛОГІКИ

Наведений аналіз способів керування процесом буріння під час поглиблення свердловини і обґрунтована можливість і актуальність розробки нечіткого способу оптимізації вартості метра проходки свердловини з використанням метода адаптивного керування процесом поглиблення. Розглянута базова модель процесу поглиблення свердловини і критерій оптимального буріння свердловини. Обґрунтоване введення нечіткості при визначенні часу спуско-підйомних операцій, що впливає на досягнення оптимального значення критерія. Вказані шлях розв'язання задачі нечіткої оптимізації і вибір алгоритму рішення.

Ключові слова: БУРІННЯ, ПОГЛИБЛЕННЯ СВЕРДЛОВИНИ, МАТЕМАТИЧНА МОДЕЛЬ, НЕЧІТКІСТЬ, НЕЧІТКИЙ КРИТЕРІЙ ОПТИМАЛЬНОСТІ, НЕЧІТКА ЗМІННА, ПРИНЦИП РОЗШИРЕННЯ, НЕЧІТКА ЛОГІКА.

The subject of the article is the analysis of methods of drilling process control while well deepening and proved the possibility and urgency of developing a fuzzy method for optimizing the cost per meter of the well penetration using the method of deepening process adaptive management. We consider the basic model of the process of deepening wells and the optimal drilling conditions. Reasonable introduction of ambiguities in determining the time of dip and lift operations, which has influence for the achievement of the optimal value of the criterion. Shown the way of solving the problem of fuzzy optimization and choosing the algorithm of solutions.

Keywords: DRILLING, DEEPENING WELLS, MATHEMATICAL MODEL, FUZZY, FUZZY OPTIMALITY CRITERION, FUZZY VARIABLE, EXTENSION PRINCIPLE, FUZZY LOGIC.

1. Вступ:

Поглиблення свердловин на нафту і газ є найдорожчим етапом з проекту. побудови свердловин. Оптимізація процесу поглиблення свердловин суттєво впливає на техніко-економічні показники будівництва свердловин. Складність оптимізації процесу поглиблення полягає в неоднорідності порід, що буряться і неповноті інформації про взаємодію породоруйнівного інструменту з породою. В силу природної складності більшість технологічних та геофізичних параметрів в процесі поглиблення мають нечіткий характер.

Для оптимізації процесу поглиблення розроблено математичні моделі [1,2], які в різному сенсі задовольняють потреби виробництва, але останнім часом багато уваги приділяють сучасним способам оперативного управління на основі нечіткої логіки [2], які дозволять підвищити ефективність управління і рентабельність за рахунок інтелектуалізації моделей і алгоритмів оптимізації.

Таким чином, розробка нечіткої моделі поглиблення свердловини для оптимізації вартості метра проходки і побудова на цій основі способу оптимізації процесу буріння є актуальною.

2. Постановка задачі:

Процес механічного буріння є стохастичним, нестационарним і таким, що розвивається у часі. Це значно ускладнює визначення цілого ряду параметрів, що впливають на процес руйнування породи. Базовими параметрами технологічного процесу поглиблення при роторному бурінні є осьове навантаження на долото, частота обертання долота, витрата промивної рідини, зношення опор та озброєння долота та ін.

Оскільки вище наведені параметри є нечіткими при вимірюванні їх безпосередньо в процесі поглиблення, то в роботі розглядається питання розробки нечіткої моделі на основі розробленої диференційної моделі [3] і можливість використання нечіткої моделі в способі нечіткого оптимального керування процесом поглиблення свердловини.

Мета дослідження – скласти нечітку модель процесу поглиблення свердловини, на базі якої є можливий спосіб нечіткого оптимального керування процесом поглиблення свердловини з мінімальною вартістю метра проходки свердловини.

Для цього необхідно скласти нечітку модель для оптимального керування процесом поглиблення, перевірити чи є вона адекватна існуючій диференційній моделі [3] і чи дає це дослідження право на використання її в способі нечіткого керування.

Розробити алгоритми та програмне забезпечення для способу нечіткого керування, перевірити на адекватність результати при порівнянні способу нечіткого керування і керування на базі диференційної моделі процесу поглиблення свердловини.

3. Розгляд існуючих рішень:

Існує багато математичних моделей, які розроблені вченими на протязі багатьох років. Аналіз моделей з точки зору їх використання в оперативному керуванні процесом поглиблення свердловин на момент 1988 року наведений в роботі [3]. За наступний період до аналізу існує більше робіт, однак по суті поставленої мети найбільшу увагу визивають наступні роботи.

Модель Елдрета [1] призначена для оптимізованого виконання різних операцій буріння, вона заснована на свердловинних вимірах. Модель процесу буріння представляє комбінований вплив умов на заборі свердловини і роботи колони бурильних труб. На основі реакції зворотного зв'язку система використовує змінні моделі процесу буріння і моделі пластів для визначення оптимальної механічної швидкості проходки для пласта що в даний момент буриться. Результати вимірювань як реакції зворотного зв'язку одночасно використовуються для підтвердження достовірності діючої в даний момент моделі пластів. Якщо виявлені відхилення, модель пластів оновлюється так, щоб вона відображала нові результати вимірювань. Цей процес відбувається безперервно протягом всього процесу буріння пласта.

Недоліки: спосіб передбачає побудову в ході роботи в свердловині адаптивної моделі буріння, яка містить параметри моделі пластів і параметри гідравлічної моделі промивання,

що дозволяє підлаштовувати її під умови на заборі, проте модель не враховує можливості вибору оптимального режиму буріння – вона призначена для роботи в рамках одного режиму.

Відома також модель процесу буріння Самсоненко-Бойченко [4], що представляє собою експонентний тренд механічної швидкості проходки, яку отримують в результаті шести і більше експериментальних даних буріння в інтервалі пласта однаковою буримості з подальшою їх апроксимацією методом найменших квадратів. Далі коефіцієнти моделі буріння підставляють в критерій «мінімум вартості метра проходки», що містить вираз залежності часу буріння t від осьового навантаження на долото G , частоти його обертання n і витрати бурового розчину Q у вигляді полінома першого ступеня, і виробляють пошук мінімуму критерію методом штрафних функцій. Отримані оптимальні параметри встановлюють на буровій установці і подальше регулювання буріння виробляють на оптимальних уставках G , n і Q .

Недоліки: спосіб вимагає попереднього проведення факторного експерименту для отримання поліноміальної залежності часу буріння від навантаження на долото, частоти його обертання і витрати бурового розчину; використовує показник роботи долота на неоднорідних пластах, що не є найбільш оптимальним.

4. Базова математична модель:

Для розробки способу нечіткого керування найбільш прийнятна математична модель, яка була розроблена в Івано-Франківському інституті нафти і газу колективом дослідників під керівництвом професора Семенцова Г.Н. [5] і використана для метода адаптивного оптимального керування процесом поглиблення свердловини [3]. Вона дозволяє визначити оптимальне осьове навантаження на долото в процесі його роботи з урахуванням мінливих умов. Експериментально і теоретично підтверджено, що математична модель справедлива при різних формах зносу зубів долота.

Математична модель:

$$\begin{cases} \frac{dh}{dt} = \frac{V_0(F, n)}{\varepsilon} \\ \frac{d\varepsilon}{dt} = K_\varepsilon(F, n) \\ \frac{d\xi_{\text{оп}}}{dt} = K_{\xi_{\text{оп}}}(F, n) \end{cases} \quad (1)$$

при граничних умовах:

$$\begin{cases} h(0) = \xi_{\text{оп}}(0) = 0, h(t_\delta) > 0; \\ \varepsilon(0) = 1, 1 \leq \varepsilon(t_\delta) \leq (1+m)^2; \\ 0 \leq \xi_{\text{оп}}(t_\delta) \leq 1; \\ F_{\text{min}} \leq F \leq F_{\text{max}}, \\ n_{\text{min}} \leq n \leq n_{\text{max}}, \\ Q = \text{const} \end{cases} \quad (2)$$

де h - проходка; F - осьове навантаження на долото, n - частота обертання долота; $\xi_{\text{оп}}$ - відносне зношення опор долота; t_δ - час буріння; Q - витрата промивальної рідини; ε - оцінка відносного зносу озброєння долоті; K_ε - швидкість зміни оцінки відносного зношення озброєння долота.

$$V_0(F, n) = K_1 F^{\alpha_1} n^{\beta_1} \quad (3)$$

$$K_\varepsilon(F, n) = K_2 F^{\alpha_2} n^{\beta_2} \quad (4)$$

$$K_{\xi_{\text{оп}}}(F, n) = K_3 F^{\alpha_3} n^{\beta_3} \quad (5)$$

де K, α, β - залежать від типу долота та фізико-механічних властивостей гірських порід.

При цьому $K_\varepsilon(F, n)$ описує ситуацію коли зношується озброєння долота, $K_{\xi_{\text{оп}}}(F, n)$ - коли зношуються опори.

Оптимізація режимів роботи бурової установки проводиться за умов підтримки осьового навантаження F протягом деякого проміжку часу Δt і встановленої частоти обертання $n = \text{const}$. Припускається, що за час Δt властивості породи не змінюються або змінюються в незначній мірі, тому початкова швидкість проходки $V_0(F, n) = V_0 = \text{const}$.

Для рішення поставленої задачі оптимізації був вибраний критерій мінімальної вартості метра проходки.

$$q = \frac{C_\delta(t_\delta - t_{\text{сп}}) + d}{h} \quad (6)$$

де C_δ - вартість години роботи бурової установки, t_δ - загальний час буріння, $t_{\text{сп}}$ - час, що витрачається на спуско-підйомні операції, d - вартість долота.

5. Введення нечіткості:

Значення $t_{\text{сп}}$ зазвичай передбачається відповідно до експериментальних вимірів на інших свердловинах або в попередніх рейсах на тій самій свердловині. Таке передбачення є

досить складним якщо додатково до технічних характеристик свердловини та обладнання додати такі фактори як кваліфікація бригади, погодні умови та ін.

Постановка задачі: знайти осьове навантаження F із допустимої області A_U , таке що буде незмінним протягом часу Δt та критерій оптимальності набудатиме мінімального значення.

Тому розглядатимемо $t_{\text{сп}}$ як нечітку величину, що має функцію належності $\mu(t_{\text{сп}})$.

$$\mu(t_{\text{сп}}) = \exp\left\{-\frac{(t_{\text{сп}} - m)^2}{2\sigma^2}\right\} \quad (7)$$

де m, σ - числа, що характеризують розмитість нечіткої величини $t_{\text{сп}}$.

Критерій оптимальності (6) зручно подати так:

$$q = \frac{C_\delta t_\delta + d}{h} + \frac{C_\delta}{h} t_{\text{сп}} \quad (8)$$

Оскільки $\mu(t_{\text{сп}})$ - функція належності нечіткої величини x і за умови, що якщо між множинами X і Y існує однозначна залежність $y = \varphi(x)$ така, що $x \in X, y \in Y$ - визначається $\varphi^{-1}(y) = \{x: x \in X, \varphi(x) = y\}$ звідки отримаємо:

$$\mu(y) = \mu(\varphi^{-1}(y)) \quad (9)$$

Для випадку з критерієм що розглядається (9) отримає вигляд:

$$\mu(q) = \mu(\varphi^{-1}(q)) \quad (10)$$

Із рівняння (8) $\varphi^{-1}(q) = t_{\text{сп}} = \frac{q-a}{b}$, де $a = \frac{C_\delta t_\delta + d}{h}, b = \frac{C_\delta}{h}$

Підставимо у (7):

$$\mu(q) = \exp\left\{-\frac{(q - a - mb)^2}{2\sigma^2 b^2}\right\} \quad (11)$$

Виберемо певне значення $\mu(q) = \alpha$ та йому відповідне $q = q^*$ яке можна визначити з (7) при умові $\alpha \in [0,1]$.

Врахувавши попередні умови та допущення можна визначити:

$$q = \frac{C_\delta(t_\delta + m) + d}{h} + \frac{\sigma C_\delta}{h} \sqrt{\ln \frac{1}{\alpha^2}} \quad (12)$$

за умови що q, h, t_δ визначені на універсумі U для якого виконуються обмеження

$$F_{\text{min}} < F < F_{\text{max}}, n_{\text{min}} < n < n_{\text{max}}$$

Час буріння визначається відповідно до чіткої моделі коли зношення озброєння долота випереджує зношення опор $t_\delta = \frac{\varepsilon^* - 1}{K_\varepsilon}$.

Тоді маємо критерій оптимальності для випадку коли озброєння долота зношується швидше ніж його опори:

$$q = \frac{C_{\delta}A_{\varepsilon} + K_{\varepsilon}(C_{\delta}m + d)}{V_0 \ln(A_{\varepsilon} + 1)} + \frac{\sigma K_{\varepsilon} C_{\delta}}{V_0 \ln(A_{\varepsilon} + 1)} \sqrt{\ln \frac{1}{\alpha^2}} \quad (13)$$

де $A_{\varepsilon} = \varepsilon^* - 1$, де ε^* – значення оцінки ε в кінцевий момент часу.

6. Результати:

Задача (13) є задачею нелінійного програмування з обмеженнями (2). Такі задачі, як правило, розв'язуються числовими методами, які для своєї реалізації вимагають, щоб критерій оптимальності і обмеження мали похідні до другого порядку включно; критерій оптимальності повинен бути унімодальною функцією, а обмеження – випуклими. Тільки у випадку виконання цих умов існують необхідні і достатні умови існування мінімуму скалярної функції на множині значень її аргументів, які задовольняють певним обмеженням.

Крім того критерій оптимальності (13) може мати складну топологію, наприклад, у вигляді яру, що також затрудняє розв'язок задачі оптимізації за допомогою класичних методів. Як альтернативу таким методам можна використати генетичні алгоритми.

7. Висновки:

Показано, що через технологічні та суб'єктивні фактори тривалість спуско-підйомних операцій може розглядатись як нечітка величина з певною функцією належності. Тому задача розробки способу нечіткого оптимального керування процесом поглиблення свердловини може бути сформульована у термінах нечіткої математики. «Розмитість» $t_{сп}$ може суттєво впливати на величину вартості метра поглиблення яка розглядалась як критерій оптимальності. Вказані шляхи розв'язання для визначення значень нечітких параметрів.

Список літератури:

1. Пат. 2244117 РФ. Способ управления работой в скважине и система бурения скважины / ЭЛДРЕД Уолтер Д. // Бюл. - 2004. - № 1. - С. 96.
2. Влацкая И.В., Литвинов М.А. Разработка нечеткой модели и алгоритма оптимизации выбора управляющих воздействий для технологического процесса бурения скважин // Компьютерная интеграция производства и ИПИ(CALS) технологии / Сборник статей всероссийской научно-практической конференции. - Оренбург: ИПК ОГУ, 2005. - 158 – 162 с.
3. Телишева Т.О. Разработка адаптивного метода управления технологическим процессом углубления нефтяных скважин: Автореферат до дис. канд. техн. наук. – Івано-Франківськ: Івано-Франківського нац. техн. ун-т нафти і газу, 1988. - 18 с.
4. Пат. 1231946 СССР. Способ регулирования процесса бурения / Самсоненко В. И, Бойченко В. А // Бюл. - 1995. - № 3. - С. 34.
5. Горбійчук М. І. Оптимізація процесу буріння глибоких свердловин : монографія / М. І. Горбійчук, Г. Н. Семенцов. – Івано- Франківськ : Факел, 2006. – 493 с.

ОГЛЯД АЛГОРИТМІВ ФОРМУВАННЯ РЕКОМЕНДАЦІЙ З ОБМІУ ПОДАРУНКАМИ

В цій статті проведено огляд основних методів, які застосовуються для створення рекомендацій. Зроблено аналіз наявних алгоритмів реалізації цих методів, які можна застосовувати при побудові рекомендацій на основі інформації про користувача (профіль, вимоги тощо) та характері його поведінки (оцінки). Представлена порівняльна характеристика алгоритмів відносно їх точності.

This article provides an overview of the main methods used to create recommendations. The analysis of existing algorithms implement methods that can be used in the construction recommendations based on user information and the nature of its conduct. Presented comparative characteristic of algorithms based on their accuracy.

1. Вступ

В цій статті проведено огляд основних методів, які застосовуються для створення рекомендацій. Зроблено аналіз наявних алгоритмів реалізації цих методів, які можна застосовувати при побудові рекомендацій на основі інформації про користувача та характері його поведінки. Представлена порівняльна характеристика алгоритмів на основі їх точності.

2. Методи побудови рекомендацій

Вирізняють три основних підходи для побудови рекомендаційних систем:

1. колаборативна фільтрація (collaborative filtering);
2. підхід, заснований на знаннях (knowledge-based);
3. гібридний (hybrid).

Метод колаборативної фільтрації заснований на аналізі великої кількості інформації про поведінку користувача, його активності та вподобань, та прогнозує, що сподобається користувачеві, виходячи із його схожості з іншими користувачами. Поведінка, як і активність в конкретній предметній області може бути представлена різними способами, наприклад, для системи обміну подарунків, за поведінку можна взяти оцінки отриманих подарунків. На основі цієї інформації від багатьох користувачів, можна згрупувати їх за вподобаннями (група буде включати декілька користувачів, які отримали однакові або дуже схожі подарунки). В кожній групі ідентифікується найбільші за рейтингом

подарунки. На основі зібраної інформації, конкретному дарувальнику, можна рекомендувати найвище оцінений подарунок з тих, які отримувач ще не отримував. Досить ефективний метод, який має як переваги (наприклад, різноманітність рекомендацій, відносно висока вірогідність точної рекомендації), так і недоліки, основний недолік – вирішення задачі холодного старту. Ця задача буде описана в наступному розділі.

Метод, заснований на знаннях про предметну область. На відміну від попереднього методу, робить рекомендації на основі додаткових знань, таких як:

1. вимоги користувачів, які представлені через окрему сутність (case-based);
2. властивості користувача (demographic-based);
3. відносна корисність кожного товару безпосередньо для користувача, яка окрема розраховується (utility-based);
4. схожість об'єктів рекомендації (content-based). Цей вид рекомендацій набув широкого розповсюдження в останні часи, здебільшого за рахунок своєї простоти, тому його часто виносять як окремий вид. В чистому виді він спирається на попередню поведінку користувача, наприклад, переглянутий фільм, та рекомендує далі фільми, схожі за жанром, режисером або іншим критерієм з переглянутим. Тобто створює рекомендації в досить вузькому напрямі. Тому до використання даного методу треба підходити досить обачливо, та враховувати предметну область, оскільки для інтернет магазину вже

даний підхід не буде доречним, і принесе більше шкоди ніж користі.

Гібридний метод полягає в комбінуванні попередніх двох методів. Найбільш складний в реалізації, але й найбільш ефективний.

3. Алгоритми колаборативної фільтрації

Формально задачу колаборативної фільтрації можна представити наступним чином. На вході відомо:

- множина користувачів

$$U = \{u_1, u_2, \dots, u_n\}, \quad (1)$$

- множина об'єктів (подарунки, фільми, блоги тощо)

$$I = \{i_1, i_2, \dots, i_m\}, \quad (2)$$

- множина подій (оцінок фільмів, подарунків, покупки товарів)

$$D = \{d_1, d_2, \dots, d_{nm}\}. \quad (3)$$

Кожна подія представляє собою вектор виду:

$$d_j = (r_{ui}, u, i \dots), \quad (4)$$

де r_{ui} - результат оцінки, яку поставив користувач u об'єкту i . Три крапки означають, що набір елементів вектору можна доповнювати при необхідності, наприклад, такими значеннями, як час оцінки, пора року тощо.

Необхідно:

- передбачити уподобання:

$$\hat{r}_{u,i} = Predict(u, i \dots) \approx r_{ui}, \quad (5)$$

- створити кожному користувачеві список персональних рекомендацій з k елементів:

$$u \mapsto (i_1 \dots i_k) = Recommend_k(u \dots) \quad (6)$$

- знайти схожі об'єкти:

$$i \mapsto (i_1 \dots i_m) = Similar_m(i) \quad (7)$$

Існує декілька алгоритмів для вирішення поставленої задачі.

Перший і найбільш простий – алгоритм кластеризації, полягає з знаходженні схожості користувачів за їх історією оцінок через деяку функцію:

$$sim(u, v), \quad (8)$$

об'єднанні користувачів у кластери за схожістю:

$$u \rightarrow F(u), \quad (9)$$

надання оцінки користувачеві, як середню оцінку всього кластера даному об'єкту, за формулою:

$$\hat{r}_{u,i} = \frac{1}{|F(u)|} \sum_{r \in F(u)} r_{vi}. \quad (10)$$

Але цей метод має ряд недоліків:

- немає що рекомендувати новим;
- не враховується специфіка кожного користувача;
- якщо в кластері ніхто не оцінював об'єкт, то неможливо зробити передбачення.

При чому цей алгоритм має низьку точність. Точність частіше всього вимірюється через розрахунок середньоквадратичної помилки - RMSE (Root Mean Square Error).

$$RMSE = \sqrt{\frac{1}{|T|} \sum_{(u,i) \in T} (p_{ui} - r_{ui})^2}, \quad (11)$$

де u – користувач,

i – об'єкт,

r – оцінка,

p – прогнозована оцінка,

T – загальна кількість тестових оцінок.

Rmse для розглянутого алгоритму рівна 1.53.

Другий алгоритм - user-based. Він прогнозує оцінки на основі схожості користувачів між собою, при чому більш схожим користувачам надається більша вага. Детальний опис алгоритму можна знайти в [1]. Rmse цього алгоритму рівна 1.49. Але все одно залишаються наступні проблеми:

- немає що рекомендувати новим або нетиповим користувачам;
- холодний старт - нові об'єкти нікому не рекомендуються.

Третій – алгоритм item-based. Дуже схожий на попередній, але надає рекомендації на основі схожості об'єктів між собою, а не користувачів. Детальний опис алгоритму можна знайти в [1]. Точність при цьому рівна 1.35. Проблеми, які залишилися:

- холодний старт;
- рекомендації часто тривіальні(оскільки враховується середній рейтинг фільму).

Четвертий – алгоритм SVD (Singular Value Decomposition). Хоча він і найбільш складний для розуміння та в реалізації, але має найбільшу точність з усіх розглянутих алгоритмів – 1.30, високу продуктивність та низьке споживання пам'яті. Ще однією перевагою даного алгоритму є його самонавчання. Детальний опис алгоритму можна знайти в [2].

4. Алгоритми методу, заснованого на знаннях

Найчастіше для цього методу застосовують алгоритм it-idf, який дозволяє визначити схожість двох текстів, а також знайти ключові слова тексту, які потім можуть

використовуватись як теги. Даний алгоритм складається з трьох етапів:

1. визначення ваги слів;
2. вирахування частоти терму;
3. ранжування слів за важливістю.

Детальний опис алгоритму можна знайти в [3].

Якщо ранжувати слова за низхідними значеннями, то перші декілька слів і будуть давати адекватне уявлення про вміст документу.

Перед застосуванням цього алгоритму бажано привести всі слова тексту до словарних форм (алгоритм Портера). Його точність на тестових даних становила 1.46.

Також використовують і більш прості алгоритми, але вони дають гірший результат точності.

5. Висновок

З розглянутих алгоритмів, в чистому виді найбільш точним виявився SVD. Для нього є модернізації – SVD++ та time SVD ++. Їх rmse менше від оригінального алгоритму на 0.01, але вони мають значно меншу продуктивність, що не бажано для невеликих систем.

Хоча SVD і найбільш точний, алгоритми методу, заснованого на знаннях часто використовують, тому що вони, на відміну від методів колаборативної фільтрації, не мають проблеми холодного старту та простіші в реалізації. А для покращення точності застосовують їх комбінації.

Список літератури

1. Ricci F. Recommender Systems Handbook // Springer New York– 2011. - №2 – P.837
2. Jannach D. Recommender Systems: An Introduction // Cambridge University Press -2010. – P.333
3. Segaran T. Programming Collective Intelligence// O'Reilly Media – 2007 – P.323

ВИКОРИСТАННЯ ТРІЙКОВОЇ ФУНКЦІЇ ВТРАТ У ЗАДАЧІ РОЗПІЗНАВАННЯ ОБЛИЧ

Стаття присвячена опису трійкової функції втрат та її використання у навчанні нейронної мережі для задачі розпізнавання облич.

The subject of the article is definition of triplet loss function and its usage in neural network training for face recognition.

1. Вступ

Використовуючи глибоку згорткову нейронну мережу, можна відобразити зображення облич у компактний евклідовий простір, відстані у якому відповідають за міру схожості облич [1]. Нейронна мережа тренується за допомогою функції втрат так, що квадратичні L2 відстані у відображеному просторі повинні відповідати наступному: обличчя однієї і тієї ж самої людини мають малі відстані, обличчя різних людей мають великі відстані.

2. Основна частина

На відміну від звичайного використання пар позитивних (тобто, зображення, що є схожими на задане) та негативних (не схожих) зображень, трійкова функція втрат намагається встановити відступ між кожною парою облич однієї людини та інших облич. Таким чином, це дозволяє обличчям однієї людини знаходитися на топологічному многовиді, в той же час дотримуючись відстані і розрізнення від інших людей.

Відображення представлено функцією $f(x) \in \mathbb{R}^d$. Вона відображає зображення обличчя x у d -вимірний евклідовий простір. Також, відображення повинно знаходитися у d -вимірній сфері [2], тому маємо обмеження: $\|f(x)\|_2 = 1$. Далі необхідно забезпечити, щоб опорне зображення (якір) конкретної людини x_i^a було ближчим до всіх інших x_i^p (позитивних) тієї ж людини, аніж до x_i^n зображень (негативних) будь-якої іншої особи. Маємо,

$$\|x_i^a - x_i^p\|_2^2 + \alpha < \|x_i^a - x_i^n\|_2^2, \forall (x_i^a, x_i^p, x_i^n) \in T, \quad (1)$$

де α - відступ між позитивною і негативною парою, T - множина всіх можливих трійок у тренувальній множині.

Отже, трійкова функція втрат, яку необхідно мінімізувати:

$$L = \sum_i^N [\|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha]_+, \quad (2)$$

Множина всіх можливих трійок буде містити трійки, що задовольняють обмеженню (1). Ці трійки ніяк не поліпшать навчання мережі, тільки сповільнять збіжність. Тому слід вибирати строгі трійки, які будуть брати безпосередню участь у навчанні мережі. Для цього навчальна вибірка розбивається на менші вибірки, з яких вибираються строго позитивні $\operatorname{argmax}_{x_i^p} \|f(x_i^a) - f(x_i^p)\|_2^2$ та строго негативні $\operatorname{argmin}_{x_i^n} \|f(x_i^a) - f(x_i^n)\|_2^2$ зображення.

3. Основні результати і висновки

Трійкова функція втрат сприяє більшому розрізненню зображень конкретної особи з іншими особами за допомогою відступу, а отже, впливає на більшу точність та швидкість навчання нейронної мережі для розпізнавання облич.

Список літератури

1. F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In Proc. CVPR, 2015.
2. K. Q. Weinberger, J. Blitzer, and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. In NIPS. MIT Press, 2006. 2, 3

СКЛАД

Організаційного комітету з проведення на ФІОТ наукової конференції студентів «Інформатика та обчислювальна техніка – ІОТ-2016»

Голова організаційного комітету: О.А. Павлов – декан.

Заст. голови організаційного комітету: О.М. Долголенко.

Співголови організаційного комітету:

Г.М. Луцький – завідувач кафедрою ОТ;

О.М. Жданова – в.о. завідувача кафедрою АСОІУ.

Члени організаційного комітету:

П.І.П.	Посада
Ю.О. Кулаков	Професор кафедри ОТ
В.І. Жабін	Професор кафедри ОТ
В.П. Симоненко	Професор кафедри ОТ
С.Г. Стіренко	Професор кафедри ОТ
О.П. Марковський	Доцент кафедри ОТ
О.В. Гриша	Доцент кафедри АСОІУ
О.В. Гавриленко	Доцент кафедри АСОІУ
Ю.О. Олійник	Старший викладач кафедри АСОІУ
М.О. Сперкач	Старший викладач кафедри АСОІУ
О.А. Халус	Старший викладач кафедри АСОІУ
К.Ю. Ларіна	Старший викладач кафедри АСОІУ

Секретарі конференції:

Н.Є. Куц	Пров. інженер кафедри ОТ
С. П. Якуніна	Інженер кафедри АСОІУ

Публікується у відповідності до розпорядження розпорядження декана факультету ІОТ Павлова О.А. № 11 від «_24_»__03_ 2016 р.

Наукова конференція студентів, магістрантів та аспірантів «Інформатика та обчислювальна техніка – ІОТ-2016». Секція кафедри автоматизованих систем обробки інформації і управління. Матеріали конференції. – Київ. – 2016. – 25-27 квітня 2016р. – 138 с.

У збірник включені тези доповідей, які були представлені на конференції “Інформатика та обчислювальна техніка» – ІОТ-2016” в секції кафедри автоматизованих систем обробки інформації і управління. В доповідях розглянуті наукові та методичні питання щодо сучасних аспектів інформатики та обчислювальної техніки.

Редакційна колегія:

Жданова О.Г., к.т.н., доцент, кафедра АСОІУ НТУУ «КПІ» – співголова

Олійник Ю.О., кафедра АСОІУ НТУУ «КПІ»

Гавриленко О.Г., к.ф.-м.н., кафедра АСОІУ НТУУ «КПІ»