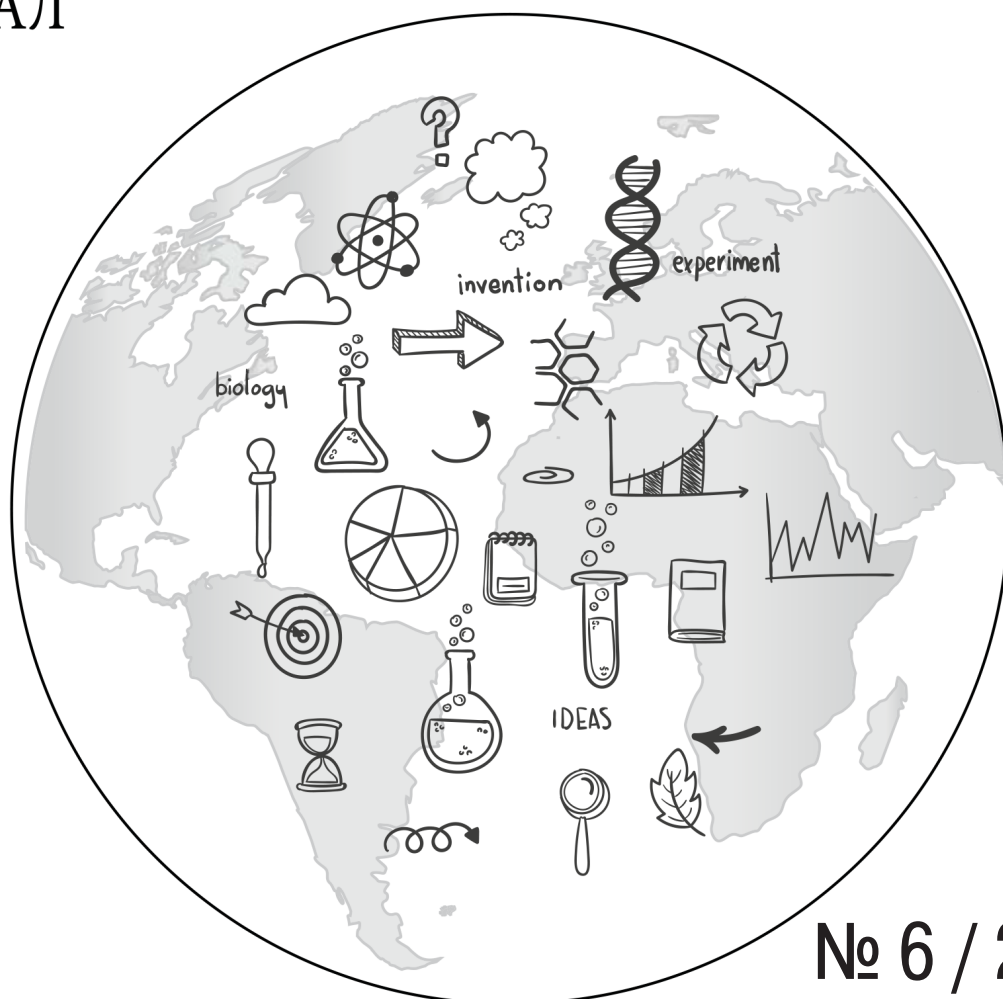


МІЖНАРОДНИЙ НАУКОВИЙ ЖУРНАЛ

INTERNATIONAL
SCIENTIFIC
JOURNAL

МЕЖДУНАРОДНЫЙ
НАУЧНЫЙ
ЖУРНАЛ



№ 6 / 2016

1 ТОМ

МІЖНАРОДНИЙ НАУКОВИЙ ЖУРНАЛ
INTERNATIONAL SCIENTIFIC JOURNAL
МЕЖДУНАРОДНЫЙ НАУЧНЫЙ ЖУРНАЛ

*Свидетельство
о государственной регистрации
печатного средства массовой информации
КВ № 20971-10771Р*

Сборник научных трудов

Выпуск 6

1 том

Киев 2016

ББК 1
УДК 001
М-43

В журнале опубликованы научные статьи по актуальным проблемам современной науки.
Материалы публикуются на языке оригинала в авторской редакции.
Редакция не всегда разделяет мнения и взгляды авторов. Ответственность за достоверность фактов, имен, географических названий, цитат, цифр и других сведений несут авторы публикаций.
При использовании научных идей и материалов этого сборника, ссылки на авторов и издания являются обязательными.

© Авторы статей, 2016
© Международный научный журнал, 2016

Полное библиографическое описание всех статей Международного научного журнала представлено в: НЭБ Elibrary.ru, Polish Scholarly Bibliography.

Журнал зарегистрирован в международных каталогах научных изданий и наукометрических базах данных: РИНЦ; Open Academic Journals Index; ResearchBib; Scientific Indexing Services; Turkish Education Index; Electronic Journals Library; Staats- und Universitätsbibliothek Hamburg Carl von Ossietzky; RePEc; InfoBase Index; International Institute of Organized Research; CiteFactor; Open J-Gate, Cosmos Impact Factor.

Редакционная коллегия

Главный редактор: **Коваленко Дмитрий Иванович** — кандидат экономических наук, доцент

Заместитель главного редактора: **Золковер Андрей Александрович** — кандидат экономических наук, доцент

Заместитель главного редактора: **Безверхий Константин Викторович** — кандидат экономических наук, доцент

Глава редакционной коллегии: **Тарасенко Ирина Алексеевна** — доктор экономических наук, профессор

Член редакционной коллегии: **Чабан Виталий Васильевич** — доктор технических наук, профессор

Член редакционной коллегии: **Румянцев Анатолий Александрович** — доктор технических наук, профессор

Член редакционной коллегии: **Сергейчук Олег Васильевич** — доктор технических наук, профессор

Член редакционной коллегии: **Беликов Анатолий Серафимович** — доктор технических наук, профессор

Член редакционной коллегии: **Мельник Виктория Николаевна** — доктор технических наук, профессор

Член редакционной коллегии: **Наумов Владимир Аркадьевич** — доктор технических наук, профессор (Российская Федерация)

Член редакционной коллегии: **Луценко Игорь Анатольевич** — доктор технических наук, профессор

Член редакционной коллегии: **Пахрутдинов Шукриддин Илесович** — доктор политических наук, профессор (Республика Узбекистан)

Член редакционной коллегии: **Степанов Виктор Юрьевич** — доктор наук по государственному управлению, профессор

Член редакционной коллегии: **Дегтярь Андрей Олегович** — доктор наук по государственному управлению, профессор

Член редакционной коллегии: **Дегтярь Олег Андреевич** — доктор наук по государственному управлению, доцент

Член редакционной коллегии: **Колтун Виктория Семеновна** — доктор наук по государственному управлению, доцент

Член редакционной коллегии: **Щербан Татьяна Дмитриевна** — доктор психологических наук, профессор, Заслуженный работник образования Украины, ректор Мукачевского государственного университета

Член редакционной коллегии: **Цахаева Анжелика Амировна** — доктор психологических наук, профессор (Российская Федерация, Республика Дагестан)

Член редакционной коллегии: **Сунцова Алеся Александровна** — доктор экономических наук, профессор, академик Академии экономических наук Украины

Член редакционной коллегии: **Денисенко Николай Павлович** — доктор экономических наук, профессор, член-корреспондент Международной академии инвестиций и экономики строительства, академик Академии строительства Украины и Украинской технологической академии

Член редакционной коллегии: **Кухленко Олег Васильевич** — доктор экономических наук, профессор

Член редакционной коллегии: **Чубукова Ольга Юрьевна** — доктор экономических наук, профессор

Член редакционной коллегии: **Драган Елена Ивановна** — доктор экономических наук, профессор

Член редакционной коллегии: **Захарин Сергей Владимирович** — доктор экономических наук, старший научный сотрудник, профессор

Член редакционной коллегии: **Лойко Валерия Викторовна** — доктор экономических наук, профессор

Член редакционной коллегии: **Скрипник Маргарита Ивановна** — доктор экономических наук, профессор

Член редакционной коллегии: **Селиверстова Людмила Сергеевна** — доктор экономических наук, профессор

Член редакционной коллегии: **Ефименко Надежда Анатольевна** — доктор экономических наук, профессор

Член редакционной коллегии: **Мигус Ирина Петровна** — доктор экономических наук, профессор

Член редакционной коллегии: **Смолин Игорь Валентинович** — доктор экономических наук, профессор

Член редакционной коллегии: **Шинкарук Лидия Васильевна** — доктор экономических наук, профессор, член-корреспондент Национальной академии наук Украины

Член редакционной коллегии: **Гоблик Владимир Васильевич** — доктор экономических наук, кандидат философских наук, доцент, Заслуженный экономист Украины

Член редакционной коллегии: **Заруцкая Елена Павловна** — доктор экономических наук, профессор

Член редакционной коллегии: **Вдовенко Наталия Михайловна** — доктор экономических наук, профессор

Член редакционной коллегии: **Копилук Оксана Ивановна** — доктор экономических наук, профессор

- Член редакционной коллегии: **Ниценко Виталий Сергеевич** — доктор экономических наук, доцент
- Член редакционной коллегии: **Шевчук Ярослав Васильевич** — доктор экономических наук, старший научный сотрудник, доцент
- Член редакционной коллегии: **Мухсинова Лейла Хасановна** — доктор экономических наук, доцент (Российская Федерация)
- Член редакционной коллегии: **Задерей Петр Васильевич** — доктор физико-математических наук, профессор
- Член редакционной коллегии: **Байчоров Александр Мухтарович** — доктор философских наук, профессор (Республика Беларусь)
- Член редакционной коллегии: **Ильина Антонина Анатольевна** — доктор философских наук, доцент
- Член редакционной коллегии: **Сутужко Валерий Валериевич** — доктор философских наук, доцент (Российская Федерация)
- Член редакционной коллегии: **Стеблук Всеволод Владимирович** — доктор медицинских наук, профессор криминалистики и судебной медицины, Народный Герой Украины, Заслуженный врач Украины
- Член редакционной коллегии: **Щуров Владимир Алексеевич** — доктор медицинских наук, профессор, главный научный сотрудник лаборатории коррекции деформаций и удлинения конечностей (Российская Федерация)
- Член редакционной коллегии: **Свиридов Николай Васильевич** — доктор медицинских наук, главный научный сотрудник отдела эндокринологичной хирургии, руководитель Центра диабетической стопы
- Член редакционной коллегии: **Иоелович Михаил Яковлевич** — доктор химических наук, профессор (Израиль)
- Член редакционной коллегии: **Сопов Александр Валентинович** — доктор исторических наук, профессор (Российская Федерация)
- Член редакционной коллегии: **Свинухов Владимир Геннадьевич** — доктор географических наук, профессор (Российская Федерация)
- Член редакционной коллегии: **Сенотрсова Светлана Валентиновна** — доктор биологических наук, доцент (Российская Федерация)
- Член редакционной коллегии: **Кузава Ирина Борисовна** — доктор педагогических наук, доцент
- Член редакционной коллегии: **Бабина Юлианна Ивановна** — докторантка (Республика Молдова)
- Член редакционной коллегии: **Коньков Георгий Игоревич** — кандидат технических наук, профессор
- Член редакционной коллегии: **Рамский Андрей Юрьевич** — кандидат экономических наук, профессор
- Член редакционной коллегии: **Чаленко Надежда Владимировна** — кандидат экономических наук, доцент
- Член редакционной коллегии: **Бугас Наталия Валериевна** — кандидат экономических наук, доцент
- Член редакционной коллегии: **Русина Юлия Александровна** — кандидат экономических наук, доцент
- Член редакционной коллегии: **Белялов Талат Энверович** — кандидат экономических наук, доцент
- Член редакционной коллегии: **Бадзым Александр Сергеевич** — кандидат экономических наук, доцент
- Член редакционной коллегии: **Квасова Ольга Петровна** — кандидат экономических наук, доцент
- Член редакционной коллегии: **Сибирянская Юлия Владимировна** — кандидат экономических наук, доцент
- Член редакционной коллегии: **Щепанский Эдуард Валерьевич** — кандидат экономических наук, доцент
- Член редакционной коллегии: **Криволапов Василий Сергеевич** — кандидат экономических наук, доцент (Российская Федерация)
- Член редакционной коллегии: **Саньков Петр Николаевич** — кандидат технических наук, доцент
- Член редакционной коллегии: **Артюхов Артем Евгеньевич** — кандидат технических наук, доцент
- Член редакционной коллегии: **Филева-Русева Красимира Георгиева** — кандидат психологических наук, доцент (Республика Болгария)
- Член редакционной коллегии: **Баула Ольга Петровна** — кандидат химических наук, доцент
- Член редакционной коллегии: **Вавилова Елена Васильевна** — кандидат сельскохозяйственных наук, доцент (Российская Федерация)
- Член редакционной коллегии: **Вицентий Александр Владимирович** — кандидат математических наук, доцент (Российская Федерация)
- Член редакционной коллегии: **Мулик Екатерина Витальевна** — кандидат наук по физическому воспитанию и спорту, доцент
- Член редакционной коллегии: **Олейник Анатолий Ефимович** — кандидат юридических наук, профессор
- Член редакционной коллегии: **Химич Ольга Николаевна** — кандидат юридических наук
- Член редакционной коллегии: **Фархитдинова Ольга Михайловна** — кандидат философских наук

CONTENTS**СОДЕРЖАНИЕ****ТЕХНИЧЕСКИЕ НАУКИ**

Дорошенко Юлія Іванівна ДОСЛІДЖЕННЯ ВПЛИВУ ТЕПЛОТИ ЗГОРАННЯ ГАЗУНА ЕКСПЛУАТАЦІЙНІ ПАРАМЕТРИ ГАЗОВИХ МЕРЕЖ	8
Дичко Віктор Миколайович, Ляхевич Сергій Володимирович ОПТИЧНА НАВІГАЦІЙНА СИСТЕМА БПЛА: ПОРІВНЯЛЬНИЙ АНАЛІЗ АЛГОРИТМІВ ЗНАХОДЖЕННЯ КЛЮЧОВИХ ТОЧОК.....	11
Горносталя Олександр Миколайович СИСТЕМА КЛАСТЕРНОГО АНАЛІЗУ ТА ВІЗУАЛІЗАЦІЇ ВЕЛИКИХ ОБСЯГІВ ДАНИХ.....	22
Журавльов Павло Володимирович, Медведський Андрій Миколайович ПРОБЛЕМА ВИБОРУ ТИПУ СХОВИЩА ДАНИХ ПІДПРИЄМСТВА.....	25
Івченко Дмитро Анатолійович МОБІЛЬНІ ДОДАТКИ ДЛЯ АСТМАТИКІВ.....	28
Качаловський Андрій Сергійович МЕТОДИ І МОДЕЛІ ПРОГНОЗУВАННЯ ЛІНІЙНИХ ТА НЕЛІНІЙНИХ НЕСТАЦІОНАРНИХ ПРОЦЕСІВ.....	33
Ковтун Володимир Валентинович СИСТЕМА ЗАХИСТУ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ВІД НЕСАНКЦІОНОВАНОГО ДОСТУПУ ДО КОДУ	37
Кравченко Анатолій Михайлович МОДЕЛІ ТА ОЦІНКИ ПРОГНОЗІВ ПРОЦЕСІВ ЕКОНОМІКИ ПЕРЕХІДНОГО ПЕРІОДУ	40
Кравчук Євгеній Сергійович АНАЛІЗ ПРОТОКОЛІВ ВЗАЄМОДІЇУ КЛІЄНТ-СЕРВЕРНІЙ АРХІТЕКТУРІ.....	44

Лісафін Володимир Петрович ОЦІНЮВАННЯ ВПЛИВУ АТМОСФЕРНИХ ОПАДІВ НА ОБЛІК НАФТИ В РЕЗЕРВУАРАХ З ПЛАВАЮЧОЮ ПОКРІВЛЕЮ	49
Магас Валентин Васильович КЛАСТЕРНИЙ АНАЛІЗ ПРОСТОРОВИХ БАЗ ДАНИХ ОНКОХВОРИХ.....	52
Мельник Вікторія Миколаївна, Карачун Володимир Володимирович, Форостянко Віталій Сергійович ДИСТАНЦІЙНЕ УПРАВЛІННЯ ТЕПЛОМАСООБМІНОМ В БІОРЕАКТОРАХ НА РЕЗОНАНСНОМУ РІВНІ.....	56
Мельник Александр Петрович МОДЕЛИРОВАНИЕ ПРОЦЕССА ГОРЯЧЕГО ПРЕССОВАНИЯ БУМАЖНОГО ПОЛОТНА	62
Метелиця Марія Андріївна МЕТОДИ ФОРМУВАННЯ ТА ОЦІНЮВАННЯ ГРУПОВИХ РЕКОМЕНДАЦІЙ	64
Мислюк Артур Юрійович СИСТЕМИ ДІАГНОСТКИ ЕЛЕКТРОКАРДІОГРАМ ЗА ДОПОМОГОЮ ГЛИБИННИХ НЕЙРОННИХ МЕРЕЖ	68
Исаев Сайфулла Маматович, Туйчиев Бекзод Оромович, Норкобилов Салим Абдухамидович ПРИНЦИПЫ УСТРОЙСТВА И ЛОГИЧЕСКАЯ СТРУКТУРА АВТОМАТИЧЕСКОЙ СИСТЕМЫ ОБНАРУЖЕНИЯ СИГНАЛОВ.....	71
Петрішенко Сергій Олександрович УПРАВЛІННЯ ПАМ'ЯТТЮ В IOS ЗАСТОСУВАННЯХ	75
Петрішенко Сергій Олександрович ПОРІВНЯЛЬНИЙ АНАЛІЗ SWIFT І OBJECTIVE-C	78
Піпко Анна Сергіївна ДОСЛІДЖЕННЯ ВИКОРИСТАННЯ КАСКАДНОГО КЛАСИФІКАТОРА ДЛЯ РОЗПІЗНАВАННЯ ПІШХОДІВ У ВІДЕОПОТОЦІ.....	81
Пирожков Олексій Юрійович ВИКОРИСТАННЯ ТЕХНОЛОГІЇ ВІРТУАЛІЗАЦІЇ ДЛЯ АВТОМАТИЗОВАНОГО ТЕСТУВАННЯ ЗМІШАНИХ КРИПТОГРАФІЧНИХ СИСТЕМ	84
Проценко Михайло Михайлович АНАЛІЗ ФРЕЙМВОРКІВ ЯК ЗАСОБІВ РОЗРОБКИ WEB-ДОДАТКІВ	86
Radkewitsch M., Garirow A. D. PROJEKTIERUNG DER ANLAGE FÜR GEWINNUNG DER ERDÖLPRODUKTENDÄMPFE VON DER TANKSTELLE	90
Рижко Борис Володимирович СИСТЕМА АВТОМАТИЗОВАНОГО ЗБОРУ, ОБРОБКИ ТА АНАЛІЗУ ВЕЛИКИХ ОБСЯГІВ ДАНИХ.....	93
Светлова Олена Костянтинівна ЗАСТОСУВАННЯ КЛІТИННИХ АВТОМАТІВ ДЛЯ ОБРОБКИ ЗОБРАЖЕНЬ.....	96

Светлова Олена Костянтинівна ГЕНЕРАТОР ПСЕВДОВИПАДКОВИХ ЧИСЕЛ ДЛЯ МОДЕЛЮВАННЯ ДОСЛІДУ ШТЕРНА.....	99
Севідов Павло Миколайович ДОСЛІДЖЕННЯ АЛГОРИТМІВ ОПТИМІЗАЦІЇ ІНДИВІДУАЛЬНИХ КАЛЕНДАРНИХ ПЛАНІВ НАВЧАЛЬНОГО ПРОЦЕСУ.....	102
Севідов Павло Миколайович ОПТИМІЗАЦІЯ ЗАПИТІВ ЗА ДОПОМОГОЮ ГЕНЕТИЧНОГО АЛГОРИТМУ	106
Слухай Яна Олександрівна ПРИНЦИПИ ПОБУДОВИ ТА ПОРІВНЯЛЬНИЙ АНАЛІЗ СЕМАНТИЧНИХ РІЗОНЕРІВ	113

Дорошенко Юлія Іванівна*кандидат технічних наук, доцент, доцент кафедри транспорту і зберігання нафти і газу
Івано-Франківський національний технічний університет нафти і газу***Дорошенко Юлія Ивановна***кандидат технических наук, доцент, доцент кафедры транспорта и хранения нефти и газа
Ивано-Франковский национальный технический университет нефти и газа***Doroshenko Juliya Ivanivna***Ph.D., Associate Professor, Department of oil and gas transport and storage
Ivano-Fankivsk National Technical University of Oil and Gas*

ДОСЛІДЖЕННЯ ВПЛИВУ ТЕПЛОТИ ЗГОРАННЯ ГАЗУ НА ЕКСПЛУАТАЦІЙНІ ПАРАМЕТРИ ГАЗОВИХ МЕРЕЖ

STUDY OF IMPACT OF GAS COMBUSTION HEAT ON OPERATING PARAMETERS OF GAS NETWORKS

Анотація. Досліджені теоретичні питання впливу теплоти згорання газу на експлуатаційні параметри газових мереж населених пунктів.

Ключові слова: система опалення, нижча теплота згорання газу, індивідуальна норма витрати газу, коефіцієнт корисної дії котла, вологість.

Аннотация. Исследованы теоретические вопросы влияния теплоты сгорания газа на эксплуатационные параметры газовых сетей населенных пунктов.

Ключевые слова: система отопления, нижняя теплота сгорания газа, индивидуальная норма расхода газа, коэффициент полезного действия котла, влажность.

Summary. Theoretical questions of the impact of gas combustion heat on operating parameters of gas networks of settlements where studied.

Key words: heating system, gas lower calorific value, individual gas flow rate, boiler efficiency, humidity.

Система опалення — це сукупність зв'язаних між собою конструктивних елементів, призначених для одержання, транспортування та передачі необхідної кількості теплоти опалюваному приміщенню.

Будь-яку теплоенергетичну систему з метою аналізу можна умовно розбити на три основні ділянки: ділянка виробництва теплової енергії (котельня); ділянку транспортування теплової енергії споживачу (трубопроводи теплових мереж); ділянка споживання теплової енергії (опалювальний об'єкт). Кожна з наведених ділянок володіє характерними непродуктивними втратами, зниження яких і є основною функцією енергозбереження. Головною ланкою на ділянці виробництва теплової енергії є котлоагрегат, функціями якого є перетворення хімічної енергії палива в теплову і передача цієї енергії теплоносію. У котлоагрегаті відбувається ряд фізико-хімічних процесів, кожен з яких має свій ККД. І будь-який котлоагрегат, яким би досконалим він не був, обов'язково втрачає частину енергії палива в цих процесах.

Розрізняють наступні види споживання газу: в квартирах, в комунальних та суспільних закладах, на опалення і вентиляцію будівель, на промислових підприємствах.

Нормування витрати палива — це встановлення планової міри його виробничого споживання. Метою нормування витрати газу є підвищення ефективності використання газу. Нормуванню підлягає увесь розхід газу котельними незалежно від об'єму його споживання.

Індивідуальна норма витрати газу — питома витрата газу на вироблення 1 Гкал (1 ГДж) тепла котлом даного типорозміру визначається за нормативними характеристиками котла при паспортній теплопродуктивності.

Детальний аналіз літературних джерел стосовно питання дослідження дозволив окреслити наступні чинники впливу: тип об'єкту; геометричні параметри будівлі; температурні параметри приміщення та навколишнього середовища; конструктивні особливості будівлі; функціональне призначення приміщень;

тривалість опалювального сезону; кількість проживаючих в будинку; число точок гарячої води; якість газу.

Більшість цих факторів врахувати із необхідним ступенем точності не вдається, тому норми витрат розробляються на основі дослідних даних по споживанню газу та їх статистичному аналізу. Особливо важко нормувати витрати газу у квартирах, так як він залежить від степені обслуговування міських жителів комунальними послугами. Норми враховують усереднені умови. У будівельних нормах та правилах наводяться норми річної витрати газу для усіх категорій споживачів, які вказуються по витраті тепла у газі віднесеній до 1 людини або умовної одиниці. Тому нормування витрати газу залежить від визначення теплотворної здатності газу. Знаючи витрату тепла визначають витрату газу із врахуванням коефіцієнту корисної дії джерела тепла. При перерахунку на газове паливо необхідно враховувати можливості покращення чи погіршення технології та відповідно скорочення або збільшення витрати газу за рахунок зміни його якості. Для проектних розрахунків необхідно визначити індивідуальну норму витрати палива окремим газовим обладнанням при оптимальних експлуатаційних умовах.

Газ відіграє важливу роль в енергетичному балансі нашої країни. При цьому, від його якості напряму залежить ефективність і надійність роботи енергетичних пристроїв, їх технічні показники та енерговіддача.

Тому загальну тему досліджень можна умовно поділити на декілька частин і провести окремі дослідження по кожному питанню. В даній частині роботи дослідимо вплив теплоти згорання газу (якості газу) на енергоефективність системи газопостачання будинку.

Фізичні властивості газу в значній мірі залежать від температури і тиску, його складу і вологості. Вологість — це один із показників газу, який стараються зменшити усіма способами. Тому тема даного дослідження є актуальною на сьогоднішній день.

Фізичні властивості вологого газу характеризується параметрами його стану, якими є температура t , вологовміст d , абсолютна γ_n і відносна вологість φ , густина ρ , температура точки роси $t_{рос}$, ступінь насичення ψ , парціальний тиск водяної пари p_n , ентальпія I . Для визначення усіх величин, характеризуючи стан вологого газу, як правило, достатньо знати хоча би два параметри.

При використанні газу для опалення необхідно враховувати його характеристики, у тому числі і вологість, яка залежить від температури і тиску.

Підвищена вологість газу негативно впливає на віддачу тепла. Погіршення параметрів, яке виникає під час впливу перепаду температур, призводить до утворення конденсату у трубах. Як наслідок передчасна корозія газопроводу та арматури.

Слід зазначити, що випадання конденсату є значною проблемою, оскільки в результаті постійного впливу вологи в деяких місцях газотранспортної системи буквально за кілька років можуть утворюватися свищі або корозійна розгерметизація. Що в свою чергу, веде до витоку газу. А враховуючи, що велика частина підземних, та й внутрішньобудинкових газових комунікацій недоступна для ретельного огляду, а значить, місця витоків важко діагностувати, боротьба з високою вологістю газу набуває особливого значення.

Другим наслідком підвищення вологості є низька питома теплота згорання, яка впливає на витрату палива і якість обігрівання приміщення.

У газопромисловій практиці, як показник вологості, частіше використовують значення температури точки роси вологи — температури початку конденсації(кристалізації) пари вологи, яка є присутньою в складі газу при визначеному тиску. У нормативних документах, в більшості, регламентується саме значення температури точки роси вологи з огляду на її найбільшу інформативність.

При споживанні природного газу основними показниками його якості є показники його теплотворної здатності, вміст корозійно-активних компонентів та домішок у твердому стані. Крім того рідкі фази води та вуглеводневих фракцій мають бути відсутніми. Відповідність показників якості газу у споживача нормам, можна гарантувати тільки при виконанні вимог, які висуваються щодо газу, який подається у газопроводи для транспортування після його підготовки на промислі.

Особливе значення при підготовці газу приділяється точці роси, тому що в присутності вологи в трубопроводі створюються сприятливі умови і для утворення гідратів, і для корозії обладнання. Через це газ, що подається в газопровід для транспортування, має бути осушеним до такого стану, при якому за будь-яких термогазодинамічних умов в газопроводі на шляху транспортування газу до споживача вологи, яка є присутньою в газі в пароподібному стані, не буде переходити в рідкий стан. Забезпечення цієї умови визначається значенням температури точки роси, яка обумовлює мінімально можливу температуру рівноважного стану «газ-волога». Також важливим показником є температура точки роси вуглеводнів, значення якої показує глибину вилучення важких вуглеводневих фракцій з газу і, при дотриманні її нормованого значення, гарантується транспортування газу без утворення вуглеводневого конденсату. Як правило, нормативні вимоги до якості газу, що подається для транспортування, встановлюються в окремих документах державного рівня.

Виконання вимоги приведення значення температури точки роси до умовного тиску порівняння є

особливо важливим для транспортування, адже значна частка українських родовищ перебуває на завершальній стадії експлуатації і видобування газу проводиться при низьких тисках. При цьому не меншої значимості набуває питання встановлення обґрунтованого значення умовного тиску порівняння, адже встановлення надмірно жорстких вимог до температури точки роси вологи призведе до зайвих затрат на осушування газу. Це питання було опрацьовано при розробці нових технічних умов на газ, що подається з родовищ, газопереробних заводів та підземних сховищ України до магістральних газопроводів, в якому були встановлені вимоги щодо вологості газу.

У роботі були розраховані фізичні властивості для сухого газу, який використовується в системі газопостачання сільського населеного пункту. Також були розраховані фізичні властивості газу з врахуванням поправки на вологість. Ця поправка розраховувалась за двома методиками. Згідно першої методики волога, яка наявна в газі впливає на молярну масу газу, а це в свою чергу на густину, в'язкість, та нижчу теплоту згорання газу. Від нижчої теплоти згорання газу залежить необхідний об'єм постачання газу. Згідно другої методики, вологовміст газу впливає на молярну масу і густину.

Проведено розрахунок зміни властивостей природного газу в залежності від наявної кількості у ньому вологи. Максимальний вміст вологи, який допускається у системах газопостачання становить 200 г/м³. Для розрахунків задавались інтервалом від 1 до 155 г/м³.

Порівнюючи результати отримані при розрахунку сухого і волого газу мною було виявлено, що вологий газ має більшу густину і дещо меншу в'язкість (0,792 кг/м³, 12,91 мм²/с), ніж сухий газ (0,786 кг/м³, 13,05 мм²/с), а також зменшується його питома нижча теплота згорання ($Q_{н.р.} = 36419$ кДж/м³, при 37893 кДж/м³ для сухого). Наслідком таких змін фізичних властивостей є зміна необхідних об'ємів постачання газу, і геометричних параметрів мережі. Розрахунок термодинамічних властивостей газу проводився для сухого і вологого газу. Порівнюючи отримані результати розрахунку можна дійти висновку,

що вологість в газі впливає на два основних показники газової мережі: необхідний об'єм транспортованого газу та на втрати тиску під час транспортування. Так згідно розрахунку годинна витрата сухого газу на ГРП становить 343 м³/год, а для вологого — 356 м³/год.

Наявність у газі вологи негативно впливає на його фізичні властивості: густину, в'язкість, теплотворну здатність. Вона сприяє корозії, та утворенню гідратів, що блокують трубопровід, запірну арматуру, виводять з ладу контрольно-вимірювальні прилади.

Таким чином пропонується ввести коефіцієнт ε_p , який характеризує відхилення норм витрат палива та ККД котла [1, с. 107; 2, с. 282] при роботі в режимі, відмінному від номінального. Реальний коефіцієнт корисної дії котла визначається за формулою

$$\varepsilon_p = \frac{Q_n^{\text{відм}}}{Q_n^{\text{норм}}}, \quad (1)$$

$$\eta_{\text{реал}} = \eta_{\text{пасп}} \cdot \eta_p, \quad (2)$$

де $Q_n^{\text{відм}}$ — нижча теплота згорання із врахуванням відхилення властивостей газу від нормативних параметрів;

$Q_n^{\text{норм}}$ — нижча теплота згорання газу;

$\eta_{\text{пасп}}$ — паспортне значення ККД котла.

Індивідуальну норму витрати газу розраховуємо за формулою

$$H = \frac{34,121}{\eta_{\text{реал}}}, \text{ [кг у.п./ГДж]}. \quad (3)$$

Отримані залежності в розробленій методиці дозволять врахувати показники якості газу при розрахунку індивідуальної норми витрати газу на виробництво тепла при опаленні приміщень, а також при проведенні проектних гідравлічних розрахунків внутрішніх газових мереж, а також для розроблення техніко-економічного обґрунтування інвестиційних проектів технічної модернізації систем та обладнання. Норми витрат палива, визначені згідно з цією методикою, мають сприяти ефективному використанню палива та супроводжуватись планом організаційно-технічних заходів щодо подальшої його економії.

Література

1. Дорошенко Ю.І. Огляд сучасних систем опалення та методик дослідження впливу температурних режимів на ефективність їх експлуатації в умовах сільського населеного пункту / Ю.І. Дорошенко // Розвідка та розробка нафтових і газових родовищ. Івано-Франківський національний технічний університет нафти і газу. — 2014. — № 1 (50). — С. 107–118.
2. Дорошенко Ю.І. Дослідження енергозбереження традиційними методами в умовах населених пунктів / Ю.І. Дорошенко, М.М. Засідко // Прикарпатський вісник наукового товариства ім. Шевченка. — 2016. — № 1 (33). — С. 282–295.

Дичко Віктор Миколайович

студент

Київський Національний Технічний Університет «КПІ»

Ляhevич Сергій Володимирович

аспірант ИПСА

Київський Національний Технічний Університет «КПІ»

Дычко Виктор Николаевич

студент

Киевський Національний Технічний Університет «КПІ»

Ляhevич Сергей Владимирович

аспірант ИПСА

Киевський Національний Технічний Університет «КПІ»

Dychko V.

student

Kyiv National Technical University «KPI»

Liakhevych S.

postgraduate

Kyiv National Technical University «KPI»

**ОПТИЧНА НАВІГАЦІЙНА СИСТЕМА БПЛА: ПОРІВНЯЛЬНИЙ
АНАЛІЗ АЛГОРИТМІВ ЗНАХОДЖЕННЯ КЛЮЧОВИХ ТОЧОК
ОПТИЧЕСКАЯ НАВИГАЦИОННАЯ СИСТЕМА БПЛА: СРАВНИТЕЛЬНЫЙ
АНАЛИЗ АЛГОРИТМОВ НАХОЖДЕНИЯ КЛЮЧЕВЫХ ТОЧЕК
OPTICAL NAVIGATION SYSTEM: KEY-POINTS FINDING
ALGORITHMS COMPARATIVE ANALYSIS**

Анотація. Використання безпілотних літальних апаратів (БПЛА) для проведення аерофотозйомки є надзвичайно актуальним напрямком розвитку методів збору геопросторових даних на сьогодні. Особливо у військовий час – для ведення розвідки супротивника, проте існує можливість створення радіоелектронних заваг, що призводить до перехоплення БПЛА, або його «засліплення». В даній роботі досліджені методи автономної навігації за оптичними орієнтирами.

Ключові слова: БПЛА, автономна навігація.

Аннотация. Использование беспилотных летательных аппаратов (БПЛА) для проведения аэрофотосъемки является чрезвычайно актуальным направлением развития методов сбора геопространственных данных на сегодня. Особенно в военное время – для ведения разведки противника, однако существует возможность создания радиоэлектронных помех, что приводит к перехвату БПЛА, или его «ослепления». В данной работе исследованы методы автономной навигации по оптическим ориентирам.

Ключові слова: БПЛА, автономная навигация.

Abstract. The use of unmanned aerial vehicles (UAVs) for aerial photography is an extremely important area of geospatial data collection methods today. Especially in wartime – for reconnaissance of the enemy, but the creation of electronic interference that results in intercepting or «blinding» the UAV is possible. This paper investigated methods of autonomous navigation optical orientyramy.

Key words: UAV, autonomous navigation.

Вступ

Використання безпілотних літальних апаратів (БПЛА) для проведення аерофотозйомки є надзвичайно актуальним напрямком розвитку методів збору геопросторових даних на сьогодні. В мирні часи цю інформацію можна використовувати для пошуку лісових пожеж, моніторингу стану доріг та трубопроводів, а у військові — для ведення розвідки супротивника. Проте саме у військовий час проведення місій БПЛА обмежено через недоліки системи керування.

БПЛА можуть бути керовані оператором за допомогою пульта, проте такий підхід суттєво звужує радіус їхньої дії. Найбільш поширеними на сьогодні є автономні БПЛА, які при здійсненні своєї місії використовують координати GPS, проте існує можливість створення потенційним супротивником фальшивого сигналу GPS чи радіоелектронних завад, що призводить відповідно до перехоплення БПЛА, або його «засліплення».

Одним з способів вирішення цієї проблеми є використання допоміжних систем, які будуть працювати на фізичних принципах, що відрізняються від принципів роботи основної навігаційної системи. У випадку, коли основна навігаційна система працює на базі сигналів GPS, допоміжною може бути оптична навігаційна система, яка визначає положення шляхом співставлення зображення, отриманого з відеокамери БПЛА з картою, що прошита в Базі знань БПЛА. Проте такий підхід містить ряд технічних труднощів. По-перше поточний ландшафт та той, що міститься на прошитій карті можуть відрізнитися, особливо якщо місія здійснюється в зоні бойових дій — об'єкти, такі як будівлі, дороги можуть бути знищені. По-друге, зазвичай висота, на якій здійснюють свої місії БПЛА та висота на якій здійснюється аерофотозйомка для формування прошитої карти суттєво відрізняються. Отже, необхідно співставляти зображення з урахуванням масштабування, проективного перетворення та часткової зміни об'єктів. Незважаючи на відмінності зображень, деякі частини залишаються незмінними, тому для розпізнавання спільних точок доцільно застосовувати алгоритми, які знаходять ключові точки зображень та їх дескриптори [1–4], за допомогою яких ці точки можна порівнювати.

Мета даної статті полягає в дослідженні та порівняльному аналізі алгоритмів знаходження ключових точок та їхніх дескрипторів з ціллю використання в оптичній системі навігації БПЛА.

1. Алгоритми виявлення ключових точок та співставлення дескрипторів

1.1. SIFT (Scale-invariant feature transform)

Алгоритм SIFT знаходить характеристики зображення, які інваріантні до масштабування зображення

і обертання, та частково інваріантні до змін в освітленні і точки знімання 3D камери [1, 2]. Виявлені характеристики значно відрізняються одна від одної, що дозволяє з високою ймовірністю коректно знайти відповідність між однаковими характеристиками та різних зображеннях, забезпечуючи основу для розпізнавання об'єктів.

Алгоритм включає такі основні етапи:

1. Знаходження екстремумів у просторі масштабів. На першому етапі відбувається пошук по всіх масштабах і розташуваннях на зображенні. Він реалізується, за допомогою функції різниці гаусіанів для виявлення потенційних точок які не залежать від масштабу і орієнтації.

Простір масштабів зображення визначається як функція $L(x, y, \sigma)$, що отримується як згортка гаусіана зі змінним масштабом $G(x, y, \sigma)$ та вхідного зображення $I(x, y)$:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y),$$

де $*$ — операція згортки на x і y , і

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\left(\frac{x^2+y^2}{2\sigma^2}\right)}.$$

Функція згортки різниці гаусіанів із зображенням $D(x, y, \sigma)$, що може бути обчислена як різниця двох сусідніх масштабів, розділених константним множником k :

$$\begin{aligned} D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma) \end{aligned}$$

2. Локалізація ключових точок. Для кожного місцезорозташування точки-кандидата знаходиться детальна модель для визначення розташування і масштабу. Ключові точки вибираються на основі міри їх стійкості.

3. Визначення орієнтації. На основі локальних напрямків градієнта кожній ключовій точці призначаються одна чи декілька орієнтацій. Всі подальші операції виконуються на даних зображення, які були перетворені по відношенню до призначеної орієнтації, масштабу і місцеположення кожної характеристики, забезпечуючи тим самим інваріантність цих перетворень.

Масштаб ключової точки використовується для того, щоб вибрати зображення L , згладжене гаусіаном. Для кожного зразка зображення $L(x, y)$, значення градієнта $m(x, y)$ і орієнтація $\theta(x, y)$ обчислюється як різниця пікселів:

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2}$$

$$\theta(x, y) = \tan^{-1} \left(\frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)} \right).$$

4. Дескриптори ключових точок. В області навколо кожної характерної точки вимірюються локальні градієнти зображення в обраному масштабі. Вони перетворюються в представлення, що дозволяє значні спотворення локальних форм і зміни освітлення [2].

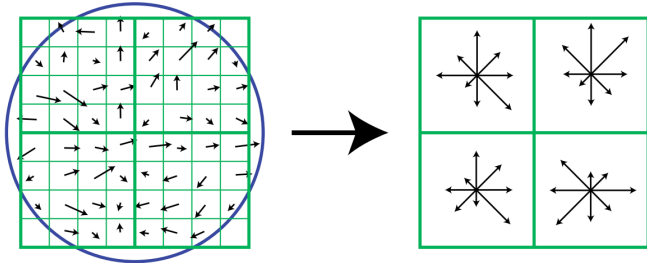


Рисунок 1. Дескриптор ключової точки

1.2. SURF (Speeded Up Robust Features)

Алгоритм SURF базується на таких самих принципах і кроках, як і SIFT, але конкретна реалізація на кожному кроці відрізняється [3]. Алгоритм має три основні частини: виявлення характерних точок, опис і співставлення характеристик локального околу.

Виявлення ключових точок

SURF використовує детектор на основі матриці Гессе для знаходження ключових точок. Визначник матриці Гессе використовується в якості міри локальної зміни навколо точки. Обираються точки, де цей визначник максимальний. SURF також використовує визначник матриці Гессе для вибору масштабу.

Матриця Гессе в точці $p=(x,y)$ зображення I масштабу σ $H(p,\sigma)$ визначається наступним чином:

$$H(p,\sigma) = \begin{pmatrix} L_{xx}(p,\sigma) & L_{xy}(p,\sigma) \\ L_{xy}(p,\sigma) & L_{yy}(p,\sigma) \end{pmatrix},$$

де $L_{xx}(p,\sigma)$ – згортка похідної гаусіана другого порядку $\frac{\partial^2}{\partial x^2}g(\sigma)$ із зображенням I в точці p .

Аналогічно для $L_{xy}(p,\sigma)$ і $L_{yy}(p,\sigma)$.

В алгоритмі використовуються квадратні фільтри, що апроксимують похідні гаусіана другого порядку і можуть бути розраховані дуже швидко за допомогою

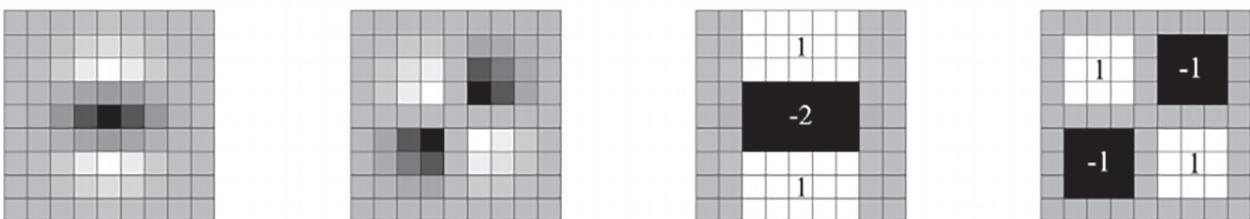


Рисунок 2. Зліва на право: (дискретизовані і обрізані) часткові похідні гаусіана другого порядку в напрямках y , x_y і їх апроксимація за допомогою квадратних фільтрів. Сірі області дорівнюють нулю

інтегрального представлення зображення, незалежно від розміру.

Дескриптори

Дескриптор SURF базується на подібних до SIFT властивостей, але при цьому має ще меншу складність. Перший крок складається з фіксації відтворюваної орієнтації на основі інформації з кругової області навколо ключової точки. Далі будується квадратна область, вирівняна до вибраної орієнтації, з якої отримується SURF-дескриптор.

1.3. ASIFT (Affine-SIFT)

ASIFT імітує з достатньою точністю всі спотворення викликані варіацією напрямлень оптичної осі камери [4]. Потім застосовується метод SIFT. Іншими словами, ASIFT імітує три параметри: масштаб, кут довготи камери і кут широти і нормалізує інші (поворот і зміщення).

ASIFT включає наступні кроки:

1. Кожне зображення перетворюється шляхом імітації всіх можливих афінних перетворень спричиненими зміною орієнтації оптичних осей камери з фронтальної позиції. Ці перетворення залежать від двох параметрів: довготи ϕ і широти θ .

2. Ці повороти і нахили виконуються для кінцевої і невеликої кількості кутів широти і довготи, кроки дискретизації цих параметрів забезпечують, що модельовані зображення ближчі до будь-якого іншого можливого зображення згенерованого за допомогою інших значень ϕ і θ .

3. Усі змодельовані зображення порівнюються за допомогою алгоритму співставлення (SIFT). [4]

2. Алгоритм співставлення зображень

Для співставлення супутникових знімків, що зроблені з різних позицій та мають спотворення перспективи і деякі локальні спотворення, такі як зміна будинків, доріг, дерев (наприклад, у залежності від пори року) та ін. будемо використовувати алгоритми SIFT, SURF, ASIFT як базові.

Алгоритм співставлення знімку та зображення зі супутника складається з наступних етапів.

1. Знаходження ключових точок на обох зображеннях.

Приклад знаходження ключових точок за допомогою алгоритмів SIFT, SURF, ASIFT на повному зображенні, яке необхідно перетворити (роздільна здатність: 500x667 пікселів) представлено на рисунку 3. Під час дослідження використовувалися зображення з роздільною здатністю 1944x2592 із значно більшою кількістю ключових точок.

Дослідження показало, що при класичному застосуванні SIFT, SURF та ASIFT, тобто при спробі знаходження перетворення зображення на основі всіх знайдених точок, результат виходить незадовільний внаслідок великої кількості неправильних співставлень точок. Приклад співставлення усіх точок зображень показано на рисунку 4.

2. Виділення областей відповідностей

Внаслідок помилок співставлення всіх точок, описаних на етапі 1, для подальшої роботи алгоритму запропоновано на другому кроці алгоритму співставлення виділити декілька (2–4) відповідних областей на парі зображень. Приклад виділення областей показано на рисунках 8 та 12.

3. Виключення із подальшого розгляду всіх ключових точок, що не потрапили у відповідні області.

На даному етапі всі ключові точки, що не потрапили у відповідні області виключалися із розгляду, таким чином, зменшуючи кількість неправильних співставлень. Приклад зображень з співставленими ключовими точками на виділених областях отриманими за допомогою алгоритму ASIFT наведено на рисунку 5.

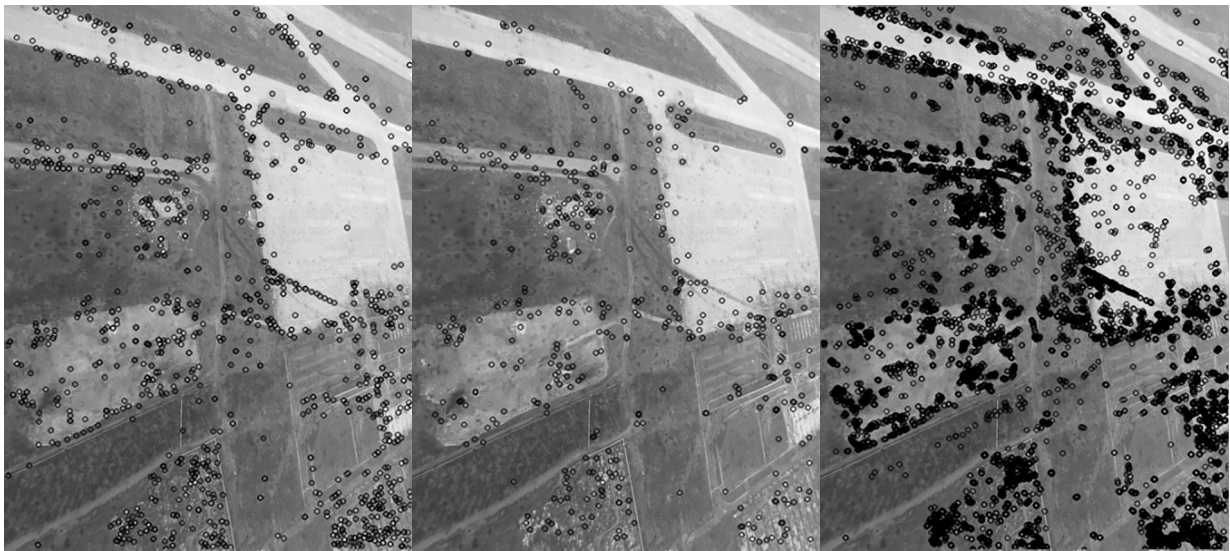


Рисунок 3. Ключові точки (SIFT, SURF, ASIFT)

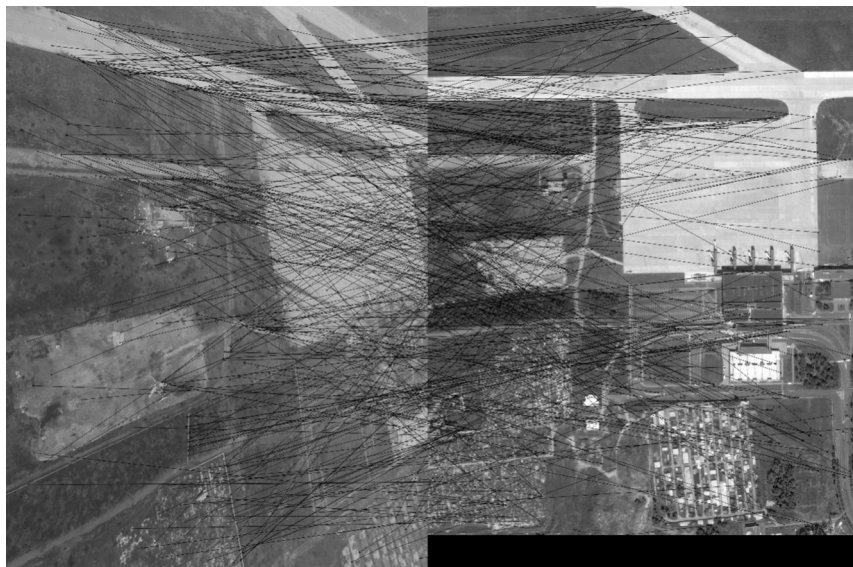


Рисунок 4. Співставлення всіх ключових точок

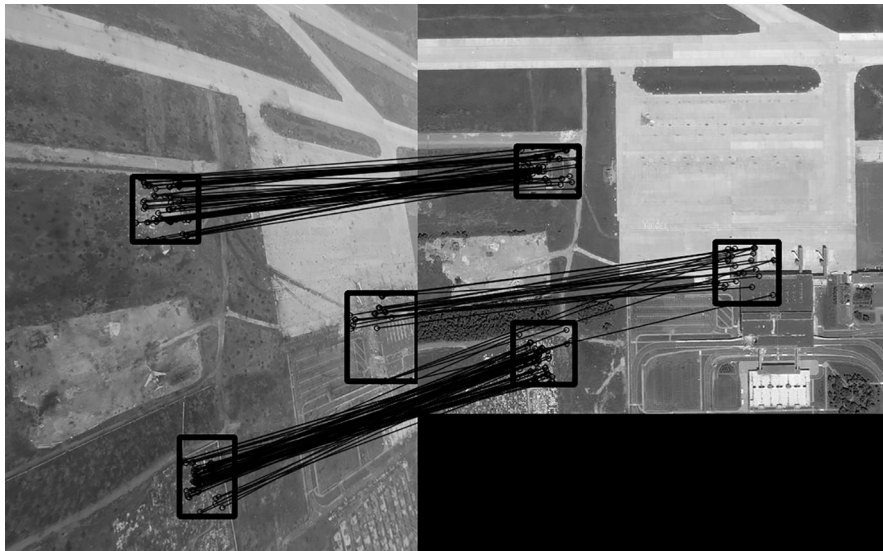


Рисунок 5. Відповідні області та співставлення ключових точок

4. Знаходження проєктивного перетворення (перетворення гомографії) одного зображення до іншого на основі відповідних точок обох зображень.

Відображення точок двовимірної плоскої поверхні на зображення камери є прикладом гомографії. Це відображення можна виразити в термінах множення матриць при використанні однорідних координат, щоб виразити спостережувану точку Q і точку на зображенні q , у яку Q відображається. Гомографія може бути виражена як $\tilde{q} = sH\tilde{Q}$, де $\tilde{Q} = [X \ Y \ Z \ 1]^T$, $\tilde{q} = [x \ y \ 1]^T$, де s – масштабний коефіцієнт.

Приклад знаходження перетворення гомографії для зображення знімку представлено на рисунку 6 (посередині).

5. Накладання зображень.

Результуюче зображення отримується як зважена сума отриманого після перетворення гомографії та зображення зі супутника. Приклад представлено на рисунку 6.

Процедура визначення якості перетворення зображення складається з наступних кроків:

1. Загальна перевірка коректності перетворення.

На цьому етапі перевіряється наявність дзеркального відбиття для пар вершин зображення по горизонталі та вертикалі. При його наявності, якість перетворення зображення встановлюється рівною нулю. В іншому випадку якість розраховується, як описано нижче (етап 2, 3). Приклад некоректного перетворення з нульовою якістю наведено на рисунках 10 та 14 (перші зображення).

2. Виділення спільної прямокутної області.

Після перетворення гомографії змінюється форма зображення. Тому, для порівняння потенційно подібних частин зображень, на перетвореному зображенні виділялась прямокутна область. Приклад виділення такої області наведено на рисунку 7.



Рисунок 6. Зліва на право – оригінальне зображення, перетворене та результат накладання

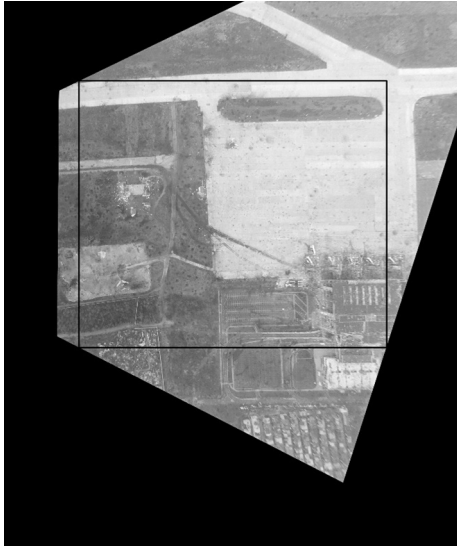


Рисунок 7. Виділення внутрішньої прямокутної області

3. Знаходження значення SSIM (Structure similarity).

Якість коректного перетворення (ступінь подібності перетвореного зображення до зображення зі супутника) вимірювалась за допомогою SSIM.

Відмінною особливістю індексу структурної подібності (SSIM) є те, що метод враховує «сприйняття помилки» завдяки врахуванню структурної зміни інформації. Ідея полягає в тому, що пікселі мають сильний взаємозв'язок, особливо коли вони близькі просторово. Дані залежності містять важливу інформацію про структуру об'єктів і про зображення загалом. Метрика SSIM розрахована на різні розміри вікна. Різниця між двома вікнами x і y , що мають однаковий розмір $N \times N$:

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

де

- μ_x — середнє x ;
- μ_y — середнє y ;
- σ_x^2 — дисперсія x ;
- σ_y^2 — дисперсія y ;
- σ_{xy} — коваріація x і y ;

$$c_1 = (k_1L)^2, c_2 = (k_2L)^2$$

L — динамічний діапазон пікселів (зазвичай $2^{\text{кількість бітів на піксель}} - 1$)

$k_1 = 0,01, k_2 = 0,03$ — константи. [5]

Приклади обчислення значень SSIM для різних зображень представлено на рисунках 10, 11, 14, 15.

3. Аналіз результатів дослідження порівняння алгоритмів

В якості вибірки для порівняльного аналізу використовувалися зображення, зняті з БПЛА в зоні донецького аеропорту, в якості карти використовувалася карта Google. При проведенні дослідження, на кожному зображенні виділялись відповідні області, які потім подавались на вхід вищеописаному алгоритму для перетворення зображення.

Для кожного експерименту та кожного набору областей застосовувалося зміщення кожної області (на 20% від розміру області) для кожного з чотирьох діагональних напрямків. Зміщення застосовувалося до областей зображення, яке необхідно перетворити; області зображення супутника залишалися незмінними. Середня якість чутливості — середня якість для перетворених зображень, для областей яких застосовувалося таке зміщення.

Також, експерименти проводилися для збільшених розмірів областей на 75%. Результати представлені в стовпці «Середня якість зі збільшеними областями».

В таблицях 1 та 3 наведено час виявлення ключових точок для різних експериментів, що є важливим при побудові систем реального часу. В таблицях 2 та 4 представлені дані щодо якості накладання зображень в залежності від обраного алгоритму та кількості областей співставлення.

Експеримент № 1

Розглянемо приклад пари зображень із двома виділеними областями наведений на Рисунку 8.

Таблиця 1

Час виявлення ключових точок

Алгоритм	Час виявлення ключових точок (сек.)	Час співставлення дескрипторів (сек.)
SIFT	12,12	2,915
SURF	4,183	1,406
ASIFT	252,762	21,205

З графіків представлених на рисунку 9 можна побачити, що найкращу якість дає алгоритм ASIFT, проте при 4 областях його результати та результати SURF близькі.

На Рисунках 10 та 11 наведено приклади перетворення зображень із різною якістю.



Рисунок 8. Зображення з виділеними відповідними областями

Таблиця 2

Якість накладання зображень

Алгоритм	Кількість областей	Кількість експериментів	Середня якість	Максимальна якість	Відсоток з перетворень з ненульовою якістю	Середня якість чутливості	Середня якість зі збільшеними областями
SIFT	2	10	0,00	0,00	0,00	0,00	0,05
	3	10	0,17	0,59	0,40	0,02	0,06
	4	5	0,00	0,00	0,00	0,08	0,22
SURF	2	10	0,05	0,49	0,10	0,00	0,00
	3	10	0,28	0,58	0,50	0,04	0,12
	4	5	0,57	0,58	1,00	0,16	0,43
ASIFT	2	10	0,34	0,61	0,70	0,29	0,37
	3	10	0,50	0,58	0,90	0,46	0,44
	4	5	0,57	0,58	1,00	0,56	0,57

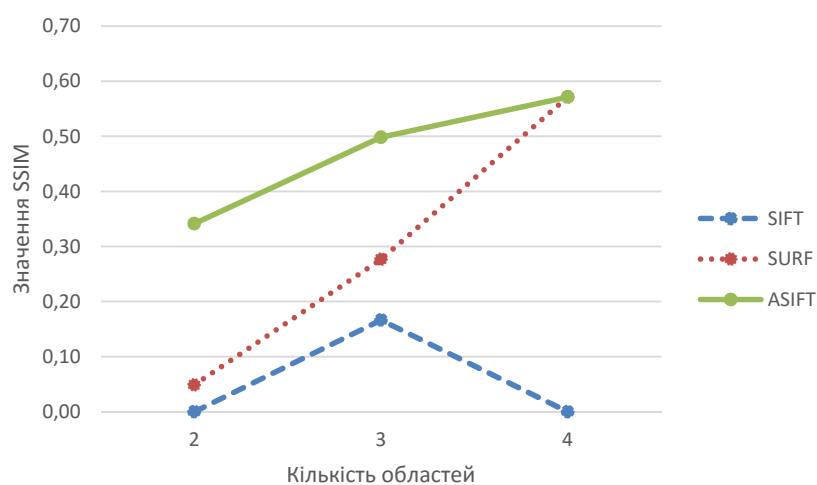


Рисунок 9. Порівняння якості співставлення зображень (рисунок 8) різних алгоритмів виділення ключових точок з різною кількістю виділених областей

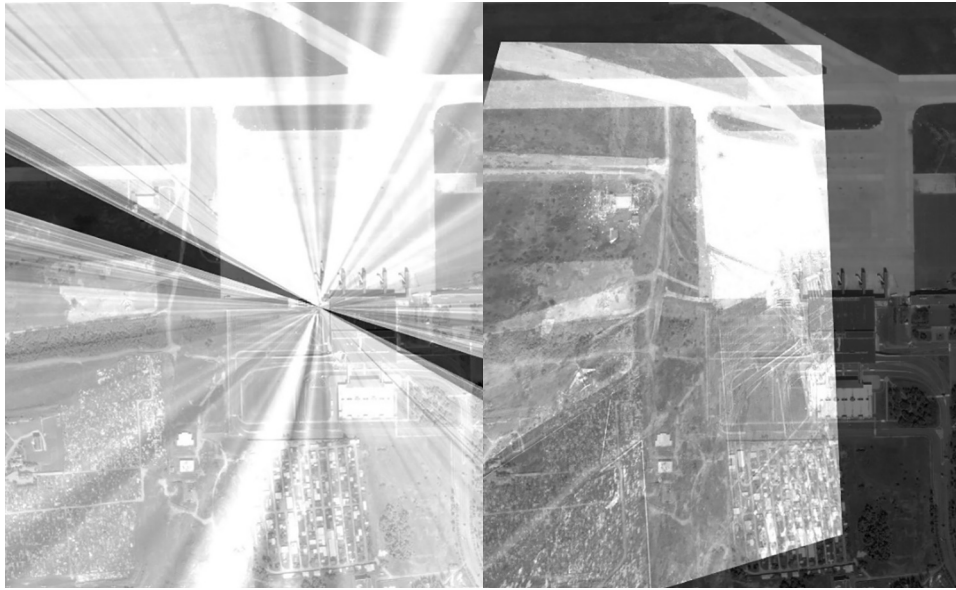


Рисунок 10. Приклади перетворення зображень з різною якістю. Зліва на право: 0; 0,47



Рисунок 11. Приклади перетворення зображень з різною якістю. Зліва на право: 0,50; 0,57

Експеримент № 2

Дослідимо ще один приклад пари зображень із двома виділеними областями наведених на Рисунок 12.

Таблиця 3

Час виявлення ключових точок

Алгоритм	Час виявлення ключових точок (сек.)	Час співставлення дескрипторів (сек.)
SURF	4,144	0,964
ASIFT	271,837	47,62

Результати для SIFT вийшли нульові і не були внесені в таблицю 4. Зважаючи на якість роботи SURF, на рисунку 13 наведено порівняння якості співставлення в залежності від кількості виділених областей лише для ASIFT.

На Рисунках 14 та 15 наведено приклади перетворення зображень із різною якістю.

Отже, за результатами досліджень найкраща якість досягається при використанні алгоритму знаходження ключових точок ASIFT та при кількості виділених відповідних областей, що дорівнює 4. При цьому не спостерігається значного погіршення якості навіть для зображень з поворотом відносно зображення із супутника. Недоліком використання алгоритму ASIFT є його швидкість, використовувати його в системах реального часу не є доцільним. Пропонується використовувати ASIFT для співставлення карти з певним базовим зображенням з БПЛА, а з інших фото з БПЛА створювати панораму за допомогою SURF.



Рисунок 12 – Зображення з виділеними відповідними областями

Таблиця 4

Якість накладання зображень

Алгоритм	Кількість областей	Кількість експериментів	Середня якість	Максимальна якість	Відсоток з перетворень з ненульовою якістю	Середня якість чутливості	Середня якість зі збільшеними областями
SURF	2	10	0,00	0,00	0,00	0,00	0,04
	3	10	0,00	0,00	0,00	0,03	0,21
	4	5	0,00	0,00	0,00	0,10	0,23
ASIFT	2	10	0,30	0,63	0,50	0,23	0,18
	3	10	0,56	0,66	0,90	0,55	0,64
	4	5	0,65	0,66	1,00	0,64	0,52

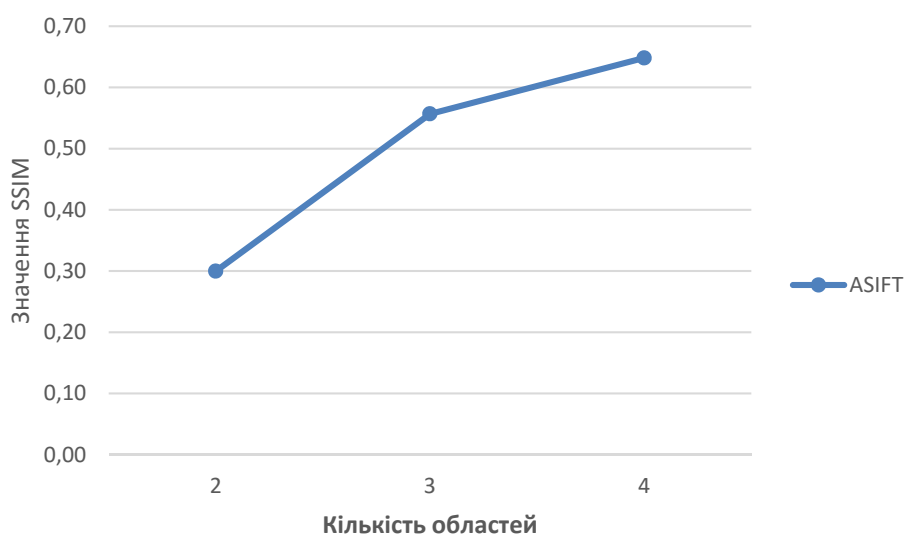


Рисунок 13. Порівняння якості співставлення зображень (рисунок 12) з різною кількістю виділених областей



Рисунок 14. Приклади перетворення зображень з різною якістю. Зліва на право: 0; 0,53



Рисунок 15. Приклади перетворення зображень з різною якістю. Зліва на право: 0,63; 0,66

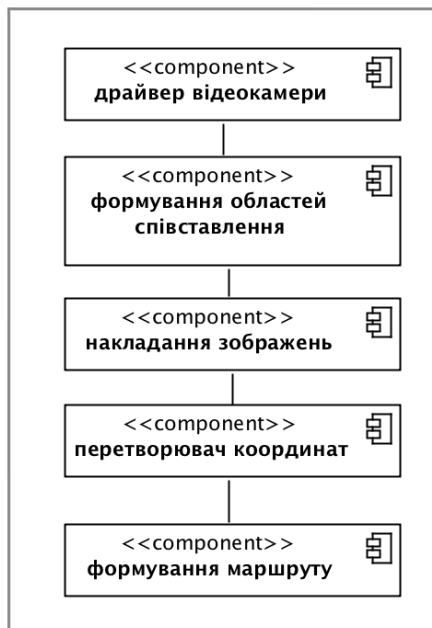


Рисунок 16. Основні компоненти оптичної навігаційної системи БПЛА

Розглянемо структуру оптичної навігаційної системи, побудованої на основі запропонованого алгоритму (Рисунок 16).

4. Оптична навігаційна система

За допомогою драйвера відеокамери система отримує базове зображення. Оператор виділяє на зображенні та карті 4 області співставлення. Базове зображення та карта накладаються за допомогою ASIFT. Нові зображення з БПЛА зшиваються між собою (SURF) та на основі перетворення, отриманого з накладання карти та базового зображення співставляються з картою з метою отримання координат. Маючи координати, система може сформувати маршрут руху.

Висновки

В ході роботи було запропоновано структуру оптичної системи навігації БПЛА та алгоритм співставлення зображень який може бути в ній використаний. Запропонована система дозволяє вирішити

проблеми навігації на основі пульта чи GPS і забезпечує можливість БПЛА здійснити свою місію як в мирний так і військовий час.

При аналізі запропонованого алгоритму співставлення знімків з БПЛА та супутникових зображень було визначено, що найкраща якість досягається при використанні алгоритму знаходження ключових точок ASIFT для базового зображення та при кількості виділених відповідних областей, що дорівнює чоти-

рьом. При цьому не спостерігається значного погіршення якості навіть для зображень з поворотом відносно зйомки із супутника.

Подальші дослідження плануються щодо автоматизації вибору областей співставлення, модифікації алгоритму для підвищення якості перетворення зображень та застосування комбінацій алгоритмів для підвищення швидкості роботи.

Список літератури

1. Lowe D. G. Object Recognition from Local Scale-Invariant Features / D. G. Lowe // International conference on Computer Vision. — 1999. — P. 1150–1157.
2. Lowe D. G. Distinctive Image Features from Scale-Invariant Keypoints / D. G. Lowe // International Journal of Computer Vision. — 2004. — Vol. 60, № 2. — P. 91–110.
3. Bay H. Surf: Speeded up robust features / H. Bay, T. Tuytelaars, L. Van Gool // European Conference on Computer Vision. — 2006. — P. 1:404–417.
4. Morel J. ASIFT: A New Framework for Fully Affine Invariant Image Comparison / J. Morel, G. Yu // SIAM Journal on Imaging Sciences — 2009. — Vol. 2, № 2. — P. 438–469.
5. Wang Z. Image Quality Assessment: From Error Measurement to Structural Similarity / Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli // IEEE Transactions on image processing. — 2004. — Vol. 13, № 1. — P. 600–612.

Горносталь Олександр Миколайович

студент

НТУУ «Київський політехнічний інститут»

Горносталь Александр Николаевич

студент

НТУУ «Киевский политехнический институт»

Hornostal O.

student

NTUU «Kyiv Polytechnic Institute»

**СИСТЕМА КЛАСТЕРНОГО АНАЛІЗУ ТА ВІЗУАЛІЗАЦІЇ
ВЕЛИКИХ ОБСЯГІВ ДАНИХ
СИСТЕМА КЛАСТЕРНОГО АНАЛИЗА И ВИЗУАЛИЗАЦИИ
БОЛЬШИХ ОБЪЕМОВ ДАННЫХ
SYSTEM OF CLUSTER ANALYSIS AND
VISUALIZATION OF BIG DATA**

Анотація. В роботі проведено огляд існуючих систем кластерного аналізу та візуалізації великих обсягів даних. Після дослідження та проектування було розроблено систему, призначену для: завантаження, нормалізації та аналізу вхідних даних; здійснення кластерного аналізу різними алгоритмами; візуалізації та збереження результатів проведеного кластерного аналізу.

Ключові слова: кластерний аналіз, аналіз даних, big data, кластер, міра відстані.

Аннотация. В работе проведен осмотр существующих решений систем кластерного анализа и визуализации больших объемов данных. После исследования и проектирования была разработана система, предназначенная для: загрузки, нормализации и анализа входных данных; проведения кластерного анализа разными алгоритмами; визуализация и сохранение результатов проведенного кластерного анализа.

Ключевые слова: кластерный анализ, анализ данных, big data, кластер, мера расстояния.

Summary. In this work an overview of existed systems of cluster analysis and data visualization was made. After analysis and projecting was made system with such functionality: uploading, normalizing and analysis of input data; cluster analysis using various types of algorithms; visualization and saving results of cluster analysis.

Key words: cluster analysis, data analysis, big data, cluster, distance measure.

Вступ

Останнім часом інформація, що зростає в колосальних обсягах, породжує необхідність опрацювання великих обсягів даних. В цьому напрямку своє місце відведено інтелектуальному аналізу даних. Даний напрямок включає в кластерний аналіз та методи, основані на моделюванні, ймовірнісних узагальненнях, асоціюванні та пошуках закономірностей. Кластерний аналіз або автоматичне групування об'єктів є частковим випадком такого аналізу. В великій мірі розвитку цієї дисципліни сприяло проникнення в сферу аналізу даних ідей, що виникли в теорії штучного інтелекту.

Головним завданням кластерного аналізу є виділення необхідної кількості груп об'єктів, що об'єднані за певними критеріями між собою всередині групи і максимальньо відрізняються від екземплярів інших груп.

Аналіз математичного та алгоритмічного забезпечення систем кластерного аналізу

Кластерний аналіз групує об'єкти даних базуючись тільки на інформації, знайденої в даних, що описують об'єкти та їх взаємозв'язки. Мета в тому, щоб об'єкти в групі були схожими один до одного, але відрізнялися від об'єктів інших груп. Чим більша схожість об'єктів в групі, тим більше відрізняються групи, тим краща кластеризація [1].

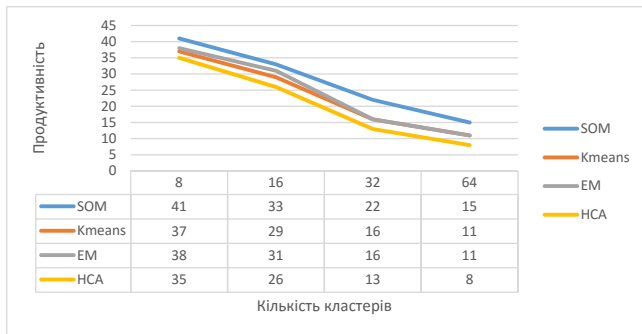


Рис. 1. Графік залежності продуктивності алгоритмів в залежності від кількості кластерів (розробка автора)

В роботі було обрано три прості, але важливі техніки для представлення багатьох концепцій в кластерному аналізі:

- к-середніх;
- агломеративна ієрархічна кластеризація;
- DBSCAN.

Проведено порівняння залежності продуктивності алгоритмів в залежності від кількості кластерів на наступних алгоритмах (рис. 1): к-середніх; алгоритм ієрархічної кластеризації; SOM (Self-Organization Map); EM (Expectation Maximization) [3].

Зі зростанням кількості кластерів продуктивність SOM алгоритму падає. Продуктивність EM та К-середніх стає кращою ніж в ієрархічного алгоритму.

Кількість кластерів впливає також на якість кластеризації, тобто достовірність отриманих результатів.

Структура системи

Система кластерного аналізу складається з чотирьох основних блоків: блок аналізу, блок кластеризації, блок оцінки якості та інтерпретації кластерного рішення, засоби візуалізації [4]. Структура системи приведена на рисунку 2.

Блок аналізу необхідний для виконання попереднього аналізу вхідних даних. Блок кластеризації використовується для безпосереднього розподілу об'єктів до кластерів. Планується реалізація різних типів алгоритмів кластерного аналізу даних. Засіб оцінки якості кластеризації призначений для оцінки степеня достовірності кластерних рішень, на основі яких будуть видані відповідні рекомендації. Засіб візуалізації представляє користувачу можливість взаємодії з системою.



Рис. 2. Структура системи (розробка автора)

Модель розробленого програмного забезпечення

Виходячи із побудованої концептуальної моделі була спроектована діаграма класів, що зображена на рисунку 3.

Діаграма класів представляє логічне моделювання системи.

Результати роботи

Для демонстрації роботи програми застосуємо реалізований алгоритм кластеризації до набору даних, що можуть бути як згенеровані додатком автоматично, так і завантажені користувачем з файлу.

Набір даних містить кластери не сферичної форми з наявністю шумів в вигляді поодиноких точок. Такий набір кластерів найкраще виявляється алгоритмом DBSCAN при параметрах MinPts = 10, Eps = 4 (Рис. 4).

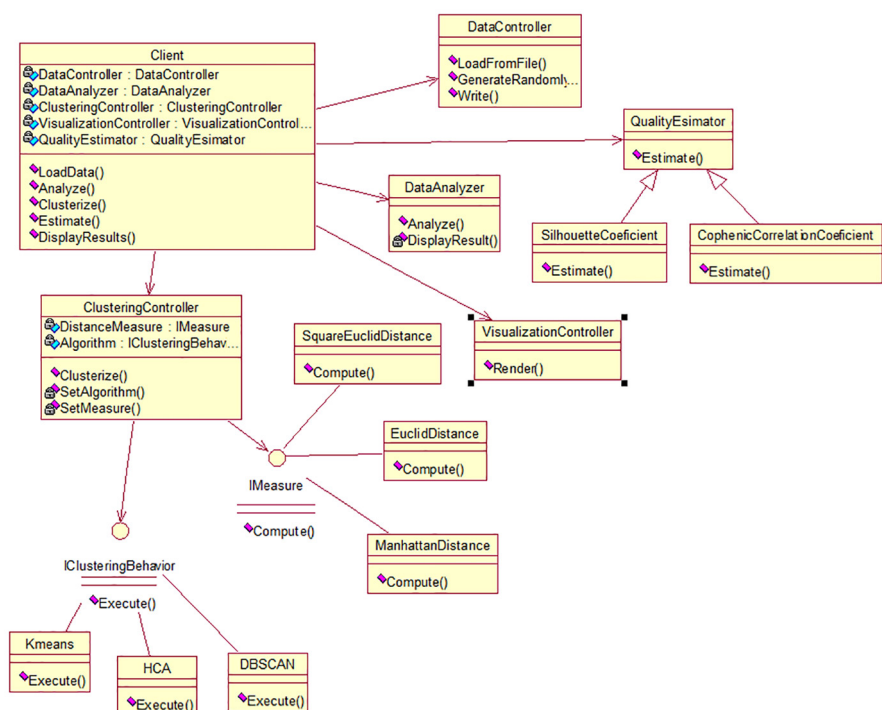


Рис. 3. Діаграма класів системи (розробка автора)

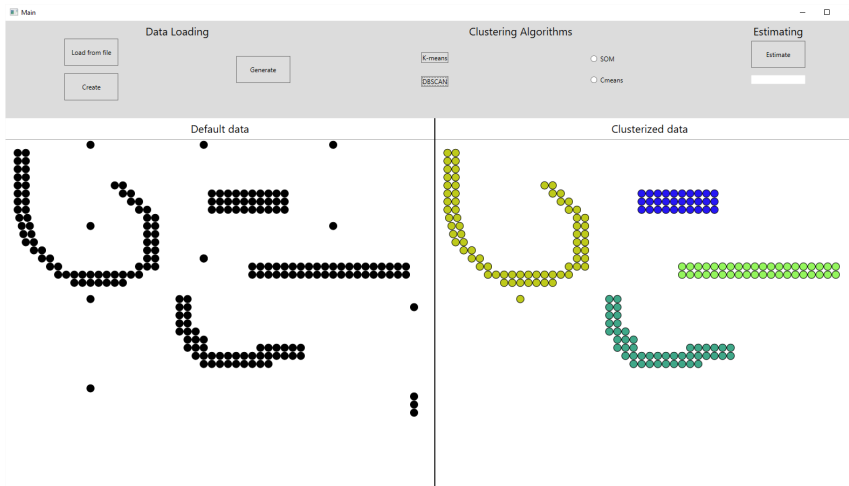


Рис. 4. Результаты работы алгоритма DBSCAN

Висновки

Було виконано дослідження предметної області, аналіз теоретичних засад та математичних методів кластерного аналізу, проектування, реалізацію та тестування програмного додатку системи кластерного

аналізу та візуалізації великих обсягів даних.

З метою можливості подальшої підтримки програмного додатку код був написаний у відповідності до сучасних технологічних рішень, таких як паттерни проектування та з дотриманням SOLID принципів.

Розроблена система кластерного аналізу та візуалізації даних виконує поставлені перед нею задачі: аналіз даних, виконання алгоритмів кластеризації та візуалізація отриманих результатів.

Наступними напрямками розвитку системи можна визначити: реалізація більшої кількості алгоритмів кластеризації, розширення функціоналу для роботи не тільки з числовими, а й з категоріальними даними, покращення продуктивності при виконанні візуалізації.

Література

1. A. K. Jain R. Algorithms for Clustering Data / R. C. Dubes A. K. Jain. — New Jersey: Prentice Hall, 1988. — 334 с.
2. Нейский И. М. Классификация и сравнение методов кластеризации [Электронный ресурс] / Нейский И. М. — Режим доступа до ресурсу: http://it-claim.ru/Persons/Neyskiy/Article2_Neyskiy.pdf (дата звернення 18.05.2016). — Назва з екрану.
3. L. Kaufman. Finding Groups in Data: An Introduction to Cluster Analysis. Wiley Series in Probability and Statistics / L. Kaufman, P. J. Rousseeuw. — New York: John Wiley and Sons, 1990.
4. Програмная система кластерного анализа данных смешанного типа [Электронный ресурс] — Режим доступа до ресурсу: <http://www.jurnal.nips.ru/sites/default/files/Paper-2013-1-11.pdf> (дата звернення 30.05.2016). — Назва з екрану.

Журавльов Павло Володимирович

студент

Національний технічний університет України «Київський політехнічний інститут»

Медведський Андрій Миколайович

студент

Національний технічний університет України «Київський політехнічний інститут»

Журавлев Павел Владимирович

студент

Национальный технический университет Украины «Киевский политехнический институт»

Медведский Андрей Николаевич

студент

Национальный технический университет Украины «Киевский политехнический институт»

Zhurzvlyov P.

student

National Technical University of Ukraine «Kyiv Polytechnic Institute»

Medvedskyi A.

student

National Technical University of Ukraine «Kyiv Polytechnic Institute»

ПРОБЛЕМА ВИБОРУ ТИПУ СХОВИЩА ДАНИХ ПІДПРИЄМСТВА

ПРОБЛЕМА ВЫБОРА ТИПА ХРАНИЛИЩА ДАННЫХ ПРЕДПРИЯТИЯ

CHOOSING THE RIGHT TYPE OF ENTERPRISE STORAGE ARCHITECTURE

Анотація. Досліджені різні типи файлових сховищ, їх проблеми та різниця між ними. Також сформована відповідність між вимогами до сховища та сховищами, що їх задовільняють.

Ключові слова: сховище даних, DAS, NAS, SAN, об'єктне сховище.

Аннотация: Исследованы разные типы файловых хранилищ, их проблемы и разница между ними. Так же сформировано соответствие между требованиями к хранилищам и хранилищами, которые их удовлетворяют.

Ключевые слова: хранилище данных, DAS, SAN, SAN, объектное хранилище.

Summary. Different types of file storage, their problems and the difference between them studied. Also storage requirements and storage types use cases studied.

Key Words: data storage, object storage, DAS, SAN, SAN.

Закон Мура свідчить, що «за всю історію обчислювальної техніки, кількість транзисторів у щільній інтегральній схемі подвоювалась приблизно кожні два роки». В той час як ця думка, головним чином, стосується обчислювальної потужності комп'ютерів, цю ж закономірність можна побачити у світлі росту та зберігання даних. Об'єм даних, що генеруються сьогодні величезний. До 2020 р. він, як очікується, досягне 40000 екзабайт (EB), що більше, ніж 5,2 ТБ даних на одну людину на планеті.

Діяльність сучасних компаній заснована на роботі з даними. Користувацькі дані збираються, використовуються та зберігаються у великих кількостях. При

такій великій кількості даних стає складно підтримувати їх на стандартних засобах зберігання. Більшість даних існує в публічних та внутрішніх «хмарах», віддалених сховищах, організованих з використанням технологій віртуалізації.

Оскільки сховище даних повинно справлятися зі зростом кількості різноманітних даних, виникає задача порівняння та вибору серед різних архітектур сховищ з урахуванням їх переваг та потреб підприємства. Існує чотири основних типи сховищ:

- DAS (англ. Direct-attached storage) — система зберігання даних з прямим підключенням;

- NAS (англ. Network Attached Storage) — мережеве зберігання даних;
- SAN (англ. Storage Area Network) — мережа зберігання даних;
- Об'єктне сховище.

Direct Attached Storage

Direct Attached Storage — цифрове сховище, яке безпосередньо підключено до сервера, який отримує доступ до пам'яті, на противагу віддаленого доступу через мережу. Найбільш поширені типи пристроїв DAS — внутрішні / зовнішні жорсткі диски і оптичні приводи дисків (наприклад, CD / DVD-ROM). Сховище DAS — це фізичний диск, який може бути безпосередньо підключеним до сервера всередині або зовні, так само, як зовнішній диск для ПК.

До основних переваг DAS-систем можна віднести їх низьку вартість (у порівнянні з іншими рішеннями СЗД), простоту розгортання і адміністрування, а також високу швидкість обміну даними між системою зберігання і сервером. Власне, саме з цієї причини вони стали дуже популярні в сегменті малих офісів і невеликих корпоративних мереж. У той же час DAS-системи мають і свої недоліки — в першу чергу це висока вартість зберігання і управління даними внаслідок їх розкиданості по організації, а також вимушений простий мережі в момент додавання нових дисків і необхідність нарощування пам'яті або процесорної потужності сервера при перевищенні певного розміру дискового простору. Перевантаженість мережевого трафіку з додаванням нових серверів ускладнює проблему захисту даних, перешкоджає ефективному використанню ресурсів і т.д.

Хоча деякі DAS системи можуть бути розширені, вони мають більш низьку межу масштабованості, ніж інші типи пристроїв.

Network Attached Storage

Network Attached Storage — Зовнішній пристрій, який фізично з'єднаний з серверним оточенням по LAN або WAN мережі. На відміну від DAS пристроїв, які підключаються безпосередньо до сервера, NAS пристрої підключаються через існуючу мережу. Типові пристрої NAS є повністю автономними, і поставляються з власною операційною системою і інтерфейсом адміністрування. Крім оптимізованої ОС, звільненої від всіх функцій, не пов'язаних з обслуговуванням файлової системи і реалізацією введення-виведення даних, NAS-системи мають оптимізовану по швидкості доступу файловою системою. NAS-системи проєктуються таким чином, що вся їх обчислювальна потужність фокусується виключно на операціях обслуговування і зберігання файлів. Сама операційна система розташовується у флеш-пам'яті та встанов-

люється фірмою-виробником. У порівнянні з традиційними файловими серверами, NAS-пристрої є більш продуктивними і менш дорогими. В даний час практично всі NAS-пристрої орієнтовані на використання в мережах Ethernet (Fast Ethernet, Gigabit Ethernet) на основі протоколів TCP / IP. Доступ до пристроїв NAS здійснюється за допомогою спеціальних протоколів доступу до файлів. Найбільш поширеними протоколами файлового доступу є протоколи CIFS та NFS.

Так як пристрої NAS під'єднуються через мережу, вони не залежать від працездатності серверів, як DAS сховища.

Пристрої NAS ідеально пасують для людей чи компаній, що прагнуть обмінюватися файлами через мережу. Операції передачі даних можуть оброблюватися декількома серверами одночасно з великою кількістю з'єднань.

Простіший та зручніший метод доступу та керування, як наслідок, наявність своєї операційної системи та графічного інтерфейсу користувача.

Серед недоліків можна вказати відсутність додаткових функцій серверів NAS та залежність продуктивності пристроїв NAS від пропускну здатності мережі.

Storage Area Network

SAN — це спеціалізована мережева інфраструктура для зберігання даних (мережа зберігання даних). Ці мережі інтегруються у вигляді окремих спеціалізованих підмереж до складу локальної (LAN) або глобальної (WAN) мережі. По суті, SAN-мережі пов'язують один або кілька серверів (SAN-серверів) з одним або декількома пристроями зберігання даних. SAN-мережі дозволяють будь-якому SAN-сервера отримувати доступ до будь-якого пристрою зберігання даних, не завантажуючи при цьому ні інші сервери, ні локальну мережу. Крім того, можливий обмін даними між пристроями зберігання даних без участі серверів. SAN-мережі дозволяють дуже великому числу користувачів зберігати інформацію в одному місці (з швидким централізованим доступом) і спільно використовувати її. Як пристрої зберігання даних можуть застосовуватися RAID-масиви, різні бібліотеки (стрічкові, магнітооптичні і ін.), А також JBOD-системи (масиви дисків, не об'єднані в RAID). Для побудови мереж SAN використовується або стандарт Fibre Channel (FC), або стандарт iSCSI.

Основними перевагами SAN систем є висока продуктивність, додаткові набори функцій, таких як реплікація, дедуплікація, висока доступність, висока масштабованість, відсутність обмежень на кількість дисків, які можна додати.

Також забезпечується безперебійна робота, оскільки системи SAN не потребують перезавантаження для додавання нових дисків, для заміни дисків чи для налаштування RAID-групи.

Недоліком даного типу сховищ є висока собівартість зберігання даних.

Об'єктне сховище

Об'єктне сховище зберігає дані у вигляді об'єктів. Кожен «об'єкт» включає в себе власне дані, деяку кількість метаданих, а також глобально унікальний ідентифікатор. Об'єктне сховище є недорогим і зручним у масштабуванні способом для зберігання великих обсягів неструктурованих даних.

Особливістю об'єктного сховища є те, що кожен файл має ще й унікальний глобальний ідентифікатор крім користувачього. Якщо фізичне розташування об'єкта змінюється, зміни обробляються внутрішнім механізмом об'єктного сховища, а ідентифікатор, призначений користувачем або додатком, залишається незмінним, що дозволяє легко отримати доступ до нього в будь-який інший час.

Об'єктне сховище може зберігати велику кількість даних та доступ до них є більш простим, оскільки воно має декілька незалежних вузлів, які зберігають дані і центральний вузол не має знати місцезнаходження об'єкта для того, щоб отримати його.

Ідентифікатори можна використовувати для того, щоб легко порівняти два файли без необхідності завантаження. Об'єктне сховище також дає можливість використання великої кількості, простого, більш дешевого апаратного забезпечення різних типів, пов'язуючи все разом в єдину систему.

Кожен об'єкт, як правило, має кілька реплік, можливо в географічно розподілених кластерах. Кожен об'єкт також містить контрольну суму, яка дозволяє легко виявити пошкодження даних, в цьому випадку можлива генерація нової копії об'єкта для заміни пошкодженої.

Іншою характерною особливістю зберігання об'єкта є використання метаданих, або даних про дані. Кожен об'єкт при зберіганні маркований не тільки своїм унікальним ідентифікатором, але також інформацією про дані в об'єкті. Наприклад, оцифрована пісня поміщена в об'єктне сховище може зберігатися з додатковими метаданими про назву пісні, виконавця, альбомом, тривалістю, роком і так далі. Важливо відзначити, що метадані кожного об'єкта фіксуються і зберігаються з ним. Немає необхідності тримати їх в окремій централізованій базі даних, яка може некеровано збільшитись. Проте недоліком даного типу сховищ є те, що пошук по метаданим є складним в реалізації.

Критерії вибору сховищ

Перед тим як вибрати, яке рішення для зберігання найкраще працює для ваших потреб, є кілька факторів, які необхідно врахувати.

1. Чи необхідно захищати дані, підвищуючи надлишковість даних?

Найкращим варіантом за таких умов є об'єктне сховище, адже саме воно надає функцію зберігання кількох реплік одного об'єкта в декількох місцях, що забезпечує максимальний час безперебійної роботи. Однак серед SAN пристроїв також є типи, що можуть виконувати цю функцію.

2. Наскільки важлива продуктивність операцій читання / запису?

При важливості такого критерію найкращим вибором є SAN сховища які забезпечують найвищу продуктивність.

3. Скільки людей повинні отримати доступ до даних і як часто?

При порівнянні сховищ за цим критерієм, найкращий показник забезпечення доступності мають об'єктні і деякі SAN сховища. Щодо DAS та NAS сховищ, то найнижчий рівень забезпечують DAS, а NAS, хоч і забезпечують низький рівень, однак він кращий, ніж у DAS.

4. Скільки коштів і ресурсів планується витратити на сховище?

У разі обмеженості бюджету відповідними варіантами будуть DAS або NAS сховища.

Висновки

В кінцевому рахунку немає одного типу сховищ, яке буде відповідати всім вимогам зберігання даних, і тому бажано визначити, які вимоги є найбільш важливими у тому чи іншому випадку. Це може стати частиною загальної стратегії організації зберігання даних, що має вирішальне значення для забезпечення ефективного виконання бізнес процесів. Правильний вибір типу сховища даних може допомогти підприємствам впоратись із зростаючими об'ємами даних.

В результаті дослідження було розглянуто чотири основні типи сховищ даних. Кожен з них пропонує величезні переваги в певних обставинах. Не можна однозначно визначити «правильне» або «неправильне» рішення — є тільки те, що найкраще підходить до потреб підприємства. Було зазначено основні критерії до вибору найбільш прийняттого рішення та відповідність розглянутих типів сховищ даним критеріям.

Література

1. Gnanasundaram, S., Shrivastava, A., & Services, E. E. (2012). Information storage and management: Storing, managing, and protecting digital information in classic, virtualized, and cloud environments (2nd ed.). Indianapolis, IN: Wiley, John & Sons. — pp. 64–69.
2. Poulton, Nigel. Data Storage Networking. N.p.: Sybex, 2014. — pp. 145–156.
3. Toigo, J. W. The holy Grail of enterprise data storage. / Prentice-Hall, 1999 — pp. 45–47.

Івченко Дмитро Анатолійович

студент

Національний технічний університет України «Київський політехнічний інститут»

Ивченко Дмитрий Анатоліевич

студент

Национальный технический университет Украины «Киевский политехнический институт»

Ivchenko D.

student

National Technical University of Ukraine «Kyiv Polytechnic Institute»

МОБІЛЬНІ ДОДАТКИ ДЛЯ АСТМАТИКІВ

МОБИЛЬНЫЕ ПРИЛОЖЕНИЯ ДЛЯ АСТМАТИКОВ

MOBILE APPLICATION FOR ASTHMATICS

Анотація. Досліджено основні особливості функціонування мобільних додатків для астматиків.

Ключові слова: астма, мобільний, додаток, лікування.

Аннотация. Исследованы основные особенности функционирования мобильных приложений для астматиков.

Ключевые слова: астма, мобильный, приложение, лечение.

Summary. Were investigated main features of the functioning of mobile applications for asthmatics.

Key words: asthma, mobile, application, treatment.

Вступ

Разом з діабетом і серцево-судинними захворюваннями хронічні легеневі захворювання у вигляді бронхіальної астми є одними з найпоширеніших на планеті. За наявними оцінками 300 мільйонів чоловік у всьому світі страждають від астми. Близько 10% від загальної кількості — це діти. Тільки в США їх налічується 25 млн., А в Сполученому Королівстві — 5,4 млн.: 4,3 млн дорослих і 1,1 млн дітей. Щороку Національна служба охорони здоров'я Великобританії направляє на лікування пацієнтів даної групи близько мільярда фунтів; щорічно через легеневі захворювання країна втрачає до 1,1 млн трудових людино-днів. До 2025 року очікується збільшення числа пацієнтів з астмою більш ніж на 100 млн. З ростом поширеності астми ростуть витрати системи охорони здоров'я на стаціонарне лікування, зростають навантаження на медичний персонал. Цілі лікування астматиків за допомогою мобільних додатків вдома:

- Попередження пацієнтів про потенційно можливий напад, виходячи з аналізу даних про стан пацієнта, наявності алергенів і забруднювачів.
- Надання користувачеві інформації про те, як лікується астма, де і що викликає напади астми, з ме-

тою більш ефективного попередження і лікування ускладнень астми.

- Моніторинг застосування пацієнтом інгаляторів (небулайзерів) з вбудованими системами GPS, призначених для розпилення лікарського препарату на дисперсні частинки, які за допомогою трубки або маски потрапляють в дихальні органи пацієнта. Саме розмір часток лікувального розчину дозволяє йому проникнути в глибокі відділи органів дихання (бронхи і навіть бронхіоли), що підсилює ефект від процедури і прискорює дію препарату.
- Підтримка щоденного здоров'я і активності з урахуванням індивідуальних особливостей пацієнта і стану навколишнього середовища (рівня шкідливих викидів в атмосфері, наявності квіткового пилку, параметрів температури і вологості повітря і т.д.)
- Запобігання або усунення подальших ускладнень від хвороби.

Астма і технології визначення критичних станів хворого

Астма визначається як хронічна гіперчутливість (реактивність) дихальних шляхів, яка проявляється у вигляді повторюваних періодів загострень (нападів)

і заспокійливих періодів. Симптоми нападів включають задишку, утруднене дихання, важкість у грудях, кашель. Вони виникають як наслідок скорочення гладкої бронхіальної мускулатури, набряку та гіперсекреції слизу, що призводять до звуження дихальних шляхів (бронхоспазмів). Близько 70% астматиків страждають від алергії, індукованої алергенами з навколишнього середовища: пилом з дерев або трави, димом, пилом, кліщами, пір'ям, хутром, метеорологічними умовами та ін. Астма по частоті нападів розділяється на чотири стадії відповідно до основних принципів GINA (Global Initiative on Asthma):

1) початкова: переривчастий легкий кашель і хрипи не частіше, ніж два рази на тиждень, нічні напади менш, ніж два рази на місяць;

2) м'яка: напади трапляються, по меншій мірі, один раз в тиждень.

3) середня: звужуються проходи в легенях і трахеї, активізуються кашель і свистяче дихання;

4) важка: напади постійно трапляються як вдень, так і вночі і можуть тривати більше декількох днів, постійний кашель і хрипи.

Ще більш ситуацію погіршує той факт, що, згідно з даними недавніх опитувань приблизно половина астматиків ніяк не контролюють свою хворобу. Переломити цю тенденцію і допомогти пацієнтам виробити звичку відповідального ставлення до власного здоров'я як раз і покликані відповідні мобільні додатки. Вимірювання пікового обсягу повітря (ПОП), що видихається з легенів пацієнта (Peak expiratory flow Meteror «PEF») з використанням портативного пікфлометра. Прилад цей дуже простий, він фіксує максимальну швидкість потоку (МШП), що видихається, яка змінюється лише при зміні внутрішнього діаметра бронхів. Визначення ПОП з використанням пікфлометра іменується пікфлометрією. Всі дані, які видає прилад, можна порівняти зі спеціальною кольоровою шкалою ризику. Кольорова шкала являє собою аналог світлофора і розділена на три кольори: зелений, лимонний і червоний, що показують ступінь небезпеки недуги. Пікфлометрія попереджає про звуження дихальних шляхів часто заздалегідь за кілька годин або навіть днів, перш ніж у пацієнта проявляться симптоми астми. Використовуючи додаток, в якому знаходяться всі необхідні рекомендації при астмі, він буде знати, коли приймати рятувальний інгалятор або інші ліки проти астми. Слідуючи інструкціям цього плану дій при астмі, пацієнт може швидко зупинити звуження дихальних шляхів і уникнути важких нападів астми. Люди з середньою і важкою стадіями астми повинні мати пікфлометр в будинку і вміти ним користуватися. Лікуючим лікарем виконується розробка плану дій при астмі. Цей план може

допомогти пацієнтові контролювати своє захворювання і уникнути нападів бронхіальної астми. План дій при астмі призначений, щоб проінструктувати пацієнта про його дії, коли він відчуває зміни в симптомах хвороби і спостерігає зміни показника пікфлометра. Наприклад, в плані дій при астмі можуть бути перераховані алергени, які загострюють перебіг протікання астми у пацієнта, і деякі способи їх уникнення. План дій при астмі може також перераховувати рутинні симптоми астми і те, що пацієнт повинен робити, якщо ці симптоми виникають. План дій при астмі дає пацієнтові і його родині інформацію, яка може бути використана в тому випадку, якщо він відчуває приступ астми в дуже важкій формі.

Таким чином, план дій при астмі може включати:

1. Список збудників астми (алергенів), які провокують прояв симптомів астми, і як їх уникнути.

2. Список пікових показань пікфлометра і його зон, що враховує особистісні фізіологічні особливості пацієнта.

3. Список симптомів астми, таких як кашель, свистяче дихання, важкість у грудях, задишка і надлишок слизу, а також те, що повинен робити пацієнт, якщо ці симптоми виникають.

4. Найменування і щоденну дозу ліків проти астми, яка повинна бути прийнята, навіть якщо у пацієнта немає симптомів астми.

5. Найменування і дози швидкодійного або рятувальних ліків, які повинні бути прийняті при появі симптомів астми.

6. Найменування і дози препарату невідкладної допомоги, який повинен бути прийнятим, коли пацієнт відчуває приступ астми.

7. Телефони екстреного зв'язку і місця надання невідкладної допомоги. Інструкції про те, коли звернутися до лікаря, кому дзвонити, якщо лікар недоступний, і список, де можна отримати екстрену допомогу. Для зручності план дій при астмі, як зазначалося вище, часто розподіляється на три зони на пікфлометрію пацієнта: зелену, жовту і червону. Для кожної зони лікар формує в письмовим вигляді інструкції про те, як і що повинен робити пацієнт.

8. Ведення щоденника при астмі (symptoms diary) дозволяє записувати симптоми астми пацієнта, алергени і проведені лікування, так що він може контролювати хід астми. Там же фіксуються показники пікової швидкості видиху, порівняння цих показань з зонами плану дій при астмі, частота застосування ліків при раптовому нападі астми. Ведення такого щоденника астми (наприклад, в електронному вигляді) допоможе пацієнтові розпізнавати напади астми і застосовувати заходи щодо їх врегулювання, перш ніж серйозно захворіти. Лікар може використовувати

цей щоденник, щоб оцінити, наскільки правильні дії пацієнта при реалізації плану дій при астмі. Шляхом моніторингу наближаються симптомів нападів астми за допомогою пікфлометра і виконання інструкцій лікаря для різних зон показників МСП пацієнт може жити активним життям, вільної від симптомів астми.

9. Віддалений моніторинг стану пацієнта лікарем (медсестрою) і своєчасні нагадування йому про необхідність прийняти ліки або пройти відповідну процедуру, а також його діях у виниклих невідкладних ситуаціях. Якщо життєві показники, за якими ведеться моніторинг, наблизяться до небезпечної межі, лікар тут же буде сповіщений про це. Моніторинг може здійснюватися як за записами щоденника при астмі, так і спеціальних пристроїв (інгаляторів з GPS, портативних вимірювачів дихання пацієнта та ін.). При цьому з метою контролю концентрації алергенів пацієнт може носити з собою поясний бандаж з портативних приладів, що вимірюють якість навколишнього середовища.

10. Додаток враховує відомості, що характеризують як індивідуальні особливості пацієнта, так і стан навколишнього середовища — рівень шкідливих викидів в атмосфері, наявність квіткового пилку, параметри температури і вологості повітря і т.д. На основі отриманих відомостей додаток вираховує для свого користувача індивідуальний план дій, і, слідуючи цим планом, пацієнт може вибирати стратегію подальших дій і підтримувати більш осмислений діалог під час консультацій з лікарем. Крім того, в залежності від показників пацієнта і навколишнього середовища додаток пропонує користувачеві ознайомитися з відібраними персонально для нього матеріалами медичної тематики.

11. Інтегрування в соціальні мережі, які можуть надавати своїм учасникам можливість отримувати всебічну інформацію про методи лікування астми, ділитися своїм досвідом в цій сфері. Користувач може створювати свій обліковий запис (використовуючи для цього, зокрема, акаунти в інших соціальних мережах), вступати в групи за інтересами, розміщувати свої матеріали і коментувати публікації інших учасників, завантажувати фотографії, обмінюватися електронними листами з іншими членами мережі. Прикладом може служити соціальна мережа Asthma UK.

12. Синхронізацію записів щоденника при астмі з персональної електронної медичної картою ПЕМК (personal health record — PHR), якій належить важлива роль в забезпеченні мобільності медичної інформації та яку ведуть самі пацієнти за допомогою онлайн-сервісів, подібних Google Health і лікарів.

Мобільні додатки для контролю астми

Мобільні додатки для контролю за астмою включають нові технології, які дозволяють пацієнтові зрозуміти

ти що саме викликає напади астми, щоб запобігти і не допускати загострення астми. Ці інструменти включають GPS додатки до інгалятора, які фіксують дані про час і місце прийому препарату, використовують, мобільні додатки подібні блокноту, в які пацієнти можуть власноруч вводити дані про свій стан. А також ці програми мають програмне забезпечення для прогнозування можливої кризи, який може привести до потенціалом нападу астми, що базується на таких факторах навколишнього середовища, як алергени і забруднювачі.

Варіанти використання: •

Згідно CDC, майже 25 мільйонів чоловік в США були діагностовані з астмою, яка становить приблизно 8 відсотків населення.

- Астма — хронічна хвороба, яка проявляється в залежності від:
 - Астма поширена частіше серед багатонаціональних (14.8%), латиноамериканських (14.2%) і афроамериканців (9.5%), у порівнянні з європейцями (7.8%).
 - Відмінності також помічені на вік, стать і соціально-економічному статусі: поточне поширення астми вище серед дітей (9.3%), у порівнянні з дорослими (7.3%); вище серед жінок (8.6%), ніж чоловіки (6.9%); і вище серед бідних (11.2%).
- За даними Behavioral Risk Factor Surveillance System (BRFSS) за 2010 рік, у 5.9% дітей ($n = 530,690$) і 7.7% дорослих ($n = 2,155,879$) в Каліфорнії в даний час хворіють на астму[9,10].
- Мобільні додатки для управління астмою призначаються для тих, хто страждає від нападів астми, а особливо для дітей. Щоб допомогти їм уникнути областей викликають напад алергену і допомогти їм краще відстежувати і контролювати свої симптоми.
- Існує безліч мобільних інструментів для хворих на астму, які в даний час представлені на ринку, у всіх з яких є подібні, але трохи відмінні підходи: — Інгалятори з технологією GPS: підключають GPS і інтернет до інгалятора, щоб допомогти визначити місце розташування і можливу причину виходячи з даних про місцезнаходження, наприклад це може бути район з великою кількістю дерев липи і відповідно великою кількістю вірогідних алергенів в повітрі [11].
- Інгалятори з аудіовізуальними нагадуваннями: використовують аудіовізуальні нагадування на інгаляторах, щоб допомогти поліпшити лікування астми, наприклад приходить повідомлення про необхідності прийняття препарату [7].
- Додатки для смартфона: дозволяють користувачам швидко і легко вести облік етапів лікування астми, прийом ліків і список алергенів в щоденнику. Ці дані можуть бути легко проаналізовані їх лікарями в режимі онлайн.

Складнощі і ризики використання програми

1. Простота в використанні: багато хто з цих технологій використовують програми смартфона, що може викликати деякі проблеми для літніх людей.
2. Обмежені дані: ці програми в основному не перевірені, особливо щодо результатів та можливого ROI і зниження витрат з використанням цих технологій.
3. Проблеми приватного життя: відсутність чітких рекомендацій простеження GPS і передача медичної інформації по інтернету, можуть привести до розголошення особистих даних. Але коли ти хворий, вибрати не доводиться.
4. Вартість підтримки пристрою: можуть виникнути проблеми з придбанням у літніх людей і малозабезпечених людей.
5. Проблеми компенсації: широко поширене прийняття цієї технології дуже залежить від моделі компенсації. Якщо курс лікування накриття страховкою, малоймовірно, що пацієнти куплять їх за власні кошти, мається на увазі покупка самого додатка і необхідного устаткування та препарати.

Наступні кроки до впровадження

1. Розробити стратегію, яка буде адресована для терплячого виконання умов: багато хто з цих технологій можуть бути обтяжливими для користувача. Наприклад, пацієнти повинні вручну вводити дані про хід хвороби в додаток смартфона багаторазово в день як частина їх розпорядку дня.
2. В результаті стратегії повинні бути розроблені разом з пацієнтом, щоб мінімізувати зміни до розпорядку дня пацієнта і побоюванням про порушення конфіденційності особистого життя з приводу приватного життя.
3. Leverage Technologies для втручання охорони здоров'я: у цих програм є ясна епідеміологічна вартість, оскільки дані про астму можуть тепер збиратися протягом довгого часу. У свою чергу, охорону здоров'я і академічні спільноти повинні працювати у співпраці, для розвитку підхід, щоб зібрати й інтерпретувати дані для користування осіб охорони здоров'я.

Мобільні додатки

Революція в світі смартфонів дуже сильно змінила наше життя. Так само ця революція і дійшла до хворих астмою людей, щоб полегшити перехід їх хвороби. Зовсім недавно були додатки, які допомагали людям відслідковувати скільки разів вони використовують їх інгалятори і які дали можливість лікарям спостерігати за лікуванням їх пацієнтів режимі онлайн. Тепер же кількість різноманітних додатків дуже велике. При виборі відповідного для вас важливо знати чим вони відрізняються і для чого потрібні і який функціонал

вам буде доступний, а також не менш важливою є їх вартість.

1. Мобільні додатки, які пов'язані з інгаляторами з використанням технології Bluetooth та автоматично зберігають інформацію про показання приладу в телефон, щоб лікар мав можливість відслідкувати динаміку проходження захворювання.
2. Мобільні додатки, які записують інформацію про місцезнаходження GPS для оповіщення астматиків про екологічних умови, які можуть спричинити напад астми.
3. Мобільні додатки, які використовують відео записи і ігри, щоб мотивувати пацієнтів робити свої вправи фізичної терапії вдома;
4. Мобільні додатки, які допомагають астматикам здійснювати контроль за використанням інгалятора та нападами астми;
5. Мобільні додатки, що запитують у користувача вручну ввести симптоми і екологічні фактори, специфіку яких попередньо визначає лікар, і зберегти інформацію для подальшого розгляду (додатки щоденники);
6. Мобільні додатки, які використовують дані пацієнта, такі як вік, стать і симптоми пацієнта для того, щоб давати рекомендації по прийому ліків.
7. Мобільні додатки, які покликані дозволити користувачеві ініціювати виклик медсестри або екстреної допомоги за допомогою технології інтернет або стільникового телефону;
8. Мобільні додатки, що вимірюють спірометрію легенів за допомогою власного динаміка.

Таблиця 1

Порівняльна таблиця мобільних додатків

Характеристики додатку/ Назва	Запис даних вручну	Автоматично Bluetooth	Нагадування про прийом ліків	Аналіз пацієнта	GPS Інф.	Тривимірний висілки лікаря	Програма тренувань	Святок з сол. місяцями	Android	IOS	Ціна
Asthma New Zealand	•		•							•	Free
My Asthma Log	•		•					•	•		Free
AsthmaMD	•		•	•					•	•	
AsthmaCheck	•		•	•					•	•	
AsthmaTracker	•		•	•							
AsthmaSense	•		•			•			•	•	Free
Propellerhealth		•	•		•	•				•	Free
Respi		•	•	•	•				•	•	
Cohero Health		•	•	•					•	•	
BreatheEasy							•			•	Free

Висновки

Було проведено аналіз всіх існуючих мобільних додатків для хворих на астму та описано функціонал, якими вони володіють. Приведено порівняльну таблицю, в якій наведено найпопулярніші додатки.

Література

1. Sveum R., Diagnosis and Management of Asthma / Sveum R. Bergstrom J, Brottman G, Hanson M, Heiman M, Johns K, Malkiewicz J, Manney S, Moyer L, Myers C, Myers N, O'Brien M, Rethwill M, Schaefer K, Uden D. Institute for Clinical Systems Improvement, Updated July 2012. P. 1–2.
2. Sveum R., Diagnosis and Management of Asthma / Sveum R. Bergstrom J, Brottman G, Hanson M, Heiman M, Johns K, Malkiewicz J, Manney S, Moyer L, Myers C, Myers N, O'Brien M, Rethwill M, Schaefer K, Uden D. Institute for Clinical Systems Improvement. 86 p.
3. National Heart, Lung, Blood Institute. Expert panel report 3: guidelines for the diagnosis and management of asthma, 2007. P. 74.
4. National Heart, Lung, and Blood Institute National Asthma Education and Prevention Program Expert Panel Report 3: Guidelines for the Diagnosis and Management of Asthma, 2007. P. 60.
5. Newman J. High-tech inhaler from Madison company would help doctors track asthma attacks. June 15, 2011. Wisconsin State Journal [Электронный ресурс]. — Режим доступа: http://host.madison.com/wsj/business/article_e79d070e-9791-11e0-a329-001cc4c002e0.html. Дата доступа: грудень 2012.
6. Behavioral Risk Factor Surveillance System (BRFSS). Child Current Asthma. Prevalence Rate (Percent) and Prevalence (Number) by State or Territory: BRFSS2010. [Электронный ресурс]. — Режим доступа: <http://www.cdc.gov/asthma/brfss/2010/child/current/tableC1.htm>. Дата доступа: грудень 2011.
7. Halepis, H. GPS Inhalers Offer A Breath of Fresh Air, 2011. [Электронный ресурс]. — Режим доступа: <http://www.rm-tracking.com/blog/2011/04/18/gps-inhalers-offer-a-breath-of-fresh-air/>. Дата доступа: грудень 2011.
8. Charles, T. An audiovisual reminder function improves adherence with inhaled corticosteroid therapy in asthma / Charles, T., Quinn, D., Weatherall, M., et al. Allergy Clin Immunol, 2007. P 811–816.
9. AsthmaMD. About AsthmaMD. [Электронный ресурс]. — Режим доступа: http://www.asthmamd.org/about/#/resources/iphone_chart.jpg. Дата доступа: грудень 2013.
10. Asthma Signals. Asthma Signals, Inc. and the Asthma and Allergy Foundation of New England collaborate on an innovative mobile technology designed to tackle Pediatric Asthma. Press Release: May 24, 2011. [Электронный ресурс]. — Режим доступа: <http://www.asthmasignals.com/partners.htm>. Дата доступа: грудень 2013.
11. Klein, S. Quality Matters Improving Asthma Control with Mobile Technology. The Commonwealth Fund. October/November 2011. [Электронный ресурс]. — Режим доступа: <http://www.commonwealthfund.org/Newsletters/Quality-Matters/2011/October-November-2011/QA.aspx>. Дата доступа: квітень 2015.

Качаловський Андрій Сергійович

Студент

Національний технічний університет України «Київський політехнічний інститут»

Качаловский Андрей Сергеевич

Студент

Национальный технический университет Украины «Киевский политехнический институт»

Kachalovskyi A.

Student

National Technical University of Ukraine Kyiv Polytechnic Institute

**МЕТОДИ І МОДЕЛІ ПРОГНОЗУВАННЯ ЛІНІЙНИХ
ТА НЕЛІНІЙНИХ НЕСТАЦІОНАРНИХ ПРОЦЕСІВ
МЕТОДЫ И МОДЕЛИ ПРОГНОЗИРОВАНИЯ ЛИНЕЙНЫХ
И НЕЛИНЕЙНЫХ НЕСТАЦИОНАРНЫХ ПРОЦЕССОВ
METHODS AND PREDICTION MODELS LINEAR
AND NONLINEAR NON-STATIONARY PROCESSES**

Анотація. В роботі досліджується можливість використання регресійних моделей для опису та прогнозування фінансово-економічних процесів.

Ключові слова: Модель, тренд, регресія, гетероскедастичність, нафта.

Аннотация. В работе исследуется возможность использования регрессионных моделей для описания и прогнозирования финансово-экономических процессов.

Ключевые слова: Модель, тренд, регрессия, гетероскедастичность, нефть.

Summary. This paper explores the possibility of using regression models to describe and predict the financial and economic processes.

Keywords: model, trend, regression, heteroskedasticity, oil.

Вступ

В сучасних умовах загальної економічної нестабільності, коли у світі відбуваються фінансово-економічні кризи, виникає необхідність удосконалення світової фінансової системи, а також удосконалення методів і технологій управління фінансовими ризиками. Актуальною задачею є виконання поглиблених наукових досліджень в напрямі математичного моделювання і прогнозування фінансових процесів. Це стосується, безумовно, і українського фінансового ринку, на якому фінансовий ризик-менеджмент знаходиться на ранній стадії свого становлення.

Прагнення побудувати модель, яка б найбільш точно відповідала реальній поведінці фінансових ринків, та необхідність підвищення якості прогнозів ведуть до появи нових класів моделей і до модифікацій вже існуючих. Таким чином, побудова і дослідження ма-

тематичних моделей, які адекватно описують динаміку таких фінансових інструментів як акції, облігації, опціони, котирування валют та інших, є актуальним напрямом сучасних досліджень.

Актуальність наведених задач, пов'язаних з аналізом нестационарних фінансових процесів, визначила тему і напрям магістерського дослідження.

Постановка задачі

Необхідно вибрати клас математичних моделей для фінансово — економічних процесів, виконати їх розробку, виконати оцінювання якісних показників для побудованої моделі та прогнозу, та виконати порівняльний аналіз побудованих моделей.

Гетероскедастичність означає, що дисперсія процесу зменшується чи збільшується в часі, або є більш складною функцією часу. Тобто, вона може змінюватися за

досить складним законом, який і потрібно знайти при створенні моделі процесу. Іноді використовують припущення, що гетероскедастичність має наступну форму:

$$\sigma_{\varepsilon(k)}^2 = k^2 x^2,$$

де k — константа, яку необхідно оцінити за допомогою експериментальних даних та вибраного методу оцінювання параметрів.

Лінійний та нелінійний тренд

Існує два типи тренду: детермінований (регулярний) та стохастичний (нерегулярний). Процеси зі змінною дисперсією (змінюється з часом) називають гетероскедастичними. Дисперсія може змінюватися за досить складним законом, який і потрібно знайти при створенні моделі процесу. Процеси з трендами або зі змінною дисперсією особливо характерні для нестійкої швидкозростаючої перехідної економіки. Наявність нелінійностей. Однією з проблем визначення структури моделі є встановлення факту нелінійностей в досліджуваному процесі та їх типу. Економічна динаміка являє собою нелінійне явище. Нелінійність означає можливість неочікуваних змін напрямлення процесів. В динамічних економічних системах нелінійність виявляється як підвищена реакція на зміну одних факторів і повною байдужістю до інших.

В загальному випадку прогноз може бути представлений одним (точковим) значенням змінної, інтервалом, в який попадає випадкова змінна, а також ймовірністю прийняття змінною (чи подією) деякого значення у вибраному інтервалі. Якщо для опису процесу застосовують лінгвістичні змінні, то прогнозом буде нечітке значення, але його також можна перетворити в чітке число.

Можна по-різному ставити задачу прогнозування в залежності від рівня прийняття рішення та конкретної поставленої задачі управління чи контролю. Прогнозування може стосуватись наступних складових процесу:

- детермінованого тренду, як індикатора довгострокових змін процесу;
- випадкового (нерегулярного) тренду, як показника коротко- та середньострокових змін;
- короткострокових змін, тобто, прогнозування коливань (відхилень), що накладаються на тренд;
- сезонних ефектів;
- приростів (швидкості) зміни процесу, які визначаються першими різницями;
- дисперсії або стандартного відхилення, як міри розсіювання процесу (наприклад, волатильність, яку часто використовують за міру ризику у інвестуванні або міру якості на виробництві).

Відповідно до того, які складові процесу необхідно прогнозувати, ставиться задача побудови математич-

ної, ймовірнісної (статистичної) або логічної моделі, що має меті забезпечити високу якість прогнозу на заданий часовий горизонт. Розглянемо деякі можливості створення математичного опису складових процесів різної природи.

Детермінований тренд

Якщо описати детермінований тренд за допомогою полінома від часу довільного порядку $y(k) = a_0 + a_1 * k + a_2 * k^2 + \dots + a_p * k^p + \varepsilon(k)$, $E[\varepsilon(k)] = 0$ то визначення прогнозу тренду зводиться до підстановки в це рівняння бажаного значення часу k і застосування безумовного математичного сподівання. Прогнозування значень тренду вважається довгостроковим прогнозом, оскільки детермінований тренд вказує на довгострокові зміни процесів. Обмеження на випадковий процес $E[\varepsilon(k)] = 0$ необхідне для коректного застосування методів оцінювання моделей, а також для виконання подальшого аналізу якості оцінок прогнозів.

Стохастичний тренд

Для опису і прогнозування стохастичного тренду можна скористатись, наприклад, відомим рівнянням випадкового кроку із перетином (константою):

$$y(k) = a_0 + y(k-1) + \varepsilon(k),$$

розв'язок якого має вигляд:

$$y(k) = y_0 + ka_0 + \sum_{i=1}^k \varepsilon(i)$$

Сума $\sum_{i=1}^k \varepsilon(i)$ в правій частині останнього рівняння описує випадкову складову тренду. Цю складову називають ще нерегулярною.

Прогнозування коливань, що накладаються на тренд.

Колівання, що накладаються на тренд, або короткострокові зміни можна описати рівняннями авторегресії з ковзним середнім

$$y(k) = \sum_{i=1}^p a_i y(k-i) + \sum_{j=1}^q b_j \varepsilon(k-j) + \varepsilon(k).$$

Техніку обчислення такого прогнозу за допомогою умовного та безумовного математичного сподівання розглянемо нижче в цьому розділі.

При переході до перших та різних вищих порядків з процесу вилучається тренд відповідного порядку. Наприклад, якщо процес містить лінійний тренд, то перші різниці вилучають його і після переходу до різниць ми маємо справу з коливаннями, що накладаються на тренд.

Можливість прогнозування сезонних ефектів досягається за рахунок введення в модель процесу відповідних значень основної змінної із затримками (ла-

гами), що відповідають періодичності ефекту. Як буде показано нижче, сезонний ефект може бути врахований як за допомогою основної змінної, так і за допомогою процесу ковзного середнього.

Прогнозування дисперсії

Якщо дисперсія процесу змінюється в часі, то для її описання можна вибрати рівняння для формування відповідної функції прогнозування та обчислення в подальшому оцінки прогнозованого значення. Процеси зі змінною дисперсією отримали назву гетероскедастичних. Методика побудови моделей гетероскедастичних процесів передбачає наступні кроки:

- математичний опис самої змінної рівнянням авторегресії невисокого порядку (наприклад, першого);
- математичний опис умовної дисперсії, як динамічної змінної, за допомогою рівняння прийнятної (за якістю прогнозу) структури.

Тепер проаналізуємо та побудуємо моделі для опису показників нафти. Це обумовлено тим що нафта досить нестабільна. Однією з головних проблем при побудові моделей цього ряду, те що вибірка містить лише 60 вимірів даних. На рисунку 1.1 зображені показники AR(1) для нафти.

Dependent Variable: TRAFFIC
Method: Least Squares
Date: 06/07/16 Time: 21:00
Sample (adjusted): 2 60
Included observations: 59 after adjustments
Convergence achieved after 3 iterations

Variable	Coefficient	Std. Error	t-Statistic	Prob.
AR(1)	1.048987	0.010195	102.8925	0.0000

R-squared	0.985218	Mean dependent var	237.8305
Adjusted R-squared	0.985218	S.D. dependent var	183.2784
S.E. of regression	22.28298	Akaike info criterion	9.062327
Sum squared resid	28798.81	Schwarz criterion	9.097540
Log likelihood	-266.3387	Hannan-Quinn criter.	9.076073
Durbin-Watson stat	1.944284		

Inverted AR Roots	1.05
-------------------	------

Estimated AR process is nonstationary

Рисунок 1.1. Показники для процесу AR(1) для нафти

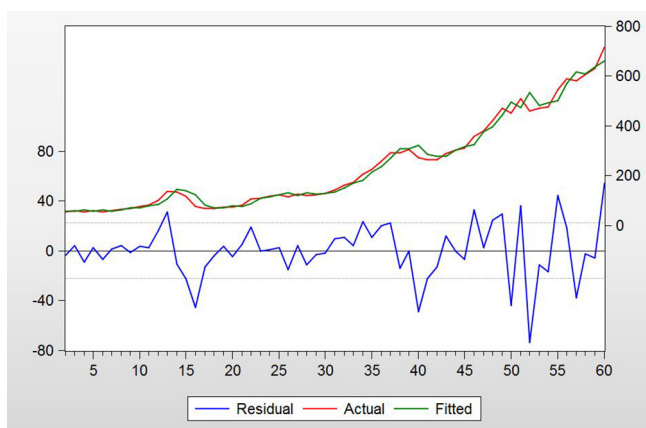


Рисунок 1.2. Графічне зображення моделі з величиною помилки

На рисунку 1.3 побудована модель AR + тренд для нафти.

Dependent Variable: TRAFFIC
Method: Least Squares
Date: 06/07/16 Time: 21:07
Sample (adjusted): 2 60
Included observations: 59 after adjustments
TRAFFIC=C(1)+C(2)*TRAFFIC(-1)+C(3)*K

	Coefficient	Std. Error	t-Statistic	Prob.
C(1)	-5.039655	6.702047	-0.751958	0.4552
C(2)	1.000539	0.045012	22.22846	0.0000
C(3)	0.520031	0.454823	1.143370	0.2577

R-squared	0.985556	Mean dependent var	237.8305
Adjusted R-squared	0.985041	S.D. dependent var	183.2784
S.E. of regression	22.41660	Akaike info criterion	9.106990
Sum squared resid	28140.23	Schwarz criterion	9.212628
Log likelihood	-265.6562	Hannan-Quinn criter.	9.148227
F-statistic	1910.567	Durbin-Watson stat	1.904447
Prob(F-statistic)	0.000000		

Рисунок 1.3. Показники для процесу AR(1) + кв. тренд для нафти

Аналіз результатів моделювання для показників нафти

Таблиця 1.1

Модель	SEE	R2	DW	СКП	САПП	U
AR(1)	28798,81	0,985218	1,944284	22,09270	7,831744	0,0369
AR(2)	28764,7	0,984968	1,887666	22,26977	7,881067	0,036946
AR(1)+тренд	28140,23	0,985556	1,904447	21,83926	7,993027	0,036532
AR(1)+кв. тренд	25755,31	0,98678	1,789986	20,89332	7,856288	0,034945

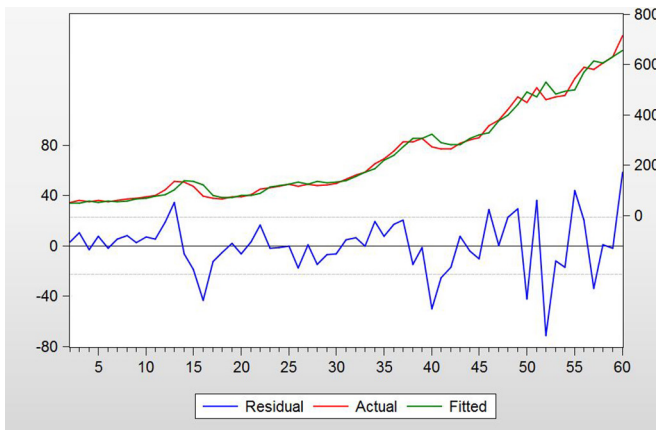


Рисунок 1.4. Графічне зображення моделі $AR(1)$ + кв. тренд з величиною помилки

В таблиці 1.1 наведено порівняльний аналіз результатів моделювання показників нафти за допомогою різних моделей.

Висновки

У даній роботі були розглянуті основні підходи до побудови моделей та оцінок прогнозів фінансово — економічних процесів. Побудована модель для опису показників нафти. Найкраще ряд описує модель $AR(1)$ + квадратичний тренд. Можна звернути увагу на графік. У останні роки на графіку чітко виражений спад, а потім поступовий підйом. Саме вплив таких неконтрольованих збурень впливає на побудову моделі та прогнозування, що набагато ускладнює цей процес.

Список використаних джерел

1. Бідюк П. І. Аналіз часових рядів / П. І. Бідюк, В. Д. Романенко, О. Л. Тимошук. — К.: Політехніка, 2010. — 317 с.
2. Молчанов И. Н. Компьютерный практикум по начальному курсу эконометрики (реализация на Eviews): Практикум / Ростовский государственный экономический университет. / Молчанов И. Н., Герасимова И. А. — Ростов-н/Д., — 2001. — 58 с.
3. Бідюк П. І. Часові ряди: моделювання і прогнозування / П. І. Бідюк, О. І. Савенков, І. В. Баклан. — К.: ЕКМО, 2003. — 144 с.

Ковтун Володимир Валентинович

Студент

Національний технічний університет України «Київський політехнічний інститут»

Ковтун Владимир Валентинович

Студент

Национальный технический университет Украины «Киевский политехнический институт»

Kovtun V.

Student

National Technical University of Ukraine «Kyiv Polytechnic Institute»

**СИСТЕМА ЗАХИСТУ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ
ВІД НЕСАНКЦІОНОВАНОГО ДОСТУПУ ДО КОДУ
СИСТЕМА ЗАЩИТЫ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ОТ
НЕСАНКЦИОНИРОВАННОГО ДОСТУПА К ИСХОДНОМУ КОДУ
SYSTEM SOFTWARE PROTECTION AGAINST
UNAUTHORIZED ACCESS TO THE SOURCE CODE**

Анотація. Розглянуто проблеми вже використовуваних захистів програм, а також програмних засобів, які використовують для обстеження програмного коду і висунута нова модель захисту коду програми.

Ключові слова: захист інформації, системи захисту від взлому.

Аннотация. Рассмотрены проблемы уже используемых защит программ, а также приложений, которые используют для обследования программного кода и выдвинута новая модель защиты кода программы.

Ключевые слова: защита информации, системы защиты от взлома.

Summary. The article considers problems are used for applications and also software tools used for inspection of code and launched a new security model code.

Key words: information security, protection system against hacking.

Вступ

Вданий час існує величезна кількість програм, які допомагають фахівцям у всіх сферах їхньої діяльності. Наприклад, текстові редактори, які спрощують роботу з текстом, або кошти для розробки програмного забезпечення, які дозволяють програмістам створювати нові програми. Кожна з програм містить унікальні напрацювання або нововведення, яка відрізняє її від інших. Багато з розробників намагалися зберегти в секреті свою інтелектуальну власність і отримувати прибуток від розробок свого програмного забезпечення. З цією метою і були створені перші системи захисту, які вбудовувалися в програми на етапі проектування і представляли собою прості системи ідентифікації і автентифікації. Вони ґрунтувалися на знанні користувачем ключа (серійного номера), який поставлявся разом з програмним забезпеченням. Користувач при запуску програми вводив відомий йому

ключ, потім система захисту звіряла його з еталонним, і якщо ключі збігалися, то користувач міг працювати з програмою, якщо ж ні, то виводилося повідомлення про те, що ключі неоднакові і програма не виконувалась. У перший час розвитку методів і систем захистів приділялося мало уваги, так як не було способів обійти існуючі. Але з розвитком інформаційних технологій стали з'являтися програмні засоби, що дозволяють зловмисникам вивчати роботу програм без знання вихідних кодів. Це були перші налагодчики, за допомогою яких користувачі стали обходити існуючі методи захисту, в результаті чого багато розробників стали зазнавати збитків. З появою налагодчиків стали активно розвиватися методи і системи захисту, які ускладнюють або унеможливають роботу налагодчика. Незважаючи на безліч розроблених методів захисту програм, програмні засоби для вивчення коду також активно розвивалися, з'являлися нові засоби, такі як

дизасемблер, декомпілятор і інші, які вже дозволяли краще розуміти логіку роботи програми і отримувати її вихідний текст.

На даний момент розвиток методів і технік захисту відстає від розвитку програмних засобів для вивчення коду. Розробники для захисту свого програмного забезпечення використовують або застарілі методи і техніки захисту, або комерційні системи захисту. Деякі з цих систем надають ефективний захист і ускладнюють вивчення програм, інші ж спроектовані з використанням застарілих методів і технік, які швидко обходяться кваліфікованими програмістами. Тому необхідне створення нової системи захисту. Модель системи протидії програмних засобів вивчення коду програмного забезпечення, яка повинна враховувати особливості програмних засобів для вивчення коду та існуючі способи протидії їм інших систем захистів або приклади їх обходу, так як це дозволить не допустити подібні помилки в проектуванні і розробити більш стійку систему.

Огляд основних типів програмних засобів вивчення коду

Для створення ефективної системи захисту необхідно ретельно дослідити засоби, які будуть застосовуватись зловмисниками для її вивчення. Адже в кожному програмному засобі є свої переваги і недоліки, які необхідно враховувати. Одним з програмних засобів, яке застосовується для вивчення роботи програми, є наладчик [1. С. 188–216].

Він дозволяє виконувати динамічний аналіз, в ході якого зловмисник отримує більше інформації про роботу програми. Це стає можливим, тому що наладчик створює нескінченний цикл всередині програми і чекає системної події налагодження. Коли таке повідомлення генерується, цикл переривається і викликається відповідний обробник подій. Подіями, які слугують командою для наладчика, є точки зупину, що дозволяють зупинити процес. Таким чином, можна вивчити кожну інструкцію, виконувану програмою, перевірити змінні, аргументи стека і дізнатися дані, що знаходяться в цей момент в пам'яті, до того, як вони будуть перезаписані. Це дозволяє програмістам отримувати більше інформації про роботу програми, а також детально вивчати механізми захисту.

Існують три основні види точок зупину [2]:

1. Програмні точки зупину — часто використовуються, так як дозволяють зупинити процесор при виконанні інструкцій програми, представляють собою однієї інструкції, яка зупиняє процес і передає управління обробнику переривань. Кількість встановлених програмних точок в програмі необмежено. Але при їх установці модифікується код програ-

ми, яку може відстежити механізм захисту цілісності і припинити роботу програми.

2. Апаратні точки зупину — даний тип точок встановлюється на рівні процесора в спеціальних регістрах і не модифікує програму, що вивчається. Всього можна встановити до чотирьох точок зупину.

3. Точки зупинки пам'яті — дозволяють змінювати дозвіл на сторінці пам'яті, в якій можуть міститися важливі дані. Завдяки цьому такі точки дозволяють детально вивчати механізми захисту, засновані на порівнянні введеного пароля з еталонним, так як еталонний пароль під час порівняння знаходиться в пам'яті у відкритому вигляді.

Стає очевидно, що наладчик являє собою сильний програмний засіб для вивчення програм, так як дозволяє отримувати багато цінної інформації про їхню роботу.

Наступний тип програмних засобів, так званий дизасемблер [3], дозволяє перетворювати машинний код в текст програми на мові асемблера, який дає можливість програмістам дізнатися архітектуру програми, а також визначити місце розміщення захисних механізмів. Ще один тип програмних засобів для вивчення коду називається шістнадцятирічним редактором [4]. Він дозволяє редагувати дані, такі як образ диска, вміст оперативної пам'яті і інші. Варто відзначити, що дані представлені у вигляді ланцюжків байтів в шістнадцятирічній системі числення. Завдяки цьому типу засобів стає можливим знайти в досліджуваній програмі впроваджений еталонний ключ без використання наладчика або дизасемблера. Однак у даний момент ключі генеруються в самій програмі і в відкритому вигляді не зберігаються. З цієї причини в існуючих системах захисту застосовується кілька методів і технік, які підвищують стійкість до вивчення програмного забезпечення, але разом з тим мають свої недоліки. Наприклад, багато систем захисту спроектовані за загальним шаблоном, розкриття якого призводить до нейтралізації захисту. Під шаблоном розуміється застосування однотипних методів і технік протидії програмних засобів. З цієї причини багато хто з систем захисту є неефективними, так як алгоритм протидії їм, в більшості випадків, є схожим.

Модель системи протидії програмних засобів вивчення коду

Для усунення наведених недоліків в системах захисту була розроблена нова модель системи, яка дозволить створити унікальну систему захисту і більш ефективно протидіяти програмним засобам для вивчення коду.

Система являє собою систему впровадження модулів в програму яка захищається. Система впрова-

дження на основі відомих алгоритмів впровадження коду вбудовує захисні модулі в програму. Захисні модулі створюються самим розробником і являють собою об'єктні файли, написані на мовах програмування C/C ++, асемблера і ін., Які містять в собі свої методи і техніки або застосовують існуючі, що дозволить усунути однотипність захисту, так як в разі вивчення зловмисником модуля захисту та розробки алгоритму протидії даний модуль може бути швидко замінений іншим, а вивчення системи впровадження із застосуванням програмних засобів також не надасть зловмисникові ніякої корисної інформації. Ще однією перевагою такої системи є тісна інтеграція захисних механізмів з самою програмою, що дозволяє

усувати критичні помилки на етапі проектування. До того ж такий підхід дозволяє створювати кілька модулів, які можна буде впроваджувати в програму поступово, у міру вивчення і обходу існуючих механізмів захисту.

Варто відзначити, що модулі можуть містити різноманітні захисні техніки, наприклад текст програми може бути зашифрований і розшифровуватися за допомогою системи впровадження або мати власний дешифратор. Тому система спроектована з урахуванням можливих модифікацій під потреби розробників, дозволить їм самостійно створювати більш ефективний захист і зберігати в секреті свою інтелектуальну власність.

Література

1. Dang B., Gazet A., Bachaalany E. Practical Reverse Engineering: x86, x64, ARM, Windows Kernel, Reversing Tools, and Obfuscation. Indianapolis, 2014. — 383 p.
2. Общие сведения об отладке: точки останова. URL: <http://msdn.microsoft.com/ru-ru/library/vstudio/4607yxb0%28v=vs.100%29.aspx> (дата обращения: 20.12.2014).
3. Disassembler // Wikipedia, the free encyclopedia. URL: en.wikipedia.org/Disassembler (дата обращения: 18.12.2014).
4. Hex editor // Wikipedia, the free encyclopedia. URL: en.wikipedia.org/Hex_editor (дата обращения: 18.12.2014).

Кравченко Анатолий Михайлович

Студент

Національний технічний університет України «Київський політехнічний інститут»

Кравченко Анатолий Михайлович

Студент

Национальный технический университет Украины «Киевский политехнический институт»

Kravchenko A.

Student

National Technical University of Ukraine «Kyiv Polytechnic Institute»

**МОДЕЛІ ТА ОЦІНКИ ПРОГНОЗІВ ПРОЦЕСІВ
ЕКОНОМІКИ ПЕРЕХІДНОГО ПЕРІОДУ
МОДЕЛИ И ОЦЕНКИ ПРОГНОЗОВ ПРОЦЕССОВ
ЕКОНОМІКИ ПЕРЕХОДНОГО ПЕРИОДА
MODELS AND FORECASTS ESTIMATES
FOR ECONOMY IN TRANSITION**

Анотація. В роботі досліджується можливість використання регресійних моделей для опису та прогнозування процесів економіки перехідного періоду.

Ключові слова: модель, регресія, тренд, ВВП.

Аннотация. В работе исследуется возможность использования регрессионных моделей для описания и прогнозирования процессов экономики переходного периода.

Ключевые слова: модель, регрессия, тренд, ВВП.

Summary. This paper investigate regression usage for modeling and forecast for economy in transition.

Key word: model, regression, trend, GDP.

Вступ

Протікання сучасних фінансово-економічних процесів характеризується великою кількістю специфічних особливостей, які треба враховувати при побудовах адекватних моделей та їх прогнозування. Можна виділити розглянуті нижче особливості.

Висока нестаціонарність процесів. Під нестаціонарними процесами будемо розуміти такі процеси для яких властиво наявність тренду або змінної дисперсії (гетероскедастичні процеси). Під трендом будемо розуміти загальний довгостроковий напрям розвитку процесу. Фактично він співпадає з поточним середнім значенням. Існує два типи тренду: детермінований (регулярний) та стохастичний (нерегулярний). Визначення типу тренду відбувається в подальшому аналізі — побудові моделі. Процеси зі змінною дисперсією (змінюється з часом) називають гетероске-

дастичними. Дисперсія може змінюватися за досить складним законом, який і потрібно знайти при створенні моделі процесу. Процеси з трендами або зі змінною дисперсією особливо характерні для нестійкої швидкозростаючої перехідної економіки.

Наявність нелінійностей. Однією з проблем визначення структури моделі є встановлення факту нелінійностей в досліджуваному процесі та їх типу. Економічна динаміка являє собою нелінійне явище. Нелінійність означає можливість неочікуваних змін напрямів розвитку процесів. В динамічних економічних системах нелінійність виявляється як підвищена реакція на зміну одних факторів і повною нечутливістю до інших.

Вплив сильних економік на слабкі. Однією із невід'ємних складових перехідного процесу є вплив на економічні процеси держави зі слабкою економікою

держав з вищим рівнем економічного розвитку. Слабкість ринку (як складової частини економіки) проявляється неможливістю ринковими механізмами розв'язувати деякі економічні задачі взагалі або найкращим способом. Більш розвинені економіки напряму впливають на ситуацію розвитку фінансово-економічних процесів. Встановлення ціни на природні ресурси, залучення інвестицій, підтримання експортно-імпортних відносин, надання кредитів відіграє значний вплив на формування процесів, тому при їх моделюванні потрібно враховувати фактори впливу цих країн.

Наприклад Росія має великий вплив на макроекономіку країн колишнього Радянського Союзу, а рівень промислового розвитку Сполучених Штатів Америки має прямий вплив на рівень промислового розвитку Канади та Великобританії.

Висока динаміка змін економічних процесів. На різних ділянках часового ряду може спостерігатись різна динаміка, тобто різна швидкість падіння або зростання значень змінної.

1. Постановка задачі

Необхідно вибрати клас математичних моделей для фінансово – економічних процесів, виконати їх розробку, виконати оцінювання якісних показників для побудованої моделі та прогнозу, та виконати порівняльний аналіз побудованих моделей.

2. Авторегресія з ковзним середнім порядком (p, q) (АРКС)

В загальному вигляді авто регресійне рівняння має вигляд:

$$y(k) = a_0 + \sum_{i=1}^p a_i y(k-i) + \sum_{j=1}^q b_j \varepsilon(k-j) + \varepsilon(k)$$

де $a_0 + \sum_{i=1}^p a_i y(k-i)$ – авторегресійна складова, $\sum_{j=1}^q b_j \varepsilon(k-j)$ – ковзне середнє, $\varepsilon(k)$ – випадкова величина.

Послідовність побудови рівняння авторегресії з ковзним середнім

Необхідно чітко розуміти, що АРКС може будуватися двома способами, які за своєю глибинною суттю відрізняються. Так трапилося, що фактично дві різні речі називають однією назвою.

Підхід № 1. Побудова АРКС (p, q) коли ковзне середнє будується по залишкам АР(p) рівняння моделі.

Крок 1. Оцінюються коефіцієнти рівняння моделі АР(p)

$$y(k) = a_0 + \sum_{i=1}^p a_i \cdot y(k-i) + e \tag{2.1}$$

де e – залишки моделі (resid).

Крок 2. По залишкам АР(p) моделі формують КС.

Крок 3. Визначення порядку КС(q).

Для цього будують ЧАКФ вектора для зберігання ковзного середнього та визначають q.

Крок 4. Оцінювання коефіцієнтів КС(q).

Фактично треба визначити коефіцієнти рівняння

$$resid(k) = mv(k) + \sum_{j=1}^q b_j \cdot mv(k-j) \tag{2.2}$$

Використовуючи рівняння (2.1) та (2.2) здійснюється побудова АРКС(p, q) по частинам, тобто спочатку коефіцієнти авто регресійної частини а після коефіцієнти при ковзних середніх. Можна одразу оцінити всі коефіцієнти $a_0, \dots, a_p, b_1, \dots, b_q$ рівняння (2.3). За теорією вважається що оцінювання коефіцієнтів окремо (спочатку авто регресійної частини, а потім КС) дає більш якісний результат.

$$y(k) = a_0 + \sum_{i=1}^p a_i \cdot y(k-i) + mv(k) + \sum_{j=1}^q b_j \cdot mv(k-j) \tag{2.3}$$

Підхід № 2. Побудова АРКС(p, q) коли КС будується по вихідному сигналу y.

Крок 1. Побудова КС по вихідному сигналу y – створюємо вектор mv.

Крок 2. Визначення порядку КС(q).

Для цього будують ЧАКФ вектора для зберігання ковзного середнього та визначають q. Треба зауважити, що на практиці в даному випадку (коли КС будується по вихідному сигналу y) дуже часто $p = q$.

Крок 3. Оцінювання коефіцієнтів $a_0, \dots, a_p, b_1, \dots, b_q$ АРКС(p, q), дивись рівняння (2.3).

При цьому бажано здійснити проміжне перетворення, перенести mv(k) в ліву частину рівняння (2.3). Інакше це відобразиться на якості отриманих коефіцієнтів.

$$y1(k) = y(k) - mvk = a_0 + \sum_{i=1}^p a_i \cdot y(k-i) + \sum_{j=1}^q b_j \cdot mv(k-j)$$

Обчисленні коефіцієнти за теорією повинні бути

$$\sum_{i=1}^p a_i < 1, \text{ а } \sum_{j=1}^q b_j \rightarrow 1.$$

Але на практиці ці умови не завжди виконуються. Погано це чи ні в кожному окремому випадку можна визначити проаналізувавши результати прогнозування за допомогою отриманої моделі на декілька кроків вперед. Якщо результати прогнозування прийнятні, то на невиконання умов, щодо коефіцієнтів моделі, можна закрити очі, якби мовити – для досягнення мети всі моделі гарні, за умови що вони дають гарні результати.

3. Моделі процесів з трендами

У випадку, коли $E[x(k)] \neq const$, тобто математичне сподівання змінюється в часі, то такий процес називають процесом з трендом або інтегрованим процесом (по аналогії із характером зміни сигналу на виході інтегратора) або процесом з одиничними коренями (відповідного характеристичного рівняння).

Під трендом будемо розуміти загальний довгостроковий напрям розвитку процесу, фактично він співпадає з поточним середнім значенням. Тренд може бути зростаючим або спадаючим, а за характером зміни в часі може бути детермінованим або стохастичним.

Детермінований тренд описують вибраною функцією, наприклад, поліномом від часу, сплайном, експонентою, комбінацією тригонометричних функцій та інше. Часто використовують поліноми від часу вигляду:

$$y(k) = a_0 + a_1 \cdot k + a_2 \cdot k^2 + \dots + a_m \cdot k^m + \varepsilon(k),$$

де k – дискретний час, який зв'язаний з неперервним реальним часом t через період реєстрації (дискретизації) даних: $t=kT_s$; $\varepsilon(k)$ – випадкова змінна, оцінку якої можна знайти після оцінювання рівняння: $\hat{\varepsilon}(k) = e(k)$, де $e(k)$ – похибка моделі. Очевидно, що після оцінювання моделі послідовність значень $\{e(k)\}$ буде містити всі коливання, що накладаються на тренд.

Якщо тренд відносно швидко змінює свій напрям розвитку і для нього важко знайти адекватне функціональне описання, то застосовують моделі випадкових трендів, які ґрунтуються на комбінаціях випадкових величин.

Для опису і прогнозування стохастичного тренду можна скористатись, наприклад, відомим рівнянням випадкового кроку із перетином (константою):

$$y(k) = a_0 + y(k-1) + \varepsilon(k)$$

розв'язок якого має вигляд:

$$y(k) = y_0 + ka_0 + \sum_{i=1}^k \varepsilon(i).$$

Сума $\sum_{i=1}^k \varepsilon(i)$ в правій частині останнього рівняння

описує випадкову складову тренду.

Прогнозування коливань, що накладаються на тренд. Коливання, що накладаються на тренд, або короткострокові зміни можна описати рівнянням авторегресії з ковзним середнім

$$y(k) = \sum_{i=1}^p a_i y(k-i) + \sum_{j=1}^q b_j \cdot \varepsilon(k-j) + \varepsilon(k)$$

4. Моделювання та прогнозування показника ВВП України

Валовий внутрішній продукт (ВВП) є основним показником результатів економічної діяльності на макrorівні, який характеризує сукупну ринкову вартість кінцевих товарів і послуг, вироблених підприємствами, організаціями і установами в певному періоді на економічній території країни.

ВВП є найбільш повним показником сумарного обсягу виробництва товарів та послуг в певній країні за певний період.

ВВП характеризує кінцеве споживання товарів та послуг, валове накопичення основного капіталу, змінення запасів матеріальних оборотних коштів, чисте надбання цінностей і сальдо експорту та імпорту товарів і послуг.

Тепер проаналізуємо та побудуємо моделі для опису ВВП України. Однією з головних проблем при побудові моделей цього ряду, те що вибірка містить лише 26 вимірів даних.

На рисунку 1 зображена АКФ ряду ВВП України.

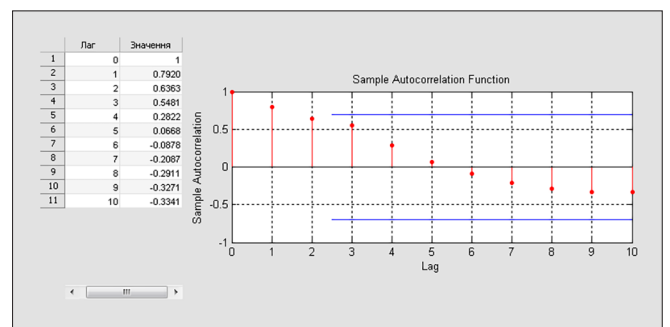


Рисунок 1. АКФ для ряду ВВП України

У зв'язку з малим об'ємом вибірки прогноз будемо робити лише на 1 крок вперед. На рисунку 2 зображена AR(1) для ряду ВВП України.



Рисунок 2. AR(1) модель для ряду ВВП України

Побудована модель АРКС(1.1) зображена на рисунку 3.

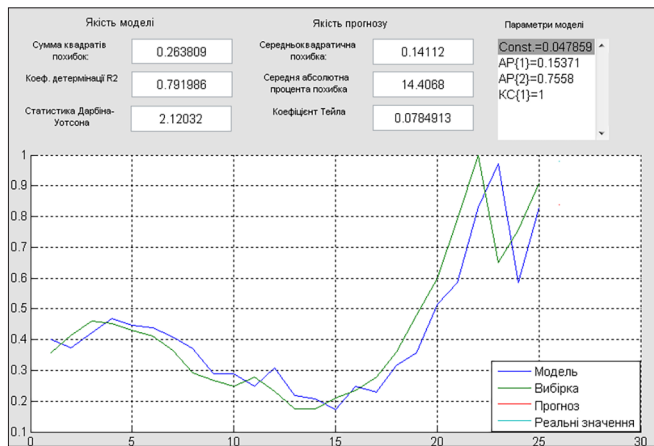


Рисунок 3. АРКС(1,1) для ряду ВВП України

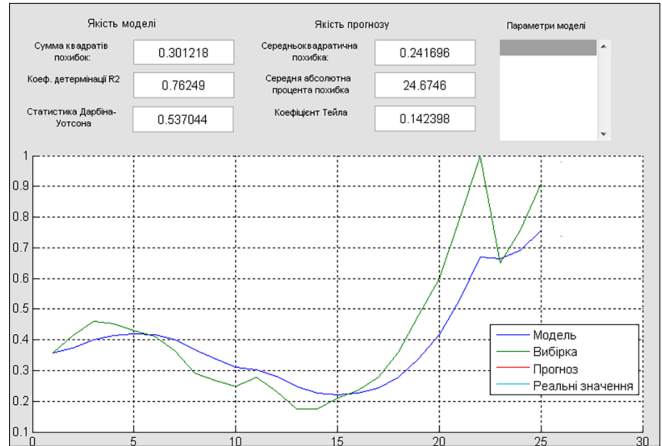


Рисунок 4. ЕЗ(0,3) модель для ряду ВВП України

Побудована модель ЕЗ з параметром альфа 0.3 для ряду ВВП України зображена на рисунку 4.

В таблиці 1 наведено порівняльний аналіз результатів моделювання ВВП України за допомогою різних моделей.

Таблиця 1

Аналіз результатів моделювання для ВВП України

Модель	SEE	R2	DW	СКП	САПП	U
АР(1)	0,2846	0,7754	1,902	0,0633	6,463	0,03375
АР(2)	0,2844	0,7757	1,9523	0,06167	6,296	0,03285
АРКС(2,1)	0,2638	0,7919	2,12	0,14112	14,41	0,07849
АРКС(2,3)	0,1853	0,8539	2,2179	0,3735	38,32	0,24005
ЕЗ(0,1)	0,9128	0,28	0,266	0,445	45,43	0,29787
ЕЗ(0,3)	0,3012	0,7625	0,537	0,2417	24,67	0,14239
АР(1) тренд	0,1896	0,8336	1,894	2,668	42,45	0,149

5. Висновки

У даній роботі були розглянуті основні підходи до побудови моделей та оцінок прогнозів фінансово – економічних процесів перехідного періоду.

Побудована модель для опису ВВП України. Для цього показника складно вибрати найкращу модель тому, що однієї моделі високі параметри якості для побудованої моделі, то в неї погані параметри оцінок якості прогнозу або навпаки. На мою думку, найкраще

ряд ВВП України описує модель АР(2). Можна звернути увагу на графік ВВП. У останні роки на графіку чітко виражений спад, а потім поступовий підйом. Це спричинено впливом світової фінансово – економічної кризи 2008–2009 років, яка не оминула і економіку України. Саме вплив таких неконтрольованих збурень впливає на побудову моделі та прогнозування, що набагато ускладнює цей процес.

Література

1. Бідюк П. І. Аналіз часових рядів / П. І. Бідюк, В. Д. Романенко, О. Л. Тимошук. — К.: Політехніка, 2010. — 317 с.
2. Бідюк П. І. Часові ряди: моделювання і прогнозування / П. І. Бідюк, О. І. Савенков, І. В. Баклан. — К.: ЕКМО, 2003. — 144 с.
3. Молчанов И. Н. Компьютерный практикум по начальному курсу эконометрики (реализация на Eviews): Практикум / Ростовский государственный экономический университет. / Молчанов И. Н., Герасимова И. А. — Ростов-н/Д., — 2001. — 58 с.

Кравчук Євгеній Сергійович

студент

Національний технічний університет України «Київський політехнічний інститут»

Кравчук Евгений Сергеевич

студент

Национальный технический университет Украины «Киевский политехнический институт»

Kravchuk Y.

student

National Technical University of Ukraine «Kyiv Polytechnic Institute»

**АНАЛІЗ ПРОТОКОЛІВ ВЗАЄМОДІЇ
У КЛІЄНТ-СЕРВЕРНІЙ АРХІТЕКТУРІ**
**АНАЛИЗ ПРОТОКОЛОВ ВЗАИМОДЕЙСТВИЯ
В КЛИЕНТ-СЕРВЕРНОЙ АРХИТЕКТУРЕ**
**ANALYSIS OF COMMUNICATION PROTOCOLS
IN CLIENT-SERVER ARCHITECTURE**

Анотація. Метою даної роботи є аналіз існуючих протоколів взаємодії веб-сервісів з подальшим виявленням переваг та недоліків кожного з протоколів та опис найбільш пасуючих випадків використання. Дана робота зосереджується як на будові кожного протоколу, так і на порівнянні його з іншими протоколами для ефективного аналізу.

Ключові слова: веб-сервіс, протокол взаємодії, SOAP, REST, RPC, XML-RPC.

Анотация. Целью данной работы является анализ существующих протоколов взаимодействия веб-сервисов с последующим выявлением преимуществ и недостатков каждого из протоколов и описание наиболее подходящих случаев использования. Данная работа сосредоточена как на строении каждого протокола, так и на сравнении его с другими протоколами для эффективного анализа.

Ключевые слова: веб-сервис, протокол взаимодействия, SOAP, REST, RPC, XML-RPC.

Summary. The aim of this research is the analysis of existing web-service communication protocols with subsequent detection of advantages and disadvantages of each of the protocols and the description of the most suitable use cases. This study focuses on both structure of each protocol and comparison with other protocols to achieve effective analysis.

Key words: web-service, communication protocol, SOAP, REST, RPC, XML-RPC.

Вступ

Сучасній мережі Інтернет характерна різноманітність додатків, що функціонують на різних вузлах мережі, апаратно-програмних платформах та використовують різноманітні технології та мови програмування. Якщо гіпотетично поділити Інтернет на декілька шарів, то виділяються щонайменше два концептуальних типа додатків — обчислювальні вузли, котрі реалізують нетривіальні функції, і прикладні веб-ресурси. В цей же час, другі частіше за все використовують послуги, що надають перші.

Для того, щоб зв'язати вищеописані сутності і надати одним додаткам можливість обмінюватися дани-

ми з іншими, були придумані веб-сервіси. Веб-сервіси по своїй суті — це реалізація абсолютно чітких інтерфейсів обміну даними між різними додатками, котрі написані не тільки різними мовами, але й розподілені на різних вузлах мережі.

Саме з появою веб-сервісів розвинулася ідея Service Oriented Architecture — сервіс-орієнтованої архітектури веб-додатків. Веб-сервіси виступають серверами, постачальниками послуг, для інших сервісів, котрі в цей час виступають клієнтами або споживачами.

Мета роботи

З огляду на ріст популярності веб-сервісів та сервіс-орієнтованої архітектури загалом, зростає важливість вибору протоколів взаємодії. Метою даною роботи є вибір найбільш доцільного протоколу взаємодії веб-сервісів для проектів різного масштабу та складності серед існуючих популярних протоколів, виявлення їх переваг та недоліків загалом та у контексті конкретних типів додатків.

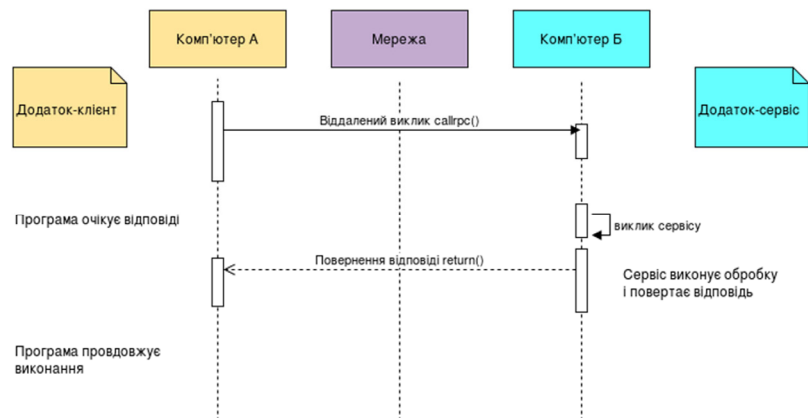


Рисунок 1. Діаграма послідовності RPC

Існуючі протоколи взаємодії веб-сервісів

На сьогоднішній день найбільш широке розповсюдження отримали SOAP, XML-RPC та REST. Вони відрізняються як контекстом, так і використанням. У світі XML існує два основних шляхи реалізації віддаленого виклику процедур: XML-RPC та SOAP. XML-RPC — протокол, котрий є пращуром SOAP і виділяється виключною простотою в застосуванні та використанні. SOAP — це протокол, в той час як REST — це архітектурний стиль. SOAP є добре продуманим протоколом, що використовується в веб індустрії і стандартизується World Wide Web Consortiumом (W3C). REST в свою чергу — це результат дисертації «Architectural Styles and the Design of Network-Based Software Architecture» 2000-го року, автором якої є Рой Філдинг. REST набирає популярності через свою простоту, масштабованість і архітектурну залежність від World Wide Web. Великі корпорації, як-то Google і Amazon, зокрема, використовують REST у переважній більшості своїх продуктів. Найголовнішою відмінністю між цими моделями є те, що SOAP є сильно-зв'язаною системою, водночас REST — слабозв'язана система.

RPC. XML-RPC

RPC (XML Remote Procedure Call) — віддалений виклик процедур з допомогою XML. Сама методика віддаленого виклику відома давно і використовується в таких технологіях як CORBA, DCOM, SOAP. RPC слугує для побудови розподілених клієнт-серверних обчислень.

На дуже спрощеному прикладі механізм роботи виглядає наступним чином: додаток, виконучи обробку деяких даних на локальному комп'ютері звертається до деякої процедури. Якщо її реалізація присутня в програмі, то процедура приймає параметри, виконує дію і повертає деякі дані. Якщо це віддалений виклик, то треба знати де буде виконуватися процедура.

Діаграма послідовності на рисунку 1 описує процес віддаленого виклику процедури між програмами на різних комп'ютерах у мережі.

Запит на виконання процедури разом з параметрами записується у вигляді XML-документа і передається по мережі посередництвом HTTP на інший комп'ютер, де з XML-документа вилучається ім'я процедури, параметри та інша потрібна інформація. Після завершення роботи процедури формується відповідь, і вона передається комп'ютеру, що надіслав запит.

Але формат обміну даними при класичній моделі RPC залишається бінарним, що означає складність у роботі з ним, якщо, наприклад, потрібно організувати роботу розподіленої системи, де між окремими ділянками мережі сотять фаєрволи та проксі-сервери. До того ж різні платформи мають свої тонкощі у використанні.

Ця проблема передувала створенню похідної технології XML-RPC компанією UserLand Software Inc. Передача даних у них виконується протоколом HTTP, формат даних — XML. Це знімає обмеження, накладені на конфігурацію мережі, так і на маршрут слідування пакетів, — виклики XML-RPC являють собою простий тип даних text/xml і вільно проходять через шлюзи там, де допускається ретрансляція HTTP-трафіку.

Повідомлення XML-RPC передаються методом POST протокола HTTP. Вони бувають трьох типів: запит, відповідь і повідомлення про помилку. Протокол передбачає сім простих типів даних і два складних, для передачі параметрів методу і повертаємих значень.

Більш складні типи даних, наприклад об'єкти, треба передавати у бінарному вигляді чи замінити структурами. Не дивлячись на те, що протокол є дещо застарілим, але він ще й досі використовується і має певні переваги та недоліки.

Таблиця 1

Основні характеристики технології XML-RPC

Характеристика	XML-RPC
Скалярні типи даних	Є
Структури	Є
Масиви	Є
Іменовані масиви та структури	Відсутні
Кодування, визначені розробником	Відсутні
Типи даних, визначені розробником	Відсутні
Деталізація помилок	Є
Легкість практичного застосування та засвоєння	Легкий для розуміння і практичного використання

SOAP

Це протокол для обміну веб-сервісами структурованою інформацією в комп'ютерних мережах. Мова для опису веб-сервісів WSDL (Web Service Description Language) використовується разом з SOAP, щоб зробити повідомлення доступними у Web посередництвом веб-сервісів. Загалом, SOAP сумісно з WSDL називається SOAP Web services.

Без протокола SOAP веб-сервіси не можуть працювати кросплатформенно, і зазвичай несумісні. SOAP був розроблений для заповнення цієї прогалини, полегшуючи розробнику конструювання веб-програми незалежно від вищезгаданих обмежень. SOAP в даний момент залежить від HTTP, що лежить в основі, але може використовувати інші протоколи передачі даних. Microsoft були першими, хто почав розроблювати SOAP, а далі й інші відомі корпорації почали вносити свій вклад у розробку.

Раніше в історії персональних комп'ютерів, база обробка даних складалася з одного персонального комп'ютера з використанням простих додатків, що залежали від локальних ресурсів. Розробки у сфері комп'ютерних мереж поступово популяризували концепції комп'ютерних мереж LAN, MAN та WAN, даючи можливість спілкуватися комп'ютерам, що фізично знаходяться у різних місцях.

Протоколи веб-сервісів дали можливість комп'ютерам взаємодіяти на однакових платформах, технологіях та операційних системах. Потреба у протоколі, незалежному від цих обмежень породила протокол SOAP, котрий використовує XML для спілкування поверх транспортного рівня.

Призначення SOAP-а — надати мінімальний транспортний функціонал, поверх якого можуть бути побудовані більш складні протоколи та процеси взаємодії.

Для розробки клієнта треба знати URL кінцевої точки сервера, яка відповідає за обробку повідомлень. Це стає ще легшим з використанням WSDL, форматом XML схеми, який надає можливість представити

мережеві сервіси у вигляді множини кінцевих точок, що обробляють повідомлення.

Офіційно протокол був стандартизований консорціумом W3C у 2003 році, використовується для представлення бізнес даних через інтерфейси і сервіси, що обмінюються цілими документами чи відображають дані на об'єкт, використовуючи назви методів, вхідні та вихідні параметри. Наступний уривок коду є прикладом типового SOAP повідомлення:

POST /InStock HTTP/1.1

Host: www.example.org

Content-Type: application/soap+xml; charset="utf-8"

Content-Length: 299

<?xml version="1.0">

<soap: Envelope xmlns: soap="http://w3.org/2003/05/soap-envelope">

<soap: Header>

</soap: Header>

<soap: Body>

<m: GetStockPrice xmlns: m=

"http://www.example.org/stock">

<m: StockName>IBM

</m: StockName>

</m: GetStockPrice>

</soap: Body>

</soap: Envelope>

Використані елементи мають наступні значення:

- *Envelope* — є індикатором початку та кінця повідомлення;
- *Header* — містить опціональні допоміжні дані для обробки повідомлень;
- *Body* — складається з головних XML даних;

Щоб бути незалежними від HTTP, SOAP має перевагу, а саме, елемент *attachment* — надсилання прикріплених файлів, що часто вважають найкращим рішенням для сервісів, що мають справу з прикріпленими файлами.

Через свою надійність та рівень безпеки протоколи SOAP широко використовуються у банківських системах та корпоративних додатках.

REST

REST — це архітектура для розробки веб-сервісів, що намагається імітувати архітектури, котрі використовують HTTP чи схожі протоколи, шляхом обмеження інтерфейсів набором стандартних і добре знайомих розробникам операцій (наприклад, GET, PUT/PATCH, POST та DELETE для HTTP).

Акцент зроблений на взаємодії з ресурсами, що мають збережений стан, а не з повідомленнями та

операціями. Можна сказати, що ця архітектура розроблена для того, щоб показати, що існуючого HTTP достатньо для побудови веб-сервісу, та демонстрації масштабування.

Рой Філдинг, один із основних авторів HTTP, визначив у своїй докторській дисертації набір архітектурних принципів для Web, котрі називаються REST. Ці принципи фокусуються на системних ресурсах, визначаючи як ресурси можуть бути адресовані та передані по HTTP.

Головною ідеєю REST було скоріш використання добре розвинуеного HTTP для передачі даних між пристроями, ніж використання протоколу, побудованого поверх рівня HTTP, для передачі повідомлень. Додаток, розроблений слідуючи принципам даного архітектурного підходу, буде використовувати HTTP для виконання викликів між пристроями не спираючись на складні об'єктні механізми на кшталт CORBA (Common Object Request Broker Architecture), RPC (Remote Procedure Call), чи SOAP. Відповідно, REST-додатки використовують функції HTTP запиту для надсилання, зчитування та видалення даних, використовуючи повну функціональність HTTP CRUD (Create, Read, Update і Delete) операцій. До того ж REST може використовувати HTTPS, надаючи безпечне передавання даних.

На відміну від SOAP, REST принципи легкі для розуміння та застосування розробнику, котрий знайомий з використанням HTTP. CRUD операції разом з функціями HTTP REST та відповідними SQL операціями показані у таблиці 2.

Таблиця 2

Відповідність HTTP-функцій SQL-операціям

| CRUD-операції | Ключові слова REST (HTTP) | Оператори SQL |
|---|---------------------------|---------------|
| CREATE — створити чи додати нові сутності | POST | INSERT |
| UPDATE — оновити чи редагувати існуючі дані | PUT | UPDATE |
| READ — зчитати, отримати дані | GET | SELECT |
| DELETE — видалити існуючі дані | DELETE | DELETE |

Порівняльний аналіз протоколів

З огляду на те, що SOAP є похідним від XML-RPC і, водночас, виступає основною альтернативою REST, доцільним є порівняння технології SOAP окремо з XML-RPC та REST.

Порівняння SOAP та XML-RPC

Порівняно з SOAP, XML-RPC має простішу архітектуру, простіший для розуміння, ніж SOAP. По-

рівняння за основними характеристиками наведені у таблиці 3.

Таблиця 3

Порівняння SOAP та XML-RPC

| Характеристика | SOAP | XML-RPC |
|--|--|---|
| Скалярні типи даних | Є | Є |
| Структури | Є | Є |
| Масиви | Є | Є |
| Іменовані масиви та структури | Є | Відсутні |
| Детальний опис помилок | Є | Є |
| Кодування, визначені розробником | Є (US-ASCII, UTF-8, UTF-16) | Відсутні |
| Типи даних, визначені розробником | Є | Відсутні |
| Можливість вказати отримувача | Є | Відсутня |
| Деталізація помилок | Є | Є |
| Потребує «розуміння» клієнтом | Так | Ні |
| Легкість практичного застосування та засвоєння | Потребує час для вивчення та розуміння | Легкий для розуміння і практичного використання |

Порівняння SOAP та REST

У таблиці 4 наведено відповідності основних властивостей REST та SOAP.

Таблиця 4

Порівняння REST та SOAP

| REST | SOAP |
|--|---|
| Передбачає модель обміну даними типу точка-точка, що виключає можливість використання у розподіленому середовищі з вузлами-посередниками | Розроблений для роботи у розподілених середовищах |
| Потребує мінімального набору інструментів, необхідна лише підтримка протоколу HTTP. | Потребує широкого набору інструментів та проміжного шару програмного забезпечення |
| URL ідентифікує ресурс, який отримується/видаляється/оновлюється | Вміст повідомлення визначає операцію |
| Формальні стандарти опису не набули широкого використання | Добре визначені механізми опису інтерфейсів, наприклад WSDL+XSD |
| Відсутність обмежень щодо корисного змісту | Корисний зміст має відповідати SOAP-схемі |
| Вбудована обробка помилок | Відсутність обробки помилок |
| Працює тільки з HTTP | Може використовувати різні транспортні протоколи |
| Менш детальний | Високий рівень деталізації |

Надодачу, у архітектурному стилі REST застосовуються лише добре зарекомендовані стандарти, наприклад HTTP, SSL. Методи DELETE та PUT часто вимкнені фаєрволами, що призводить до складнощів з безпекою. Водночас SOAP підтримує широкий спектр стандартів для організації безпеки, надійності, підтримки транзакцій.

Використання протоколів

Так RPC та XML-RPC роками успішно демонструють доречність простоти своєї концепції в додатках, де акцент зроблено на швидкість передачі простих структур та виконання простих логічних операцій. В цьому випадку простота та обмеження виступають перевагами, оскільки вони значно зменшують труднощі реалізації протоколів і тестування їх сумісності. До того ж низький поріг складності вивчення дозволяє створювати функціональні додатки навіть початківцю. Надодачу, спираючись на стрімкий ріст популярності функціонального стилю програмування, можна зробити примушення, що стиль запитів у вигляді операцій, котрий надає RPC та XML-RPC, дає можливість описаним протоколам повернути популярність, властиву їм раніше чи набути ще більшого розповсюдження, ніж дотепер.

При виникненні потреби обміну складними структурами даних, формалізувати взаємодію клієнта та сервера, ввівши чіткі її контракти, доцільним є розглянути SOAP. Даний протокол слідує формальним корпоративним підходам, добре сумісний з популярними великою кількістю транспортних протоколів, підтримка безпеки та авторизації є вбудованою функцією протокола, присутня можливість повного опису з використанням WSDL, що дозволяє реалізувати взаємодію типу точка-точка у мережі і надає можливість іншим пристроям у мережі «розуміти» веб-сервіс без втручання людини. Але даний протокол не позбавлений недоліків: не є доречним його впровадження у системах з повільним з'єднанням; протокол має труднощі реалізації, і, як наслідок, віне непопулярний серед веб-розробників та розробників мобільних додатків. Тобто не варто використовувати SOAP, коли треба надати API стороннім розробникам або коли пропускна здатність Інтернет-з'єднання низька. SOAP широко використовується у фінансових сервісах, платіжних системах та комунікаційних сервісах. Повсюдне використання даного протоколу у цих сферах дає впевненість у подальшій його популяризації та використанні у майбутньому.

Альтернативою протоколу SOAP є архітектурний стиль REST, що слідує філософії «Відкритого Інтернету». Його перевагами є відносна легкість у реалізації, чітке розділення серверної та клієнтської реалізацій,

можливість кешування відповідей та різні формати відповідей на запити, як-то JSON чи XML. Так як REST на пряму залежить від HTTP, для передачі даних між клієнтом та сервером треба менше зусиль, порівняно з SOAP чи RPC–XML. Переваги REST зробили його вибором багатьох розробників соціальних мереж, медіа-сервісів, мобільних додатків. Корпорації на кшталт Twitter, LinkedIn та Slack використовують REST, як основу своїх додатків. Недоліками REST є те, що він працює лише поверх HTTP та складність впровадження безпеки та авторизації поверх нього. Цей підхід не варто використовувати, коли вимагається наявність чіткого контракту між клієнтом та сервером або відбувається виконання транзакцій, котрі включають багаторазові виклики.

Висновки

Вибір протоколів взаємодії є важливим архітектурним рішенням і потребує значної уваги. Результат цього вибору має великий вплив на межі подальшого розвитку проекту та висуває ряд вимог до архітектури додатку. Безперечно, кожна з технологій має переваги та недоліки, але аналіз привів до висновку, що розглянуті протоколи є різними підходами і всі вони можуть бути доречними та принести користь, розмір якої залежить в першу чергу від контексту.

Сфери використання XML-RPC, SOAP та REST варіюються від невеликих систем з простою логікою до масштабних корпоративних додатків. Часто використовується комбінування даних технологій для досягнення оптимальної продуктивності чи надання різних API для задоволення потреб як розробників, котрі його використовують, так і інших систем, що взаємодіють з веб-сервісом.

На основі аналізу визначено у яких випадках доречніше використовувати ту чи іншу технологію, описані переваги та недоліки кожного з підходів. З огляду на це, можна сказати, що вибір пасуючої технології залежить в першу чергу від цілей та задач, котрі буде реалізувати система, довгостроковості її роботи.

Підсумовуючи проведений аналіз, можна сказати, що не існує наріжного каменя у виборі протоколу взаємодії. Для вибору максимально пасуючого протоколу варто аналізувати задачі, котрі буде вирішувати система, визначити перспективи розвитку розроблюваного проекту.

Лісафін Володимир Петрович

*кандидат технічних наук, доцент, доцент кафедри транспорту і зберігання нафти і газу
Івано-Франківський національний технічний університет нафти і газу*

Лисафин Владимир Петрович

*кандидат технических наук, доцент, доцент кафедры транспорта и хранения нефти и газа
Ивано-Франковский национальный технический университет нефти и газа*

Lisafin V.

*Ph.D., associate professor of the department of transportation and storing of oil and gas Ivano-Frankivsk
National Technical University of Oil and Gas*

ОЦІНЮВАННЯ ВПЛИВУ АТМОСФЕРНИХ ОПАДІВ НА ОБЛІК НАФТИ В РЕЗЕРВУАРАХ З ПЛАВАЮЧОЮ ПОКРІВЛЕЮ

ОЦЕНКА ВЛИЯНИЯ АТМОСФЕРНЫХ ОСАДКОВ НА УЧЕТ НЕФТИ В РЕЗЕРВУАРАХ С ПЛАВАЮЩЕЙ КРЫШЕЙ

EVALUATION OF PRECIPITATION IMPACT ON OIL GAUGING IN FLOATING ROOF TANKS

Анотація. Досліджено вплив атмосферних опадів на облік нафти в резервуарах з плаваючою покрівлею на нафтоперекачувальних станціях магістральних нафтопроводів.

Аннотация. Исследовано влияние атмосферных осадков на учет нефти в резервуарах с плавающей крышей на нефтеперекачивающих станциях магистральных нефтепроводов.

Summary. Precipitation impact on oil gauging in floating roof tanks at oil pumping stations of main oil pipelines was investigated.

Облікові операції є важливою складовою процесу транспортування нафти магістральними нафтопроводами — нафта є цінним вантажем. Підвищення точності вимірювання, контролю та обліку маси нафти потребує рішення правових, організаційних та технічних питань щодо розробки оптимально вигідних процедур виконання операцій з приймання та зберігання продукту в резервуарах. Без обліку нафти неможливо забезпечити оптимальний технологічний режим роботи магістрального нафтопроводу, особливо за неповного завантаження.

Згідно з нормативними документами, при виконанні обліково-розрахункових операцій застосовують, як правило, непрямі методи вимірювання об'єму та маси нафти та нафтопродуктів.

Облік нафти при прийманні (здаванні) проводять за допомогою вузлів обліку динамічним об'ємно-масовим методом (на потоці), що є основним методом (комерційний облік нафти). При оперативному контролі

зокрема застосовується об'ємно — масовий статичний метод. Суть цього методу полягає у визначенні об'єму за результатами вимірювання рівня нафти та даних таблиць поінтервального градуювання резервуара, а також густини продукту за однакових (або зведених до однакових) умовах. Масу продукту визначають як добуток об'єму і густини [1, с. 452–462].

На нафтоперекачувальних станціях магістральних нафтопроводів використовуються різні типи резервуарів — вертикальні сталеві, з понтоном та плавучою покрівлею [1, с. 271–278].

З метою скорочення втрат нафти і нафтопродуктів від випаровування широке розповсюдження знайшли резервуари з плаваючою покрівлею (РВСПП).

Прикладом застосування РВСПП можуть бути резервуари, що експлуатуються в системі трубопроводного транспорту нафти України, при чому спостерігається загальна тенденція до збільшення частки таких резервуарів. Це, в основному резервуари

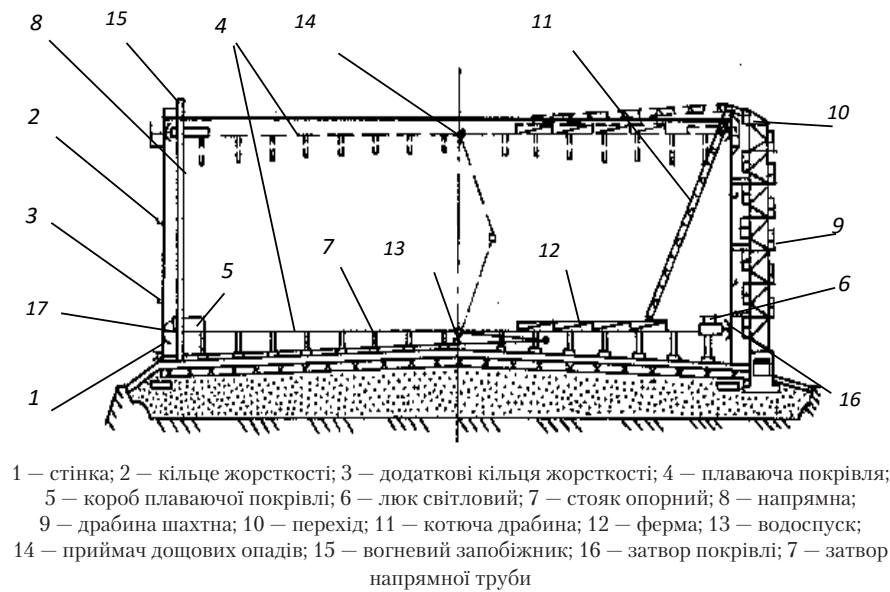


Рис. 1 — Схема резервуара з плаваючою покрівлею місткістю 50000 м³ (складено автором на підставі [1, с. 277])

об'ємом 20000 м³ (МНТ «Південний», ЛВДС «Броди» та ін.), 50000 м³ (нафтоперекачувальна станція «Лисичанськ»), 75000 м³ (ЛВДС «Броди») [2, с. 145–148].

На рисунку 1 наведено схему резервуара РВСПП — 50000.

Плаваюча покрівля, яка запобігає випаровуванню нафти, складається з понтонного кільця і плоскої центральної частини. Це найбільш поширений тип плаваючої покрівлі, досить простої і достатньо надійної.

Покрівля складається з плоскої центральної частини 4 і зовнішнього понтонного кільця з пустотілими герметичними коробами 5, що утворюють відсіки. Плоска центральна частина покрівлі виготовлена з листової сталі товщиною 5 мм. Вона має нахил 1:100 від понтонного кільця до приймача опадів.

З метою запобігання повороту плаваючої покрівлі при її пересуванні передбачені дві діаметрально розташовані напрямні стійки 8 з труб діаметром 530 мм.

На резеруарі установлений комплект обладнання, що забезпечує виконання необхідних технологічних операцій, прилади дистанційного вимірювання рівня, сигналізатор максимального рівня нафти і знижений пробовідбірник та ін.

Діаметр плаваючої покрівлі на 400 мм менший від діаметра резервуара. Зазор між покрівлею і стінкою ущільнюється затвором, конструкція якого підбирається, виходячи з умов експлуатації. Також ущільнюється зазор між напрямними трубами і покрівлею.

Для відведення атмосферних опадів з поверхні плаваючої покрівлі резеруар обладнаний

системою водоспуску. Водоспуск складається з приймача опадів, системи труб, що з'єднані між собою самоущільнюючими сальниковими шарнірами і нижнього поворотного вузла.

Дощові опади потрапляючи на покрівлю за рахунок нахилу покрівлі концентруються у її центральній частині, де встановлений приймач опадів DN150. Далі через систему водоспуску вони видаляються за межі резервуара.

Інтенсивність опадів на території України нерівномірна. Їх добова кількість знаходиться у межах від 58 мм/д до 178 мм/д [3, додаток А].

Звичайно, що опади з покрівлі не можуть миттєво потрапити у водоспускну систему (необхідний час для їх підтікання до приймача

опадів). Окрім того, слід врахувати нерівності покрівлі (хлопуни), через які частина води залишиться на покрівлі, плівку вологи, яка поступово висихатиме, тобто вага покрівлі під час опадів буде відрізнятись від її ваги у сухому стані.

При визначенні рівня нафти в сучасних резеруарах типу РВСПП використовують радарні пристрої, які, скануючи рівень нафти у напрямній трубі, перетворюють його значення у рівень наливу нафти.

Розглянемо схему затвору між плаваючою покрівлею і стінкою резервуара (рис. 2).

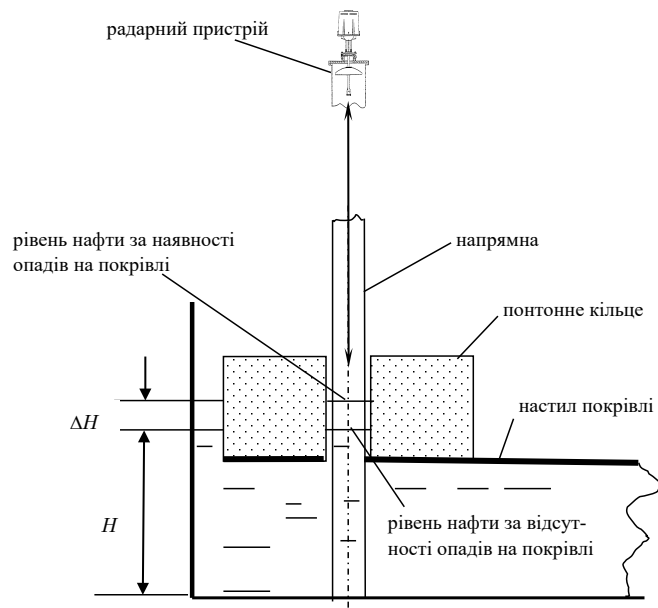


Рис. 2. Схема затвору плаваючої покрівлі (ущільнення умовно не показані) (складено автором на підставі [3, с. 147])

Введемо наступні позначення:

G — вага покрівлі;

V_1 — об'єм нафти, що витискається покрівлею за відсутності дощових опадів;

V_2 — об'єм нафти, що витискається покрівлею за наявності води на ній;

ρ_n — густина нафти;

ρ_w — густина води.

Застосовуючи закон Архімеда, можемо записати баланс сил:

за відсутності води на покрівлі

$$G = V_1 \rho_n g, \quad (1)$$

за наявності води на покрівлі

$$G + V_w \rho_w g = V_2 \rho_n g. \quad (2)$$

Шляхом математичних перетворень знаходимо

$$V_2 - V_1 = V_w \frac{\rho_w}{\rho_n}. \quad (3)$$

Обчислимо площі зазору між покрівлею і корпусом резервуара, а також площу двох напрямних

$$S_3 = \frac{\pi}{4} (D_1^2 - D_2^2); \quad (4)$$

$$S_n = \frac{\pi}{2} d^2, \quad (5)$$

де D_1 , D_2 , d — діаметри резервуара, покрівлі та напрямних відповідно.

Тоді ΔH (рис. 2) буде дорівнювати

$$\Delta H = \frac{V_2 - V_1}{S_3 + S_n}. \quad (6)$$

Оцінімо цю величину для резервуара з плаваючою покрівлею РВСПП-50000 за наступних даних:

– густина нафти — 860 кг/м³;

– густина води — 1000 кг/м³;

– діаметр покрівлі — 60,3 м;

– діаметр резервуара — 60,7 м;

– діаметр прямої труби — 0,5 м;

– об'єм води на покрівлі — 1 м³.

Виконуємо обчислення.

Площа зазору

$$S_3 = \frac{\pi}{4} (60,7^2 - 60,3^2) = 37,994 \text{ м}^2.$$

Площа напрямних

$$S_n = \frac{\pi}{2} 0,5^2 = 0,3925 \text{ м}^2.$$

Різниця об'ємів

$$V_2 - V_1 = 1 \frac{1000}{860} = 1,16 \text{ м}^3.$$

Різниця рівнів

$$\Delta H = \frac{1,16}{37,994 + 0,3925} = 0,03 \text{ м}.$$

Як бачимо, різниця рівнів у напрямній трубі за відсутності води на покрівлі і за її наявності складає значну величину (точність сканування рівня рівнеміром становить ± 1 мм), що слід враховувати при обліку нафти об'ємно — масовим методом. Також слід враховувати наявність снігу за від'ємних температур повітря, за яких він при сильних опадах не встигає розтанути і тим самим створює додаткове навантаження на покрівлю, що в свою чергу призводить до підвищення рівня нафти у напрямній трубі. Таким чином необхідно вводити корекцію показників пристроїв визначення рівня з метою виключення похибок при облікових операціях в резервуарах з плаваючою покрівлею.

Література

1. Середюк М. Д. Трубопровідний транспорт нафти і нафтопродуктів: [підруч. для студ. вищ. навч. закл.] / Середюк М. Д., Якимів Й. В., Лісафін В. П. — Івано-Франківськ: Кременчук, 2001. — 517 с.
2. Лісафін В. П. Проектування та експлуатація складів нафти і нафтопродуктів: [підруч. для студ. вищ. навч. закл.] / В. П. Лісафін, Д. В. Лісафін. — Івано-Франківськ: Факел, 2006. — 597 с.
3. Каналізація. Зовнішні мережі та споруди. Основні положення проектування. ДБН В.2.5-75: 2013. — Київ, 2013. — 128 с.

Магас Валентин Васильович

студент

Національний технічний університет України «Київський політехнічний інститут»

Магас Валентин Васильевич

студент

Национальный технический университет Украины «Киевский политехнический институт»

Mahas V.

student

National Technical University of Ukraine «Kyiv Polytechnic Institute»

КЛАСТЕРНИЙ АНАЛІЗ ПРОСТОРОВИХ БАЗ ДАНИХ ОНКОХВОРИХ

КЛАСТЕРНЫЙ АНАЛИЗ ПРОСТРАНСТВЕННЫХ БАЗ ДАННЫХ ОНКОБОЛЬНЫХ

CLUSTER ANALYSIS OF SPATIAL CANCER DATABASES

Анотація. Метою даної роботи є застосування засобів інтелектуального аналізу даних для виявлення прихованих закономірностей та зв'язків у просторових базах даних онкохворих пацієнтів. Дана робота зосереджується як на дискретних так і неперервних просторових медичних базах даних, до яких застосовуються кращі методи кластеризації для визначення максимально корисних кластерів.

Ключові слова: *k-means*, інтелектуальний аналіз просторових даних, алгоритми кластеризації, дендограма, просторовий розподіл.

Анотация. Целью данной работы является применение средств интеллектуального анализа данных для выявления скрытых закономерностей и связей в пространственных базах данных онкобольных пациентов.

Ключевые слова: *k-means*, интеллектуальный анализ пространственных данных, алгоритмы кластеризации, дендограмма, пространственное распределение.

Summary. The aim of this research is the application of data mining for detecting hidden patterns and relationships in spatial cancer databases. This study focuses on discrete and continuous spatial medical databases on which clustering techniques are applied and the efficient clusters were formed.

Key words: *k-means*, spatial data mining, clustering algorithm, dendrogram, spatial distribution.

Вступ

Сучасний період розвитку суспільства характеризується значним впливом на нього інформаційних технологій. Не стала винятком і медицина, яка сьогодні набула абсолютно нових рис. Поміж всієї ширини застосування новітніх технологій та засобів у даній сфері, чільне місце посідає інтелектуальний аналіз даних.

Інтелектуальний аналіз даних — збірна назва, що використовується для позначення сукупності методів виявлення в даних раніше невідомих, нетривіальних, практично корисних і доступних інтерпретації знань, необхідних для прийняття рішень в різних сферах людської діяльності.

У зв'язку бурхливим накопиченням медичних даних, особливо гостро постає проблема застосування аналізу даних для їх опрацювання та подальшого прогнозування, на їх основі. В свою чергу необхідність визначення прихованих тенденцій поширення певного захворювання на тій чи іншій території, виявлення неочевидних зв'язків між факторами що його спричиняють та подальшого прогнозування можливого розвитку подій з кожним роком набуває все чіткіших рис. Дуже актуальним виглядає дослідження просторової бази даних онкохворих, адже дана хвороба відзначається значною смертністю серед хворих, як результат, кожного року помирає понад 15 мільйонів людей у цілому світі.

Постановка задачі

Основні задачі, що були поставлені в рамках даної статті — підбір алгоритмів та програмних інструментів для аналізу бази просторових медичних даних, а також порівняння отриманих результатів для різних алгоритмів, як демонстрація можливостей застосування просторового аналізу даних в областях «e-Health» та «e-Medicine», що нині активно розвиваються.

Вибір алгоритмів для аналізу

Кластеризація (або кластерний аналіз) — це задача розбиття множини об'єктів на групи, які називаються кластерами. У середині кожної групи повинні виявитися «схожі» об'єкти, а об'єкти різних групи повинні бути якомога більш відмінні. Головна відмінність кластеризації від класифікації полягає в тому, що перелік груп чітко не заданий і визначається в процесі роботи алгоритму.

Кластер має наступні математичні характеристики: центр, радіус, середньоквадратичне відхилення, розмір кластера.

Алгоритм *k*-means

Найбільш поширений серед ітеративних методів алгоритм *k*-середніх, також званий швидким кластерним аналізом. На відміну від ієрархічних методів, які не вимагають попередніх припущень щодо числа кластерів, для можливості використання цього методу необхідно мати гіпотезу про найбільш ймовірну кількість кластерів. Алгоритм *k*-середніх будує *k* кластерів, розташованих на великих відстанях один від одного. Основний тип задач, які вирішує алгоритм *k*-середніх, — наявність припущень (гіпотез) щодо числа кластерів, при цьому вони повинні бути різні настільки, наскільки це можливо. Вибір числа *k* може базуватися на результатах попередніх досліджень, теоретичних міркуваннях або інтуїції.

Метод базується на мінімізації суми квадратів відстаней між кожним спостереженням та центром його кластера, тобто функції.

$$\sum_{i=1}^k \sum_{x \in D_i} x_i - c_i^2$$

де D_i — набір векторів що належать до *i*-го кластеру, а c_i — середнє значення цих векторів

$$c_i = \frac{\sum_{k=1}^{N_i} x_k}{N_i}, \quad x_k \in D_i$$

Основна ідея полягає в тому, що на кожній ітерації заново вираховується центр мас, для кожного кластера, потім вектори розбиваються на нові класи, відповідно до того який з отриманих центрів виявився ближчим за метрикою.

Алгоритм BIRCH

Завдяки узагальненому вигляду кластерів, швидкість кластеризації збільшується, алгоритм BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies) при цьому володіє великим масштабуванням.

У цьому алгоритмі реалізований двоетапний процес кластеризації.

В ході першого етапу формується попередній набір кластерів. На другому етапі до виявлених кластерів застосовуються інші алгоритми кластеризації — придатні для роботи в оперативній пам'яті.

Аналогія, що описує цей алгоритм: якщо кожен елемент даних уявити собі як намистину, що лежить на поверхні столу, то кластери намистин можна «замінити» тенісними кульками і перейти до більш детального вивчення кластерів тенісних кульок. Число намистин може виявитися досить велике, проте діаметр тенісних кульок можна підібрати таким чином, щоб на другому етапі можна було, застосувавши традиційні алгоритми кластеризації, визначити дійсну складну форму кластерів.

Алгоритм DBSCAN

Алгоритм DBSCAN — один з перших алгоритмів кластеризації щільнісним методом. В основі цього алгоритму лежить кілька визначень:

- ϵ -околицею об'єкта називається околиця радіуса ϵ деякого об'єкта.
- Кореневим об'єктом називається об'єкт, ϵ -околиця якого містить не менше деякого мінімального числа MinPts об'єктів.
- Об'єкт p безпосередньо щільно-досяжний з об'єкта q якщо p знаходиться в ϵ -околиці q і $q \in$ кореневим об'єктом.
- Об'єкт p щільно-досяжний з об'єкта q при заданих ϵ і MinPts, якщо існує послідовність об'єктів p_1, \dots, p_n , де $p_1 = q$ і $p_n = p$, така що $p_i + 1$ безпосередньо щільно-досяжний з p_i , $1 \leq i \leq n$.
- Об'єкт p щільно-з'єднаний з об'єктом q при заданих ϵ і MinPts, якщо існує об'єкт o такий, що p і q щільно-досяжні з o .

Для пошуку кластерів алгоритм DBSCAN перевіряє ϵ -околиця кожного об'єкта. Якщо ϵ -околиця об'єкта p містить більше точок ніж MinPts, то створюється новий кластер з кореневим об'єктом p . Потім DBSCAN ітеративно збирає об'єкти безпосередньо щільно-досяжні з кореневих об'єктів, які можуть привести до об'єднання кількох щільно-досяжних кластерів. Процес завершується, коли ні до одного кластеру не може бути додано жодного нового об'єкта. Хоча, на відміну від методів розбиття, DBSCAN не вимагає заздалегідь вказувати число одержуваних класте-

рів, виникне потреба у вказівках значень параметрів ϵ і MinPts, які безпосередньо впливають на результат кластеризації. Оптимальні значення цих параметрів складно визначити, особливо для багатовимірних просторів даних.

Отже, з цілого різноманіття методів та алгоритмів кластеризації для експериментальних досліджень було відібрано три. З ітеративних алгоритмів — k -means, з ієрархічних зупинимося на алгоритмі BIRCH, а з щільнісних оберемо DBSCAN. Такий підбір дозволить нам максимально якісно опрацювати дані та провести їх аналіз.

Дані для експерименту

В якості просторової бази онкохворих пацієнтів використаємо SEER Database, яка визначається значною кількістю записів, що являється ключовим фактором для успішного аналізу та подальшого прогнозування а його основі, а саме у ній мітяться дані про онкохворих жителів США, що збиралися протягом понад 40 років. Дані були поділені на дві групи просторові і непросторові.

Набір просторових даних містить координати розташування, зображення дистанційного зондування та іншу географічну інформацію. До основних непросторових даних відносились: стать, вік, сімейний стан, висота, вага. Досліджувались основні різновиди захворювання такі як: рак легень, рак нирок, рак горла і т.д.

Основний інтерес у вирішенні даної задачі полягає у безпосередньому застосування методів ІАПД у медицині. Таким чином, основний акцент буде зроблено на географічну прив'язку пацієнтів, їх розташування та закономірності розподілу по території.

Реалізація та результати

Для реалізації алгоритмів кластеризації були застосовано мову програмування Python з цілою низкою доповнень та бібліотек, зокрема: Pandas, NumPy, SciPy, Scikit-learn, Matplotlib. Перші три бібліотеки застосовувалися переважно для зручного представлення вхідних даних. Scikit-learn став нам у нагоді для безпосереднього аналізу даних. Остання з бібліотек стала незамінною для візуалізації результатів, а саме, при побудові діаграм та графіків. Результати роботи кожного алгоритмів наведено на рисунках 1–3, осями яких виступають географічна довгота і широта.

Як можна бачити на малюнку вище, методом k -means весь набір даних був поділений на 2 кластери. У свою чергу, методом DBSCAN було створено 3 кластери. Алгоритм ієрархічної кластеризації виділив 4 кластери на тому ж наборі вхідних даних.

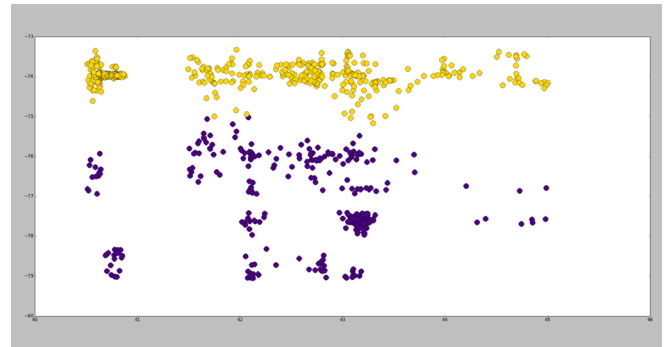


Рисунок 1. Результати кластеризації методом k -means

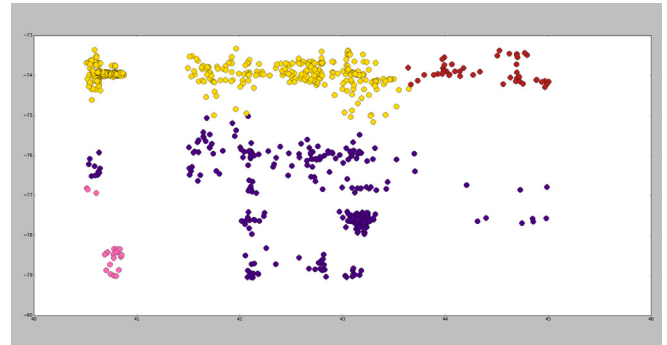


Рисунок 2. Результати кластеризації методом BIRCH

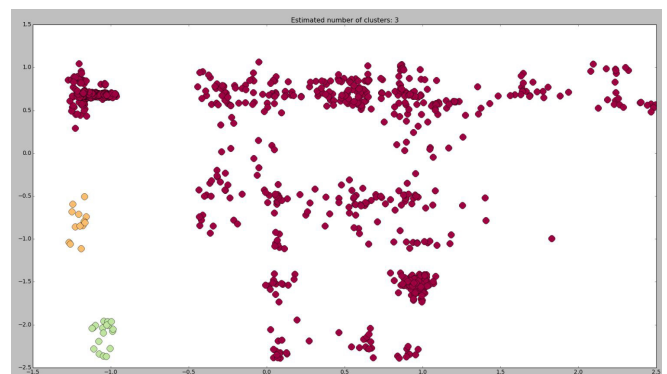


Рисунок 3. Результати кластеризації методом DBSCAN

Для оцінки роботи алгоритмів застосуємо наступні критерії порівняння:

- **Однорідність** (homogeneity) — характеристика, що показує в якій мірі кожен кластер містить члени єдиного класу. Визначається за наступною формулою.
- **Повнота** (completeness) — характеристика, що показує в якій мірі всі члени даного класу присвоєні одному кластеру.
- **V-міра** (V-Measure) — міра заснована на ентропії, яка в явному вигляді визначає наскільки критерії однорідності та повноти були задоволені. Дана характеристика обчислюється, як середнє гармонійне різниці однорідності й повноти. Дана
- **Скоригований індекс** (Adjusted Rand Index) — міра подібності між двома кластерами даних. З математичної точки зору скоригований індекс пов'язаний

с точністю і має місце навіть тоді, коли мітки класу не використовуються.

- **Взаємна інформація** (Mutal information) — це функція, що вимірює узгодженість кількох варіантів ігноруючи перестановки. Тобто, скоригована міра взаємної інформації тісно пов'язана зі зміною даних в процесі кластеризації.
- **Коефіцієнт силуету** (Silhouette Coefficient) — міра створена для оцінки моделі з чіткіше визначеним набором кластерів.

Результати порівняння алгоритмів зведемо у таблиці 1.

Таблиця 1

Результати порівняння алгоритмів

| Характеристика | K-means | DBSCAN | BIRCH |
|--------------------------------|---------|--------|-------|
| Однорідність | 0.917 | 0.953 | 0.960 |
| Повнота | 0.766 | 0.883 | 0.812 |
| V-міра | 0.881 | 0.917 | 0.855 |
| Скоригований індекс | 0.966 | 0.952 | 0.914 |
| Скоригована взаємна інформація | 0.855 | 0.883 | 0.896 |
| Коефіцієнт силуету | 0.722 | 0.626 | 0.691 |

Отже, проаналізувавши отримані результати роботи (таблиця 1) алгоритмів можна сказати, що щільніший алгоритм DBSCAN проявив себе на більшості дослідів дещо краще ніж ітеративний та ієрархічний.

В якості результатів наведемо і карту поширеності онкозахворювань серед населення США (Рисунок 4), щоб була змога краще з'ясувати закономірності розподілу.

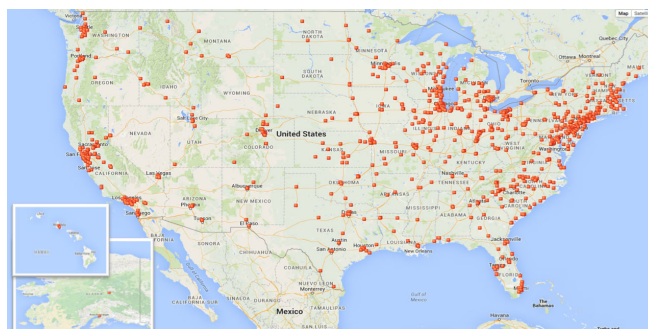


Рисунок 4. Розподіл онкохворих по території США

Висновок

Завдяки інтелектуальному аналізу просторових даних виникла можливість пошуку взаємозв'язків та особливостей у великих наборах просторових даних і моделювання поведінки відповідних об'єктів дослідження.

В даній статті було показано можливість застосування засобів кластерного аналізу для досліджень бази просторових медичних даних ідсумовуючи зроблену роботу, зазначимо, що засоби кластерного аналізу чудово проявили себе для аналізу просторової бази медичних даних. Результати роботи кожного з алгоритмів були наведені у вигляді діаграм. Також була отримана карта розподілу онкохворих людей по території США. Результат порівняння алгоритмів кластеризації засвідчив, що обрані алгоритми добре зарекомендували себе на вхідному наборі даних. Окремо варто виділити алгоритм DBSCAN, який проявив себе дещо краще у вирішенні поставленої задачі.

Застосування як алгоритмів, так і програмних засобів аналізу просторових даних у сфері медицини представляє широке поле для подальших досліджень та вдосконалень..

Література

1. Барсегян А. А. Методы и модели анализа данных: OLAP и Data Mining. / Барсегян А. А., Куприянов М. С., Степаненко В. В., Холод И. И. — СПб.: БХВ — Петербург, 2008. — 13–22 с.
2. Дубров А. М. Многомерные статистические методы: Учебник. / Дубров А. М., Мхитарян В. С., Трошин Л. И. — К.: Финансы и статистика, 2000. — 8–16 с.
3. Андрейчиков А. В. Интеллектуальные информационные системы. / Андрейчиков А. В., Андрейчикова О. Н. — К.: ФиС, 2004. — 22–45 с.
4. Мандель И. Д. Кластерный анализ. / Мандель И. Д. — К.: Финансы и статистика, 1988. — 10 с.
5. Жамбю М. Иерархический кластер-анализ и соответствия. / Жамбю М. — К.: Финансы и статистика, 1988. — 345 с.
6. Дюран Б. Кластерный анализ. / Дюран Б., Оделл П. — К.: Статистика, 1999. — 128–184 с.

Мельник Вікторія Миколаївна

*доктор технічних наук, професор, завідувач кафедри біотехніки та інженерії,
Національний технічний університет України «Київський політехнічний інститут»*

Карачун Володимир Володимирович

*доктор технічних наук, професор, професор кафедри біотехніки та інженерії,
Національний технічний університет України «Київський політехнічний інститут»*

Форостянко Віталій Сергійович

магістр

Національний технічний університет України «Київський політехнічний інститут»

Мельник Виктория Николаевна

*доктор технических наук, профессор, заведующий кафедрой биотехники и инженерии,
Национальный технический университет Украины «Киевский политехнический институт»*

Карачун Владимир Владимирович

*доктор технических наук, профессор, профессор кафедры биотехники и инженерии,
Национальный технический университет Украины «Киевский политехнический институт»*

Форостянко Виталий Сергеевич

магистр

Национальный технический университет Украины «Киевский политехнический институт»

Mel'nick V.

*doctor of technical sciences, professor,
National Technical University of Ukraine, Kiev Polytechnic Institute*

Karachun V.

*doctor of technical sciences, professor,
Head of the Department of bioengineering and biotechnics
National Technical University of Ukraine, Kiev Polytechnic Institute*

Forostyanko V.

master

National Technical University of Ukraine, Kiev Polytechnic Institute

ДИСТАНЦІЙНЕ УПРАВЛІННЯ ТЕПЛОМАСООБМІНОМ В БІОРЕАКТОРАХ НА РЕЗОНАНСНОМУ РІВНІ

ДИСТАНЦИОННОЕ УПРАВЛЕНИЕ ТЕПЛОМАСООБМЕНОМ В БИОРЕАКТОРАХ НА РЕЗОНАНСНОМ УРОВНЕ

REMOTE CONTROL TEPLOMASSOOBMEIN IN THE BIOREACTOR AT THE RESONANCE LEVEL

Анотація. Наводиться огляд з можливих технічних рішень дистанційного управління тепломасообміном в біореакторі на резонансному рівні за допомогою ультразвукового променя. Окреслюються умови виникнення хвильового співпадання, яке супроводжується різким зростом енергетичної активності поряд з мінімальною дисипацією променевих хвиль. Пояснюється природа додаткової енергетичної активності в зонах каустичних поверхонь двома типами хвиль.

Ключові слова: біореактор, енергетична активність, хвильове співпадання, кавітація, гранична частота.

Аннотация. Приводится одно из возможных технических решений дистанционного управления в биореакторе на резонансном уровне с помощью ультразвукового луча. Устанавливаются условия возникновения волнового совпадения, которое сопровождается резким ростом энергетической активности наряду с минимальной диссипацией лучевых волн. Раскрывается природа дополнительной энергетической активности в зонах каустических поверхностей двумя типами волн.

Ключевые слова: биореактор, энергетическая активность, волновое совпадение, кавитация, граничная частота.

Summary. We give one of the possible technical solutions for remote control of heat transfer in the bioreactor at resonant levels by ultrasound beam. Outlines the conditions of coincidence wave, which is accompanied by a sharp increase of energy activity, along with a minimum dissipation of radiation waves. Explained the nature of the additional energy activity in areas of caustic surfaces with two types of waves.

Keywords: bioreactor, energy activity, wave coincidence, cavitation, cutoff frequency

1. Вступ

Одним з перспективних фізичних методів впливу на речовини з метою інтенсифікації технологічних процесів є метод, заснований на використанні механічних коливань ультразвукового (УЗ) діапазону.

Ультразвукова інтенсифікація процесів заснована на введенні ультразвукових коливань високої інтенсивності (більше 3 ... 10 Вт/см²) безпосередньо в рідкі середовища. На практиці успішно реалізуються кілька варіантів контактного введення ультразвукових коливань в рідкі середовища. Один з найпоширеніших — це обробка шляхом занурення робочого інструмента ультразвукової коливальної системи в довільний технологічний об'єм або обробка в реакторах проточного типу, що містять технологічний об'єм у вигляді проточної камери і розташований в ній випромінювач ультразвукових коливань.

Традиційними середовищами для ефективної реалізації ультразвукових технологій (розчинення, екстрагування, очищення, диспергування, емульгування) є вода, органічні розчинники, рідкі середовища на водній основі з малим об'ємним вмістом твердих фракцій. Застосування водних середовищ обумовлено можливістю створення режиму розвиненої кавітації при мінімальних енергетичних витратах.

Найбільш успішно УЗ коливання використовуються в процесах, пов'язаних з рідкими станами реагентів, оскільки тільки в них виникає специфічний процес — УЗ кавітація, що забезпечує максимальні енергетичні впливи на різні речовини.

Кавітація — утворення в рідині пульсуючих бульбашок (каверн, порожнин), заповнених парою, газом або їх сумішшю. В ультразвуковій хвилі під час напівперіодів розрідження виникають кавітаційні бульбашки, які різко закриваються після переходу в область підвищеного тиску, породжуючи сильні гідродинамічні збурення в рідині, інтенсивне випромінювання акустичних хвиль. При цьому, в рідині відбувається руйнування поверхонь твердих тіл, що межують з кавітуючою рідиною.

Продуктивність і ефективність ультразвукової (УЗ) кавітаційної обробки рідких середовищ залежить головним чином від ступеня розвиненості кавітації в оброблюваній рідині, розміру кавітаційного хмари і рівномірності розподілу інтенсивності УЗ впливу по всьому технологічному об'єму. Ступінь

розвиненості кавітації визначається багатьма факторами, основними з яких є фізичні характеристики розглянутої рідини (щільність, в'язкість, початковий радіус кавітаційних зародків і т.д.) і інтенсивність УЗ коливань. А розмір кавітаційної хмари і рівномірність розподілу інтенсивності кавітації по всьому технологічному об'єму визначається як хвильовими властивостями кавітаційної зони, так і розмірами, формою технологічного об'єму.

Виникнення кавітаційних бульбашок в рідині і значна зміна їх розмірів в процесі розвитку кавітації тягне за собою зміну акустичних і фізичних властивостей оброблюваної рідини. При цьому розміри окремих бульбашок і їх скупчень, як правило, виявляються суттєво менше довжини хвилі первинного УЗ поля. Це дозволяє розглядати рідину разом з газовими краплями, як якесь нове середовище з еквівалентними акустичними характеристиками, відмінними від акустичних характеристик крапельної рідини.

При ультразвуковому кавітаційному впливі виділяють наступні режими в залежності від його інтенсивності: робота при малих інтенсивностях впливу (докавітаційний режим), при яких не порушується суцільність оброблюваного середовища; робота в режимі зародження кавітації, який характеризується виникненням в рідкому середовищі кавітаційних парогазових бульбашок, які насичують рідке середовище, змінюючи її хвильовий опір (імпеданс); робота в режимі розвиненої кавітації, який характеризується малим випромінюванням УЗ енергії, що пов'язано з наближенням імпедансу кавітуючого (водно-газового) середовища до імпедансу повітряного середовища; робота в режимі надрозвиненої кавітації, при якому бульбашки досягають критичних розмірів і відбувається їх виверження в резонансі. Теоретичний розгляд досліджуваного процесу здійснюється відповідно рис. 1.

Таким чином, вплив УЗ коливань на різні технологічні процеси дозволяє: не менш ніж у 10–1000 разів прискорити процеси, що протікають між двома або кількома неоднорідними середовищами (розчинення, очистку, знежирення, фарбування, подрібнення, просочення, емульгування, екстрагування, кристалізацію, полімеризацію, запобігання накипоутворенню, гомогенізацію, ерозію, хімічні та електрохімічні реакції та ін.); збільшити вихід корисних продуктів (наприклад, екстрактів) і додати їм додаткові властивості

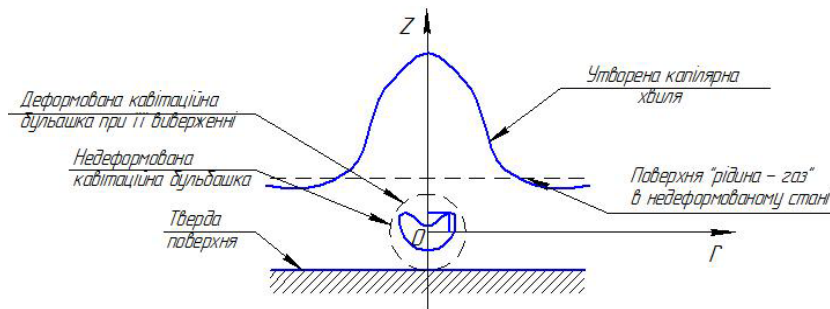


Рис. 1. Графічне представлення запропонованого способу збільшення поверхні розділу фаз

(наприклад, біологічну активність і стерильність); отримати нові речовини (наприклад, тонкодисперсні емульсії і суспензії) і реалізувати технологічні процеси, що не реалізовані традиційними методами.

Наводиться одне з можливих технічних рішень дистанційного управління тепломасообміном в біореакторі на резонансному рівні за допомогою ультразвукового променя. Окреслюються умови виникнення хвильового співпадання, яке супроводжується різким зростом енергетичної активності поряд з мінімальною дисипацією променевих хвиль. Пояснюється природа додаткової енергетичної активності в зонах каустичних поверхонь двома типами хвиль.

2. Постановка проблеми

З метою конкретизації поставлених задач технологічного процесу, обрано технічну реалізацію біореактора у вигляді газліфтного барботажного апарату та ультразвукові випромінювачі.

Відомо, що газліфтний барботажный апарат містить вертикально розташований циліндричний корпус з технологічними патрубками і розміщену в порожнині корпусу з радіальним зазором циркуляційну трубу, а також встановлений під циркуляційною трубою аератор [1].

За рахунок більш рівномірного розподілення газу та підвищення його диспергування підвищується продуктивність технологічного процесу.

Рух потоків газорідинної суміші забезпечується використанням потенціальної енергії стиснутого повітря.

Процес культивування організмів, як і більшість гетерогенних хімічних реакцій, безпосередньо залежить від кількості розчиненого в рідині газу і утворення та накопичення цільового продукту, зокрема, біомаси. Таким чином, підвищення ступеня розчинення газу в рідинній фазі реакцій призведе до скорочення його розходу та зменшення енерговитрат на аерацію.

Разом з тим недолік такого технічного рішення полягає у відносно низькій продуктивності технологічного процесу, збідненій кінематиці тепломасообміну

та постійно присутніх зонах пасивної енергетики робочої рідини в придонному прошарку та на периферії внутрішньої поверхні корпусу, невідповідності ступеня аерації і інтенсивності тепломасообміну культурального середовища вимогам технологічного процесу, а також прояв ефекту налипання маси до поверхні корпусу.

Пропонується підвищення продуктивності і якості технологічного процесу вирішити шляхом інтенсифікації тепломасообміну по всьому об'єму біореактора на резонансному рівні за допомогою штучного формування резонансного стану культурального середовища і породженої цим станом енергетичної активності робочої рідини.

Поставлена задача вирішується тим, що такий енергетичний стан усуває недоліки відомих конструкцій і пропонує нове ефективне технічне рішення з новим якісним результатом.

3. Аналіз літературних даних

Відома конструкція барботажного апарату для вирощування мікроорганізмів, яка містить ємкість з технологічними патрубками, розміщені в ємкості циркуляційний стакан і аератор, систему рециркуляції середовища, яка складається з відповідного трубопроводу, збудника розходу, підвідного трубопроводу і підключеного до нього пристрою для розбризкування культурального середовища, що розміщений у верхній частині ємкості [1].

Апарат має можливість працювати при підвищеному барботуванні культуральної рідини в ємкості, тобто за інтенсивного постачання мікроорганізмів киснем, отже і за енергійного постачання поживними речовинами. Це забезпечено усуненням стримуючої інтенсифікацію дріжжезростаючого процесу від'ємного зворотнього зв'язку між газовмістом культурального середовища в ємкості і інтенсивністю тепломасообмінних процесів в апараті загалом.

Зазначений від'ємний зв'язок ліквідується за рахунок введення в конструкцію апарату відвідного патрубка піни в складі гідроциклона та ежектора.

Недоліком цього технічного рішення постає досить висока матеріалоемність апарату, підвищене піноутворення, відносно невисока продуктивність технологічного процесу внаслідок пасивного перемішування і, практично, ламінарного поступального руху робочої рідини, і тільки паралельно осі апарату а також наявність неминучих при цьому застійних зон в придонному прошарку та на периферії апарату (на внутрішніх бічних стінках), що зумовлене обмеженими можливостями обраного технічного рішення [2].

Відомі також інші конструкції барботажних апаратів, які в тій, чи іншій формі вирішують питання технологічного процесу. Наприклад, забезпечують рух культуральної рідини із змінними швидкостями по всім рівням біореактора, завдяки опуклій формі циркуляційної труби [3]. Формують рух периферійної частини робочої рідини, що знаходиться поза циркуляційної труби, по гвинтовій траєкторії [4]. Активізують рух робочої рідини по всій висоті корпусу, а також в коловому і радіальному напрямку [5]. Кожна з них має свої переваги, але і певні вади згідно вищесказаного.

4. Фокусування енергії ультразвукового променя. Поверхні каустики

Найбільш суттєві аспекти концентрації проникної звукової енергії розглянемо на прикладі газліфтного барботажного апарату (рис. 2) та ультразвукові випромінювачі (рис. 3).

Газліфтний барботажний апарат використовується для культивування мікроорганізмів в рідинних середовищах при виготовленні вакцин та біологічно-активних речовин і містить вертикально розташований циліндричний корпус 1 з патрубком 2 для введення живильної рідини і посівного матеріалу, патрубком 3 для видалення культуральної рідини та патрубком 4 для відведення відпрацьованого газу (рис. 2, а). В порожнині корпусу 1, співвісно з ним, на днищі, розміщений аератор 5 (рис. 2, б).

У попередньо простерилізований корпус 1 крізь патрубок 2 вводять робочу рідину, після чого, через аератор 5 пускають стиснений газ (повітря) і послі-

довно включають, попередньо налаштовані на хвильовий резонанс, ультразвукові випромінювачі 6 та 7. Стиснений газ у вигляді бульбашок 8 підіймається вгору і утворює рідинно-повітряну суміш, яка набагато легша від робочої рідини на периферії внутрішньої порожнини корпусу 1. Технічний результат забезпечується обладнанням ззовні корпусу, протилежно, двома ультразвуковими випромінювачами з регулюючим напрямом променів, один з яких формує резонансні коливання з частотою вищою за граничну і функціонально замінює циркуляційну трубу циліндричною каустикою, а другий випромінювач формує резонансні коливання з частотою нижчою за граничну і будує більшого радіуса другу циліндричну каустику поблизу внутрішньої поверхні корпусу апарату, активізуючи периферійні шари.

Прийнявши сталений корпус газліфтного барботажного апарату радіусом $R = 1$ м і товщиною $2\delta = 2$ мм, окреслюємо його граничну частоту у $f_{gp} = 14,457$ кГц, якщо частота ультразвукового променя дорівнює, наприклад, $f = 42$ кГц.

На частоті ультразвукового випромінювання вищій за граничну, згинні коливання, швидкість яких залежить від частоти ω випромінювання, за певного кута θ_c променя породжують резонанс хвильового співпадання сліду швидкості згинних коливань $c_{sz}(\omega)$ і падаючої хвилі випромінювання c_0 [6]

$$c_{sz}(\omega) = \frac{c_0}{\sin \theta_c},$$

(причому, кожній частоті ω відповідає свій кут співпадання θ_c), що створює ефект акустичної прозорості, коли звукове випромінювання без втрати енергії надходить усередину корпусу апарату. Елементи внутрішньої поверхні корпусу, за умови великого його

хвильового розміру, тобто коли $kR = \frac{2\pi fR}{c_0} = 176 \gg 1$,

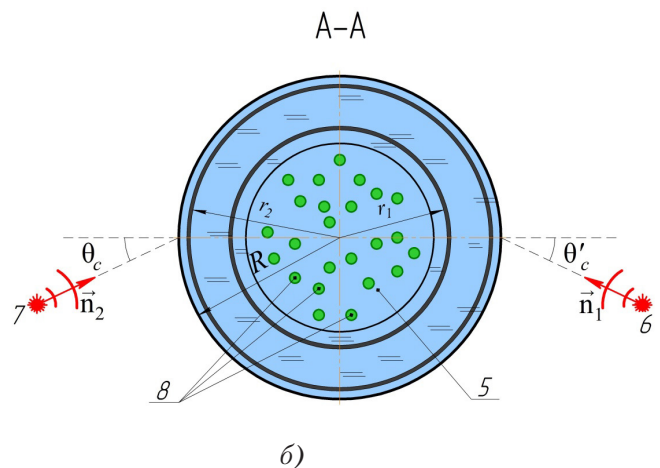
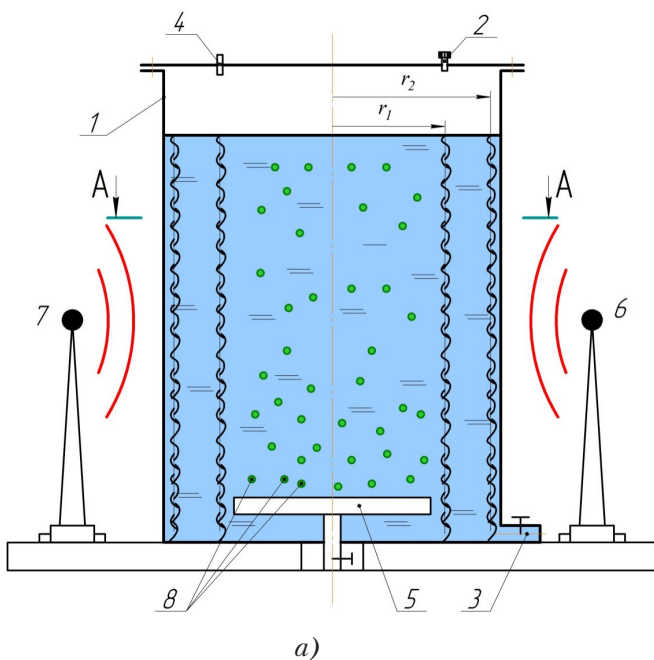


Рис. 2. Газліфтний барботажний апарат в поздовжньому (а) та поперечному (б) перерізі



а)



б)

Рис. 3. Ультразвукові випромінювачі: а) Mimi ultrasonic Cleaner Model 3560; б) блок ультразвукових випромінювачів УЗП-6-1

випромінюють в культуральну рідину звукові хвилі, які внаслідок аберації на кут β формують співвісно осі апарату циліндричну каустику, тобто зону підвищеної енергії, радіуса [7]

$$r = R \sin \beta = R \frac{c_0}{c_{sz}} = 0,30 \text{ (м)} \quad (1)$$

де $c_{sz} = \left[2E\delta^2 4\pi^2 f^2 (3m_{II})^{-1} (1 - \sigma^2)^{-1} \right]^{\frac{1}{4}}$; $c_0 = 1500 \text{ мс}^{-1}$ — швидкість звука в рідині; E — модуль Юнга матеріалу корпусу; σ — коефіцієнт Пуассона; $m_{II} = 2\rho\delta$ — питома маса; ρ — щільність.

Відповідно, ультразвуковий випромінювач УЗП-6-1, який генерує промінь набагато нижчої за граничну частоти, формує в корпусі апарату поздовжні хвилі із швидкістю c_{II} , які не залежать від частоти випромінювання і, в свою чергу, також породжують ефект хвильового співпадання, який, на відміну від згинних коливань, залежить тільки від кута θ_c співпадання і не залежить від частоти ω випромінювання

$$c_{II} = \frac{c_0}{\sin \theta_c} = 6100 \text{ (мс}^{-1}\text{)}.$$

Ці коливання, за умови, що $c_{II} > c_0$, будуть випромінювати в рідину звукові хвилі під кутом α , які утворюють також циліндричну каустику, але радіуса [8, 9]

$$r_2 = R \cos \alpha = R \sqrt{1 - \frac{c_0^2}{c_{II}^2}} = 0,97 \text{ (м)}, \quad (2)$$

яка буде знаходитися біля поверхні внутрішньої сторони корпусу і створить енергетичну активність робочої рідини на периферії.

Таким чином, формування хвильового співпадання штучно генеруючими згинними, а також поздовжніми хвилями корпусу апарату, приведе в резонансний стан культуральну рідину, збурить її по всьому об'єму, ліквідує малоактивні і застійні зони. Циліндричні каустики вздовж осі апарату і біля його сті-

нок додатково збурять робочу рідину і сприятимуть подальшій інтенсифікації процесів тепломасообміну і аерації, що, в свою чергу, підвищить продуктивність і якість технологічного процесу [10].

Якщо $c_{II} = c_0$, поверхня більшої каустики обернеться на геометричне місце точок, які знаходяться на осі корпусу.

Створення енергетично активного, турбулентного стану в робочій рідині без додаткових механічних засобів, просторовий рух повітряно-рідинної суміші, штучне збільшення часу взаємодії бульбашок кисню з робочою рідиною та відсутність в порожнині корпусу додаткових елементів конструкції для регулювання технологічного процесу, забезпечують однорідність суміші по всьому об'єму, ліквідують зони пасивної енергетики в апараті, мінімізують ефект налипання до поверхні корпусу.

Таким чином, використання ультразвукового опромінювання корпусу газліфтного барботажного апарату, завдяки новим властивостям, забезпечить енергетичну активність робочої рідини по всьому об'єму апарату, що суттєво підвищить продуктивність технологічного процесу і його якість.

5. Апробація результатів досліджень

Побудовані розрахункові схеми і отримані результати дозволяють оцінити адекватність теоретичних рішень вихідним припущенням і, таким чином, підтвердити їх доцільність для подальшої практичної реалізації. Просторовий резонанс, який створюють усередині корпусу ультразвукові випромінювачі, турбулізує всю рідиннофазову складову вмісту корпусу біореактора і примушує газові бульбашки рухатись без перешкод не тільки спрямовано вгору, але і в інших різноманітних напрямках, утримуючи їх довше в робочій рідині, збагачуючи тим самим суміш киснем і створюючи умови для інтенсивного росту клітин. Завдяки новим

властивостям, забезпечується енергетична активність робочої рідини по всьому об'єму. Що суттєво підвищує продуктивність технологічного процесу і його якість.

6. Висновки

Штучне формування енергетичної активності робочої рідини ультразвуковим променем на резонансному рівні дозволяє дистанційно керувати тепло-

масообміном, підвищувати продуктивність і якість вихідного продукту.

Технічна реалізація запропонованого рішення не обтяжена визначаючим ростом матеріалоемності, натомість, надає можливість для автоматизації всього технологічного процесу, забезпечення стерильності продукту, здійснення поточного керування енергетичним станом робочої рідини.

Література

1. А.с. 1708829 АІ СССР, С12М1/04. Газлифтний барботажний апарат [Текст]/ Ю. Г. Куляшов, В. И. Горячкин. С. П. Уткин, Ю. Н. Талызин, А. С. Васильев, В. М. Крац (СССР). — № 4612860/13; заявл. 01.12.88; опубл. 30.01.92, Бюл. № 4. — 1 с.
2. А.с. 147208 АІ СССР, С12М1/04. Апарат для вирощивання мікроорганізмів [Текст]/ Ю. Ф. Давыдов, В. М. Геллис, В. К. Погудкин, В. М. Карц, В. Н. Соловьев, С. П. Уткин (СССР). — № 4109725/28-13; заявл. 21.08.86; опубл. 30.07.89, Бюл. № 28. — 1 с.
3. Пат. № 43558 на корисну модель, Україна, МПК (2009), С 12 М 1/04. Газліфтний барботажний апарат [Електронний ресурс]/ В. В. Карачун, М. С. Тривайло, В. М. Мельник, М. А. Руденко, Д. В. Литвиненко. u 200901835; заявл. 02.03.2009; опубл. 25.08.2009, Бюл. № 16. — Режим доступу: \www/URL: <http://uapatents.com/2-43558-gazliftnijj-barbotazhnijj-aparat.html>
4. Пат. № 40230 на корисну модель, Україна, МПК (2009), С 12 М 1/04. Газліфтний барботажний апарат [Електронний ресурс]/ М. С. Тривайло, В. В. Карачун, В. М. Мельник, О. Є. Резенчук, А. О. Заброда. u 200813427; заявл. 20.11.2008; опубл. 25.03.2009, Бюл. № 6. — Режим доступу: \www/URL: <http://uapatents.com/2-40230-gazliftnijj-barbotazhnijj-aparat.html>
5. Пат. № 78382 на корисну модель, Україна, С 12 М 1/04 (2006.01) Газліфтний барботажний апарат [Електронний ресурс]/ В. М. Мельник. u 201213327; заявл. 22.11.2012; опубл. 11.03.2013, Бюл. № 5. — Режим доступу: \www/URL: <http://uapatents.com/5-78382-gazliftnijj-barbotazhnijj-aparat.html>
6. Gremer, L. Theorie der Schalldämmung dünner Wände bei schrägem Einfall [Текст] / L. Gremer // Akust. Zeitschrift, 7, 1942. Str 3–7.
7. Шендеров, Е. Л. Волновые задачи гидроакустики [Текст]: моногр. / Е. Л. Шендеров. Л.: Судостроение, 1972. — 352 с.
8. Gösele, K. Zur Körperschallausbreitung in Wohubauten [Текст]/ K. Gösele // Körperschall in Gebäuden. Berlin. 1960. Str 24–24.
9. Heckl, M. Die Schalldämmung von homogenen Einfachwänden endlicher Fläche [Текст] / M. Heckl// Acustica, 10, 1960. Str 17–21.
10. Junger, M. C. Letter to the editor [Текст] / M. C. Junger, P. W. Smith // Acustica, 5,1, 1955. Str 43–46.

Мельник Александр Петрович*ассистент кафедры машин и аппаратов химических и нефтеперерабатывающих производств
Национальный технический университет Украины «Киевский политехнический институт»***Aleksandr Melnik***Lecturer of department of machines and devices of chemical and oil refining manufacturers
National Technical University of Ukraine «Kyiv Polytechnic Institute»***МОДЕЛИРОВАНИЕ ПРОЦЕССА ГОРЯЧЕГО
ПРЕССОВАНИЯ БУМАЖНОГО ПОЛОТНА****THE MODELLING OF THE PROCESS
OF HOT PRESSING OF PAPER WEB****Аннотация:** разработана математическая модель процесса горячего прессования бумажного полотна.**Ключевые слова:** горячее прессование, бумажное полотно, сухость.**Summary:** mathematical model of process of hot pressing of paper web was developed.**Key words:** hot pressing, paper web, dryness.

В процессе производства бумаги основные объемы тепловой энергии расходуются на сушку бумажного полотна, сухость которого на украинских бумагоделательных машинах, после удаления влаги на прессовой части, составляет 35–50% [1]. При этом 60–45% воды высушивается, в результате чего производство бумаги занимает второе место по удельным энергетическим затратам после металлургической промышленности. На высушивание воды расходуются значительные объемы тепловой энергии. Существенно снизить расход тепловой энергии, на сушильной части бумагоделательной машины можно увеличивая сухость бумажного полотна перед сушильной частью машины применяя на прессовой ее части метод горячего прессования. Известно, что в процессе прессования бумаги нагретыми валами можно получить сухость бумажного полотна до 70% [2], в зависимости от массы квадратного метра бумажного полотна и параметров процесса, однако в литературных источниках практически отсутствуют данные для создания методов расчета промышленных горячих прессов бумагоделательных машин. Поэтому работа по исследованию кинетики процесса горячего прессования является актуальной.

Целью данной работы является определение кинетических закономерностей и основных параметров процесса горячего прессования.

Процесс прессования бумажного полотна происходит в захвате прессы на мягкой подложке из прессового сукна (рис. 1). Поры, образованные волокна-

ми бумажного полотна заполнены водой. В процессе прессования верхний слой 1 деформируется в результате чего вода из деформированных пор фильтруется через недеформированное бумажное полотно 2 в сукно 3, которое содержит поры большего размера и соответственно имеет больший коэффициент фильтрации, кроме того под действием градиента давления происходит фильтрация воды из бумаги в прессовое сукно.

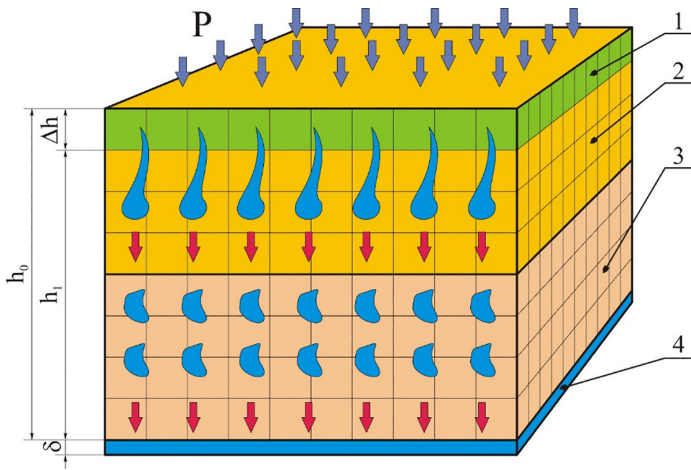
Теоретически толщина слоя отфильтрованной воды 4 соответствует величине деформации бумажного полотна.

Таким образом мы видим, что в результате прессования происходят два основных процесса. Процесс фильтрации воды, который описывается законом Дарси и процесс деформации бумажного полотна в результате которого изменяется его пористость. Скорость этих процессов описывает уравнение Дарси-Герсеванова [3]:

$$u - \varepsilon u_c = -k \frac{\partial H}{\partial z}. \quad (1)$$

Движущей силой этих процессов является градиент давления, которое создается прижимающим валом в захвате прессы. Для определения скорости изменения давления в прессовом захвате с учетом деформации бумажного полотна Новиковым Н. Е. предложено уравнение [3]:

$$\frac{\partial H}{\partial \tau} = a^2 \frac{\partial^2 H}{\partial z^2} \quad (2)$$



1 – слой бумаги, который деформируется; 2 – недеформированное бумажное полотно; 3 – прессовое сукно; 4 – слой отфильтрованной воды

Рис. 1. Физическая модель процесса горячего прессования бумажного полотна

где

$$a^2 = \frac{(1 + \varepsilon)k}{a_{сж} \gamma_v} \quad (3)$$

где ε – коэффициент пористости; k – коэффициент фильтрации, м/с; $a_{сж}$ – коэффициент сжатия, м²/Н; γ_v – удельный вес воды, Н/м³.

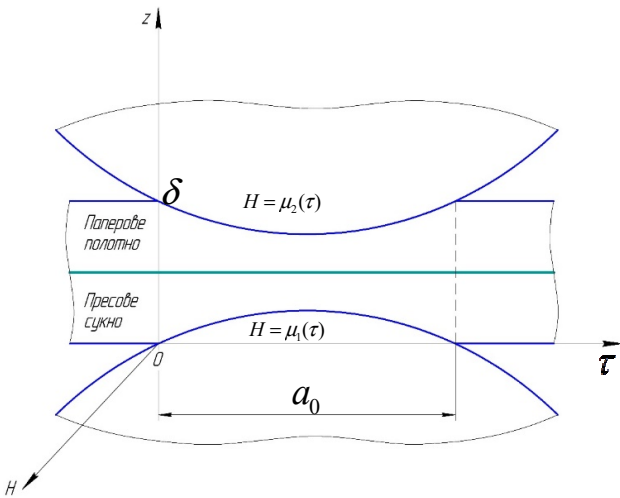


Рис. 2. Зона контакта прессовых валов

Учитывая, что давление в захвате прессы изменяется во времени по параболическому закону [3] (рис. 2)

решение уравнения (2) можно получить численным методом при следующих начальных:

$$H(z,0) = \varphi(z) \quad (4)$$

и граничных условиях:

$$\left. \begin{aligned} H &= \mu_1(\tau) + \mu_1(t) \text{ при } z=0 \\ H &= \mu_2(\tau) + \mu_2(t) \text{ при } z=\delta \end{aligned} \right\} \quad (5)$$

где $\mu(\tau)$ – зависимость напора от времени прессования; $\mu(t)$ – зависимость напора от температуры в межвалковом зазоре.

Температура испарения связана с температурой поверхности горячего вала зависимостью [4]:

$$t_s = t_b - \frac{q}{\alpha} \quad (6)$$

Температура вала может быть найдена путем интегрирования уравнения нестационарной теплопроводности горячего вала с источником тепла расположенным внутри него.

Интегрируя уравнение (2) при указанных граничных и начальных условиях получим зависимость напора от времени прессования. Дифференцируя полученную зависимость по координате z получим зависимость градиента давления от времени прессования и толщины бумаги и сукна. Найденный градиент давления используем для интегрирования уравнения Дарси с целью определения количества воды, удаленной из бумажного полотна:

$$G = -k\rho \int_0^\tau \frac{\partial H}{\partial z} d\tau \quad (7)$$

Что позволит определить конечную сухость бумажного полотна при горячем прессовании:

$$S_k = \frac{g_\delta - S_n}{g_\delta - G} \quad (8)$$

где g_δ – масса одного метра квадратного бумаги, кг/м²; S_n – начальная сухость бумажного, %.

Выводы

Представленная математическая модель позволяет определить влияние основных параметров на процесс горячего прессования бумажного полотна, определить его конечную сухость, что позволит определить пути направления снижения энергоемкости оборудования.

Литература

1. Примаков С.П., Барбаш В.А. Технология паперу і картону: Навч. посіб. / Друге видання, переробл. – Київ: ЕКМО, 2008. – 425 с.
2. Результати досліджень процесу гарячого пресування паперового полотна. В.М. Марчевський, О.П. Мельник. Вісник НТУУ «КПІ» Хімічна інженерія, екологія та ресурсозбереження. № 1 – 2008. стаття.
3. Прессование бумажного полотна. Новиков Н. Е. «Лесная промышленность», 1972 г. 240 с.
4. А. Г. Касаткин Основные процессы и аппараты химической технологии. 9-е изд. – М.: Химия, 1973. – 750 с.

Метелиця Марія Андріївна

студентка

Національний технічний університет України «Київський політехнічний інститут»

Метелиця Мария Андреевна

студентка

Национальный технический университет Украины «Киевский политехнический институт»

Metelytsia M.

student

National Technical University of Ukraine «Kyiv Polytechnic Institute»

МЕТОДИ ФОРМУВАННЯ ТА ОЦІНЮВАННЯ ГРУПОВИХ РЕКОМЕНДАЦІЙ МЕТОДЫ ФОРМИРОВАНИЯ И ОЦЕНИВАНИЯ ГРУППОВЫХ РЕКОМЕНДАЦИЙ DESIGN AND EVALUATION OF GROUP RECOMMENDATIONS

Анотація. Проаналізовано методи формування групових рекомендацій. Запропоновано метрику оцінювання якості представлених моделей. Проведено дослідження якості моделей за допомогою запропонованої метрики.

Ключові слова: групові рекомендаційні системи, теорія прийняття рішень, методи голосування, профіль користувача.

Аннотация. Проанализированы методы формирования групповых рекомендаций. Предложена метрика оценивания качества представленных моделей. Проведено исследование качества моделей с помощью предложенной метрики.

Ключевые слова: групповые рекомендательные системы, теория принятия решений, методы голосования, профиль пользователя.

Summary. The methods for group recommendations formation were investigated. A quality evaluation metric was suggested. A study of the models quality using the proposed metric was conducted.

Key words: group recommender systems, decision theory, voting methods, user profile.

Вступ. Групові рекомендаційні системи мають на меті виявлення вподобань визначеної групи користувачів, на основі їх індивідуальних вподобань та інших факторів. Широкомасштабні дослідження у цій галузі проводились фахівцями з різних країн світу. Професор Джудіт Мастгофф (Judith Masthoff), у своїх роботах [1] та [4] представляє і емпірично оцінює ряд стратегій соціального вибору для рекомендації телевізійних програм групам з невеликою кількістю користувачів. В даних роботах було проаналізовано ці стратегії з метою використання деяких з них у процесі агрегування індивідуальних вподобань користувачів. Італійський вчений Франческо Річі (Francesco Ricci) аналізує як індивідуальні рекомендаційні системи так і загальні підходи до створення групових рекомендацій [2].

Існує велика кількість галузей застосування групових рекомендаційних систем, наприклад: групи в соціальних мережах онлайн-сервіси для вибору фільмів або музики; громадські заклади як спортивні зали або ресторани та багато інших.

Мета роботи. Проаналізувати та порівняти методи формування групових рекомендацій, засновані на теорії групового вибору та запропонувати метрику оцінювання таких методів.

Основна частина. У даній роботі задача надання групових рекомендацій розглядається у контексті об'єднання (агрегування) індивідуальних вподобань користувачів групи [3]. У практичному дослідженні індивідуальні вподобання визначалися за допомогою алгоритму колаборативної фільтрації за сусідством користувачів [5].

Постановка задачі

Вихідними даними в задачі є N векторів розмірності M з передбаченими за алгоритмом колаборативної фільтрації [5] рейтингами для кожного користувача групи за всіма фільмами за бази даних:

$$\{R_j^{(M)}\}_{j=1}^N,$$

де N — кількість користувачів в групі;

M — загальна кількість фільмів;

$R_j^{(M)}$ — M -розмірний вектор, елементами якого є передбачені рейтинги для j -го користувача за кожним фільмом. Рейтинги лежать в інтервалі від 1 до 10.

В задачі необхідно отримати вектор групових вподобань — групових рейтингів для кожного фільма:

$$GR^{(M)},$$

де M — загальна кількість фільмів.

Методи формування групових рекомендацій

Для агрегування індивідуальних вподобань користувачів використовують різні методи, зокрема — методи групового вибору (стратегії соціального вибору, методи голосування) [3]. Далі приведено формалізацію методів, які було розглянуто в даному дослідженні.

Аддитивна утилітарна стратегія
(Additive utilitarian strategy)

$$gr_m = \sum_{n=1}^N r_{n,m} .$$

Мультиплікативна утилітарна стратегія
(Multiplicative utilitarian strategy)

$$gr_m = \prod_{n=1}^N r_{n,m} .$$

Стратегія середнього (Average strategy)

$$gr_m = \frac{1}{N} \sum_{n=1}^N r_{n,m} .$$

Стратегія середнього без страждань
(Average without misery strategy)

$$gr_m = \begin{cases} \frac{1}{N} \sum_{n=1}^N r_{n,m}, r_{n,m} \geq T \\ 0, r_{n,m} < T \end{cases} .$$

Стратегія мінімальних страждань
(Least misery strategy)

$$gr_m = \min_{n=1, N} r_{n,m} .$$

Стратегія максимального задоволення
(most pleasure strategy)

$$gr_m = \max_{n=1, N} r_{n,m} .$$

Стратегія справедливості (Fairness strategy)

Алгоритм визначення групової оцінки наступний.

1. Перший користувач вибирається випадковим чином. Розглядаються його перші L вподобань.

2. Для кожного з L об'єктів визначається мінімальна оцінка за всіма користувачами.

3. З усіх об'єктів, які отримали мінімальні оцінки вибирається той, мінімальна оцінка якого найбільша і він отримує групову оцінку рівну M — кількості об'єктів. На K -й ітерації оцінка буде дорівнювати $M-K$

(якщо першу ітерацію назвати нульовою). Перехід на наступну ітерацію (п. 1)

Нижче наведено формалізацію.

k-та ітерація:

$$gr_m = M - k ,$$

$$m: \begin{cases} m \in \{m_1, \dots, m_L \mid r_{nm_j} - \text{один з } L \text{ найвищих рейтингів}\} = I_n \\ r_{nm} = \max_{j \in I_n} \min_{i \in U} r_{ij} \end{cases}$$

Мажоритарна стратегія
(Plurality voting strategy)

Алгоритм наступний:

1. Перший користувач вибирається випадковим чином (тут в якості модифікації можна ввести додатковий параметр — порядок впливовості користувачів у групі). Розглядаються його перші L вподобань.

2. Для кожного з L об'єктів визначається сума оцінок за всіма користувачами, крім того, що зараз розглядається.

3. Об'єкт, сумарна оцінка якого виявилась найбільшою, отримує групову оцінку рівну M — кількості об'єктів. На K -й ітерації оцінка буде дорівнювати $M-K$ (якщо першу ітерацію назвати нульовою). Перехід на наступну ітерацію (п. 1).

Формалізацію наведено нижче.

k-та ітерація:

$$gr_m = M - k ,$$

$$m: \begin{cases} m \in \{m_1, \dots, m_L \mid r_{nm_j} - \text{один з } L \text{ найвищих рейтингів}\} = I_n \\ r_{nm} = \max_{j \in I_n} \sum_{i \in U} r_{ij} \end{cases}$$

Стратегія голосування за згодою
(Approval voting strategy)

У даному підході об'єктам нараховується один бал за кожну оцінку від користувача, яка перевищує визначений поріг. Потім ці бали сумуються та ранжуються.

$$gr_m = \sum_{n=1}^N AS_n ,$$

$$AS_n = \begin{cases} 1, r_{nm} \geq T \\ 0, r_{nm} < T \end{cases}$$

Стратегія Борда (Borda count strategy)

Алгоритм проходить в два етапи:

1. Для всіх користувачів оцінки об'єктів замінюється на нові бали за наступним принципом: об'єкт з мінімальною оцінкою отримує нуль, наступний — 1 і так далі. У тому випадку, коли користувач однаково оцінив декілька об'єктів, ставиться середня з потенційно можливих оцінок.

2. Далі отримані бали сумуються, та отримані значення і є групові оцінки.

$$gr_m = \sum_{n=1}^N BS_n,$$

$$BS_n = pos(r_{nm}, OL),$$

$$OL = \{r_{nm_1}, r_{nm_M}\}, \text{де } r_{nm_k} \leq r_{nm_{k+1}}$$

Позначення для вищенаведених формул:

- gr_m – груповий рейтинг для m -го фільму;
- N – кількість користувачів в групі;
- M – загальна кількість фільмів;
- $r_{n,m}$ – рейтинг m -го фільму для n -го користувача;
- T – порогове значення.

Метрика оцінювання якості

Представлені методи в кінцевому підсумку дають ранжований список рекомендацій. Це означає, що величини на вході і результуючі величини несумірні між собою. В такому випадку, метрики, які застосовуються для оцінки точності не підходять. Для оцінки якості методів агрегування запропоновано використувати наступну метрику ранжування. Формула для визначення значення оцінки ранжування для одного користувача має вигляд:

$$RankScore_n = \sum_{m=1}^{M_n} \frac{r_{nm} - \bar{r}_n}{2^{rank_m/M}}$$

де $RankScore_n$ – оцінка ранжування для n -го користувача;

$rank_m$ – місце в загальному рейтингу фільмів для групи;

\bar{r}_n – середній рейтинг для n -го користувача

$$\bar{r}_n = \frac{1}{M_n} \sum_{m=1}^{M_n} r_{nm};$$

M – загальна кількість фільмів у матриці вихідних даних.

$RankScore_n$ – оцінка, яка показує ступінь задоволеності ранжуванням з боку користувача n . Вираз в чи-

сельнику дозволяє нормувати значення оцінки, яку було поставлено певним користувачем. Різниця між рейтингом та середнім рейтингом для користувача визначає ступінь підвищення ступеню задоволеності користувача стосовно перегляду цього фільму. Якщо фільму було поставлено оцінку нижче за середню – скоріш за все, цей фільм понижує ступінь задоволеності користувача. Вираз в знаменнику показує те, що чим нижчим є рейтинг фільму для групи, тим нижчим є значення функції задоволеності для випадку якщо чисельник має значення вище нуля, та вищим для випадку, якщо значення чисельника нижче за нуль.

Для визначення групової оцінки ранжування застосовувались ті ж самі підходи як і для оцінок точності. Формула для групової оцінки ранжування за принципом середнього арифметичного:

$$RankScore_{gr}^{av} = \frac{1}{N} \sum_{n=1}^N \sum_{m=1}^{M_n} \frac{r_{nm} - \bar{r}_n}{2^{rank_m/M}}$$

Формула для групової оцінки ранжування за принципом мінімального задоволення:

$$RankScore_{gr}^{min} = \min_{n=1, N} \left\{ \sum_{m=1}^{M_n} \frac{r_{nm} - \bar{r}_n}{2^{rank_m/M}} \right\}$$

Це означає, що за групову задоволеність приймається ступінь задоволеності найменш щасливого її учасника.

Результати дослідження. На основі запропонованої метрики ранжування було досліджено методи агрегування індивідуальних вподобань користувачів групи. На стовчастій діаграмі на рис. 1 зображено групові оцінки ранжування при кожному з методів для груп різного розміру (3, 6 та 9 користувачів). На діаграмі на рис. 2 зображено групові оцінки ранжування при кожному з методів для вибірок різного об'єму.

З графіків видно, що при інших рівних умовах (фіксована вибірка і фіксований розмір групи) найбільш ефективними в сенсі оцінки ранжування ви-

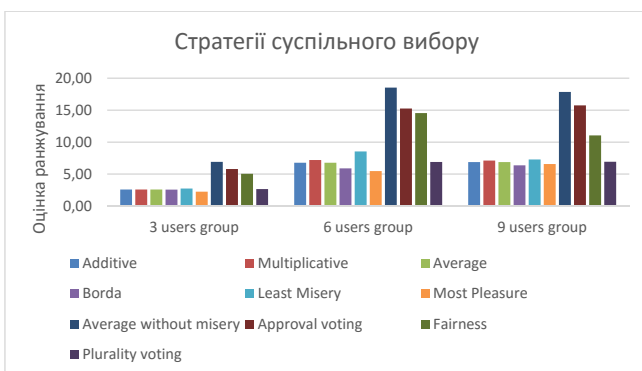


Рисунок 1 – Порівняння стратегій суспільного вибору за оцінкою ранжування для груп різного розміру

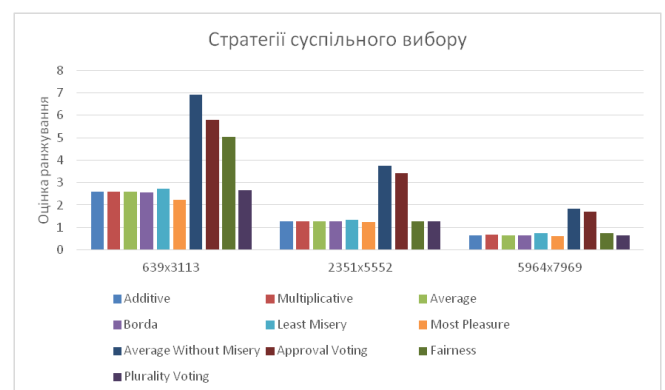


Рисунок 2 – Порівняння стратегій суспільного вибору за оцінкою ранжування для різних вибірок

явилися методи Середнього без страждань (Average without misery), Голосування за згодою (Approval Voting) і стратегія Справедливості (Fairness strategy).

Висновки. В роботі було отримано наступні результати: проаналізовано методи формування групових рекомендацій в контексті агрегування індиві-

дуальних вподобань, запропоновано метрику оцінки якості ранжування, проведено аналіз ефективності методів групового прийняття рішень в задачі формування групових рекомендацій на основі запропованої метрики.

Література

1. Masthoff, J. Group Modeling: Selecting a Sequence of Television Items to Suit a Group of Viewers / J. Masthoff // User Modeling and User-Adapted Interaction. — 2004. — № 14. — P. 37–85.
2. Recommender Systems Handbook / [Ricci, F., Rokach, L., Shapira, B., Kantor, P.B.]. — New York City: Springer, 2011. — 842 p.
3. Masthoff, J. Group Recommender Systems: Combining Individual Models / J Masthoff. // Recommender Systems Handbook. — New York: Springer, 2011. — P. 677–702.
4. Masthoff, J. The Pursuit of Satisfaction: Affective State in Group Recommender Systems / J Masthoff. — Aberdeen: University of Aberdeen, 2005. — 306 p.
5. Segaran, T. Programming Collective Intelligence / T. Segaran. — Sebastopol: O'Reilly Media, 2007. — 362 p.

Миселюк Артур Юрійович

студент

Національний технічний університет України «Київський політехнічний інститут»

Миселюк Артур Юрьевич

студент

Национальный технический университет Украины «Киевский политехнический институт»

Myseliuk A.

student

National Technical University of Ukraine «Kyiv Polytechnic Institute»

СИСТЕМИ ДІАГНОСТИКИ ЕЛЕКТРОКАРДІОГРАМ ЗА ДОПОГОЮ ГЛИБИННИХ НЕЙРОННИХ МЕРЕЖ СИСТЕМЫ ДИАГНОСТИКИ ЕЛЕКТРОКАРДИОГРАММ С ПОМОЩЬЮ ГЛУБИННЫХ НЕЙРОННЫХ СЕТЕЙ

DEEP NEURAL NETWORKS IN ELECTROCARDIOGRAM DIAGNOSING

Анотація. В роботі досліджується можливість використання глибоких нейронних мереж для діагностики електрокардіограм.

Ключові слова: глибокі нейронні мережі, машинне навчання, згорткові нейронні мережі, багатосаровий перцептрон, автокодувальник.

Аннотация. В работе исследуется возможность использования глубоких нейронных сетей для диагностики электрокардиограмм.

Ключевые слова: глубокие нейронные сети, машинное обучение, сверточные нейронные сети, многослойный перцептрон, автокодировщик.

Summary. This paper explores the possibility of using deep neural networks to diagnose electrocardiograms.

Keywords: deep neural networks, machine learning, convolution neural network, multilayer perceptron, autoencoder.

Вступ

З швидким розвитком технології стають доступними нові передові методи виявлення захворювань. Таким чином, одним з можливих способів діагностики захворювань внутрішніх органів засновано на аналізі інформації електрокардіограми сигналів. Так як ці сигнали містять інформацію про різних фізіологічних процесах, що відбуваються в організмі людини, то їх можна використовувати для оцінки стану здоров'я [1].

Таким чином, ціллю роботи є розробка системи аналізу електрокардіограм для автоматичної діагностики аномальних QRS комплексів.

Для її досягнення у роботі поставлено наступні задачі:

- Проаналізувати існуючі алгоритми, підходи та методи для тестування баз даних. Провести їх порівняльний аналіз та виявити недоліки.
- Створити алгоритм для автоматичної генерації навантажувального тесту.

- Розробити програмний продукт, в якому втілено запропонований алгоритм.

Об'єктом дослідження являються алгоритми машинного навчання на основі глибоких нейронних мереж.

Предметом дослідження є алгоритми аналізу електрокардіограм за допомогою глибоких нейронних мереж.

Мета роботи — розробити алгоритм на основі нейронних мереж для аналізу електрокардіограм.

Методом дослідження є алгоритми машинного навчання.

Науковою новизною отриманих результатів є запропонований алгоритм діагностики електрокардіограм на основі використання вейвлет перетворення та глибоких нейронних мереж.

Практична цінність роботи полягає у розробленому алгоритмі та програмному продукті, які дозволяють провести діагностику електрокардіограм.

Алгоритми глибоких нейронних мереж для аналізу електрокардіограм

Для побудови моделі на основі глибоких нейронних мереж, здатних виявляти захворювань на електрокардіограмах, використовуються згорткова нейронна мережа (англ.: Convolutional Neural Network, — CNN), багаторушні автокодувальники (англ.: Stacked Autoencoders) і глибокі машини Больцмана (англ.: Deep Boltzmann Machine — DBM) [2, 3]. Навчання таких моделей, як правило, здійснюється за допомогою двох підходів. У першому використовується навчання із вчителем. Серед різних архітектур, які використовують такий підхід, високу продуктивність мають CNN. У порівнянні зі стандартними нейронними мережами прямого поширення, вони мають згорткові шари, що складаються з невеликої кількості нейронів, які приймають на вхід обмежену область даних. Ці шари мають загальні ваги, таким чином, процес навчання еквівалентний процесу навчання фільтрів, які застосовуються для кожної області даних. Через невелику кількість з'єднань, вони навчаються більш ефективно, і в той же час ці фільтри автоматично виділяють ознаки з даних.

Нейрони згорткових нейронних мереж мають неперервні вхідні та вихідні сигнали. Функція передачі нейронів має сигмоїдну нелінійність:

$$\sigma(x) = \frac{1}{1 + e^{-x}}.$$

Вихідні сигнали нейронів визначаються таким чином:

$$y = \sigma \left(\sum_{k=1}^K w_k \cdot x_k + \theta \right).$$

де w_k — вагові коефіцієнти зв'язків нейрона, x_k — вхідні сигнали нейрона, θ — поріг нейрона. Параметри w_k та θ змінюються при навчанні нейронної мережі. Індекс k для нейронів першого шару сумація виконується в межах рецептивного поля нейрона, для нейронів з наступних шарів додається ще третій вимір — номер площини карти ознак.

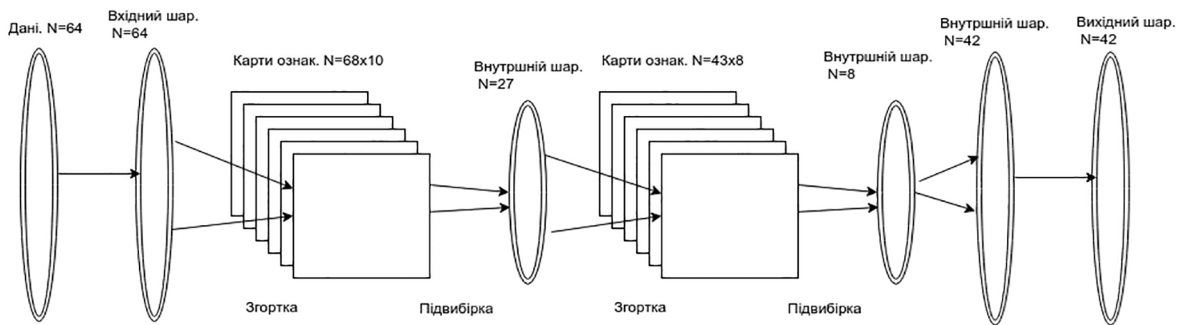


Рисунок 1.1. Архітектура згорткової нейронної мережі

У роботі було побудовано і навчено згорткову нейронну мережу, архітектуру якої показано на рис 1.1. На вході нейронної мережі фрейм із сигналу ЕКГ. На виході шар зі 42 нейронів, кожен з яких активується, якщо у сигналі присутня відповідна аномалія. Останні 2 шари являються БШП зі невноз'єднаними шарами, що виконує роль класифікатора. Попередні шари виконують роль згорткових шарів, що призначені для виявлення характерних ознак.

Згорткова нейронна мережа показала значно кращі результати ніж багатозаровий перцептрон. На рис. 1.2 показано графік загальної точності класифікацій для ЗНН від кількості ітерацій навчання. Перед навчанням було проведено ініціалізацію ваг. Як видно з графіку мережа уже при 500 ітераціях здатна правильно розпізнавати 95% даних. Після тренування згорткова нейронна мережа здатна правильно розпізнавати 98% даних.

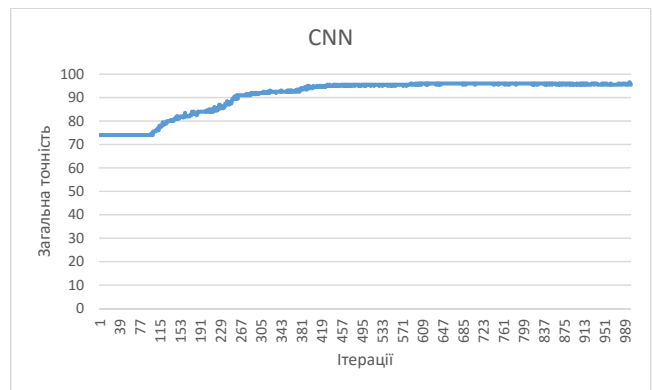


Рисунок 1.2. Графік загальної точності класифікацій для ЗНН від кількості ітерацій навчання

У таблиці 1.1 показано порівняльну характеристику різних підходів, які було досліджено у роботі. Найкраще підходять для класифікації аномалій згорткова нейронна мережа та багаторушній автокодувальник. Це пов'язано оптимізацією архітектури для класифікації із попереднім виокремлення значущих ознак. Так у ЗНМ наявні згортковий шар, який здатний знаходити карти ознак. Натомість автокодувальник містить вектор зжатих ознак, що має меншу розмірність, ніж

вхідні дані, за рахунок чого і вдається виділяти характерні ознаки і будувати класифікатор краще за БШП.

Таблиця 1.1

Порівняльна характеристика різних підходів до діагностики ЕКГ

| Модель нейронної мережі | Загальна точність класифікації |
|-------------------------|--------------------------------|
| БШП | 84,7% |
| ЗНМ | 98% |
| КА | 98% |

ЗНМ і КА проявили високу точність класифікації аномалій, що дає підстави застосовувати їх для спеціалізованих системах діагностики ЕКГ. Порівнюючи результати з іншими дослідниками також можна відзначити ефективність використання ЗНМ і КА. Так в середньому точність класифікації дещо вища, ніж в роботах інших дослідників [4], результати яких наведено у таблиці табл. 1.2.

Таблиця 1.2

Порівняльна характеристика різних підходів до діагностики ЕКГ інших дослідників

| Модель нейронної мережі | Загальна точність класифікації |
|-------------------------|--------------------------------|
| ANN | 79–96% |
| SVM | 88–96% |
| Нечіткі мережі | 95–98% |
| Модель Маркова | 70–87% |

Отож використовуючи і комбінуючи різні підходи до діагностики ЕКГ та інших параметрів пацієнтів можна з високою точністю діагностувати аномалії. Завдяки тому що найбільше обчислювальних витрат необхідно для навчання нейронних мереж, уже навчені нейронні мережі використовують мало обчислювальних ресурсів для діагностики аномалій в ЕКГ сигналі, тому описаний в роботі метод може бути використаний на переносних пристроях із малими обчислювальними потужностями.

Висновки

Стрімкий ріст кількості і смертності від серцево-судинних захворювань вимагає підвищення якості і доступності діагностики захворювань за допомогою

аналізу електрокардіограм. Під час роботи над магістерською дисертацією було проаналізовано існуючі підходи для діагностики аномалій та підходи до обробки сигналів ЕКГ. Також було досліджено глибинні нейронні мережі і їхнє застосування в медицині.

В роботі було запропоновано програмний продукт, що реалізує алгоритм обробки сигналів ЕКГ та їх діагностики, ідея якого полягає у:

- обробці фільтрами від шумів і зміщення базової лінії;
- застосування вейвлет перетворення до сигналу ЕКГ;
- пошук характерних ознак та класифікація аномалій за допомогою багатоповового перцептрону, згорткових нейронних мереж та компонованого автоекнккодера.

Даний програмний продукт було реалізовано на мові програмування Python. Для обробки сигналу і тренування нейронних мереж було використано бібліотеки PyWT, NumPy, Theano та lasagna.

Для навчання нейронних мереж було застосовано базу даних Phisionet, частина якої також використовувалась для перевірки коректності класифікації. Експериментальним шляхом було встановлено оптимальні параметри нейронних мереж, у тому числі глибину і характеристики зв'язності шарів багатоповового перцептрона, топологію згорткової нейронної мережі та компонованого автоекнккодера. Також у роботі було дослідження швидкість навчання нейронних мереж.

Також було досліджено класифікацію аномалій навченими нейронними мережами. Найкращі результати показали згорткові нейронні мережі та компонований автокодувальник. Точність класифікації становила 97–98%. Гірше себе показав багатопововий перцептрон. Для нього точність становила лише 84,7%. Це зумовлено тим, що структура багатоповового перцептрона не дає змогу виділити характерні ознаки сигналу ЕКГ і забезпечити високу точність класифікації. З іншої сторони архітектура згорткової нейронної мережі і компонованого автокодувальника містить слої, які дали змогу виділити характерні ознаки сигналу, і підвищити в свою чергу точність діагностики аномалій.

Підвищити точність класифікації та функціональний спектр роботи мережі можна за рахунок використання на вхід кількох каналів сигналу електрокардіограми та параметрів хворих як артеріальний тиск, вік, вага і т.д.

Список літератури

1. R. Martis, C. Chakraborty, A. Ray Wavelet-based Machine Learning Techniques for ECG Signal Analysis, 10 Dec 2013.
2. A. Ignatov Deep Learning in information analysis of electrocardiogram signals for disease diagnostics
3. X. Jia, K. Li, X. Li, A. Zhang A Novel Semi-supervised Deep Learning Framework for Affective State Recognition on EEG Signals http://www.cse.buffalo.edu/DBGROUP/bioinformatics/papers/cameraready_xwjiaibibe.pdf
4. Abdel-Badeeh M. Machine Learning in Electrocardiogram Diagnosis / [Електронний ресурс]. – Режим доступу: <https://fedcsis.org/proceedings/2009/PTI/pliks/10.pdf>

Исаев Сайфулла Маматович

доцент ТУИТ Каршинского филиала

Туйчиев Бекзод Оромович

старший преподаватель ТУИТ Каршинского филиала

Норкobilов Салим Абдухамидович

ассистент ТУИТ Каршинского филиала

Карши, Республика Узбекистан

ПРИНЦИПЫ УСТРОЙСТВА И ЛОГИЧЕСКАЯ СТРУКТУРА АВТОМАТИЧЕСКОЙ СИСТЕМЫ ОБНАРУЖЕНИЯ СИГНАЛОВ

Summary. Соответствующая подсистема имеет дело с измеримыми свойствами, которыми могут быть выходы ортогональных фильтров или любых других возможных операторов.

Если выходной сигнал непрерывный то он квантуется а если дискретный, то таким и воспринимается, причем каждый дискретный входной сигнал трактуется как измерение. Предложенные в данной работе структуры устройств автоматической системы обнаружения сигналов позволяет на базе выпускаемых серийных интегральных схем и микропроцессоров разработать модельные и опытные образцы системы.

The corresponding sub-system deals with measurable properties that can be orthogonal outputs of filters, or any other potential operators.

If the output signal is continuous it is quantized as if the logic, then this is perceived, and each digital input signal is treated as a dimension. Proposed in this paper, an automatic signal detection system allows the device structure based on mass-produce integrated circuits and microprocessors to develop a model and prototype systems.

Введение. Данная работа является продолжением работы [1] и обсуждается в ней логическая структура и устройство автоматической системы обнаружения сигналов основанных на результатах приведенных в той работе.

Существует три основных подсистемы автоматической системы обнаружения:

а) подсистема, которая оперирует с сигнальной средой и преобразует измеряем не параметры в объемный элемент пространства наблюдений;

б) подсистема, которая запоминает прошлые наблюдения в простейшей форме;

в) подсистема, которая оперирует с текущими результатами измерения в пространстве наблюдения и прошлыми результатами измерения в пространстве наблюдения и прошлыми результатами измерения в памяти с целью получения одного из следующих решений: сигнал наблюдается впервые или сигнал наблюдался ранее.

Пространство наблюдений. Соответствующая подсистема имеет дело с измеримыми свойствами, которыми могут быть выходы ортогональных фильтров или любых других возможных операторов. Если выходной сигнал непрерывный то он квантуется а если дискретный, то таким и воспринимается, причем

каждый дискретный входной сигнал трактуется как измерение. Каждое измерение напряжения подается в устройство, выход которого равен «единице», если свойство принимает некоторую величину, и «нулю» — в противном случае. Сигнал с выхода каждого характеристического оператора подается на бинарный преобразователь. Число замеров на параметр является переменным и зависит от числа состояний входного параметра. Процедура достижения переменной скорости квантования описывается в [1]. В соответствии с вышесказанным имеется два выходных сигнала, направляемые в устройство памяти (рис. 1).

Запоминание прошлой информации. Запоминающее устройство (рис. 2), которое запоминает «прошлое» является трехмерным запоминающим устройством на сердечниках. Обозначим координаты через X, Y, Z . Каждая точка на плоскости XU представляет собой информацию единичного наблюдения. Плоскость XU разбивается на подобласти, одна на каждое состояние. Следовательно, все системы с двумя областями, системы с семью состояниями — в системы областями и т.д.

Размер каждой области может быть выбран в соответствии с ожидаемыми числом систем, определенное число состояний которых будут наблюдаться.

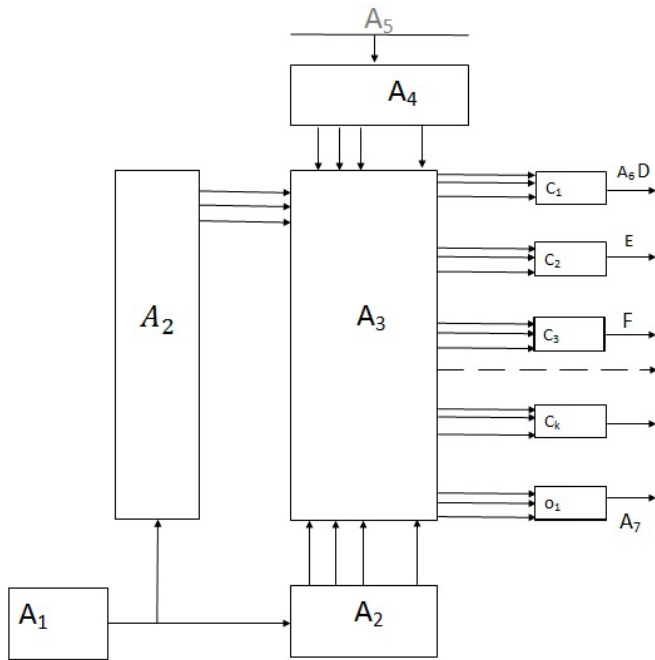


Рис. 1. Запоминающее и решающее устройство:

C_i — двоичное число, определяющее положение i -го признаке; O_1 — двоичное число, определяющее число решение в пользу частного наблюдения; A_1 — программа поиска первичной памяти; A_2 — коммутационная матрица; A_3 — первичное запоминающее устройство; A_4 — коммутационная матрица и схема переключения; A_5 — номер квантованного уровня входа; A_6 — двоичные выходные координаты; A_7 — частота решений

Этого недостаточно для хорошего определения границ областей, причем в действительности границы могут пересекаться. На Z -оси запоминающего устройства запоминаются бинарные значения данных для каждого наблюдаемого источника.

Поскольку размер выборки регулируется автоматические и точная оценка номера состояния всегда получается сразу после окончания измерения, то можно попасть в подобласть, которая соответствует измеренному номеру состояния. Эта подобласть определяется координатами X_0Y_0 являющимися началом обходе запоминающего устройства. Для данной точки X, Y вдоль Z -оси имеется K -бинарных запомненных слов, соответствующих K измеряемым параметрам. Слово O используется для определения числа решений, сделанных в пользу каждого состояния. Если измеряются другие параметры, относительно которых известно, что они не имеют разрабо-

са, то они также могут запоминаться на Z -оси каждой раз, когда (XY) -пара селектируется, множество слов $S = \{C_1, \dots, C_K\}$ считывается одновременно. Каждое слово передает адресную информацию во второе запоминающее устройство (рис. 2).

Второе запоминающее устройство (рис. 2), являющееся трехмерным, служит для запоминания распределения вероятностей, которое строится на основе прошлых наблюдений. Оно может также содержать распределения, которые предварительно вводится в него X -вход состоит из N полюсов, соответствующих N квантованным уровням на каждый параметр. Y -вход в это запоминающее устройство является просто указателем местоположения или адресным числом. С адресов передают выбранную по Y информацию в запоминающее и решающее устройство (рис. 2.).

Координата Y выбирается на основе ожидаемого числа статистик, которое будут наблюдаться. Очевидно, запомненные статистики всегда будут разделены. Итак, бинарный код C_1 выбирает линию Y_1 . Фактически наблюдаемое состояние подается на соответствующие клеммы X -оси. Пара образуется тогда и только тогда когда результат при Y_1 содержит информацию, соответствующую одному из наблюдаемых состояний. Если состояние X -входа не содержится в памяти при Y_1 или состояние Y_1 не содержится при X -входе, то это представляет собой несогласующуюся пару. Следовательно, наблюдаемое измерение не относится к сигналу, имеющему данные координаты. Тогда выбирается другая пара координат (X, Y) и процедура повторяется.

Допустим, что для каждого $\{C_i\}$ входные состояния $X_{11}, X_{12}, \dots, X_{1N}$ спарены с соответствующим рядом вдоль Y -оси в запоминающем и решающем устройстве (рис. 2). В таком случае вся запомненная информация при этих координатах считываются вдоль Z -направления. Эти числа являются вероятностями

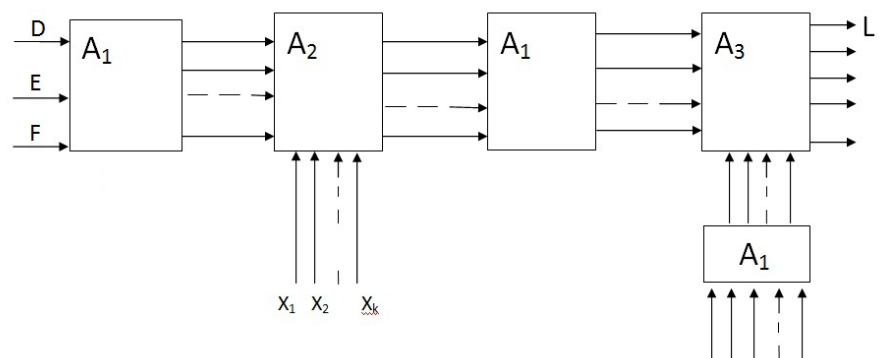


Рис. 2. Запоминающее и решающее устройство: A_1 — коммутационная матрица и схема переключения; A_2 — вторичное запоминающее устройство; A_3 — третье запоминающее устройство

состояний. После этой стадии информация поступает в третью подсистему.

Решающая подсистема. Каждое слово, которое представляет собой вероятность, связанную с данным состоянием, есть адрес для Y-координаты чисел $\{\lambda_i \log \lambda_{ij}\}$ образующих таблицу. Измеренные значения выбирают X-координату таблицы. Следовательно оперативная память используется для заполнения таблицы, для вычисления каждого члена информационной статистики. Это обеспечивает высокую скорость вычисления. Так как таблица не должна быть точной, то запоминающее устройство может быть дорогостоящим. По мере накопления последующих измерений достигается большая точность, ибо округление ошибок дает как положительную так и отрицательную величину и имеет тенденцию к взаимообогащению.

Вычисляется информационная статистика

$$H_2(S) = - \sum \lambda_i \log \lambda_{ij}$$

И это число временно запомнится в буферном устройстве. Затем исследуется следующий элемент в выбранной области блока памяти (рис. 1), процедура используется для выбора тех измерений, которые соответствуют тому же состоянию что и поступившее измерение. Если измерение соответствует $H_e(S)$, оно сравнивается с $H_2(S)$ (рис. 3). И наименьшее из двух остается в памяти, а большее отбрасывается. Такая процедура продолжается до тех пор пока наименьшее значение не останется единственным. Эта величина, скажем $H_2(S_1)$ затем сравнивается с доверительным интервалом E_0 . Если E_0 оказывается меньше или равным $H_2(S_1)$ то предполагается, что измерение пришло от чего-то источника, и этот факт может быть зафиксирован в дополнительной памяти. Таким образом может определяться частота появления каждого состояния.

Если наблюденное значение новое, эта информация запоминается и в памяти образуется новый элемент, положение которого запоминается в устройстве запоминания (рис. 1).

Логическая структура была выбрана таким не традиционным путём, как результат компромисса между скоростью вычисления и объёмом памяти. Чем сложнее система, тем больше растягивается во времени процесс измерения и принятия решения. Это закономерно так как сложная структура требует более детального анализа и с точки зрения теории

информации несёт больше потенциальной информации. В общем случае имеет место сокращение времени решения для систем с опытом по сравнению с новыми системами.

Наконец можно попытаться организовать систему обнаружения без тяготения к какой либо структуре практической системе. В частности, попытаться построить систему с такими свойствами, которые являются общими для всех систем, и индивидуальные особенности обновляются природой объекта измерения.

Нужно заметить что в устройстве 2 (рис. 2) ось Y представляет собой общее число отдельных распределений которые будут наблюдаться. В общем случае оно не известно. Однако нет необходимости сразу создавать запоминающее устройства большой ёмкости. Если потребуется, большие ёмкости могут быть добавлены в устройство 1 (рис. 1) и устройство 2 (рис. 2).

Удвоение устройства 2 (рис. 2) присоединением второй памяти потребует добавления одной биты на измеряемый параметр. Это может быть достигнуто добавлением требуемого числа плат с сердечниками, которые составляют лишь малый процент общей памяти. Это отразится в организации таблицы значений $\{\lambda \log p\}$ информационная статистика (уравнение (8) в {1}) может быть записана как

$$H2(S) = \sum_{i=1}^N n_i \log 2 \frac{n_i}{n_{i2}} < \frac{E(2)}{2}$$

где n, λ, S — целое число.

Таблица $\{\lambda \log p\}$ размером 64 на 64 имеет 9 бит на вход и требует 36,864 бит памяти. Максимальная величина входа есть $64*6=384$. Следовательно диапазон был разделен на 512 значений, а не на 384.

На первый взгляд кажется, что таблица используется только для объёма выборки $S=64$. Проведенные анализы показали, что пока $\{n_i\}$ не превышает 64, таблица является обоснованной. Нужно ещё заметить, что изменение S регулируется автоматически как

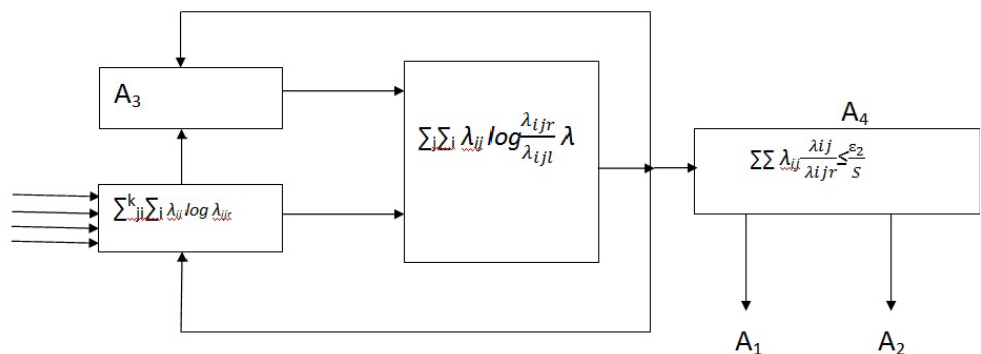


Рис. 3. Запоминающее и решающее устройство: A_1 — новый сигнал; A_2 — буферная память; A_4 — сравнение по доверительному интервалу

функция числа уровней квантования, для того чтобы сохранить фиксированный доверительный интервал и доверительный уровень. Увеличение числа уровней квантования ведёт к увеличению объёма выборки. Однако количество выборок на квантованный уровень n_i может быть почти всегда ниже 64. Очевидно, что пока $S/N-1$ (объём выборки на число степеней свободы) меньше 64, $\{n_i\}$ будет почти всегда меньше 64.

Заключение. Предложенные в данной работе структуры устройств автоматической системы обнаружения сигналов позволяет на базе выпускаемых серийных интегральных схем и микропроцессоров разработать модельные и опытные образцы системы.

Литература

1. С.М Исаев, Б. Туйчиев, Дж. Ш. Саидмурадов. Об одной обобщённой математической модели автоматической системы распознавания сигналов. Международная конференция «Актуальные проблемы развития телекоммуникаций и информационного общества» Ташкент 26–27.06.2012.

Петришенко Сергій Олександрович

студент

Національний технічний університет України «Київський політехнічний інститут»

Петришенко Сергей Александрович

студент

Национальный технический университет Украины «Киевский политехнический институт»

Petrishenko S.

student

National Technical University of Ukraine «Kyiv Polytechnic Institute»

УПРАВЛІННЯ ПАМ'ЯТТЮ В IOS ЗАСТОСУВАННЯХ

УПРАВЛЕНИЕ ПАМЯТЬЮ В IOS ПРИЛОЖЕНИЯХ

MEMORY MANAGEMENT IN IOS APPLICATIONS

Анотація. Досліджено основні особливості управління пам'яттю в мобільних застосуваннях для операційної системи iOS на основі мови програмування Swift.

Ключові слова: екземпляр класу, сильне посилання, слабе посилання, невизначене посилання, опціональний тип, ініціалізатор.

Аннотация. Исследованы основные особенности управления памятью в мобильных приложениях для операционной системы iOS на основе языка программирования Swift.

Ключевые слова: экземпляр класса, сильная ссылка, слабая ссылка, неопределенная ссылка, опциональный тип, инициализатор.

Summary. Were investigated main features of memory management in iOS applications using Swift.

Key words: class instance, strong reference, weak reference, unowned reference, optional type, initializer.

Як і в мові Objective-C, в мові Swift застосовується підрахунок посилань в якості основного механізму для відслідковування моментів, коли оперативна пам'ять, яка була динамічно виділена, більше не використовується і може бути звільнення для інших ресурсів. Протягом багатьох років (та й тепер) в мові Objective-C як і раніше застосовується ручний підрахунок посилань. В результаті нетривалого експериментування зі збирачем «сміття» в Objective-C для автоматичного управління пам'яттю, в 2011 році компанія Apple оголосила про розробку механізму ARC (Automatic Reference Counting). У ньому застосовується той же самий підхід, що і при ручному підрахунку посилань, але тільки більш суворим і детермінованим чином.

Принцип дії підрахунку посилань

ARC (automatic reference counting) використовується для відстеження та управління пам'яттю мобільного застосування. ARC автоматично звільняє

пам'ять, яка використовувалася екземпляром класу, коли ці екземпляри більше не потрібні. Кожен раз, коли створюється екземпляр класу, ARC виділяє шматок пам'яті для зберігання інформації цього екземпляра. Цей шматок пам'яті містить інформацію про тип екземпляра, про його значення і про будь-які властивості, пов'язаних з ним. Додатково, коли екземпляр більше не потрібен, ARC звільняє пам'ять, використану під цей екземпляр, і направляє цю пам'ять туди, де вона потрібна. Це свого роду гарантія того, що непотрібні екземпляри не будуть займати пам'ять.

Однак, якщо ARC звільнить пам'ять використаного екземпляра, то доступ до властивостей або методів цього екземпляра буде неможливий. Якщо спробувати отримати доступ до цього екземпляра, то додаток швидше за все видасть помилку і зупинить свою роботу. Для того, щоб потрібний екземпляр не пропав, ARC веде облік кількості властивостей, констант, змінних, які посилаються на кожен екземпляр класу. ARC не звільнить екземпляр, якщо є хоча б

одне активне посилання. Для того щоб це було можливо, кожен раз як привласнюється екземпляр властивості, константи або змінній створюється strong reference (сильний зв'язок) з цим екземпляром. Такий зв'язок називається «сильним», так як він міцно тримається за цей екземпляр і не дозволяє йому звільнитися до тих пір, поки залишаються сильні зв'язки.

Наприклад, нехай є клас Person, який визначає константну властивість name:

```
class Person {
let name: String
init(name: String) {
self.name = name
print(«\ (name) is being initialized»)
}
deinit {
print(«\ (name) is being deinitialized»)
}
}
```

Клас Person має ініціалізатор, який встановлює властивість name екземпляра і виводить повідомлення для відображення того, що йде ініціалізація. Так само клас Person має деініціалізатор, який виводить повідомлення, коли екземпляр класу звільняється.

Наступний приклад код визначає три змінні класу Person?, який використовується для установки декількох посилань до нового екземпляру Person в наступних шматках коду. Так як ці змінні опціонального типу Person?, а не Person, вони автоматично ініціалізуються зі значенням nil, і не мають жодних посилань на екземпляр Person:

```
var reference1: Person?
var reference2: Person?
var reference3: Person?
```

Тепер можна створити екземпляр класу Person і привласнити його однією з цих трьох змінних:

```
reference1 = Person(name: «John Appleseed»)
// Prints «John Appleseed is being initialized»
```

Повідомлення «John Appleseed is being initialized» виводиться під час того, як викликається ініціалізатор класу Person. Це підтверджує той факт, що відбулася ініціалізація.

Так як новий екземпляр класу Person було присвоєно змінній reference1, значить тепер існує сильне посилання між reference1 і новим екземпляром класу

Person. Тепер у цього екземпляра є як мінімум одне сильне посилання, значить ARC тримає під Person пам'ять і не звільняє її.

Якщо надати іншим змінним той же екземпляр Person, то додасться два сильних посилання до цього екземпляра:

```
reference2 = reference1
reference3 = reference1
```

Тепер екземпляр класу Person має три сильні посилання. Якщо знищити два з цих трьох посилань (включаючи і первісне посилання), присвоємо nil двом змінним, то залишиться одне сильне посилання, і екземпляр Person не буде звільнений:

```
reference1 = nil
reference2 = nil
```

ARC не звільнить екземпляр класу Person до тих пір, поки залишається останнє сильне посилання, знищивши яке вказуємо на те, що екземпляр більше не використовується:

```
reference3 = nil
// Prints «John Appleseed is being deinitialized»
```

Слабкі (weak) посилання

Слабкі посилання не утримуються за екземпляр, на який вони вказують, так що ARC не бере їх до уваги, коли підраховує посилання екземпляра. Такий підхід дозволяє уникнути ситуації, коли посилання стає частиною циклу сильних посилань. Слабке посилання вказується ключовим словом weak перед ім'ям властивості або змінною, яка об'являється. Слабкі зв'язки використовуються циклів посилань, коли є ймовірність того, що в якийсь момент часу посилання матиме значення «nil».

Через те, що слабким посиланням дозволяється мати nil, що означає «відсутність значення», то потрібно оголошувати кожне слабке посилання як те, яка має опціональне значення. Опціональні типи — найбільш пріоритетний тип для відображення, тому що змінна може не мати значення в Swift.

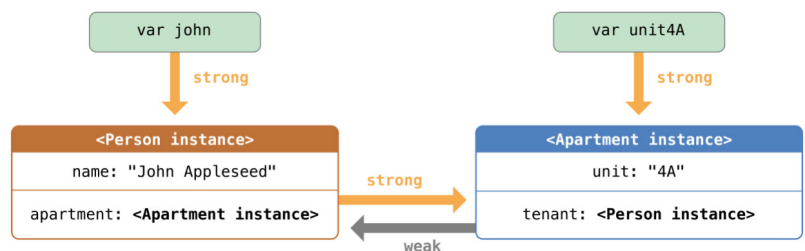


Рисунок 1. Зв'язок між двома екземплярами

Так як слабке посилання не сильно тримається за екземпляр, на який вказує, то можна звільнити екземпляр, на який вказує таке посилання. Таким чином ARC автоматично присвоює слабкому посиланню `nil`, коли екземпляр, на який воно вказує, звільняється (рис. 1) [1].

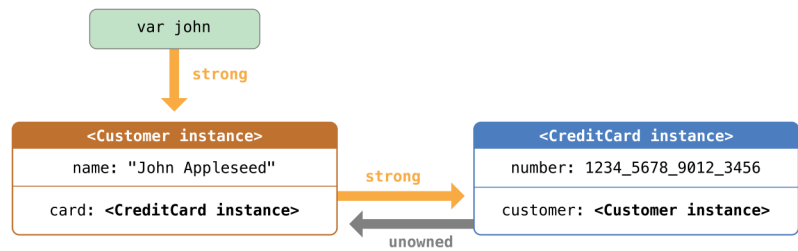


Рисунок 2. Зв'язок між двома екземплярами

Невизначені (unowned) посилання

Як і слабкі посилання, невизначені посилання так само не мають сильного зв'язку з екземпляром, на який вони вказують. На відміну від слабких посилань, невизначені посилання завжди мають значення. Через це невизначені посилання мають неопціональний тип. Слабке посилання вказується ключовим словом `unowned` перед ім'ям властивості або змінною, яка об'являється.

Так як невизначене посилання не є опціональним, то не потрібно і розгортати (`unwrap`) його кожен раз, коли воно буде використовуватися. Тому до невизначеного посилання можна звертатися безпосередньо.

Однак ARC не може встановити значення посилання на `nil`, коли екземпляр, на який воно посилається, звільнений, так як змінні неопціонального типу не можуть мати значення `nil` (рис. 2) [1].

Висновок

Отже, Automatic Reference Counting на даний момент є потужним інструментом, який дозволяє зменшити витрати на управління пам'яттю і сфокусувати їх на основній логіці програмного продукту, що розробляється, адже саме автоматичне і високопродуктивне управління пам'яттю зможе підвищити продуктивність.

Література

1. Офіційна документація мови програмування Swift. — Режим доступу: https://developer.apple.com/library/ios/documentation/Swift/Conceptual/Swift_Programming_Language/. — Дата доступу 12.05.2016.
2. Офіційний сайт Apple для розробників. — Режим доступу: <https://developer.apple.com/library/ios/releasenotes/ObjectiveC/RN-TransitioningToARC/Introduction/Introduction.html>. — Дата доступу 21.06.2016.
3. Офіційний сайт блогу RayWenderlich. — Режим доступу: <https://www.raywenderlich.com/2657/memory-management-tutorial-for-ios>. — Дата доступу 04.05.2016.

Петрішенко Сергій Олександрович

студент

Національний технічний університет України «Київський політехнічний інститут»

Петришенко Сергей Александрович

студент

Национальный технический университет Украины «Киевский политехнический институт»

Petrishenko S.

student

National Technical University of Ukraine «Kyiv Polytechnic Institute»

ПОРІВНЯЛЬНИЙ АНАЛІЗ SWIFT І OBJECTIVE-C

СРАВНИТЕЛЬНЫЙ АНАЛИЗ SWIFT И OBJECTIVE-C

COMPARATIVE ANALYSIS OF SWIFT AND OBJECTIVE-C

Анотація. Розглянуто основні мови програмування під платформу iOS. Проаналізовано основні переваги Swift.

Ключові слова: мова програмування, мобільна платформа iOS, середовище розробки, управління пам'яттю, WWDC.

Аннотация. Рассмотрены основные языки программирования под платформу iOS. Проанализированы основные преимущества Swift.

Ключевые слова: язык программирования, мобильная платформа iOS, среда разработки, управления памятью, WWDC.

Summary: Were considered main programming languages for iOS. Were analyzed basic advantages of Swift.

Програмування — основа основ комп'ютерної техніки. Корпорація Apple давно відома своїм умінням задавати тон розвитку індустрії на роки вперед, але, з огляду на поступове сходження розробників Apple з ринку професійних рішень, вона асоціюється в першу чергу з споживчими товарами. Однак чергове дослідження громадської думки показує, що розробники ПЗ більш ніж задоволені свіжою пропозицією компанії — мовою програмування Swift. Згідно з інформацією ресурсу Stack Overflow, майже 80 відсотків професіоналів із задоволенням працювали або планують працювати з випущеним не так давно інструментом розробки від Apple (рис. 1) [3]. Дослідникам вдалося опитати 26 тисяч відвідувачів з більш ніж 150 країн світу. Близько третини постійно зайнятих в сегменті написання мобільного ПЗ респондентів працюють в основному на платформі iOS, а менше половини також займаються створенням додатків для конкуруючої ОС Android. 20% опитаних не змогли визначитися, швидше за все, сюди входять фахівці, які регулярно розробляють програмне забезпечення для різних платформ.

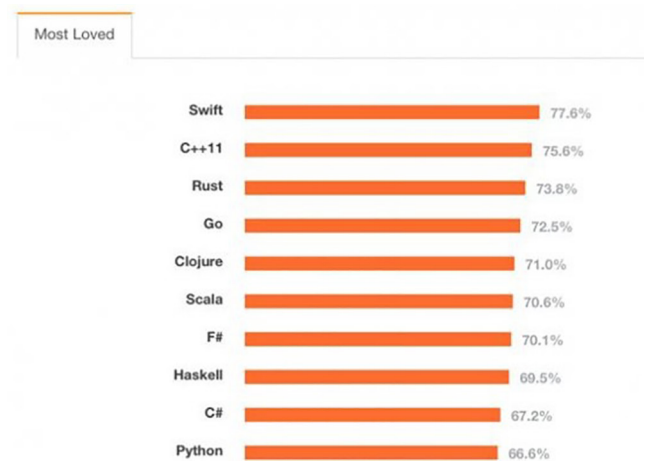


Рисунок 1.4. Оцінки мов програмування

Swift є багатопарадигмовою компільованою об'єктно-орієнтовною мовою програмування, яка була створена компанією Apple для роботи пліч-о-пліч з Objective-C. Swift була представлена 2 червня 2014 році на Всесвітній конференції розробників (WWDC). Swift була створена в першу чергу для розробників під

платформу iOS. Вона покликана повністю замінити Objective-C, адже компанія Apple покладає великі надії на свій продукт, тому Apple настільки ефективно оптимізувала компілятор і саму мову в принципі, що багато можливостей ще тільки належить розкрити. Можна сказати, що Swift «призначена для зльоту від «Hello, World» до цілої операційної системи.

Swift працює з фреймворками Cocoa і Cocoa Touch і сумісна з основною кодовою базою, написаною на Objective-C. Вона являється простою і гнучкою мовою, маючи в той же час більш читабельний код в порівнянні з Objective-C. Адже Objective-C має всі болячки, які можуть бути у мови, побудованої на C. Для диференціації ключових слів і типів, Objective-C вводить нові ключові слова, використовуючи символ «@». Так як Swift не побудована на C, то вона може об'єднати всі ключові слова і видалити численні символи «@» перед кожним Objective-C типом або перед пов'язаним з об'єктом ключовим словом. Також більше не потрібно ставити кому в кінці рядка або писати круглі дужки, щоб виділити умовні вирази всередині операторів if/else. Найбільшою перевагою Swift перед Objective-C є те, що виклики методу не розташовуються усередині один одного. Для виклику функцій і методів в Swift використовується стандартний, розділений комами, список параметрів в круглих дужках. В результаті отримуємо мову зі спрощеним синтаксисом і граматику [2].

Особливістю Swift є відмова від використання і підтримуваним програмістами одночасно двох файлів: файлу заголовка і файлу реалізації. Середовище розробки Xcode і компілятор LLVM (Low Level Virtual Machine) може з'ясувати залежність і виконати покрокові зміни автоматично в Swift. Тому, поділу змісту від тіла стає справою минулого. Swift поєднує в собі заголовок Objective-C (.h) і файли реалізації (.m) в одному файлі коду (.swift). Двох-файлова система Objective-C накладає додаткову роботу на програмістів — і це робота, яка відволікає їх від більш важливих завдань. У Objective-C потрібно вручну синхронізувати імена методів і коментарі між файлами. Xcode і компілятор LLVM можуть працювати непомітно для програміста, знижуючи його навантаження. Використовуючи Swift, програмісти роблять менше допоміжних дій і можуть витратити більше часу на створення логіки додатка. Отже, можна сказати, що Swift створена в тому числі для середнього програміста, адже має в своєму арсеналі містить досить розумні інструменти, які допомагають писати власні програми легше. Структура Swift дозволяє уникнути багатьох поширених помилок, які часто допускають розробники-початківці [1].

Swift є type-safe мовою, вона краще розуміє дії програміста і дозволяє робити більше за менший час. Од-

ним з цікавих моментів в Objective-C можна вважати спосіб обробки вказівників, особливо nil (NULL). У Objective-C нічого не трапиться, якщо викликати метод зі змінною вказівника nil. Рядок коду, де міститься виклик цього методу, не спрацює, і програма продовжить свою роботу, але насправді буде повно помилок, що призведуть до непередбачуваної поведінки. Опціональні типи дають можливість існування в Swift коді nil (опціонального значення), що говорить про можливість створення помилки компілятора при написанні поганого коду. Це створює короткий цикл зворотного зв'язку і дозволяє програмістам писати програми більш впевнено. Проблеми можуть бути усунені при вже написаному коді, що значно зменшує кількість часу і грошей, які витрачаються на виправлення помилок, а також кількість коду, в порівнянні з передачею вказівника на NSError в Objective-C [2].

Традиційно в Objective-C якщо значення було повернуто з методу, то програміст був зобов'язаний зафіксувати поведінку вказівника змінної, що повернулася. В Swift опціональні типи і типи значень дозволяють явно визначити метод, якщо значення існує, або якщо воно може бути опціональним. Для забезпечення передбачуваної поведінки, Swift викликає помилку при виконанні коду, якщо використовується опціональна змінна nil. Ця помилка забезпечує узгоджену поведінку, яка полегшує процес усунення багів, тому що змушує програміста вирішити проблему відразу. Ця помилка виникне на тому рядку коду, де була використана опціональна змінна nil. Це означає, що помилка буде виправлена своєчасно або її вдасться уникнути зовсім [2].

Головною особливістю є те, що Swift має незалежне управління пам'яттю (Automatic Reference Counting — автоматичний підрахунок посилань). Swift уніфікований так, як ніколи не була Objective-C. У Objective-C ARC підтримується всередині Cocoa API, але він не доступний для C коду і API. Це означає, що програміст буде повинен взяти на себе управління пам'яттю при роботі з API, які доступні на iOS. Величезні витрати пам'яті, які програміст може мати в Objective-C, неможливі в Swift. Програмісту не доводиться думати про виділення пам'яті для кожного об'єкта, який він створює, бо ARC обробляє всі управління пам'яттю під час компіляції, і витрати, які пішли б на управління пам'яттю, тепер можуть бути сфокусовані на основній логіці додатка і нових можливостях. Це відбувається тому, що ARC в Swift працює і на процесуальному, і на об'єктно-орієнтованому коді, і тепер не потрібно контекстних переходів для програмістів, навіть якщо вони пишуть код, який розрахований на більш старі API. Це є проблемою для поточної версії Objective-C. Apple змогли вирішити і довести, що автоматичне

і високопродуктивне управління пам'яттю може підвищити продуктивність [1].

Swift використовує меншу кількість коду. Наприклад, для додавання двох рядків Swift використовує оператор «+». Ця особливість відсутня у Objective-C. Така підтримка для об'єднаних символів і рядків має важливе значення для будь-якої мови програмування, яка використовує відображення тексту для користувача на екрані.

Система типів в Swift знижує складність в написанні коду, так як компілятор може з'ясувати типи. Наприклад, Objective-C вимагає, щоб програмісти запам'ятовували спеціальні маркери рядків (% S,% d,% @) і використовували розділений комами список змінних. Swift підтримує інтерполяцію рядків, що усуває необхідність запам'ятовування символів і дозволяє програмістам вставляти змінні, такі як назва ярлика або кнопки, безпосередньо у вбудований рядок користувача. Тип виведення системи і рядки інтерполяції зменшують кількість помилок, які поширені в Objective-C [2].

Компанія Apple велику увагу приділила швидкодії Swift. Як стверджують самі розробники, алгоритм пошуку в Swift виконується до 2,6 разів швидше, ніж в Objective-C, і до 8,4 раза, ніж в Python 2,7. Відповідно до даних Primate Labs, за показниками популярного тесту GeekBench (крос-платформний еталонний тест для вимірювання швидкодії процесора і підсистеми пам'яті комп'ютера) Swift наблизився до експлуатаційних характеристик C++ по обмеженню швидкості обчислень з використанням алгоритму Мандельброта (знаходження множини Мандельброта) в грудні 2014 року [4]. У лютому 2015 року Primate Labs виявили, що Xcode 6.3 Beta поліпшив продуктивність алгоритму GEMM в Swift (алгоритм обмеженої пам'яті з послідовним доступом великих масивів (a memory-bound algorithm with sequential access of large arrays)) до коефіцієнта 1,4. Початкова імплементація FFT (алгоритму обмеженої пам'яті з випадковим доступом

великих масивів) — поліпшення продуктивності в 2,6 рази. Swift показав і інші поліпшені показники при подальшому тестуванні: 8,5-кратне підвищення для FFT алгоритму (залишивши C++ тільки з приростом продуктивності в 1,1 рази). Також, Swift перевершує C++ для алгоритму Мандельброта з коефіцієнтом в 1,03. Практично, Swift на одному рівні з C++ для FFT і алгоритму Мандельброта. Відповідно до даних Primate Labs алгоритм GEMM показав, що компілятор Swift не може векторизувати код так, як може компілятор C++. У C++ є невелика перевага, але Apple обіцяють все виправити в наступних версіях Swift [5].

Одним із важливих нововведень Swift, в порівнянні з Objective-C, є можливість писати код і переглядати результати в реальному часі. Ми можемо внести деякі зміни в програму і відразу побачити результат, так що нам не потрібно перекомпілювати і перезапустити програму. Це має назву Interactive Playgrounds. Також тут можна використовувати Quick Look, який відображає графіку або список результатів, Timeline Assistant, який допомагає експериментувати з кодом UI (інтерфейсом користувача) або продивлятися повний цикл створення анімацій. Даний код можна перенести в основний проєкт. Починаючи з Xcode 7 можна створювати коментарі до коду, використовуючи форматований текст з жирним шрифтом або курсивом, маркований список, вбудовані зображення і посилання. Таким чином, Interactive Playgrounds розроблені для того, щоб зробити програмування більш інтерактивним і доступним [1].

Отже, Swift — це сучасні норми синтаксису, ефективно управління пам'яттю, висока швидкість роботи і інтерактивність. У Objective-C цього немає, але зате є надійність, база документації, прикладів, шаблонів і багато досвідчених програмістів. Підводячи підсумок, можна з повною впевненістю сказати, що Swift — це майбутнє. Все пізнається в порівнянні і, лише відпрацювавши певну кількість проєктів, можна стверджувати, чи зручна мова, і чи можна на ній ефективно працювати.

Література

1. ANON The Swift Programming Language (Swift 2.1) / ANON — Cupertino: Apple Inc., 2014. — 528 p.
2. Vandad Nahavandipoor iOS8 Swift Programming Cookbook / Vandad Nahavandipoor — Boston: O'Reilly Media., 2014. — 902 p.
3. Офіційний сайт Stack Overflow. — Режим доступу: <http://stackoverflow.com/research/developer-survey-2015#tech-super>. — Дата доступу 13.03.2016.
4. Офіційний сайт блогу Primate Labs. — Режим доступу: <http://www.primatelabs.com/blog/2014/12/swift-performance/>. — Дата доступу 20.04.2016.
5. Офіційний сайт блогу Primate Labs. — Режим доступу: <http://www.primatelabs.com/blog/2015/02/swift-performance-updated/>. — Дата доступу 23.04.2016.
6. Офіційна документація мови програмування Swift. — Режим доступу: https://developer.apple.com/library/ios/documentation/Swift/Conceptual/Swift_Programming_Language/. — Дата доступу 12.05.2016.

Піпко Анна Сергіївна

студентка

Національний технічний університет України «Київський Політехнічний Інститут»

Пипко Анна Сергеевна

студентка

Национальный технический университет Украины «Киевский Политехнический Институт»

Ripko A.

student

National Technical University of Ukraine «Kyiv Polytechnic Institute»

ДОСЛІДЖЕННЯ ВИКОРИСТАННЯ КАСКАДНОГО КЛАСИФІКАТОРА ДЛЯ РОЗПІЗНАВАННЯ ПІШОХОДІВ У ВІДЕОПОТОЦІ

ИССЛЕДОВАНИЕ ИСПОЛЬЗОВАНИЯ КАСКАДНОГО КЛАССИФИКАТОРА ДЛЯ РАСПОЗНАВАНИЯ ПЕШЕХОДОВ В ВИДЕОПОТОКЕ

INVESTIGATION OF USING CASCADE CLASSIFIER FOR PEDESTRIAN DETECTION

Анотація. Досліджено використання каскадного класифікатора на ознаках Хаара, HOG та LBP у задачі розпізнавання пішоходів у відео потоці. Зроблено висновки щодо залежності якості детектування від параметрів навчання.

Ключові слова: розпізнавання образів, каскадний класифікатор, ознаки Хаара, HOG, LBP.

Аннотация. Исследовано использование каскадного классификатора на признаках Хаара, HOG и LBP в задаче распознавания пешеходов в видео потоке. Сделаны выводы о зависимости качества детектирования от параметров обучения.

Ключевые слова: распознавание образов, каскадный классификатор, признаки Хаара, HOG, LBP.

Summary. Using cascade classifier with Haar features, HOG and LBP in pedestrian detection was investigated.

Key words: pattern recognition, cascade classifier, Haar features, HOG, LBP.

З кожним днем у суспільства все зростає потреба отримати засіб забезпечення максимальної безпеки людини на проїжджій частині. Причиною значної частки аварій є наїзд на пішохода. Тому вже сьогодні автовиробники пропонують покупцям моделі, оснащені бортовим комп'ютером з системою попередження про пішохода на дорозі. Проте такі автомобілі досить дорогі й мало поширені на українських дорогах. Отже, постає задача знаходження іншого шляху підвищення безпеки дорожнього руху, який міг би набутися масового поширення. Можливим рішенням може бути використання смартфона, який сьогодні є майже у кожного водія. А отже, необхідні методи, які дозволили б виявляти пішоходів, використовуючи лише відео потік з однієї камери.

Метод Віюлі-Джонса, відомий як найбільш популярний метод для детектування, початково застосовувався для виділення облич [1]. Етапами алгоритму є класифікатори бустингу над деревами рішень, що ви-

користують в якості ознак характеристики Хаара. Значна швидкість та ефективність стали причиною численної кількості модифікацій для розв'язання різноманітних задач.

Метою даної роботи є дослідження параметрів формування каскадного класифікатора, який можна було б використати у задачі виділення пішоходів на дорозі.

Для тренування було використано утиліту `opencv_traincascade` з бібліотеки OpenCV (Open Source Computer Vision Library) та вільно розповсюджуваний датасет від Даймлера (Daimler Pedestrian Detection Benchmark Dataset). Архітектура каскадного класифікатора задається такими параметрами (тут використано назви відповідних аргументів утиліти):

– `numPos`, `numNeg`: кількість позитивних та негативних зразків, які були позитивно класифіковані попередніми ступенями каскаду та використовуються для тренування наступного ступеня;

- numStages: кількість ступенів каскаду;
- w, h: ширина та висота фрагмента, що надходить класифікатору для розпізнавання, в пікселях (об'єкти меншого розміру класифікатором не розглядаються), тут були рівними 18 та 36 відповідно;
- minHitRate: мінімальна частка істинних позитивних класифікацій для ступеня каскаду, тут була рівною 0.995;
- maxFalseAlarmRate: максимальна частка хибних позитивних класифікацій для ступеня каскаду;
- maxWeakCount: максимальна кількість слабких класифікаторів в одному рівні каскаду, тут була рівною 100;
- mode: тип набору характеристик Хаара, що можуть використовуватися слабким класифікатором, тут обрано повний набір;
- bt: тип бустингу, тут було обрано Gentle AdaBoost.

Вибір архітектури класифікатора відбувався у декілька ітерацій: на кожному кроці оцінювалась якість класифікатора, отриманого внаслідок навчання із заданими параметрами, та за необхідності коректувались значення параметрів (табл. 1). Для кожного каскаду визначались оптимальні параметри детектора:

- scaleFactor: коефіцієнт зміни розмірів ковзного вікна під час сканування зображення. Чим він менший, тим більша кількість фрагментів буде розглянута і тим більше часу необхідно для обробки кадру; тут був рівний 1.2 для забезпечення допустимої швидкості роботи;
- minNeighbors: мінімальна кількість позитивних спрацювань каскаду в деякому околі фрагменту для підтвердження детекції. Збільшенням параметру можна знизити кількість хибних спрацювань, але в той же час може збільшитись кількість пропущених пішоходів.

Таблиця 1

Параметри каскадних класифікаторів

| Каскад | А | В | С | Л | Н |
|--------------------|--------------|--------------|--------------|-------|------|
| Тип ознак | ознаки Хаара | ознаки Хаара | ознаки Хаара | LBP | HOG |
| numPos | 13600 | 3500 | 5000 | 3500 | 3500 |
| numNeg | 6744 | 6744 | 10000 | 6744 | 6744 |
| maxFalse-AlarmRate | 0,7 | 0,5 | 0,6 | 0,5 | 0,5 |
| numStages | 25 | 25 | 25 | 25 | 25 |
| minNeighbors | 20 | 3 | 5 | 3 | 10 |
| TDR,% | 61,5 | 53,8 | 61,5 | 19,2 | 46,2 |
| FDR,% | 62,8 | 33,3 | 27,3 | 37,5 | 66,7 |
| t, c | 0,087 | 0,074 | 0,079 | 0,046 | 0,07 |
| T, хв | 4731 | 4278 | 3150 | 2254 | 98 |

Порівняння якості каскадів відбувалося за такими параметрами як TDR – процент виявлених унікальних пішоходів у тестовому відео, FDR – процент хибних виявлень. Для тестування було обрано відео тривалістю 44с з роздільною здатністю 640×360 пікселів та частотою 29 кадрів/с, на якому було розмічено 26 унікальних пішоходів (за умови, що мінімальний розмір пішохода, що належить до виявлення, 18×36 пікселів, тобто на відстані до 30 м від автомобіля) та 1476 відповідних еталонних обрамлюючих рамок.

На першому кроці (каскад А) хибні спрацювання (FDR) становили 62,8% (табл. 1). За такої їх кількості каскад непридатний для використання, тому для наступного кроку була змінена пропорція позитивних зразків до негативних на 1:2, щоб збільшити вклад хибних позитивних класифікацій у зваженій сумі помилок, що обчислюється під час бустингу. Також було зменшено максимальну частку хибних спрацювань для кожного рівня каскаду.

Після другого кроку (каскад В) кількість хибних спрацювань суттєво зменшилась, проте зросла кількість пропущених об'єктів (MD). Збільшення об'єму вибірки для навчання підвищило точність та повноту розпізнавання до прийнятного рівня (каскад С), тому надалі буде розглядатися саме така конфігурація.

Також було розглянуто каскади з аналогічними параметрами, але з використанням LBP (каскад Л) та HOG (каскад Н). Проте їх якість була значно нижчою, ніж у варіантах з ознаками Хаара, тому, хоч вони і вигравали у швидкості обробки кадру (t) та процесу навчання (T), далі не досліджувалися.

Розглянемо залежність детекції від кількості рівнів каскаду С (табл. 2). Із нарощенням рівнів каскаду зменшується кількість його спрацювань, за рахунок чого зменшується кількість хибних спрацювань, тому можна зменшувати значення minNeighbors, тим самим попереджуючи небажане помилкове відсіювання позитивних детекцій. Проте не варто обирати надто велику кількість рівнів, оскільки:

- кожний додатковий рівень збільшує час обробки зразка;
- можливе перенавчання;
- спрацювання можуть майже зникнути;
- навчання кожного нового рівня стає все довшим, а приріст якості, як правило, стає з певного моменту несуттєвим.

Таблиця 2

Залежність детекції від кількості рівнів каскаду С

| numStages | minNeighbors | TDR, % | FDR, % | T, хв |
|-----------|--------------|--------|--------|-------|
| 15 | 20 | 65,4 | 77,2 | 962 |
| 20 | 10 | 69,2 | 38,1 | 1846 |
| 25 | 5 | 61,5 | 27,3 | 3150 |
| 30 | 3 | 69,2 | 18,2 | 5410 |

Висновки. В результаті роботи було сформовано каскадний класифікатор з 30 ступенів на ознаках Хаара, який забезпечив найкращу точність та повноту виявлення пішоходів у відео серед отриманих конфігурацій. Для підвищення якості розпізнавання можна збільшувати об'єм навчальної вибірки, зберігаючи

пропорцію позитивних та негативних зразків як 1:2, проте це може значно збільшити і час навчання класифікатора. Використання LBP та HOG замість ознак Хаара дало значно гірший результат при аналогічних параметрах.

Література

1. Viola P. Rapid Object Detection using a Boosted Cascade of Simple Features [Електронний ресурс] / P. Viola, M. J. Jones // Proceedings IEEE Conference on Computer Vision and Pattern Recognition. — 2001. — Режим доступу: <http://www.merl.com/publications/docs/TR2004-043.pdf>

Пирожков Олексій Юрійович

студент

Національний технічний університет України «Київський політехнічний інститут»

Пирожков Алексей Юрьевич

студент

Национальный технический университет Украины «Киевский политехнический институт»

Pirozhkov Oleksiy

student

National Technical University of Ukraine «Kyiv Polytechnic Institute»

ВИКОРИСТАННЯ ТЕХНОЛОГІЇ ВІРТУАЛІЗАЦІЇ ДЛЯ АВТОМАТИЗОВАНОГО ТЕСТУВАННЯ ЗМІШАНИХ КРИПТОГРАФІЧНИХ СИСТЕМ

ИСПОЛЬЗОВАНИЕ ТЕХНОЛОГИИ ВИРТУАЛИЗАЦИИ ДЛЯ АВТОМАТИЗИРОВАННОГО ТЕСТИРОВАНИЯ СМЕШАННЫХ КРИПТОГРАФИЧЕСКИХ СИСТЕМ

USE OF VIRTUALIZATION TECHNOLOGY FOR AUTOMATED TESTING OF MIXED CRYPTOGRAPHIC SYSTEMS

Анотація. У статті розглянута проблема використання середовища віртуалізації VirtualBox для автоматизованого тестування змішаних криптографічних систем.

Ключові слова: захист інформації, тестування у віртуальному середовищі, VirtualBox, криптографічні системи.

Аннотация. В статье рассмотрена проблема использования среды виртуализации VirtualBox для автоматизированного тестирования смешанных криптографических систем.

Ключевые слова: защита информации, тестирование в виртуальной среде, VirtualBox, криптографические системы.

Summary. The article considers the problem of using VirtualBox virtualization environment for automated testing of mixed cryptographic systems.

Key words: information protection, testing in virtual environment, VirtualBox, cryptographic system.

В даний час в силу масового застосування змішаних криптографічних систем (під змішаними криптографічними системами маються на увазі програмно-апаратні засоби захисту інформації) особливої актуальності набувають питання тестування їх працездатності, рішення яких дозволяє підвищити якість продукту, тим самим збільшуючи його конкурентоспроможність. Основна мета тестування — виявлення помилок у програмній реалізації заданих властивостей програмного забезпечення (ПЗ), що в даний час практично неможливо без автоматизації, яка дозволяє істотно збільшити продуктивність процесу тестування і мінімізувати «чоловіча-ський фактор», пов'язаний з помилками тестувальника. Такий підхід зада-

ний сучасною методологією виробництва програмних засобів, при якому тестування інтегровано в життєвий цикл ПЗ, що з урахуванням автоматизації цього процесу підвищує швидкість розробки.

Зазвичай для автоматизації тестування програмного забезпечення (ПЗ) використовуються скрипти, що емулюють дії користувача [1]. Але при перенесенні даного підходу на програмно-апаратні комплекси (ПАК) ЗЗІ розробники тестів стикаються з проблемами, описаними в статті [2]. По-перше, скрипти можуть функціонувати тільки в операційній системі (ОС), тому їх неможливо використовувати для тестування ЗЗІ, що функціонують до старту ОС. По-друге, при тестуванні часто необхідне підключення / відключен-

ня апаратної частини ЗЗІ, що вимагає додаткового втручання тестувальника.

Для усунення цих проблем в даній роботі пропонується використовувати підхід, що дозволяє застосування технологій віртуалізації. Він полягає у використанні віртуальних машин, емулюючих роботу ОС, як показано на рис. 1, де приведена схема стенду для тестування ПАК ЗЗІ. Для контролю виконання і емуляції дій користувача використовується інтерфейс програмування додатків (API), що надається засобом віртуалізації. При цьому апаратна частина ЗЗІ переноситься в віртуальну машину.

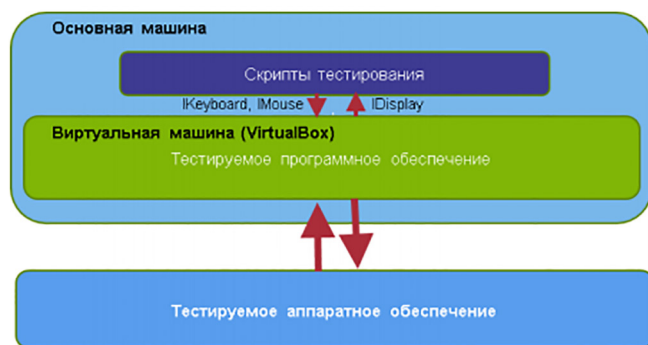


Рисунок 1. Схема стенда для тестування ПАК

Для практичної реалізації даного підходу пропонується використовувати засіб віртуалізації VirtualBox, що має багатий набір API-функцій для реалізації скриптів тестування. Емуляція дій користувача виконується за допомогою інтерфейсів IKeyboard і IMouse, що дозволяють імітувати клавіатурне і курсорне введення через відправку скан-кодів клавіш і подій миші. При цьому необхідно враховувати те, що в ПАК ЗЗІ обробка подій клавіатурного і курсорного введення вимагає певної кількості часу.

Обробку результатів виконання тесту можна здійснити двома способами. Перший спосіб полягає в запису результату в файл з подальшим відправленням в хост-систему (наприклад, по мережі або через розділяється файлову систему). Однак даний спосіб може бути не завжди реалізуємо через особливості функціонування ЗЗІ. Другий спосіб спирається на вибір інформації з інтерфейсу IDisplay, що дозволяє отримати скріншоти дисплея віртуальної машини. В такому

випадку про результат виконання тесту можна судити або шляхом порівняння еталонного і отриманого скріншота, або за допомогою розпізнавання текстової інформації, виведеної скриптом тестування.

При цьому будь-яка змінювана інформація, яка не належить до процесу тестування, наприклад, значення системних годин, відкидається.

Використання даного підходу дозволяє виконувати тестування різних програмно-апаратних ЗЗІ, а саме тих, що функціонують в ОС і тих, що стартують до запуску ОС. Іншою перевагою є те, що в даному випадку виконується повна імітація дій користувача, що дозволяє реалізувати повне функціональне тестування ПАК для оцінки коректності та надійності його функціонування.

Основним обмеженням даного підходу є недоліки сучасних засобів віртуалізації, які не дозволяють передавати деякі апаратні інтерфейси в віртуальну машину. До інших обмежень описаного методу тестування можна віднести неможливість контролю внутрішнього стану тестованого ПАК ЗЗІ, а також виконання тестових сценаріїв в контексті користувача, який може бути обмежений в правах. Крім зазначених обмежень дана технологія автоматизації тестування висуває підвищені вимоги до тестувальників, який повинен володіти навичками програмування для написання і підтримки тестових сценаріїв.

Для подолання зазначених вище обмежень в якості подальшого розвитку описаної методики організації тестування можна запропонувати такі варіанти поліпшень, як доступ до оперативної пам'яті ЗЗІ, і написання бібліотеки для спрощення написання тестів.

VirtualBox API надає інтерфейс IMachineDebugger для доступу до оперативної пам'яті і регістрів процесора. Такий доступ може бути корисний для відстеження стану внутрішніх структур ЗЗІ. Написання бібліотеки, що реалізує повторювані функції (відправка скан-кодів, порівняння скріншотів і т.д.), дозволяє знизити складність написання тестових скриптів.

Технологія віртуалізації була використана при розробці тестових сценаріїв для засобу забезпечення довіреної сеансу (ЗЗДС) «МАРШ!» [3], що дозволило добитися повної автоматизації виконання тестових сценаріїв без внесення будь-яких змін в ОС ЗЗДС «МАРШ!».

Література

1. Graham D., Fewster M. Experience of test automation / D. Graham, M. Fewster — Addison-Wesley, 2012. — 617 с.
2. Борисова Т.М. Особенности автоматизации тестирования программно-аппаратных СЗИ, 2013. Отримано з URL: http://www.okbsapr.ru/borisova_2013_1.html
3. ПАК СОДС «МАРШ!», 2014. Отримано з URL: <http://www.sodsmarsh.ru>

Проценко Михайло Михайлович

студент

Національний технічний університет України «Київський політехнічний інститут»

Проценко Михаил Михайлович

студент

Национальный технический университет Украины «Киевский политехнический институт»

Protsenko Mikhail M.

Student

АНАЛІЗ ФРЕЙМВОРКІВ ЯК ЗАСОБІВ РОЗРОБКИ WEB-ДОДАТКІВ

АНАЛИЗ ФРЕЙМВОРКОВ КАК СРЕДСТВ РАЗРАБОТКИ WEB-ПРИЛОЖЕНИЙ

ANALYSIS FRAMEWORKS AS TOOLS FOR DEVELOPING WEB-BASED APPLICATIONS

Анотація. Досліджені переваги та недоліки використання фреймворків для розробки Web-додатків. Виділено критерії вибору фреймворку для розробки Web-додатку.

Ключові слова: MVC, фреймворк, Back-End, web-додатки, база даних.

Аннотация. Исследованы преимущества и недостатки использования фреймворков для разработки Web-приложений. Выделены критерии выбора фреймворка для разработки Web-приложения.

Ключевые слова: MVC, фреймворк, Back-End, web-приложения, база данных.

Summary. The advantages and disadvantages of using frameworks for developing Web-based applications. Highlight the selection criteria framework for developing Web-based application.

Key words: MVC, Framework, Back-End, web-applications, Database.

Вступ

Швидкість, з якою поширюється інформація сьогодні, є вражаючою. Інтернет став каталізатором до поширення інформації. В Інтернеті знаходяться мільйони веб-сайтів різного спрямування. Розвиток Інтернету нерозривно пов'язано з проектуванням сайтів. Масова поява сайтів спровокувала проблему їх якості. Популярність створення веб-ресурсів сприяла розробці різних систем і програм, які спрощують процес написання сайту. Також вони допомагають підвищити ефективність роботи, а також дозволяють розробнику сфокусуватися на основною логікою програми. Такі технології, як PHP, Java, Microsoft.Net, MySQL, Oracle, Microsoft SQL Server і розроблені на їх основі фреймворки — це каркаси системи або підсистеми, що можуть включати допоміжні програми, мови сценаріїв — і все, що полегшує розробку й об'єднання різних компонентів.

Використання фреймворків стає все більше і більше популярним, і це обґрунтовується тим, що розробка з допомогою фреймворку зменшує навантаження на процес розробки web-додатків, це досягається тим,

що розробка з використанням фреймворку позбавляє від проблеми використання повторюваного коду. Без використання фреймворків, стає набагато складніше створювати web-додатки, супроводжувати і модернізувати їх. Між тим, використання фреймворків робить процес створення програми набагато більш легким і функціональним. Проаналізувавши інформацію з мережі Інтернет, можна побачити що існує сотні фреймворків для створення web-додатків. Тому досить складно зробити вибір фреймворку, так як кожен з них має велику кількість привабливих функцій та доповнень. А неправильний вибір фреймворку може стати основною причиною невдачі проекту.

Що являють собою Web-фреймворки?

WEB фреймворк — це каркас, призначений для створення динамічних веб-сайтів, мережеских додатків, сервісів або ресурсів. Він спрощує розробку і позбавляє від необхідності написання рутинного коду. Багато фреймворків спрощують доступ до баз даних, полегшують розробку інтерфейсу, а також зменшують дублювання коду.

Виділяють п'ять типів веб-фреймворків: Request-based, Component-based, Hybrid, Meta and RIA-based.

Request-based — фреймворки, які безпосередньо обробляють вхідні запити. Збереження стану відбувається за рахунок серверних сесій. Приклади: Django, Ruby on Rails, Struts, Grails.

Component-based — фреймворки, які абстрагують обробку запитів всередині стандартних компонентів і самостійно стежать за станом. Дані каркаси мають багато спільного зі стандартними програмними графічними інтерфейсами. Приклади: JSF, Tapestry, Wicket.

Hybrid-based — фреймворки, які комбінують Request-based та Component-based фреймворки, беручи під свій контроль всі дані і логічний потік в заснованій на запиті моделі. Розробники мають повний контроль над URL, формами, параметрами, cookies і pathinfos. Однак замість того, щоб відобразити дії і контролери безпосередньо до запиту, гібридні фреймворки забезпечують об'єктну модель компонентів, яка поводить себе тотожно в багатьох різних ситуаціях, таких як окремі сторінки, перервані запити, подібні порталу фрагменти сторінок та інтегровані віджети. Компоненти можуть розподілятися окремо і ефективно інтегруватися в інші проекти. Приклади: RIFE.

Meta-based — фреймворки, що мають ряд базових інтерфейсів для загального обслуговування і основу, яка легко розширюється з метою інтегрування компонентів і служб. Приклад: Keel.

RIA-based (фреймворки для розробки Rich Internet Applications (RIA) — фреймворки, що служать для розробки повноцінних додатків, які запускаються всередині браузера. Приклад: Flex.

Найбільш поширеними є Request-based і Component-based веб-фреймворки.

Зібравши і проаналізувавши інформацію, мною було виділено такі характерні компоненти web фреймворків:

- Шаблонизатор. Відповідає за незалежність верстки від програмного коду.
- Роутер. Розпізнає URL, за яким відбулося звернення до сервера.
- Модуль доступу до бази даних.
- Модуль кешування. Прискорює завантаження сторінок.
- Модуль безпеки. Аутентифікація і авторизація користувачів.
- Файли конфігурації.

WEB фреймворки також можуть керувати сесіями, вести логи, спрощувати використання технології Ajax та ін.

Переваги використання фреймворків для розробки Web-додатків

Під час вибору фреймворку можуть виникнути певні труднощі, пов'язані з визначенням завдань, які він може виконувати, та його призначення. Якщо для створення сайту потрібно знайти зручний і простий в освоєнні фреймворк, то необхідно ретельно підійти до питання його вибору та зважити всі «за» і «проти».

Розглянемо загальні переваги використання фреймворку:

- гнучкість і масштабування — завжди має гнучке рішення нестандартних завдань і можливість далі розширення функціоналу шляхом підключення сторонніх бібліотек або окремих класів; ефективне використання ресурсів сервера;
- використання підходу модель-вид-контролер (MVC) суттєво розширює функціонування та гнучкість проекту, так як використовується під час проектування та розробки програмного забезпечення. фреймворки написані розробниками для розробників, що дозволяє мати на виході прекрасно написаний код і своєчасне виправлення помилок;
- наявність детальної документації з використання фреймворку;
- безпека — забираються всі проломи в безпеці, практично немає вузьких місць для SQL-ін'єкцій; фреймворк дозволяє сконцентруватися на вирішенні архітектурних завдань, а не базових, як при розробці без його застосування;
- якість матеріалу на виході

Головна перевага фреймворків, те що вони якнайкраще підходять для створення масштабованих і унікальних сайтів. Жоден масштабний проект не розроблений на готовій CMS — вони для цього не призначені. Майже всі унікальні web-додатки розробляються з використанням фреймворків. Web-проект, розроблений за допомогою фреймворку, розвивається динамічно. При зміні вимог змінюється і сайт, для створення нового розділу або внесення новизни в дизайн, достатньо змінити окремий модуль.

Замінити окремий блок (модуль), створити новий розділ або внести новизну в дизайн.

Недоліки використання фреймворків для розробки Web-додатків

Недоліки застосування фреймворку досить умовні і незначні порівняно з перевагами:

- Важко обслуговувати — якщо проект створював один розробник, а потім з якихось причин він пропадає або просто відмовляється супроводжувати створений ним проект, то його подальший розвиток і обслуговування стає більш складним питанням і часто не вигідним заняттям;

- Ціна розробки – вартість стандартного сайту зробленого на фреймворку з нуля буде дорожче, ніж на готової CMS, тому що часу на розробку витратиться в кілька разів більше. багато коду не використовується і лежить мертвим вантажем в проєкті;
- складність в освоєнні;
- відсутність готових модулів і компонентів, які міг би встановити клієнт, в мережі Інтернет немає ані безкоштовних, ані платних. Всі доробки необхідно замовляти у розробників;

Шляхи та області використання

Для того щоб скористатися всіма можливостями фреймворка потрібен чималий багаж знань в розробці додатків. фреймворки можуть допомогти усунути дуже часту помилку при програмуванні додатків, а саме повторення коду, а також систематизувати процес розробки. Фреймворк є потужним інструментом для такої мови програмування, що швидко розвивається, як PHP, який допоможе організувати ваш код.

Кожна людина має різні вподобання і потреби. Для одного розробника використання фреймворків може допомогти у прискоренні процесу програмування, а для іншого це може здатися марною тратою часу. У більшості випадків це залежить від рівня професіоналізму, але, загалом, фреймворки призначені, щоб заощадити час і абстрагуватися від рутинних завдань.

В основному, фреймворки застосовуються для розробки проєктів складніше, ніж дво- або тресторінковий сайт з текстовими сторінками.

Критерії критерії вибору фреймворку

Фреймворки використовуються в різних областях залежно від їх основних характеристик. Виділимо основні критерії вибору фреймворку в таблиці 1

Висновки

Системи створення web-додатків або фреймворки активно використовуються розробниками при створенні web-додатків із різним функціоналом і рівнем

Таблиця 1

Критерії раціональності використання фреймворку

| Назва | Характеристика |
|---|--|
| Підтримка баз даних | Питання підтримки баз даних в фреймворк дуже важливе. Наприклад частина фреймворків мають вбудований ORM – шар, частина – ні. Залежно від використовуваної бази даних для розробки проєкту доводиться вибирати той чи інший фреймворк. |
| Підтримка спільноти | При розробці додатків, часто виникають моменти коли він сам не в змозі впоратись, з вирішенням проблеми, але при наявності великої спільноти розробників готових допомогти можна вирішити більшість проблем. |
| Документація | Деякі фреймворки не мають достатньої кількості документації, або вона є не актуальною. Тому перед вибором фреймворка необхідно переконатися в тому що документація актуальна, вчасно оновлюється і доповнюється, і що інструкція із застосування проста в розумінні. |
| Продуктивність | Ключовим фактором при виборі фреймворку все ж залишається продуктивність. Так як частина фреймворків, має низьку продуктивність. |
| Безпека | Частина фреймворків немає захисту від SQL-ін'єкцій, відсутність цього функціональ може негативно вплинути на роботу сайту вцілому і привести до краді інформації. |
| Поріг освоєння | Не всі фреймворки прості в освоєнні, це дуже важливо враховувати при виборі, так як на освоєння одного фреймворка може не вистачити й року, а на освоєння іншого – вистачить всього тижня. |
| Швидкість розробки | Доцільно виділити, що на різних фреймворках і різна швидкість розробки проєктів. Приміром, розробка із застосуванням фреймворка zend триває більше ніж із застосуванням yii . |
| Архітектура | Для зручного написання сайту фреймворк повинен використовувати MVC архітектуру. |
| Швидкість розвитку | Цей пункт так само дуже важливий, так як деякі фреймворки оновлюються раз на пару років (codeigniter), а деякі раз на пару місяців. Це дозволяє уникнути використання старого, недопрацьованого коду при розробці. |
| Використання шаблонів при створенні інтерфейсів користувача | Наявність даної функції значно спрощує створення інтерфейсу сайту, а іноді коли потрібний простий сайт то може звести написання дизайну до мінімуму. |
| Створення і перевірка форм | Наявність даного функціоналу в фреймворку полегшує створення полів для вводу тексту, логіну чи паролю до мінімуму. А також додає потрібну майже в усіх проєктах функцію валідації форм. |

складності. Аналіз основних характеристик і можливостей сучасних фреймворків дозволяє вибрати оптимальний варіант для конкретних web-додатків з урахуванням поставлених завдань. Вибір та викори-

стання фреймворків відіграє важливу роль при проектуванні, реалізації та супроводі як простих web-додатків, так і складних програмних комплексів.

Література

1. Результаты тестирования шести ведущих фреймворков на производительность [Электронный ресурс] — Режим доступа до ресурсу: <http://www.alrond.com/ru/2007/jan/25/rezultaty-testirovaniya-6-frameworks/>.
2. Використання PHP фреймворків в розробці сайту [Електронний ресурс] — Режим доступу до ресурсу: <http://ukrbukva.net/page,5,39718-Ispol-zovanie-PHP-freymvorkov-v-razrobotke-saiyta.html>.
3. Сравнение каркасов веб-приложений [Электронный ресурс] — Режим доступа до ресурсу: https://ru.wikipedia.org/wiki/Сравнение_каркасов_веб-приложений.
4. Обзоры Web-фреймворков [Электронный ресурс] — Режим доступа до ресурсу: <https://praktikatech.wordpress.com/category/обзоры-web-фреймворков/>
5. Что такое фреймворк? [Электронный ресурс] — Режим доступа до ресурсу: <http://www.dbhelp.ru/what-is-framework/page/>.
6. Десять причин избегать тяжеловесных фреймворков, а также лишних зависимостей в проекте [Электронный ресурс] — Режим доступа до ресурсу: <http://eah.me/avoid-frameworks/>.

Radkewitsch M.

*Doktor der technische Wissenschaften, Hochschuldozent,
Taschkente Institut der Kraftwagen und Autobahnen, Usbekistan*

Gapirow A. D.,

*Kandidat der technische Wissenschaften, Lehrstuhlleiter,
Taschkente Institut der Kraftwagen und Autobahnen, Usbekistan*

Radkevich M.

*doctor of technical sciences, associate professor,
Tashkent Automobile and Road Institute, Uzbekistan*

Gapirov A. D.,

*candidat of technical sciences, head of the department
Tashkent Automobile and Road Institute, Uzbekistan*

**PROJEKTIERUNG DER ANLAGE FÜR GEWINNUNG DER
ERDÖLPRODUKTENDÄMPFE VON DER TANKSTELLE
DESIGNING THE PLANT FOR OBTAINING OF PETROLEUM
VAPORS FROM GASOLINE STAND**

Abstrakt. Es wird vorgeschlagen, die Erdöldämpfe von der Tankstelle durch Kondensation mit Solar-Kühlanlagen.

Schlüsselwörter. Tankstelle, Erdölproduktendampf, Absorption der Kohlenwasserstoff, Solar-Kühlanlagen.

Abstract. Capture the vapors from gasoline station by condensation with solar cooling plant is offered.

Key words. gasoline station, petroleum products vapor, absorption of hydrocarbon, solar cooling plant.

In der Republik Usbekistan legt man großen Wert auf den Schutz der Umwelt. Im letzten Jahrzehnten verschärft die rasche Entwicklung des Straßenverkehrs wesentlich das Problem der Transportauswirkungen auf die Umwelt. Der Transport-Komplex ist eine mächtige Quelle der Verschmutzung der Umwelt. Von den 35 Mio. t. Emissionen der verschiedenen Quellen erstellen die Emissionen der Motortransport und Straßenindustrie-Unternehmen etwa 89%. Eines der drängendsten ökologischen Probleme der Welt heute ist Automobiltransport, sowie die Elementen ihrer Bedienungskomplexes, die die Umwelt belasten. Während der Verwendung von Kraftstoffen, und zwar seine Tanken, Lagerung, Ablauf und Verlaufen entweichen die Petroleumproduktdämpfen, die gewöhnlich nicht abfangen und in die Atmosphäre auswerfen werden. Z. B. möglicherweise die Gesamtemissionen von Schadstoffen von eine geringe Kapazität-Tankstelle (Tankstelle) ca. 6.57 Tonnen pro Jahr bilden. So können wir sagen, dass die Emissionen von allen Stationen als Ganzes zu ökologischen Katastrophe sind. Es ist seit langem bekannt, dass die Benzin-Dämpfe giftig sind, und neuere Studien darauf hingewiesen, dass Benzin-Dämpfe die Menschengression verursachen.

Daher ist die Verminderung der Emission der Benzindämpfen in die Atmosphäre zwingend notwendig, und die Lösung dieses Problem ist in diese Artikel entwickelt. Bei der Lösung dieser Aufgabe berücksichtigen wir die folgenden Bedingungen:

- das System wirksam sein muss;
- Stromverbrauch muss niedrig sein;
- das System muss einfach zu bedienen sein.

Inwiefern die Frage des Energiesparen gegenwärtig sehr wichtig ist, haben wir den Benzindampfgewinnung-System mit Sonnenenergieverwertung ausgearbeitet.

Bewertung der Auswirkungen der Tankstelle LTD "Umid" auf die Umwelt hat gezeigt, dass von der Tankstelle 6,575128 to p.a. der Schadstoffen, darunter Kohlenwasserstoffe, Benzin- und Dieselöldämpfen, Mineralöl-Kohlenwasserstoffe, ausstoßen [5].

Unter Berücksichtigung der vorstehenden können wir sagen, dass während jedes Ablauf (oder Einfüllen) der Erdölprodukten wirft man in die Atmosphäre etwa 1,1–1,4 Kubikmeter des Dampf-Luft-Gemisches pro 1 Kubikmeter des umgießenden Brennstoffes aus. Das nennt man "die große Atmung", die von dem Jahreszeit

und der Umgebungstemperatur ist. Außerdem gehen sich bei der Lagerung der Brennstoffen an den Tankstellen KW- Dampfemissionen aus den Lagertanks vor. Diese Emissionen sind von Tagestemperaturschwankung abhängig und nennen man "die kleine Atmung". Die Emissionsintensität macht 3–70 Kubikmeter pro Stunde aus. Mehremission (10...40% im Vergleich zum Volumen des Einpumpens) wird durch die folgenden physikalischen Erscheinungen hervorgerufen [2]:

1) molekulare Dampf-Diffusion, die mit solchen Thermoeffekt von Diffusionstransport, wie

- Wärmegleiten (4,6%);
- Diffusiongleiten (6,7%);
- und Stefanovski-Strom (37,9%) begleitet ist:

2) Turbulisierung des Dampf-Luft-Gemisches und der Oberfläche der Flüssigkeit während der Einpumpen- und Auspumpenoperationen.

Mit dem modernsten Rechenverfahren, dass allen oben genannten Faktoren und Flussdichte der verdampften Erdölprodukt berücksichtigt, ist es möglich, den Verlust während des Lagerverkehrs berechnen. Die Berechnungsfehler ist dabei ca. 16%, dass nicht schlecht ist, mit Rücksicht auf die Kompliziertheit und viel Parametern den Prozesse, die während der Einpumpen- und Auspumpenoperationen ablaufen.

Den Ausstoß der Erdöldämpfe in die Luft zu verringern und beseitigen sind verschiedene Gewinnungsmethoden vorgestellt [3, 4]:

- die Kühlung der Dämpfe in Kühlschränken (ohne Druck) bis Kondensation der Kohlenwasserstoffe (Tiefemperaturtechnologie);
- die Kompressionmischung mit Kühlung der Dämpfe Kondensation
- direkte Verbrennung von Kohlenwasserstoffen (bei ihrer hohen Konzentrationen in Dampf-Luft-Gemisch);
- die Adsorption von Kohlenwasserstoffen aus einer Mischung mit Hilfe von Adsorbens und ihrer nachfolgende Desorption;
- das Entmischen des Dampf-Luft-Gemisches mittels der Membranen mit eingestellten Selektivität;
- die Absorption der Kohlenwasserstoff aus der Gemisch und nachfolgende Desorption und Schaumfraktionieren.

Die komplizierte Forschungen zeigten, dass die Absorption der KW-Stoffen mit gekühlten Adsorbens mit dem nachfolgenden Desorption-Prozess am meistens qualitativ und aussichtsreich ist. Die Notwendigkeit der Adsorbensregenerierung aber steigert das Kosten der Anlage.

Die wirksamste Methode ist die Kühlung-Methode für Kondensierung der KW-Stoffen in den Kühlanlagen ohne Druck (Kryogen-Technologie), aber Energiekosten dieser Methode sehr hoch ist. Also unsere zukünftige Auf-

gabe muss die Entwicklung einer Schema für Gewinnung der Petroleum-Produkten-Dampf mit dieser Methode mit Verwendung der Energie aus den erneuerbaren Energiequellen sein.

Für Öl-Dampfgewinnung haben wir eine Methode, die qualitative hochwertige Gewinnung der Petroleum-Produktendämpfe von den Tankstellen mit geringem finanziellen Aufwand gewährleistet, entwickelt. Zur Verringerung oder Beseitigung der Ausstoßen von KW-Stoffen in die Atmosphäre schlagen wir dem Luft-Dampf-Gemisches abkühlen in der Kühlanlage vor. Als Energiequelle halten wir es für angebracht, die Energie der Sonne nutzen. In der Republik Usbekistan beträgt der Anzahl von Sonnentagen 300 Tage pro Jahr und die Kraft der Sonnenstrahlung etwa 13 kW/m². Weil für den Dämpfe-Rückgewinnung Abkühlen nötig ist, muss man in die Gewinnungsschema eine mit Sonnenenergie arbeitende Kühlanlage einschließen. Es sind die Anlagen zur Solar-Kühlung von zwei Arten: mit geschlossene und offene thermodynamische Zyklen bekannt [1, 6]. Zur Kühlung des Luft-Dampf-Gemisch an Tankstellen, glauben wir, ist es möglich, Solar-Absorptionskältemaschine mit einer offenen thermodynamischen Zyklus zu verwenden. Diese Anlage gewährleistet eine Temperatur 5 ... 15 C. Da die Qualität des Öl-Dampfkondensation bei dieser Temperatur nicht möglich ist, muss man den Druck in den Kühlraum zu erhöhen. Der Druck sollte etwa 50 atm sein. In diesem Fall werden die Dämpfe vollständig kondensiert und der Rückgewinnungsgrad kann 97% erreichen.

Weil der Erdöldämpfe-Ausstoß von einen Tankstelle 6.559023 Jato beträgt, kann man mit Hilfe der präsentierende Gewinnungsschema etwa 6,3 Tonnen Petroleum-Produkten wiedergewinnen. Dann wird der Erdöldämpfe-Ausstoß in die Luft um 0.259023 Jato betragen. Das macht eine bedeutende Kraftstoffsparen und gibt einen großen Nutzeffekt, der etwa \$7151,3 per Jahr beträgt. Außerdem macht die Luftverunreinigungsverminderung auch einen ökonomischer Nutzeffekt im Betrage von \$447 per Jahr. Mit Rücksicht auf die Ausgaben für den Einkauf und Montage der Solar-Kühlanlagen wird der Gesamtnutzeffekt etwa \$7473.3 per Jahr von einer Tankstelle betragen.

Auf der Grundlage der durchgeführten Arbeit kann man folgern, dass die Tankstelle eine der wichtigsten Quellen von Schadstoffen ist. Petroleum-Produkten-Dampf ist für Menschen sehr giftig und ihren Einatmen kann sowohl akute als auch chronische Vergiftung verursachen. Toxische Eigenschaften sind mit eine Narkotische Wirkung auf das zentrale Nervensystem verbunden. Vergiftung kann im Falle, daß Benzin-Dampf in die Atemwege durchdringt oder auf die großen Hautoberfläche wirkt, auftreten. Benzin oder Gas eingesaugt sich schnell in die Lunge und Magen-Darm-Trakt. Beim

Benzinsverschlucken treten die Bauchschmerzen, Erbrechen, Schlaflosigkeit, Halluzinationen, Schwitzen, Salivation ein. Bei Anwendung des Benzin auf die Haut wird sie entfettet und man verschiedene Hautkrankheiten verursachen wie Dermatitis und Ekzem erscheinen

kann. Die Anwendung der entwickelten Methode für Reduzierung der Emissionen um 97% für die Tankstelle gibt eine Möglichkeit die Bevölkerung in den angrenzenden Gebieten gesund machen und macht einen Nutzeffekt um \$7473.3 per Jahr an einer Tankstelle.

Quellennachweis

1. Авезов Р. Р. и др. Системы солнечного тепло- и хладоснабжения. М.: Стройиздат, 1990. — 328 с.
2. Автозаправочные станции: Практическое пособие. 1-е изд. — М.: Учет, 2003. — 464 с.
3. Александров А. А. и др. Деньги на ветер. Обзор действующих систем улавливания нефтепродуктов // Ж. «Современная АЭС». №№ 10, 11, 12. 2005.
4. Кулагин А. В. Прогнозирование и сокращение потерь бензинов от испарения из горизонтальных подземных резервуаров. Уфа, 2003.
5. Михеева А., Радкевич М. В. Проблемы загрязнения атмосферы автозаправочными станциями // Труды II Международной научно-практической интернет-конференции «Молодежь. Наука. Инновация». Пенза: РГУИТП, 2011. — С. 537–538.
6. Solar cooling system technology (electron resource). URL: <http://weben.dede.go.th/webmax/content/solar-cooling-system-technology>

Рижко Борис Володимирович

студент

НТУУ «Київський політехнічний інститут»

Рыжко Борис Владимирович

студент

НТУУ «Киевский политехнический институт»

Ryzhko B.

student

NTUU «Kyiv Polytechnic Institute»

**СИСТЕМА АВТОМАТИЗОВАНОГО ЗБОРУ, ОБРОБКИ
ТА АНАЛІЗУ ВЕЛИКИХ ОБСЯГІВ ДАНИХ
СИСТЕМА АВТОМАТИЗИРОВАННОГО СБОРА, ОБРАБОТКИ
И АНАЛИЗА БОЛЬШИХ ОБЪЕМОВ ДАННЫХ
AUTOMATED SYSTEM OF BIG DATA COLLECTION,
PROCESSING AND ANALYSIS**

Анотація. Робота присвячена розробці системи автоматизованого збору, обробки та аналізу великих обсягів даних.

Ключові слова: *hadoop, великі дані, flume, обробка даних.*

Аннотация. Работа посвящена разработке системы автоматизированного сбора, обработки и анализа больших объемов данных.

Ключевые слова: *hadoop, большие данные, flume, обработка данных.*

Summary. The work is dedicated to the development of an automated system of big data collection, processing and analyzing.

Key words: *hadoop, big data, flume, data processing.*

Вступ

В умовах зростання обсягів даних виникає необхідність їх збору, обробки й аналізу для вироблення прогнозувань, отримання якісних та кількісних показників на основі даних і т.д.

Для досягнення даної мети недостатньо продуктивності одного комп'ютера, оскільки через той час, за який інформація буде оброблена одним комп'ютером, вона вже може стати неактуальною.

Існуючі рішення[1] мають великі обчислювальні можливості, які здебільшого направлені на обробку даних, але виключають збереження даних та результатів обробки.

При реалізації системи були використані технології, що призначені для роботи з великими обсягами даних.

Прикладами використання таких систем можуть послугувати: моделювання ризиків, таргетування реклами, аналіз транзакцій у сфері роздрібної торгівлі,

вироблення рекомендацій, аналіз даних на предмет прогнозування помилок (мереж і т.д.), пошукові системи.

Але побудована система не обмежена вказаним переліком.

Представлена в роботі система вирішує проблеми масштабовності:

- за умов збільшення даних,
- за умов збільшення джерел інформації.

Структура системи збору, обробки та аналізу великих обсягів даних

Виходячи з зазначених вимог до системи, було визначено наступні структурні елементи: інфраструктура розподілених обчислень (ІРО), підсистема збору та обробки інформації, система керування базами даних (СКБД).

Структурна схема системи наведена на рис. 1.



Рис. 1. Структурна схема системи збору, обробки та аналізу (розробка автора)

Визначені елементи дозволяють розділити функції всієї системи, що значно полегшує налагодження системи і закладає принципи модульності. Така структура також визначає взаємозамінність кожної з підсистем у випадку необхідності.

Програмне забезпечення побудованої системи

З огляду на поставлені критерії, в якості ІРО було обрано Hadoop. Це рішення дозволяє розгорнути систему на комп'ютерах загального призначення [2]. За необхідності дана ІРО може бути розгорнута на хмарному рішенні, що дає змогу зекономити на придбанні технічного обладнання. Підсистема надає наступні можливості:

- розподілена файлова система,
- фреймворк розподілених обчислень.

Підсистемою збору та обробки даних виступає програмне забезпечення (ПЗ) Apache Flume[3]. Flume є розподіленим та надійним сервісом збору, обробки та переміщення великих обсягів даних. Він має просту та гнучку архітектуру, що базується на потоках даних. Володіє наступними характеристиками: модульний дизайн, масштабовність, сумісність з базою даних, сумісність з ІРО, гнучкі налаштування, розподіленість виконання задач, багатопоточність.

В якості СКБД виступає HBase. Дана СКБД надає наступні можливості: запис-зчитування у режимі реального

часу, інтерфейс управління СКБД, конфігурація розподіленої СКБД між вузлами, додаткові інтерфейси для доступу зі сторонніх програм [4].

Архітектура системи збору, обробки та аналізу великих обсягів даних

Побудована архітектура на основі ПЗ та структурної схеми показана на рис. 2.

В розробленій архітектурі були враховані усі вимоги, що пред'являються структурною схемою та окремим програмним забезпеченням. Отримана архітектура системи відповідає типу master-slave.

Висновки

В результаті аналізу різних ІРО було виявлено, що на сьогоднішній день, за співвідношенням вартість-ефективність для розробки автоматизованої системи

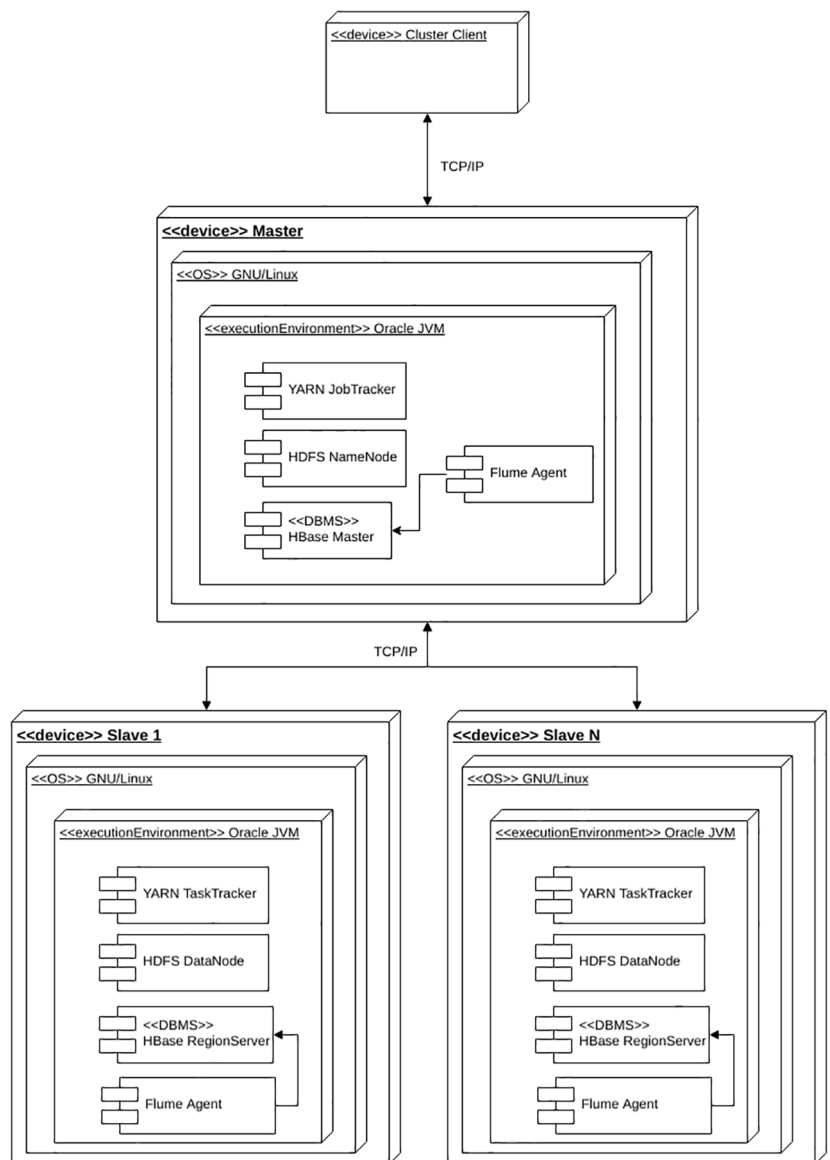


Рис. 2. Архітектура системи (розробка автора)

збору, обробки та аналізу інформації доцільно використовувати кластери, побудовані з використанням Hadoop для загального користування.

Розроблена система забезпечує: масштабовність, модульність, гнучкість у налаштуванні всіх параме-

трів. За співвідношенням вартості до ефективності розроблена система має переваги перед іншими аналогічними комерційними рішеннями.

Література

1. November 2015 | TOP500 Supercomputer Sites [Електронний ресурс] – Режим доступу до ресурсу: <http://www.top500.org/lists/2015/11/> (дата звернення 30.05.2016). – Назва з екрана.
2. How-to: Select the Right Hardware for Your New Hadoop Cluster [Електронний ресурс] // Cloudera – Режим доступу до ресурсу: <https://blog.cloudera.com/blog/2013/08/how-to-select-the-right-hardware-for-your-new-hadoop-cluster/> (дата звернення 30.05.2016). – Назва з екрана.
3. Welcome to Apache Flume – Apache Flume [Електронний ресурс] – Режим доступу до ресурсу: <https://flume.apache.org/> (дата звернення 30.05.2016). – Назва з екрана.
4. Apache HBase Reference Guide [Електронний ресурс] – Режим доступу до ресурсу: <https://hbase.apache.org/book.html> (дата звернення 31.05.2016). – Назва з екрана.

Светлова Олена Костянтинівна

студентка

Національний технічний університет України «Київський політехнічний інститут»

Светлова Елена Константиновна

студентка

Национальный технический университет Украины «Киевский политехнический институт»

Svietlova O. K.

student

National Technical University of Ukraine «Kyiv Polytechnic Institute»

**ЗАСТОСУВАННЯ КЛІТИННИХ АВТОМАТІВ
ДЛЯ ОБРОБКИ ЗОБРАЖЕНЬ
ПРИМЕНЕНИЕ КЛЕТОЧНЫХ АВТОМАТОВ
ДЛЯ ОБРАБОТКИ ЗОБРАЖЕНЬ
APPLICATION OF CELLULAR AUTOMATA
FOR IMAGE ANALYSIS**

Анотація. Проаналізовано можливість застосування клітинних автоматів для обробки зображень.

Ключові слова: клітинні автомати, обробка зображень, сегментація зображень, медіанна фільтрація, метод Оцу.

Аннотация. Проанализировано возможность применения клеточных автоматов для обработки изображений.

Ключевые слова: клеточные автоматы, обработка изображений, сегментация изображений, медианная фильтрация, метод Оцу.

Summary. The possibility of using cellular automata for image processing was analyzed.

Keywords: cellular automata, image processing, image segmentation, median filtering, Otsu method.

У сучасному світі обсяг апаратно-програмних методів, пов'язаних із захопленням, обробкою і зберіганням фото і відео зображень збільшується щорічно на 6–10%. Одним з найбільш важливих видів обробки зображень є виділення об'єктів на цифрових зображеннях. Цей вид має широкий спектр застосування в різних областях, починаючи від астрономічних фотографій, медицини та робототехніки, і закінчуючи контролем якості в промисловості, тому тема цієї статті є актуальною [1].

Що таке клітинний автомат?

Одним з перших авторів цього поняття є Джон фон Нейман. Клітинний автомат — це математична модель, що містить регулярну решітку комірок. Кожна з комірок може перебувати в кінцевому безлічі станів. Для кожної комірки визначено безліч інших, які називають околицею.

Робота клітинного автомата являє собою перехід з одного стану в інший, при цьому наступний стан

повністю визначається попереднім. Новий стан кожної клітини залежить від станів клітин в її околиці і задається певним правилом переходу. Зазвичай, правила переходу однакові для всіх комірок і застосовуються відразу до всієї решітці [2].

Вибір зображень для аналізу

Незважаючи на те, що ми живемо у ХХІ столітті, аналіз зображень цитологічних препаратів здійснюється переважно вручну, що сильно уповільнює процес та призводить до помилок, а погана якість зображення або наявність на зображенні великого числа клітин збільшує ймовірність помилки при з'ясуванні тих чи інших характеристик препарату. Розвиток алгоритмів машинного зору дозволяє поступово перейти від ручного способу аналізу клітинних структур.

Використання автоматичних методів аналізу зображень цитологічних препаратів може значно поліпшити якість аналізу і прискорити його, проте

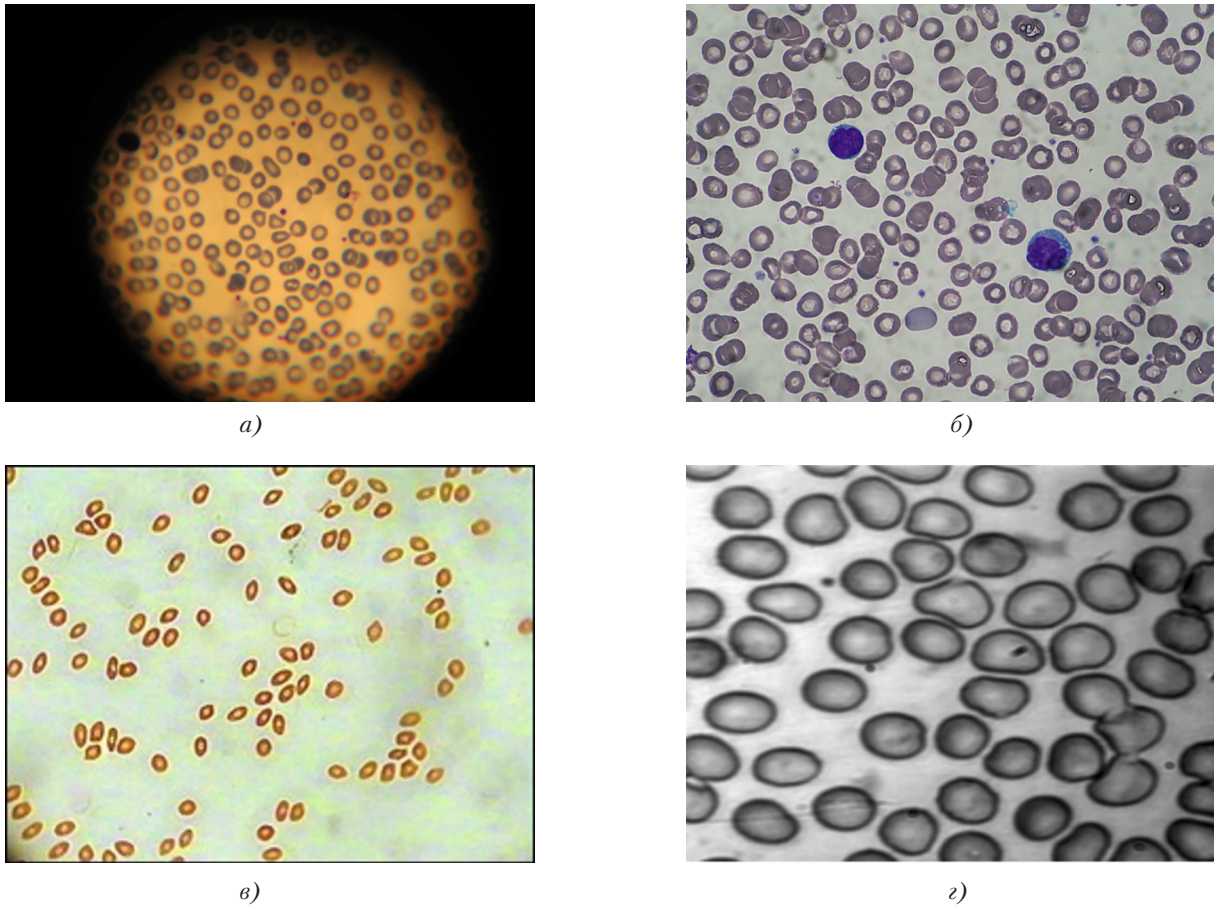


Рисунок 1. Приклади вхідних зображень [3, 4, 5]

універсальний алгоритм розробити важко в силу різноманітності зображень. Клітинні структури можуть виглядати зовсім по-різному, мати прозору, напівпрозору або непрозору структуру, володіти яскраво вираженими межами. Тому доцільно розробляти алгоритми під конкретні класи зображень, що підвищить їх ефективність, але, очевидно, застосування їх буде можливо тільки в межах відповідних класів [3, 5].

В якості вхідних даних виступають зображення кількісного аналізу крові, які були підготовлені різними способами (рис. 1). Для кожного з обраних зображень аналізів крові були розглянуті різні значення

порога (автоматично підібрані, так і підібрані вручну) і підраховано кількість клітин вручну.

Було запропоновано наступний алгоритм роботи програми, зображений на рис. 2.

Докладніше про зображені кроки алгоритму:

Медіанна фільтрація відбувається таким чином: зображення розбивається на три двовимірних масиви, кожен з яких відповідає одному з компонентів моделі RGB. Значення кожного елемента замінюється медіанної з околиці пікселя. З трьох компонентів збирається кінцеве зображення.

Для сегментації використовується бінаризація з автоматично знайденим порогом за методом Оцу. Так само поріг для бінаризації може вибиратися вручну оператором.

Клітинний автомат працює наступним чином:

- Нумерація всіх чорних крапок послідовними натуральними числами, починаючи з одиниці. Білі пікселі позначаємо нулем.
- Заміна в циклі значення номера кожного пікселя на максимальний з сусідніх. Після того, як зображення стало інваріантним щодо цього перетворення, отримуємо замкнуті області, всередині кожної з яких всі пікселі мають однаковий номер.

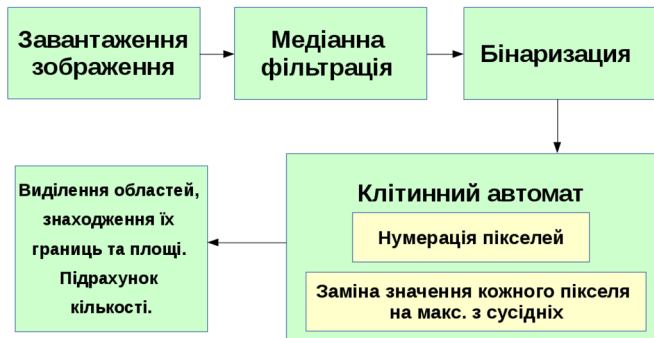


Рисунок 2. Алгоритм роботи створеного програмного продукту

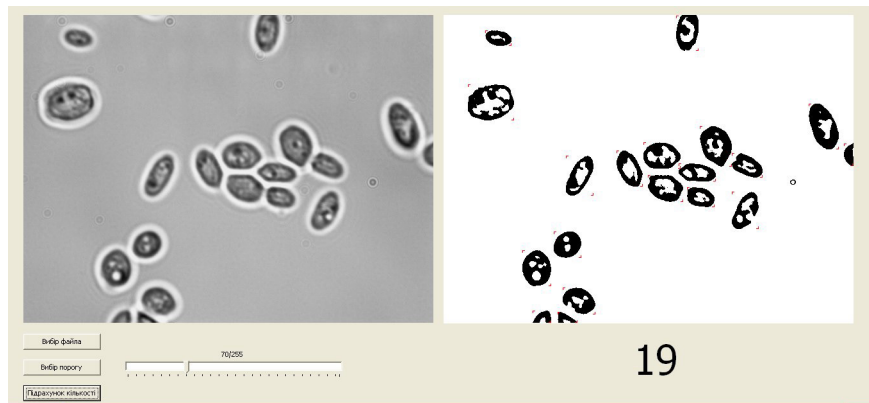


Рисунок 3. Приклад роботи створеної програми

Таблиця 1

Результати роботи програми для різного типу зображень

| Тип зображення | Значення порогу | | Кількість клітин, порахована автоматично, Ка | Кількість клітин, порахована вручну, Кв | Відхилення від результату, отриманого вручну |
|----------------|---|--|--|---|--|
| | Автоматично знайдене / підібране вручну | | | | |
| а) | 111 / 100 | | 112 / 134 | 146 | 0.08 / 0.023 |
| б) | 58 / 45 | | 208 / 198 | 202 | 0.02 / 0.02 |
| в) | 31 / 27 | | 94 / 94 | 98 | 0.04 / 0.04 |
| г) | 23 / 53 | | 35 / 50 | 48 | 0.27 / 0.04 |

- Виділяємо області, знаходимо їх границі та площу. Підраховуємо їх кількість.

Створену програму можна застосувати не тільки для обробки кількісних аналізів крові. Наприклад, при дослідженнях проб води та землі дуже часто необхідно порахувати кількість бактерій. Приклад роботи створеної програми приведено на рис. 3:

Як можна побачити з таблиці 1, результат роботи створеного програмного продукту напряму залежить від зображення, що аналізується. Також, потрібно зазначити, що для кожного різновиду зображень необхідно емпірично підбирати значення площі клітини, щоб поділяти клітини, що склеїлися. На рис. 1 б) присутні великі включення, що порахуються як декілька склеєних клітин.

Висновки

Отже, в роботі був реалізований алгоритм обробки зображень з використанням клітинних автоматів і проведено його тестування. Для подальшого розвитку необхідно вдосконалити алгоритм роботи програми, реалізувавши наступні ідеї:

- збільшити кількість можливих варіантів цитологічних зображень;
- реалізація алгоритму, який буде не тільки підраховувати кількість клітин, але і класифікувати і визначати відсоток клітин, які захворіли.

Тоді запропонований алгоритм можна буде використовувати в якості елемента додатки, що дозволить лаборанту отримати результати аналізу відразу ж після фотографування цитологічного препарату і зменшити ймовірність медичних помилок.

Література

1. Хрящев Д. О. Попередня обробка та аналіз зображень, отриманих в умовах недостатнього освітлення. автореф. дис. на здобуття наук. ступеня канд. техн. наук: спец. 05.13.01 «Системний аналіз і управління» / Д. О. Хрящев. — Астрахань, 2013. — 145 с.
2. Фон Нейман Дж. Теория самовоспроизводящихся автоматов / Дж. Фон Нейман, А. Бёркс — М.: Мир, 1971. — 382 с.
3. М. С. Тарков Оценивание числа клеток на изображениях цитологических растительных препаратах. / Тарков М. С., Осипов М. И. // Известия Томского политехнического университета. — 2007. — № 5.
4. Radicular cysts and odontogenic keratocysts epithelia classification using cascaded — Режим доступу: <http://breckon.eu/toy/publications/papers/han08cell>. — Дата доступу: 17.05.2016
5. Ковригин А. В. Применение принципов построения систем машинного зрения в задаче анализа изображений клеточных структур / А. В. Ковригин // Научный журнал КубГАУ. — 2007. — № 29.

Светлова Елена Костянтинівна

студентка

Національний технічний університет України «Київський політехнічний інститут»

Светлова Елена Константиновна

студентка

Национальный технический университет Украины «Киевский политехнический институт»

Svietlova O. K.

student

National Technical University of Ukraine «Kyiv Polytechnic Institute»

**ГЕНЕРАТОР ПСЕВДОВИПАДКОВИХ ЧИСЕЛ
ДЛЯ МОДЕЛЮВАННЯ ДОСЛІДУ ШТЕРНА
ГЕНЕРАТОР ПСЕВДОСЛУЧАЙНЫХ ЧИСЕЛ
ДЛЯ МОДЕЛИРОВАНИЯ ОПЫТА ШТЕРНА
THE PSEUDORANDOM NUMBER GENERATOR
FOR SIMULATION STERN EXPERIMENT**

Анотація. Проведено чисельний експеримент з моделювання досліду Штерна з використанням MathCAD.

Ключові слова: досвід Штерна, броунівський рух, псевдовипадковий генератор, моделювання, MathCAD.

Аннотация. Проведено численный эксперимент, моделирующий опыт Штерна с использованием MathCAD.

Ключевые слова: опыт Штерна, броуновское движение, псевдослучайный генератор, моделирование, MathCAD.

Summary. A numerical experiment, simulating Stern experiment was carried out using MathCAD.

Keywords: Stern experiment, Brownian motion, the pseudo-random generator, modeling, MathCAD.

Идея данного исследования появилась в момент решения совсем другой задачи, связанной с теорией погрешностей. Задача заключалась в том, чтобы оценить площадь поверхности шарика, зная, что измерения случайной величины — радиуса шарика — дали результат $R_0 \pm \Delta R$. В результате моделирования этой задачи в Mathcad с помощью встроенных генераторов псевдослучайных чисел было обнаружено, что *характерные значения* этих распределений (среднее, наиболее вероятное) *не связаны* соотношением $S = 4\pi R^2$.

Речь идёт о распределении молекул газа по скоростям (распределение Максвелла). Вывод закона этого распределения выходит за рамки школьной программы, однако в учебниках приводится описание экспериментальной проверки этого закона (опыт Штерна). Наше внимание привлекло то, что распределение осевших на внешнем цилиндре молекул по *координате* слишком напоминает распределение молекул по *скоростям* [1, с. 14–15].

Таким образом, целью нашего исследования стало моделирование опыта Штерна в среде Mathcad и сравнение его результатов с распределением Максвелла.

**Анализ существующих программ,
иллюстрирующих опыт Штерна**

Прежде чем начать моделировать Опыт Штерна с помощью компьютера, мы решили изучить уже существующие программы по этой теме. Рассмотрим наиболее популярные из них:

На сайте files.school-collection.edu.ru есть интересная анимированная модель опыта Штерна [2]. Общий вид страницы представлен на рис. 1.

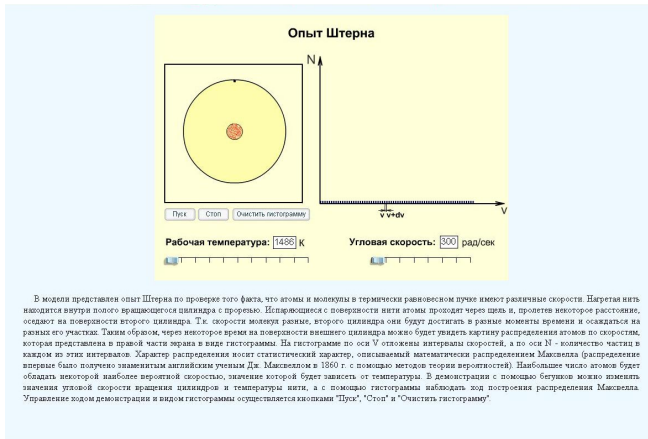


Рисунок 1. Общий вид страницы, иллюстрирующей опыт Штерна

Однако, после прочтения сопутствующего текста возникают сомнения в том, что гистограмма в правой части каким-либо образом отражает профиль молекул, осевших на поверхности цилиндра. Складывается впечатление, что она просто-напросто строится с помощью генератора случайных чисел по Максвеллу.

Ещё одну анимированную модель мы встретили на ресурсе [3]. Её внешний вид представлен на рис. 2.

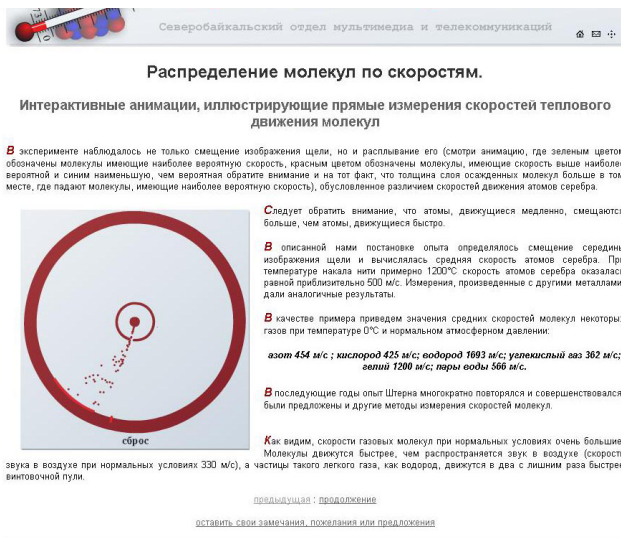


Рисунок 2. Общий вид страницы, иллюстрирующей опыт Штерна

На ней хорошо показано различие скоростей молекул, место, где они осядут в конце, но непонятно как именно будет выглядеть образовавшийся слой молекул. И уж совсем странным выглядит словосочетание «*прямые измерения* скоростей теплового движения молекул» в заголовке, потому как никакого прибора, непосредственно регистрирующего скорость молекул в опыте Штерна нет.

Всё это подтолкнуло нас к более детальному изучению теоретических основ опыта Штерна.

Моделирование опыта Штерна

Для проведения численного эксперимента были выбраны такие значения параметров, которые указаны для опыта Штерна. Масса атома серебра, температура спирали (а соответственно и вылетающих молекул), и постоянная Больцмана указаны в СИ:

$$m_{Ag} := 1.794 \cdot 10^{-25}$$

$$T_{coil} := 1500$$

$$k_{Bolman} := 1.38 \cdot 10^{-23}$$

Вначале было сгенерировано $N = 5000$ случайных чисел, которые симулируют значения модуля скорости отдельной молекулы после вылета из щели (см. пункт 2.1). Эти значения были получены с помощью метода обратного преобразования, который был реализован следующим образом:

$$Number := 10000$$

$$i := 0.. Number - 1$$

$$Velocity := \begin{cases} v \leftarrow 500 \\ \text{for } i \in 0.. Number - 1 \\ \quad r_i \leftarrow |runif(1, 0, 1)| \\ \quad M_i \leftarrow \text{root}(F_AfterSlit(v) - r_i, v) \\ \quad M \end{cases}$$

Здесь r_i — случайные числа, равномерно распределенные на отрезке $[0;1]$, функция $root$ численно решает уравнение $F_AfterSlit(v) - r_i = 0$. В результате получаем вектор-столбец со значениями скорости.

Для проверки работы генератора по полученным значениям была построена гистограмма (рис. 3). На рис. 3 также изображен маркер, соответствующий наиболее вероятной скорости для распределения Максвелла. Видно, что она не соответствует наиболее вероятной скорости вылетевших молекул.

Для перехода к координатам осевших молекул возьмем радиус внешнего цилиндра и угловую скорость системы цилиндров как в опыте Штерна (величины заданы в СИ):

$$Radius := 0.015$$

$$\omega := 2500$$

По уже сгенерированным значениям скоростей пересчитываем значения координаты места, в котором будет оседать та или иная молекула, и по полученным значениям координаты строим гистограмму (рис. 4). Сплошной линией изображена теоретически рассчитанная кривая $p(l)$.

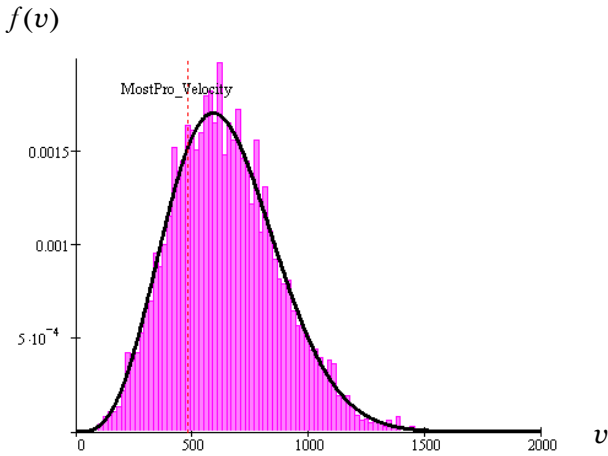


Рисунок 3. Гистограмма для скорости вылетевших молекул

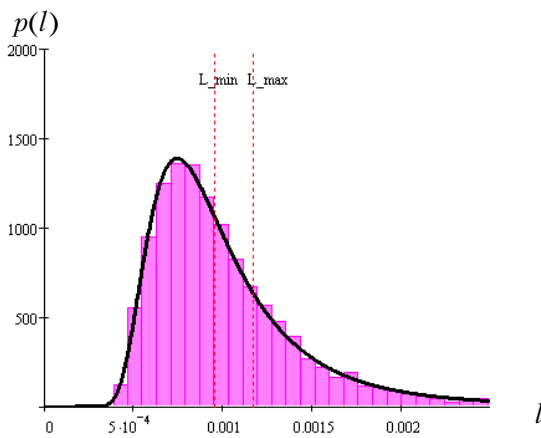


Рисунок 4. Гистограмма для координаты осевших молекул

Перспективы развития

По итогам проведенного исследования можно определить дальнейшие перспективные направления его развития.

Во-первых, можно продолжить исследование в сторону создания *динамического симулятора* опыта Штерна. Хотелось бы увидеть развернутую во времени согласованную динамику изменений гистограммы оседания молекул на поверхности внешнего цилиндра и гистограммы распределения вылетающих молекул по скоростям.

Во-вторых, можно развивать дальше рассмотренную идею создания *генератора псевдослучайных чисел с наперед заданным законом распределения* (метод обратного преобразования). Такой генератор будет полезен при изучении и демонстрации законов теории вероятностей и математической статистики.

Выводы

Проведенное исследование позволяет сделать следующие выводы:

1. На конкретном примере было показано «смещение» наиболее вероятного значения при переходе от распределения одной случайной величины к другой (если эти величины связаны функциональной зависимостью).

2. Были созданы генераторы псевдослучайных чисел с наперед заданными распределениями:

а) распределение Максвелла

$$f(v) = 4\pi v^2 \left(\frac{m}{2\pi kT} \right)^{3/2} \exp\left(-\frac{mv^2}{2kT} \right),$$

б) распределение молекул по энергиям

$$g(E) = \frac{2\pi \cdot \sqrt{E}}{(\pi kT)^{3/2}} \cdot e^{-\frac{E}{kT}},$$

в) распределение молекул после вылета из щели

$$f(v) = \frac{1}{2} \left(\frac{m}{kT} \right)^2 v^3 \cdot \exp\left(-\frac{mv^2}{2kT} \right),$$

г) распределение осевших молекул по координате (в опыте Штерна)

$$p(l) = \frac{\omega R^2}{2} \cdot \left(\frac{m}{kT} \right)^2 \cdot \frac{\exp\left(-\frac{m\omega^2 R^4}{2kT l^2} \right)}{l^5}.$$

3. Был проведен численный эксперимент, моделирующий опыт Штерна на компьютере и показано, что наиболее вероятное значение координаты не соответствует ни одной из характерных скоростей распределения Максвелла, если пользоваться формулой связи для отдельно взятой молекулы.

Литература

1. Вентцель Е. С. Теория вероятностей: Учебник для студентов Вузов / Е. С. Вентцель. — М.: Мир, 1969. — 576 с.
2. Молекулярная физика — Режим доступа: <http://files.school-collection.edu.ru/dlrstore/d4204a73-7eb4-4f97-91c3-ac19453f933a/molek1.htm> — Дата доступа: 07.05.2016.
3. Распределение молекул по скоростям — Режим доступа: http://somit.ru/mkt/skorosti2_0.htm — Дата доступа: 05.05.2016.

Севідов Павло Миколайович

студент

Національний технічний університет України «Київський політехнічний інститут»

Севидов Павел Николаевич

студент

Национальный технический университет Украины «Киевский политехнический институт»

Sevidov P.

student

National Technical University of Ukraine «Kyiv Polytechnic Institute»

**ДОСЛІДЖЕННЯ АЛГОРИТМІВ ОПТИМІЗАЦІЇ ІНДИВІДУАЛЬНИХ
КАЛЕНДАРНИХ ПЛАНІВ НАВЧАЛЬНОГО ПРОЦЕСУ**

**ИССЛЕДОВАНИЕ АЛГОРИТМОВ ОПТИМИЗАЦИИ ИНДИВИДУАЛЬНЫХ
КАЛЕНДАРНЫХ ПЛАНОВ УЧЕБНОГО ПРОЦЕССА**

ALGORITHMS FOR INDIVIDUAL TRAINING SCHEDULE OPTIMIZATION

Анотація. У даній роботі ставиться завдання розглянути алгоритми оптимізації індивідуальних планів-графіків навчання, для вирішення проблеми управління часу студента. Було досліджено оптимальність використання генетичних алгоритмів для вирішення задачі багатокритеріальної оптимізації календарного плану навчального процесу.

Ключові слова: алгоритми оптимізації, генетичний алгоритм, план-графік, фітнес функція, алгоритм імітації вогпалу, жадібний алгоритм.

Аннотация. В данной работе ставится задача рассмотреть алгоритмы оптимизации индивидуальных план-графиков обучения, для решения проблемы управлением времени студента. Было исследовано оптимальность использования генетических алгоритмов для решения задачи многокритериальной оптимизации календарного плана учебного процесса.

Ключевые слова: алгоритмы оптимизации, генетический алгоритм, план-график, фитнес функция, алгоритм имитации отжига, жадный алгоритм.

Summary. In this paper was considered the task of optimization algorithms of individual training schedules to solve the problem of student time management. Investigated the optimal use of genetic algorithms for solving the problem of multi-criteria optimization schedule of the educational process.

Key Words: optimization algorithms, genetic algorithm, a schedule, a fitness function, simulated annealing, a greedy algorithm.

Люди протягом усього свого життя стикаються з задачею оптимального планування свого часу. Кожен з щодня займається плануванням власного часу. Майже всі люди керуються при цьому «схожим алгоритмами», намагаючись виконати всі важливі справи та вчасно, виділивши при цьому максимальний час на дозвілля та відпочинок.

Різноманітність в навчальному плані предметів і видів навантаження призводить до неоптимального використання студентом власного часу. Наслідком даної проблеми є так званий «синдром студента». Це призводить до втрати часу, яке виділяється студентом

при оцінці трудомісткості і ризиків навчальних робіт, і до збільшення рівня стресу в кінці семестру. Рішення даної проблеми є актуальним і дозволить зменшити навантаження на студентів і викладачів. Найвні рішення не враховують індивідуальний розклад студента і рівень його щоденної завантаженості власними справами при формуванні графіка виконання навчального плану.

Даний тип задач можна віднести до задач з теорії розкладів. Суть якого полягає в створенні певного порядку дій та оптимальному розподілі ресурсів (вільний час студента), що необхідні для досягнення

поставленої мети (успішне та своєчасне виконання студентом усіх робіт, що передбачені навчальним процесом з врахуванням власного графіку).

Критерії до задачі

Складання індивідуального розкладу для студента передбачає ряд критеріїв, з яких виділено два основних типи: *hard stop* (критерії, що заборонено порушувати для даної задачі) і *soft stop* (критерії, що не бажано порушувати, порушення призводить до погіршення результату).

До *Hard Stop* критеріїв віднесено:

1. Кожна робота повинна бути закінчена і здана до крайнього строку (*deadline*).
2. Виконану роботу можна здати за наявності запиття з викладачем, який приймає роботу.
3. Початок виконання роботи можливо за умови, якщо немає інших справ в цей час.

Перелік *Soft Stop* критеріїв:

1. Показник завантаженості не повинен перевищувати допустиму норму.
2. Задачі з високим пріоритетом, виконуються якомога раніше.
3. Рівномірність завантаженості кожного робочого дня.
4. Сума дат завершення робіт є мінімальною.

Вибір даного набору критеріїв є критичним для поставленої задачі. Наприклад, порушення крайнього строку суперечить меті – виконання всіх робіт вчасно, використовуючи мінімальну кількість ресурсів. Перелік *soft stop* критеріїв представлений в порядку ваги критичності порушення. Тобто порушення норми завантаженості, призводить роботи поза нормований робочий день і зменшенню продуктивності. Мінімальна сума дат завершення робіт – показник знаходження оптимального рішення. Результатом може бути знайдене гарне рішення, що не суперечить всім критеріям, крім останнього. Тобто порушення даного критерію не є критичним.

Пошук рішення поставленої задачі можливо шукати на базі таких існуючих методів: евристичних, стохастичних, детермінованих. Даний тип завдань є NP складною і не існує точної математичної моделі для знаходження оптимального рішення. Тому доцільно використання стохастичних і евристичних методів. Для дослідження розглядалися алгоритми: імітації відпалу, як представник стохастичного методу, генетичний алгоритм та жадібний алгоритм – евристичного.

Алгоритми оптимізації

Жадібний алгоритм. Простий і прямолінійний евристичний алгоритм, який приймає найкраще рішення, виходячи з наявних на поточному етапі даних,

не турбуючись про можливі наслідки, сподіваючись врешті-решт отримати оптимальне рішення. Легкий в реалізації і часто дуже ефективний за часом виконання. Мінус – багато задач не можуть бути розв’язані з його допомогою.

Алгоритм імітації відпалу. Алгоритм ґрунтується на імітації фізичного процесу, який відбувається при кристалізації речовини, в тому числі при відпалі металів. Передбачається, що атоми вже вишикувалися в кристалічну решітку, але ще допустимі переходи окремих атомів з одного осередку в іншу. Передбачається, що процес протікає при поступово знижується температурі. Перехід атома з одного осередку в іншу відбувається з певною ймовірністю, причому ймовірність зменшується з пониженням температури. Стійка кристалічна решітка відповідає мінімуму енергії атомів, тому атом або переходить в стан з меншим рівнем енергії, або залишається на місці.

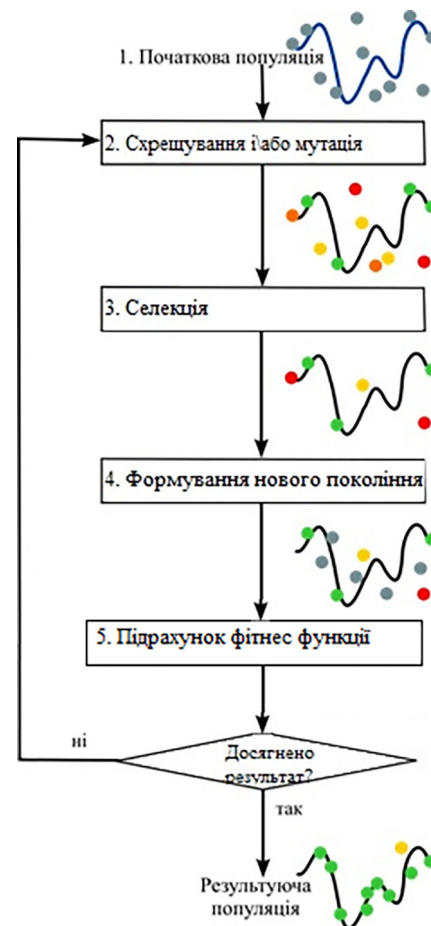


Рис. 1. Блок-схема роботи генетичного алгоритму

Генетичний алгоритм. Завдання формалізується таким чином, щоб її рішення могло бути закодовано у вигляді вектора («генотипу») генів, де кожен ген може бути бітом, числом або якимось іншим об’єктом. У класичних реалізаціях генетичного алгоритму (ГА) передбачається, що генотип має фіксовану

довжину. Однак існують варіації ГА, вільні від цього обмеження.

Деяким, звичайно випадковим, чином створюється безліч генотипів початкової популяції. Вони оцінюються з використанням «функції пристосованості», в результаті чого з кожним генотипом асоціюється певне значення («пристосованість»), яке визначає наскільки добре фенотип, їм описуваний, вирішує поставлене завдання.

Для поставленої проблеми, рішення за допомогою ГА критичним є вибір методів мутації і кросинговеру, також налаштування значень двох змінних ймовірності мутації і розміру популяції. Провівши дослідження було обрано:

- **Впорядкований кросинговер (OX1):** частина одного з батьків береться за основу нащадку. Далі в отриманий нащадок недостаючи гени беруться з другого батька зберігаючи порядок та пропускаючи ті, що вже є в частині від першого батька. Таким чином не порушуються жодне з обмежень.
- **Метод мутації – «обмін»**, що має найкращі показники простоти реалізації і швидкодії, суттю якого є зміна в послідовності двох сусідів одного випадково обраного гена.

Далі на основі тестових даних, для яких взято учбовий план зі всіма п'ятого семестру, зроблено моделювання індивідуальні план-графіки. Провівши аналіз індивідуальних розкладів з різними значеннями

параметрів розміру популяції і ймовірності мутацій, отримано такі результати.

Популяція складається з певної кількості різних особин. При збільшенні популяції, збільшується варіативність особин, що в свою чергу збільшує шанси досягти оптимального результату. Мінусом збільшення популяції є уповільнення роботи алгоритму, тому потрібно знайти компромісне рішення. Як бачимо з рисунку 2 таким значенням є розмір популяції в межах від 50 до 70.

При збільшенні ймовірності, збільшується «випадковість» генерації значень нового покоління, що в свою чергу унеможлиблює досягти оптимального значення. Результатом даних досліджень є вибір значення ймовірності мутації в межах від 0.1 до 0.25.

Результуючі результати значення функції оптимізації по відношенні до початкового приближення склали для досліджених алгоритмів:

1. Генетичний алгоритм – 26%.
2. Алгоритм імітації відпаду – 29%.
3. Жадібний алгоритм – 17%.

Висновки

Результатом даної роботи є аналіз особливостей та ефективності використання генетичного алгоритму для вирішення задачі створення та оптимізації календарного плану навчального процесу. Продуктом даного аналізу є порівняльна характеристика показників

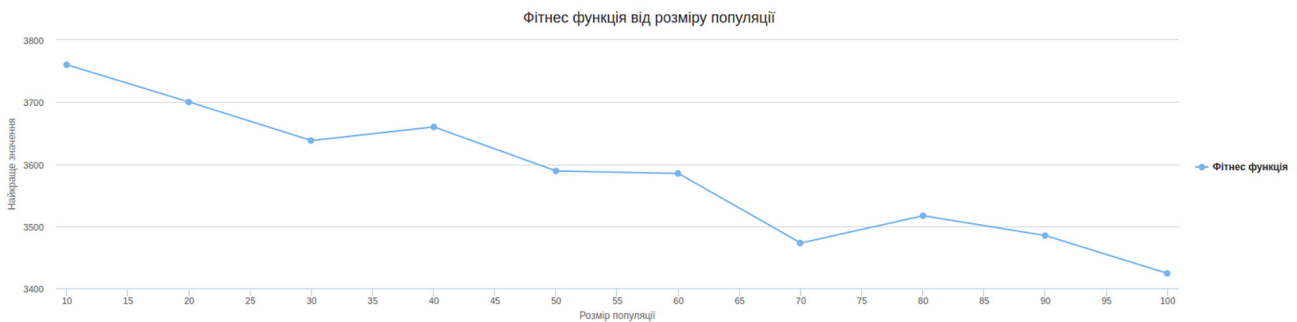


Рис. 2. Залежність найкращого значення фітнес функції від розміру популяції

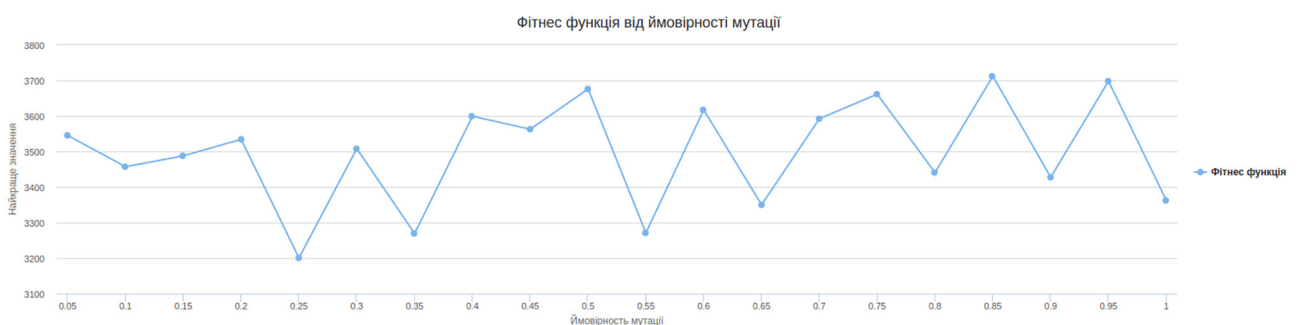


Рис. 3. Залежність найкращого значення фітнес функції з популяції до ймовірності мутації



Рис. 4. Порівняння значень функцій оптимізації

роботи та результатів оптимізації розкладу на один навчальний семестр.

Варіації етапів генетичного алгоритму дали можливість підібрати варіант, що не порушує наявних критеріїв. Було обрано впорядкований кросингвер (OX1). В якому частина одного з батьків береться за основу нащадку. Далі в отриманий нащадок недостаючи гени беруться з другого батька зберігаючи порядок та пропускаючи ті, що вже є в частині від першого батька. Таким чином не порушуються жодне з наявних критеріїв. Найкращими показниками простоти реалізації і швидкодії відповідає метод мутації **обміну**, що змінює в послідовності двох сусідів одного випадково обраного гена.

Для генетичного алгоритму було відображено залежність параметрів від найкращого значення функції. При збільшенні розміру популяції, зростає ймовірність досягнути оптимального рішення і досягається краще значення фітнес функції. Генетичний алгоритм має явну перевагу, використовуючи його створення

індивідуального план-графіку на базі для веб-додатку тому, що отримання «гарного» рішення досягнуто приблизно на 200 ітерації роботи і витрачається мала кількість часу.

Недоліки алгоритму: час виконання функції оптимізації велике, необхідність знайти всі рішення задачі, а не одне з них, багато тимчасових даних, недоведеність збіжності.

Також було наведено порівняльну характеристику з алгоритмом імітації відпалу. Фінальна оптимізація генетичного алгоритму менша на 3%. Можливо досягнути кращу мінімізацію збільшуючи розмір популяції, але постає інша проблема — задіяння неймовірної кількості пам'яті. Це призводить до частих запусків збирача сміття, в результаті чого «просідання» в швидкості видачі готового результату кінцевому користувачеві.

Надалі можливо поліпшити результати, використовуючи гібридні алгоритми, комбінуючи переваги розглянутих алгоритмів.

Література

1. Лазарев А. А. Решение NP-трудной задачи теории расписаний минимизации суммарного запаздывания // Журнал Вычислительной математики и математической физики — 2007. том 47, N.6. — С. 1087–1098.
2. Лазарев А. А., Гафаров Е. Р. Теория расписаний. Минимизация суммарного запаздывания для одного прибора. // Научное издание, М.: Вычислительный центр им. А. А. Дороницына РАН, 2006. — 134 с.
3. Alon N., Woeginger G.J., Yadid T. Approximation schemes for scheduling on parallel machines // J. of Scheduling. — 1998. — V. 1. — P. 55–66.
4. Brucker P., Knust S. Complex scheduling Springer-Verlag Berlin, Heidelberg, Germany, 2006.
5. Nouredin Sadawi [Електронний ресурс]. — Режим доступу: <https://www.youtube.com/channel/UCNYv4HA3WjV3gZGLfBehRWQ>. — Дата доступу: 27.05.2016.

Севідов Павло Миколайович

студент

Національний технічний університет України «Київський політехнічний інститут»

Севидов Павел Николаевич

студент

Национальный технический университет Украины «Киевский политехнический институт»

Sevidov P.

student

National Technical University of Ukraine "Kyiv Polytechnic Institute"

ОПТИМІЗАЦІЯ ЗАПИТІВ ЗА ДОПОМОГОЮ ГЕНЕТИЧНОГО АЛГОРИТМУ ОПТИМИЗАЦИЯ ЗАПРОСОВ С ПОМОЩЬЮ ГЕНЕТИЧЕСКОГО АЛГОРИТМА QUERY OPTIMIZATION BY GENETIC ALGORITHM

Анотація. Генетичні алгоритми являють собою потужний метод пошуку, заснований на механіці природного відбору і природного генетики, які успішно використовуються для вирішення завдань в самих різних дисциплінах.

У цій статті представляється використання генетичних алгоритмів в одній з найважливіших завдань оптимізації в області комп'ютерної науки, оптимізація запитів бази даних для великого join запиту.

Сучасні методи оптимізації запитів є недостатніми для підтримки, деяких формується додатків, що використовують бази даних. У цій статті описано проблему оптимізації запитів до бази даних і описано адаптацію генетичного алгоритму. І порівняння між простими SQL запитами, що мають п'ять join і той же запит з використанням генетичного підходу. А також дано основний огляд Carquinyoli Genetic Optimizer, заснованого на генетичному програмуванні.

Ключові слова: Генетичні алгоритми, оптимізація запитів.

Аннотация. Генетические алгоритмы представляют собой мощный метод поиска, основанный на механике естественного отбора и природной генетики, которые успешно используются для решения задач в самых разных дисциплинах.

В этой статье представляется использование генетических алгоритмов в одной из самых важных задач оптимизации в области компьютерной науки, оптимизация запросов базы данных для большого join запроса.

Современные методы оптимизации запросов являются недостаточными для поддержки, некоторых формирующейся приложений, использующих базы данных. В этой статье описано проблему оптимизации запросов к базе данных и описано адаптацию генетического алгоритма. И сравнение между простыми SQL запросами, имеющих пять join и тот же запрос с использованием генетического подхода. А также дано основной обзор Carquinyoli Genetic Optimizer, основанного на генетическом программировании.

Ключевые слова: Генетические алгоритмы, оптимизация запросов.

Abstract. Genetic Algorithms are a powerful search technique based on the mechanics of natural selection and natural genetics that are used successfully to solve problems in many different disciplines.

In this paper we present genetic algorithms in one of the most important optimization problems in computer science, database query optimization for large join query.

Current query optimization techniques are inadequate to support some of the emerging database application. In this paper, we outline a database query optimization problem and describe the adaptation of genetic algorithm. And comparison between simple SQL queries having five join and same query using genetic approach. And also give basic overview of the Carquinyoli Genetic Optimizer based on Genetic Programming.

Keywords: Genetic Algorithms, Query Optimization.

Введение

Генетический алгоритм становится широко используемым и принятым методом для решений очень сложных задач оптимизации. Он был использован для решения широкого спектра проблем, таких как оптимизации, интеллектуальный анализ данных, игры, эволюционировали поведение в биологических сообществах и т.д.

Пользователи действительно нужно использовать очень большие запросы на соединение (join queries), чтобы поддержать их в своих бизнес-решениях [1]. Кроме того, сложность этих запросов будет увеличиваться, если СУБД может справиться с ними легко. С помощью этих общих сценариев, в настоящее время коммерческие СУБД становятся не в состоянии поддерживать удовлетворительные результаты, сохраняющие минимальные требования к производительности [2]. А именно, динамические методы программирования, примененные к оптимизации запросов, накладывают ограничения на время и память. Так как пространство поиска растет экспоненциально с линейным увеличением числа соотношений, участвующих в запросе, эти алгоритмы должны сохранить экспоненциально большее число частичных планов в памяти. Этот процесс отнимает очень много времени и обычно заканчивается без решения, когда у оптимизатора заканчивается память. Различные подходы были предложены для исправления этой ситуации [3, 5, 6].

Эта статья организована следующим образом. Раздел 2 содержит вводный материал, обеспечивающий некоторые общие принципы работы генетического алгоритма. Раздел 3 посвящен разнообразным алгоритмам оптимизации для оптимизации проблемы большого запроса на соединение, со сравнением с точки зрения оценки работы времени между простым SQL-запросом, имеющего пять запросов на соединение и тот же запрос с использованием генетического подхода. И раздел 4 посвящен основному обзор Carquinyoli Genetic Optimizer на основе генетического программирования.

2. Генетическое программирование в оптимизации запросов

В Генетическое программирование (GP) Основная идея заключается в том, чтобы получить лучшее решение с помощью эволюционных методов [3, 4].

Основное поведение этого типа алгоритмов заключается в следующем.

Первоначальный набор программ создается с нуля. В данной работе мы представляем их в виде древовидных структур, так как они являются наиболее приемлемым подходом, учитывая, что план выполнения за-

проса (QEP) в СУБД, как правило представлен в виде tree-shaped структуры. Этот набор также называется начальной популяцией.

После того, как начальная популяция создана, мы итерационно применяем набор генетических преобразований на членов популяции. Операторами первичного преобразования являются **Кроссинговер** и **Мутация**.

Первый работает путем изменения двух (или более) программ (или древовидных структур) и объединения их каким-либо образом; последний путем изменения одной древовидной структуры.

Каждая итерация алгоритма называется **Поколением**. В конце каждого поколения, третья генетическая операция, называемая **Отбор** применяется для того, чтобы устранить худших членов в популяции. После применения этих операций алгоритм получает следующее поколение членов.

Условие остановки гарантирует, что алгоритм завершается. После того, как критерий остановки удовлетворяется, мы берем лучшее решение от конечной популяции. Одним из типичных применений этого типа алгоритма для решения задач оптимального поиска пути. В этих проблемах, каждый член в популяции представляет собой путь для достижения конкретной цели и имеет соответствующую стоимость.

Оптимизация запросов может быть сведена к задаче поиска, где СУБД должна найти оптимальный QEP в обширном пространстве поиска. Каждый план выполнения может рассматриваться в качестве возможной программы для решения проблемы поиска хорошего пути доступа для получения необходимых данных. Таким образом, в генетическом оптимизаторе, каждый член популяции является действительным планом выполнения запроса. Интуитивно понятно, что по мере развития населения, средняя стоимость плана членов уменьшается [3, 5].

3. Проблема большого запроса на соединение

Значительный рост объема данных, необходимых для того, чтобы принимать правильные решения, делает текущие оптимизаторы запросов неадекватные в некоторых ситуациях. Компании, предоставляющие банковские услуги являются хорошим примером типичного клиента, который нуждается в очень большой емкости для хранения очень больших и сложных конструкций баз данных. В таких пользователей действительно нужно использовать очень большие запросы на соединение (join queries), чтобы поддержать их в своих бизнес-решениях. Различные подходы были предложены для исправления этой ситуации [3, 5, 6].

Эвристические алгоритмы: пространство поиска сокращается с использованием оценок. Они, как

правило, очень быстрые, но редко находят оптимальное решение.

Случайные алгоритмы: случайная прогулка по пространству поиска выполняется для того, чтобы найти почти оптимальное решение. Различные политики приводят к различным алгоритмам, а именно итеративного усовершенствования, имитации отжига, гибридных алгоритмов и т.д.

Генетические алгоритмы: вдохновленные в естественном отборе, генетические алгоритмы пытаются найти оптимальное среди популяции. Эта популяция страдает от постоянных преобразований, осуществляется с использованием трех основных видов деятельности: отбор, сочетание и мутации.

Генетическое программирование и оптимизация запросов

Оптимизация запросов может быть сведена к задаче поиска, где СУБД должна найти оптимальный план выполнения запроса (QEP) в обширном пространстве поиска. Каждый QEP можно рассматривать в качестве возможного решения (или программы) для задачи нахождения хорошего пути доступа для извлечения данных, необходимых в запросе. Таким образом, в генетическом оптимизаторе запросов, каждый член популяции является QEP. Оптимизация запросов с участием большого количества объединений с использованием генетических алгоритмов был введен Беннетом Эль-Аль и испытано позже Steinbrunn и др. Показали, что это очень конкурентный подход [5, 7].

Генетический алгоритм и SQL запрос на соединение с 5 join

Имеется реляционные таблицы со следующими таблицами: Person, Orders, Product, Department, и Quantity с помощью подходящего набора данных инициализации. Ниже приводится простой запрос с соединением для извлечения из него данные на заданных определенных условиях:

```
select
  p.LastName, p.FirstName, o.OrderNo, o.p_id,
  pd.productname, pd.prd_id, s.sales_id,
  dept.departmentname, qty.quantity
from Person p
inner join Orders o on p.p_id = o.p_id inner join
Product pd on o.prd_id = pd.prd_id inner join
Sales s on s.prd_id = pd.prd_id, inner join
Department dpt on s.sales_id = dpt.sales_id inner join
Quantity qty on dpt.dept_id = qty.dept_id
and qty.prd_id = qty.prd_id order by p.LastName
group by dpt.dept_id;
```

Запустив простое соединение запроса в SQL Server 2012, получаем график, показывающий время, необходимое для выполнения запроса:

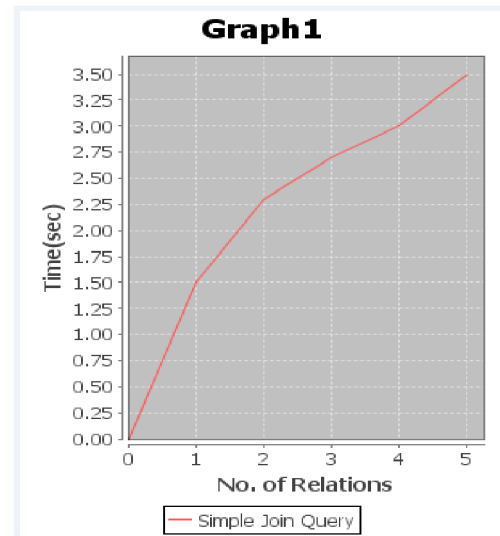


Рисунок 1. Зависимость количества join до времени выполнения запроса

Применение генетического алгоритма

Теперь применим генетический алгоритм на том же запросе с использованием 5 join.

Процедура:

Здесь мы рассмотрим приведенный выше запрос с N отношений, где (N = 5)

1. Принцип работы генетического алгоритма является создание популяции (решение пространства).

2. После создания населения установить No. Поколения и No. Потомка, который создает основу для населения.

3. Затем, взяв цикл для всех отношений в запросе создаем пространство решений т.е. популяцию путем случайного выбора соотношения, используя rand(). Здесь для его отношения должны быть соединены случайным образом выбирается другое соотношение учитывая Left Deep дерево [8].

4. После того, что в соответствии с принципом ГА выбор родителей из популяции и подсчет их хромосом.

5. Вычислить фитнес значение каждой хромосомы и выбрать наиболее пригодную хромосому

6. Для расчета в фитнес-функцию следует рассматривать как $P(x) = x^2$.

7. После того, как фитнес значения рассчитываются для хромосом эти хромосомы скрещиваются друг с другом.

8. После завершения кроссинговера происходит мутация.

Все вышеперечисленные шаги реализуются в SQL хранимой процедуре.

Заметка. Для следующей процедуры создайте таблица 'results', чтобы сохранить результат вычисления ГА, на котором выполняется QEP и вычисляется граф. Создать процедуру GeneticAlgoWithJoins.

После выполнения вышеописанной процедуры в Sql Server 2010 время, необходимое для QEP для вышеуказанного запроса SQL с 5 операциями на присоединение показано на следующем графике:

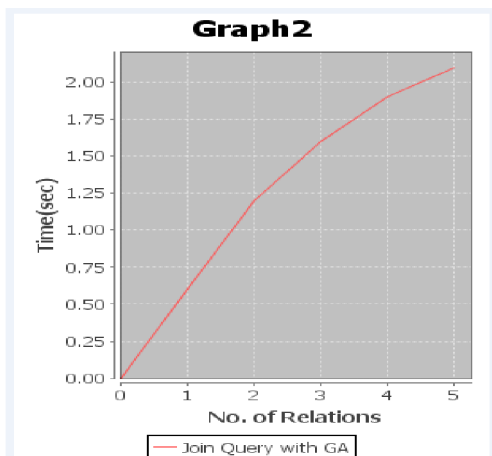


Рисунок 2. Зависимость количества join до времени выполнения запроса после применения ГА

Далее представлен график сравнения времени, необходимого для выполнения простого запроса до и после применения ГА.

На приведенном выше графике показано применение GA.

1. Фитнес значения показывают здоровье хромосом, отображенных в каждом Поколении.

2. График показывает значение фитнес функции как 1,99 за 100 итераций.

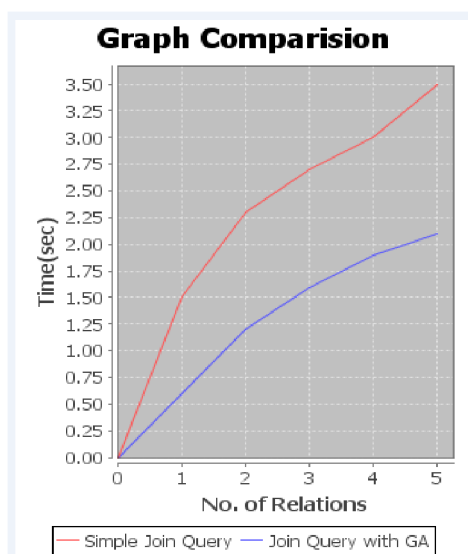
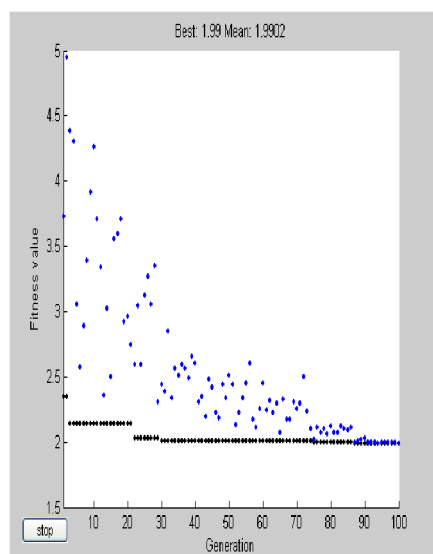


Рисунок 3 и 4. Применение ГА на данный запрос и сравнение с результатами без использования

3. В графике выше точки в нижней части (черные цветные точки) обозначают наилучшие значения пригодности, а точки над ними (синего цвета) обозначают средние фитнес значения в каждом поколении. Также показывает лучшие и средние значения в текущем поколении численно на самом верху.

4. На приведенном выше графике показано наилучшую пригодность в каждом поколении, показывает незначительный прогресс в снижении значения пригодности и среднее расстояние между отдельными генами в каждом поколении, что является хорошим показателем разнообразия населения.

5. Установка начального диапазона [1; 100], слишком мало разнообразия для алгоритма, чтобы добиться прогресса.

6. Фитнес значение индивидуума является значение функции пригодности для данного индивида. Лучшее значение пригодности для популяции является наименьшим значением пригодности для любого индивидуума в популяции.

7. В этом случае значение целевой функции каждой отдельной (хромосомы) является вычисление с помощью функции

$$P(x) = x^2$$

8. Где x здоровье случайно выбранного индивида (хромосомы) в каждом поколении.

9. На каждом поколении случайным образом выбираются отдельные индивиды и фитнес значение рассчитывается.

10. Индивиды (хромосомы) с наилучшими значениями пригодности быстро размножаются, и также предотвращают генетический алгоритм от поиска других областей пространства решений, пропуская слабые неподходящие хромосомами.

11. Число индивидуумов с наилучшими значениями пригодности в текущем поколении, которые гарантированно выживут в следующем поколении. Генетический алгоритм использует индивидуумов в текущем поколении, чтобы создать детей, которые составляют следующее поколение.

12. Увеличение численности населения позволяет генетическому алгоритму поиск большего точек и тем самым получить лучший результат.

13. Лучшее соотношение фитнес быстро улучшается в начале поколений, когда особи находятся дальше от оптимального.

Лучшее значение пригодности улучшается медленнее в последующих поколениях, население которых ближе к оптимальной точке.

14. Фитнес масштабирование преобразует сырые фитнес оценки, которые возвращаются функции пригодности для значений в диапазоне, который подходит для функции выбора. Функция выбора использует масштабируемые значения пригодности для выбора родителей следующего поколения.

4. Carquinyoli genetic optimizer (CGO)

Дается краткий обзор CGO оптимизаторов и доказывается, что они могут превзойти классические оптимизаторы, когда число запросов на объединение (join queries) в SQL-запросе велико. Объектом исследования являются оптимизаторы на основе генетического программирования. Для того чтобы выполнить анализ и представить новые идеи для улучшения оптимизации больших запросов на объединения, внедрено новый генетический оптимизатор, основанный на генетическом программировании. Данный оптимизатор называется Carquinyoli Genetic Optimizer (CGO) [9].

Основные компоненты CGO

Представлено четыре базовых структур, используемых CGO:

База данных. CGO предполагается работа на схеме реляционной базы данных. Как обычно, схема содержит набор отношений, которые связаны через первичные и внешние ключи (PK и FK, соответственно). Каждое отношение имеет связанный с ним мощность, которая соответствует количеству кортежей или записей, сохраненных в этом отношении.

Запрос. CGO предполагает SQL, является стандартным языком, используемым для выражения запроса. Тем не менее, для того, чтобы упростить оптимизатор, CGO работает с пониженным подмножеством стандартного языка SQL, который позволяет выбрать объект, проекции, соединения и операций сортировки. Язык псевдо-SQL, используемый CGO называется CGO-SQL.

План выполнения запроса (QEP). В QEPs являются членами популяции в генетическом оптимизатора. CGO предполагает QEP структурирован в виде tree-shaped дерева.

Популяция. В QEPs организованы в популяциях. Первоначально первая популяция создается с нуля случайным образом и, после этого QEPs в популяции изменяется или удаляется.

CGO Оперативное Описание

Процесс оптимизации начинается с создания исходного набора программ (в нашем случае, программа яв-

ляется QEP), как правило, называют членом исходной популяции по CGO. Начальная популяция содержит N элементов или программ. Каждый член в популяции представляет собой способ для достижения конкретной цели и имеет соответствующую стоимость. В случае оптимизации запросов каждый QEP представляет собой способ решения запроса предоставленной системе. Начиная с этой исходной популяции, как правило, создаются с нуля, две операции используются для производства новых членов в популяции:

Операции Кроссинговера, которые сочетают в себе свойства существующих членов в популяции, а также **операции Мутации**, которые вводят новые свойства в популяцию.

Для того, чтобы сохранить размер постоянной популяции. CGO выполняет операции кроссинговер C , выбирая два случайных QEPs в популяции каждый раз, и операции мутации, выбирая одну QEP, на поколение.

После применения генетических операций, операции кроссинговера породили новые $2C$ потомств и операции мутации породили M новых потомств. Таким образом, численность населения увеличилась, содержащая $N + 2C + M$ элементов.

После этого шага, стоимость для каждого нового QEP в популяции вычисляется, а QEPs сортируются по их стоимости. Третья операция, как правило, называют, как выбор, используется, чтобы отбросить худшие подогнанные элементы, с помощью этой функции пригодности.

Этот процесс порождает новую популяцию, содержащую элементы N , называемый также поколение, которое включает в себя как старые и новые члены, которые выжили операцию выбора. Этот процесс повторяется итеративно для G поколений, пока условие остановки не заканчивает выполнение.

Каждая итерация возвращает новое население, которое эволюция популяции вернулась с предыдущей итерации.

После того, как критерий остановки удовлетворяется, лучшее решение берется из конечной популяции. Интуитивно понятно, что по мере развития населения, средняя стоимость QEP в популяции уменьшается.

Алгоритм 1, представляет собой простое описание главной процедуры, выполняемой CGO.

Во-первых, начальная популяция создается с нуля (строка 3).

После того, как P заполняется, мы входим в основной цикл в строке 4 и итерацию для G поколений.

Каждая итерация в этом цикле представляет эволюцию и создание нового поколения QEPs.

Для каждого поколения, кроссинговер и мутация операции применяются (строки 5 и 6).

После того, как все операции выполняются, текущая популяция объединяется с новыми QEPs порожденных операций (строка 7).

И операция выбора применяется сортировка членов в популяции по стоимости и отбрасывая наивысшим сметой расходов QEPs из полученного населения

Algorithm 1 CGO псевдо код

```

1: procedure CGO
  main function
2: Population P, P1, P2, P3;
3: P = creatInitialPopulation ();
4: while (stop criterion is not met) do
5: P1 ← applyCrossoverOperations (P);
6: P2 ← applyMutationOperations (P);
7: P ← P ∪ P1 ∪ P2;
8: P ← applySelectionOperation (P);
9: end while
10: end procedure
    
```

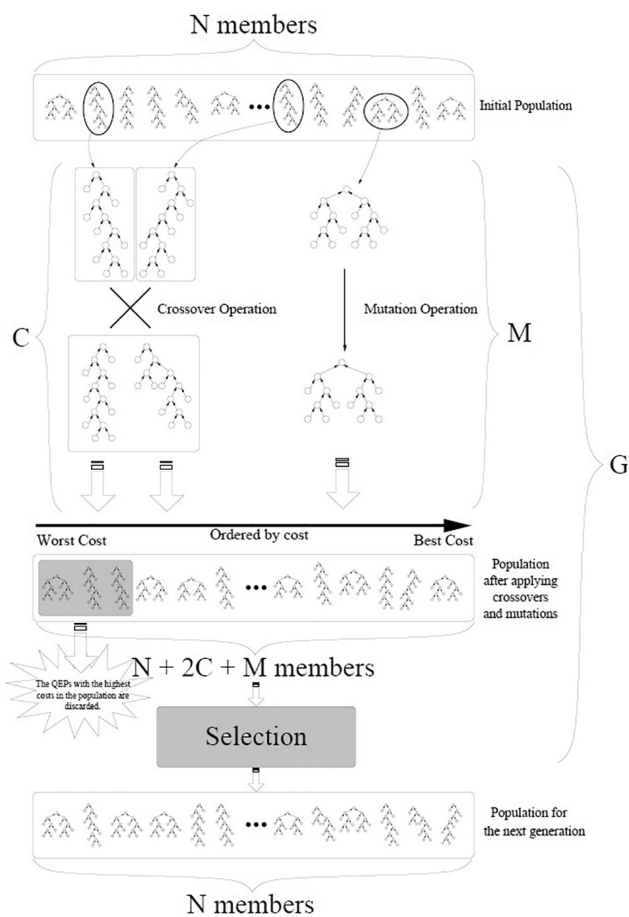


Рисунок 5. Общий вид поведения CGO

Выводы

В этой статье рассмотрено применение генетических алгоритмов в одной из самых важных задач оптимизации в области компьютерной науки, оптимизация запросов базы данных для большого join запроса.

Современные методы оптимизации запросов недостаточные для поддержки, некоторых формирующейся приложений, использующих базы данных. В этой статье решено проблему оптимизации запросов к базе данных с использованием генетического алгоритма.

Такой подход ГА подходит для больших баз данных, которые имеют огромный объем информации, и их цель состоит в том, чтобы выполнить выполнение больших запросов, имеющих большое количество запросов на соединение (более 16 join). Метод отбора и лучшее значение функции пригодности используется для обработки хромосом (особей), и мутационный процесс, который сокращает время и объем работы процессора в зависимости количества отношений.

Функция выбора присваивает более высокую вероятность отбора особям с более высокими значениями фитнес функции. Диапазон вариаций значений влияет на производительность генетического алгоритма.

Наконец график, который показывает лучшее фитнес значение для хромосом в каждом поколении, может быть использован для изучения поведения функции и может быть полезным для определения значения функции. Этот метод можно использовать для оптимизации QEPs и затрат времени и средств, необходимых для выполнения соединения. CGO оптимизатор на основе генетического программирования способен справиться с большой проблемой запросов на соединение.

Будущие области применения

Современные методы оптимизации запросов являются недостаточными для поддержки некоторых новых приложений баз данных. Генетические алгоритмы, однако, идеально подходят для обработки, классификации и контроля сложных запросов для очень-больших и разнообразных данных.

Такой подход ГА может быть использован для оптимизации запроса, имеющий большое количество запросов на соединение (более чем 16 join). Опять же можно использовать, чтобы минимизировать объем памяти, необходимый для QEP.

Литература

1. A. Swami and A. Gupta. Optimization of large join queries. In Proc. of the 1988 ACM-SIGMOD Conference on the Management of Data, pages 8–17, Chicago, IL, June 1988.
2. M. Jarke and J. Koch. Query optimization in database systems. ACM Computing Surveys, 16(2): pages 111–152, June 1984.
3. Melanie Mitchell, “An introduction to Genetic Algorithms”, Prentice Hall of India, 2004.
4. Hsiung Sam, Matthews James, “An introduction to Genetic Algorithms”, 2008.
5. Y. E. Ioannidis and Y. Kang. Randomized algorithms for optimizing large join queries. In Proc. of the 1990 ACM-SIGMOD Conference on the Management of Data, pages 312–321, Atlantic City, NJ, May 1990.
6. Optimization algorithms [Электронный ресурс]. — Режим доступа: [http://en.wikipedia.org/wiki/Category: Optimization_algorithms](http://en.wikipedia.org/wiki/Category:Optimization_algorithms). — Дата доступа: 10.07.2016.
7. A. Swami and A. Gupta. Optimization of large join queries. In Proc. of the 1988 ACM-SIGMOD Conference on the Management of Data, pages 8–17, Chicago, IL, June 1988.
8. Left-depp vs. Bushy trees [Электронный ресурс]. — Режим доступа: <http://www.deepdyve.com/lp/association-for-computing-machinery/left-deep-vs-bushy-trees-an-analysis-ofstrategy-spaces-and-its-2GTEPmpJUv>. — Дата доступа: 05.06.2016.
9. V. Muntès-Mulero, J. Aguilar-Saborit, C. Zuzarte, and J-L. Larriba-Pey. Cgo: a sound genetic optimizer for cyclic query graphs. In Proceedings of the International Conference on Computer Science.

Слухай Яна Олександрівна

бакалавр комп'ютерних наук,

Національний технічний університет України «Київський політехнічний інститут»

Слухай Яна Александровна

бакалавр компьютерных наук,

Национальный технический университет Украины «Киевский политехнический институт»

Slukhai Y.

Bachelor of computer science

The National Technical University of Ukraine «Kyiv Polytechnic Institute»

ПРИНЦИПИ ПОБУДОВИ ТА ПОРІВНЯЛЬНИЙ АНАЛІЗ СЕМАНТИЧНИХ РІЗОНЕРІВ

ПРИНЦИПЫ РАБОТЫ И СРАВНИТЕЛЬНЫЙ АНАЛИЗ СЕМАНТИЧЕСКИХ РИЗОНЕРОВ

PRINCIPLES OF CONSTRUCTION AND COMPARATIVE ANALYSIS OF SEMANTIC REASONERS

Анотація. Досліджено основні принципи побудови семантичних ризонерів та проведено їх порівняльний аналіз

Ключові слова: ризонер, *Structural reasoner*, *Pellet*, *Hermit*, семантична таблиця, гіпертаблиця, пряме виведення, зворотне виведення.

Аннотация. Исследованы основные принципы построения семантических ризонеров и проведен их сравнительный анализ

Ключевые слова: ризонер, *Structural reasoner*, *Pellet*, *Hermit*, семантическая таблица, гипертблица, прямой вывод, обратный вывод.

Abstract. The basic principles of the semantic reasoners were investigated and a comparative analysis was made.

Keywords: reasoner, *Structural reasoner*, *Pellet*, *Hermit*, semantic tableau, hypertableau, forward chaining, backward chaining.

Семантичний ризонер — частина програмного забезпечення, що здатна виводити логічні висновки з набору вибраних фактів та аксіом, а також надає можливість автоматичної підтримки таких завдань як: класифікація, налагодження, формування запитів. Правила виведення зазвичай задаються засобами мови онтологій, і часто засобами мов описової логіки [1, с. 115]. При проектуванні дуже великих онтологій (понад декілька тисяч класів) без ризонера дуже важко обслуговувати великі, складні онтології і зберігати в коректному, належному вигляді.

Серед основних задач ризонерів можна виділити такі: проводити класифікацію і виводити ієрархію класів, перевіряти консистентність онтології, визначати тип індивіда — належність до певного класу, визначати класи, що не перетинаються з заданим класом, визначати підкласи вибраних класів.

Перейдемо до алгоритмів, на базі яких найчастіше будується логіка ризонерів.

Пряме виведення (*forward chaining*) є одним з двох основних методів логічного виведення. Алгоритм починається з формування ланцюжка з наявними даними і використовує правила виведення для вилучення більшої кількості даних (від кінцевого користувача, наприклад), поки мета не буде досягнута. У системах з прямим виведенням за відомими фактами відшукується факт, який з них впливає. Якщо такий факт вдається знайти, то він записується в базу фактів. Пряме виведення називають також виведенням, керованим даними або виведенням, керованим посиланнями правил.

Зворотний вивід (або зворотне міркування) — це метод отримання висновку, який працює в зворотному напрямку від мети. Він використовується

в автоматичному доведенні теорем, машинному виведенні та інших напрямках штучного інтелекту [2, с. 146–147].

Цілям завжди відповідають висновки, у яких пізніше антецеденти розглядаються в якості нової мети. В кінцевому рахунку антецедентам повинні відповідати відомі факти (вони, як правило, визначаються як висновки, у яких завжди істинний антецедент). Таким чином, правилом виведення, яке використовується є *modus ponens*. Зворотний вивід часто використовується в експертних системах [3, с. 234–236].

Метод семантичних таблиць — це формальна роздільна процедура для формул логіки висловлювань і логіки предикатів, що дозволяє чисто синтаксичними засобами вирішувати семантичні проблеми формалізованих обчислень. Семантична таблиця — це дерево, вершинами якого є досліджувана формула і всі її підформули [4, с. 20–21]. Останні вершини кожної гілки — це обов'язково атоми, і такі вершини називаються листками. Семантична таблиця складного висловлювання *K* будується індуктивно, виходячи з семантичних таблиць підформул, що входять в висловлювання *K*. Кожній логічній зв'язці, що виконується у відповідній даній вершині-підформулі, зіставляється елементарна семантична таблиця у вигляді дерева, що розкриває логічну інтерпретацію зв'язки.

Алгоритм побудови гіпертаблиць базується на особливостях побудови аналітичних таблиць, але з використанням переваг головних ідей алгоритму гіпер-резолюцій [5, с. 106–107].

Оскільки метод є розширенням табличних методів, він забезпечує багату структуру для всього процесу виведення; важливі частини історії виведення зберігаються в таблиці і можуть бути використані для подальших логічних виведень [6, с. 7–8].

Було проаналізовано швидкодію трьох ризонерів та якість виконання ними їх головних функцій.

Таблиця 1

Відповідність між ризонерами та алгоритмами (складено автором на основі [7, с. 5])

| Ризонер | Алгоритм |
|---------------------|-------------------------------|
| Structural Reasoner | Прямого та зворотнього виводу |
| Pellet | Семантичних таблиць |
| Hermit | Гіпертаблиць |

Було використано 6 тестових онтологій для оцінки роботи ризонерів.

Для всіх тестових онтологій найкоротший час виконання логічного виведення класової ієрархії, ієрархії властивостей даних та ієрархії властивостей об'єктів показав Pellet. Час отримання всіх зв'язків «клас — підклас» онтології найкоротшим був для Structural ризонера, але він для тестової онтології № 4 виявив меншу кількість зв'язків, ніж Pellet та Hermit. Час отримання всіх зв'язків «клас — непересічний клас» найкоротшим також виявився для Structural, але кількість таких виявлених ним зв'язків у всіх шести випадках була меншою ніж у Pellet та Hermit. Серед останніх двох швидшим виявився Pellet. Часу на те, щоб показати, що онтологія не має індивідуалів, найменше витратив Hermit, однак на вирахування їх кількості у екземплярних онтологіях менше часу витратив Structural, але він знову ж таки, зв'язків «індивідуал — клас» він нарахував менше, ніж Pellet чи Hermit. Перевірка онтології на консистентність найменше часу у п'яти випадках з шести зайняла у Hermit-а, лише один раз у Structural. Отже оптимальним варіантом для виведення логічних фактів на основі даного дослідження є Pellet — ризонер, що реалізовує метод семантичних таблиць. Але перевірку на консистентність системи варто проводити, використовуючи Hermit.

Таблиця 2

Тестові онтології (використані для тестування ризонерів, наявні по відповідним посиланням у вільному доступі)

| | IRI онтології |
|-----------------------|---|
| Тестова онтологія № 1 | http://ontology.dumontierlab.com/molecule-complex |
| Тестова онтологія № 2 | http://ontology.dumontierlab.com/unit-individuals |
| Тестова онтологія № 3 | http://ontology.dumontierlab.com/time-interval-primitive |
| Тестова онтологія № 4 | http://owl.man.ac.uk/2006/07/ssw/people.owl |
| Тестова онтологія № 5 | http://www.biopax.org/release/biopax-level1.owl |
| Тестова онтологія № 6 | http://ontology.dumontierlab.com/physics-complex-1.0.owl |

Література

1. Cornet R. Non-standard reasoning services for the debugging of description logic terminologies. / Cornet R., Schlobach S. — Gottlob, G., Walsh, T., eds.: IJCAI, Morgan Kaufmann, 2003. — 529 с.
2. Hayes-Roth F. — Building Expert Systems / Hayes-Roth F., Waterman D., Lenat D. — Addison-Wesley, 1983. — 254 с.
3. Kaczor K. Overview of Expert System Shells / Kaczor K., Szymon B., Grzegorz J. — Krakow, Poland: Institute of Automatics: AGH University of Science and Technology, Poland, 5 December 2010. — 334 с.
4. Бет Э. Математическая теория логического вывода / Бет Э. — М.: Наука, 1967. — 523 с.
5. Hähnle, R. Tableaux and Related Methods. Handbook of Automated Reasoning / Hähnle, R. — Volume I. Elsevier science, 2001. — 277 с.
6. Baumgartner P. Hyper Tableaux / Baumgartner P., Furbach U. — Niemela Universitat Koblenz Institut für Informatik Rheinau 1, 56075 Koblenz, Germany, 2013. — 18 с.
7. Gardiner Tom. Automated Benchmarking of Description Logic Reasoners / Gardiner Tom, Ian Horrocks, Dmitry Tsarkov. — Description Logics Workshop 2006. — 8 с.

МІЖНАРОДНИЙ НАУКОВИЙ ЖУРНАЛ
INTERNATIONAL SCIENTIFIC JOURNAL
МЕЖДУНАРОДНЫЙ НАУЧНЫЙ ЖУРНАЛ

Сборник научных статей

Выпуск 6

1 том

Глава редакционной коллегии — д.э.н., профессор *Тарасенко И.А.*

Киев 2016

Издано в авторской редакции

Адрес: Украина, г. Киев, ул. Павловская, 22, оф. 12
Контактный телефон: +38(067) 401-8435
E-mail: editor@inter-nauka.com
www.inter-nauka.com

Подписано в печать 19.07.2016. Формат 60×84/8
Бумага офсетная. Гарнитура PetersburgC.
Условно-печатных листов 13,49. Тираж 100. Заказ № 397.
Цена договорная. Напечатано с готового оригинал-макета.

Напечатано в ООО «Спринт-Сервис»
г. Киев, ул. Почайнинская, 28б
Свидетельство: Серия ДК №4365 от 17.07.2012
Контактный телефон: +38(050) 647-1543