

ТОЛОЧКО ДМИТРО

Івано-Франківський національний технічний університет нафти і газу
<https://orcid.org/0009-0003-1171-9133>
e-mail: dimentor8683@gmail.com

ДМИТРІВ ІГОР

Івано-Франківський національний технічний університет нафти і газу
<https://orcid.org/0009-0005-3727-7593>
e-mail: dmtihor@gmail.com

ІВАНЦІВ НАЗАРІЙ

Івано-Франківський національний технічний університет нафти і газу
<https://orcid.org/0009-0009-4197-2766>
e-mail: Titop.vector@gmail.com

ДЯЧУК АНДРІЙ

Івано-Франківський національний технічний університет нафти і газу
<https://orcid.org/0009-0008-3956-570X>
e-mail: aenef28ad@gmail.com

ДМИТРЕНОК СЕРГІЙ

Івано-Франківський національний технічний університет нафти і газу
<https://orcid.org/0009-0009-7910-1923>
e-mail: sieergi0@gmail.com

КОНЦЕПТУАЛЬНІ ОСНОВИ МОДЕЛЮВАННЯ КОНТЕНТУ ІНТЕЛЕКТУАЛЬНИХ СИСТЕМ

Дана стаття присвячена дослідженню методологій задоволення обмежень як способу формування контенту інтелектуальних систем, використовуючи теорію нечітких обмежень та її застосувань у інтелектуальних системах. Розглядається впровадження формальних метаструктур для вираження преференцій, ієрархій обмежень та їх використання для оптимізації вибору рішень. Автори аналізують різні підходи до задач пошуку на основі обмежень, включаючи методи звуження, пошуку та синтезу, і зосереджуються на ролі абдуктивного міркування в логічному програмуванні для видобування даних та вирішення проблем на основі обмежень. Стаття також обговорює використання індуктивного та абдуктивного логічного програмування для підвищення ефективності алгоритмів та їх можливі додатки в реальних технологічних задачах. Отримане формальне розширення зберігає класичні визначення для міток, розмірів, інтерпретації графів, простору розв'язків і присвоєння. Окреслено нову інтерпретацію процесів задоволення та порушення обмежень, часткового задоволення обмежень у частині керування процесом розв'язування пошукових задач шляхом виділення наборів помилок (незадоволених обмежень), що, відповідно, дозволяє ідентифікувати помилкові знання та дії оператора технологічного процесу. Відповідно, процес пошуку розв'язку розглядається у вигляді задач інформаційного пошуку за кількістю розв'язків, які поділяються на недообмежені та надобмежені. Наведено формулювання пошукового завдання як стратегії автоматизованої інтелектуальної системи за рівнем знань оператора, який визначається кількістю помилок, що може бути відповідно як посилено, так і послаблено. Запропоновано інтерпретацію технологічних задач на основі методу контролю множини порушених і задовольняючих обмежень, що дозволяє присвоїти кожному обмеженню оціночне значення, яке задає загальні преференції обмежень. Введено поняття впорядкованої множини для множини оцінок, що дозволяє описати рівні порушення та задоволення обмежень, а також метод побудови структури оцінки, що визначає залежність оціночних значень від множин порушених та задовольняючих обмежень на рівні наборів, систем та ієрархій накладених обмежень.

Ключові слова: обмеження, помилки, оцінки, знання, технологічні проблеми, компаратори доменів, висновки, міркування, логічне програмування.

TOLOCHKO DMYTRO, DMYTRIV IHOR, IVANTSIV NAZAR, DIACHUK ANDRIY, DMYTRENOK SERHII
Ivano-Frankivsk National Technical University of Oil and Gas

CONCEPTUAL FOUNDATIONS OF MODELING CONTENT IN INTELLIGENT SYSTEMS

This article is dedicated to exploring methodologies for constraint satisfaction as a way of forming the content of intelligent systems, utilizing fuzzy constraints theory and its applications in intelligent systems. It discusses the implementation of formal meta-structures for expressing preferences, hierarchies of constraints, and their use for optimizing decision-making. The authors analyze various approaches to constraint-based search tasks, including narrowing, searching, and synthesis methods, and focus on the role of abductive reasoning in logical programming for data extraction and problem-solving based on constraints. The article also discusses the use of inductive and abductive logical programming to enhance algorithm efficiency and their potential applications in real-world technological tasks. The obtained formal extension preserves classic definitions for labels, sizes, graph interpretations, solution spaces, and assignments. A new interpretation of the processes of satisfying and violating constraints, partial constraint satisfaction in terms of managing the problem-solving process in search tasks by highlighting sets of errors (unsatisfied constraints), which allows for the identification of erroneous knowledge and actions by the operator of the technological process, is outlined. Accordingly, the search for a solution is considered in the form of information retrieval tasks based on the number of solutions, which are divided into under-constrained and over-constrained. The formulation of the search task as a strategy of an automated intelligent system at the level of the operator's knowledge, which can be correspondingly strengthened or weakened, is provided. An interpretation of technological tasks based on the method of controlling the set of violated and satisfying constraints, which allows assigning an evaluative value to each constraint, setting general preferences of

constraints, is proposed. The concept of an ordered set for a set of evaluations is introduced, allowing the description of levels of violation and satisfaction of constraints, as well as the method of constructing an evaluation structure, which determines the dependency of evaluative values on sets of violated and satisfying constraints at the level of sets, systems, and hierarchies of imposed constraints.

Keywords: constraints, errors, evaluations, knowledge, technological problems, domain comparators, conclusions, reasoning, logical programming.

Постановка задачі

На сьогоднішній день, актуальності набувають дослідження зосереджені на розробці інтелектуальних систем, які підтримують технологічні задачі, засновані на теорії нечітких обмежень і нечітких відношень. В даному контексті особливе місце займають проблеми оцінки обмежень та впровадження системи та ієрархії обмежень для керування процесом задоволення обмежень. Для рішення даної задачі релевантним є розробка методологій та інструментарію для підтримки вирішення технологічних завдань на основі теорії нечітких обмежень, яка включає оцінку обмежень і впровадження ієрархії обмежень. Цей підхід повинен дозволити ефективно керувати процесом задоволення обмежень, забезпечуючи високий рівень адаптації до динамічних змін у вимогах технологічних задач. Таке дослідження має охопити розробку формальних методів оцінки обмежень, методів введення і використання ієрархії обмежень, а також тестування цих підходів для оцінки їх ефективності та можливостей щодо автоматизації та оптимізації процесів вирішення технологічних задач. Сутність побудови контенту інтелектуальних систем на основі підходу до задач з обмеженнями (Constraint Satisfaction Problems, CSP) полягає у використанні математичної моделі для формулювання та вирішення задач, де необхідно знайти значення для змінних, які задовольняють ряд обмежень. В CSP кожна змінна може приймати значення з певної області визначення. У контексті інтелектуальних систем це можуть бути параметри, що визначають стан системи або вибір, який необхідно зробити. Для кожної змінної встановлюється допустимий набір значень, з яких можна вибирати. Обмеження це умови, які мають бути задоволені, щоб рішення було прийнятним. Обмеження можуть визначати відносини між змінними або обмежувати можливі значення окремих змінних. Вирішення задачі в стилістиці CSP зводиться до процесу пошуку значень для всіх змінних, які задовольняють усі обмеження. Це може включати алгоритми звуження області визначення, алгоритми пошуку або використання евристичних методів для знаходження оптимальних рішень. При розробці інтелектуальних систем, CSP може використовуватися для автоматизації процесу планування, складання технологічних регламентів, розробки конфігураторів рішень і т.д. Інтелектуальна система, таким чином, може автоматично генерувати пропозиції, керувати вимогами до ресурсів або відповідати на запити користувачів у реальному часі, враховуючи комплексні взаємозв'язки і обмеження. Використання CSP дозволяє системам бути більш гнучкими та адаптивними, забезпечуючи високу точність та оптимізацію у вирішенні складних технологічних завдань.

Аналіз останніх джерел

Ряд досліджень присвячено класичним підходам до проблеми представлення та задоволення обмежень (CSP – *constraint satisfaction problems*), а також задачам пошуку інформації на основі обмежень [1]. Перспективним напрямком досліджень є формальна теорія задоволення нечітких обмежень [2–4]. Ця теорія є розширенням класичних теорій ймовірностей, теорії можливостей і теорії нечіткої логіки. Дослідження в цьому напрямі виконуються в рамках теорії нечітких обмежень і нечітких відношень. З точки зору побудови ефективних інтелектуальних систем постає питання побудови нечітких висновків, висновку на основі обмежень та представлення предметної області за допомогою нечітких множин [5, 6]. З точки зору ефективності контролю процесу задоволення обмежень важливою є процедура оцінки обмежень [7–9], яка розглядається як спосіб проведення оцінки множинних обмежень і як метод представлення преференцій. Цей підхід можна розширити, ввівши оцінки для змінних, враховуючи кілька преференцій щодо кількох змінних. Введення вагових коефіцієнтів для змінних дозволяє розглядати набір присвоєнь, оцінювати процес виконання обмежень і контролювати значення вагових коефіцієнтів і їх кількість у процесі призначення та на наборі присвоєнь.

Окремим напрямком досліджень є технологія представлення та задоволення обмежень, заснована на введенні формальних метаструктур для вираження преференцій, таких як півкільця [10–12]. Представлення формальної мета-структури дозволяє вводити набори значень для кортежів присвоєння, а також використовувати формальні операції та мета-структури для побудови проєкцій наборів значень і обмежень. Ефективним методом роботи з обмеженнями є впровадження систем та ієрархії обмежень [14, 15]. Введення ієрархії обмежень дозволяє оперувати преференціями на рівні обмежень, тобто обмеження можна інтерпретувати як жорсткі (обов'язкові) і м'які (переважні). Цей підхід визначає послідовність і спосіб задоволення обмежень у процесі пошуку рішення, його уточнення, знаходження оптимального рішення тощо. Операції пошуку рішень на основі обмежень виконуються на основі концепції простору рішень [16–19]. Основними ідеями цих підходів є логічна інтерпретація м'яких і строгих ієрархічних обмежень, введення зв'язків упорядкування, представлення нечітких ієрархій обмежень, введення часткового впорядкування ієрархії і систем обмежень, визначення ефективних компараторів, моделі ієрархії і систем обмежень, оцінка ефективності ієрархії і систем обмежень та їх можлива оптимізація.

Відповідно до цього будується пошук певного фіксованого рішення, усіх можливих рішень і найкращого (оптимального) рішення за заданим критерієм (системою критеріїв). Класично існує три основні методи розв'язування задач пошуку на основі обмежень: *метод звуження, метод пошуку та метод синтезу*. Найпоширенішим є метод пошуку на основі кортежів значень, який зводиться до технології зворотного

відстеження. Основна відмінність пошукових завдань, заснованих на обмеженнях, від звичайних пошукових завдань полягає в здатності звузити/розширити вихідне завдання та здатності використовувати ефективні методи пошуку. Методи сумісності та методи поширення обмежень також використовуються для вирішення задач пошуку на основі обмежень [20]. Суть методу полягає у виключенні несумісних елементів з області визначення змінних. Цей метод можна розширити підходом сумісності вершин або сумісності дуг. Досить ефективним є метод алгоритмів перегляду вперед [21], суть якого полягає у виконанні присвоєння лише після перевірки його впливу на майбутні можливі кортежі рішень.

Для існуючих алгоритмів і методів розв'язання проблем з обмеженнями важливо вибрати початковий набір змінних як основний елемент контролю ефективності зворотного відстеження та уникати надглибокого зворотного відстеження як способу досягнення великої кількості повторних присвоєнь. Ця проблема вирішується за допомогою евристики впорядкування змінних. Така евристична стратегія може залежати від кількості важливих значень у домені, кількості задоволених обмежень, кількості незадоволених обмежень, використання статистичного та динамічного впорядкування змінних [22, 23]. Ефективним методом тут є визначення порядку, в якому значення присвоюються змінним, на основі певного порядку значень. Більшість алгоритмів засновані на деревах пошуку [24]. Використовуються також структурні алгоритми, засновані на представленні графів, і методи локального пошуку, наприклад генетичні алгоритми. Пошук оптимального рішення, як правило, базується на алгоритмі розгалужень і меж із введенням евристичної функції для завдань розрахунку вартості виконаних завдань [25, 26]. Використання евристичної функції дозволяє оперувати піддеревами пошуку та виконувати часткові призначення. Метод введення переваг для обмежень є ефективним інструментом вирішення завдань накладеного пошуку на основі обмежень [27–29]. Розширенням цього методу є встановлення переваг для змінних, що дозволяє переходити за допомогою заданого методу часткового впорядкування змінних від накладених задач (*задач з обмеженнями*) до оптимізації [30–31]. Таким чином, отримано нову інтерпретацію обмежень: обмеження зі змінними з мітками, де мітки виступають як переваги [32–34]. Цінність цього підходу полягає в тому, що введені мітки є локальними по відношенню до обмежень зі змінними. Нечіткі значення також можна використовувати як мітки. Використання обмежень з позначеними змінними дозволяє розрахувати вагові значення обмежень як ступені переваг в процедурах прийняття рішень і процедурах оптимізації. Тому введення міток дозволяє ефективно застосовувати пошукові завдання з нечіткими, можливісними та імовірнісними обмеженнями.

Важливим питанням досліджень у галузі пошукових задач на основі обмежень є вибір програмної парадигми для реалізації концепції представлення та задоволення обмежень. Більшість дослідників зосереджується на парадигмі логічного програмування [35–37], оскільки вона дозволяє ефективно контролювати та розділяти змінні, використовувати декларативні представлення, механізми зворотного відстеження та уніфікації. Проекти цього класу дозволяють створювати ефективні розширення класичного логічного програмування шляхом додавання концепції задоволення обмежень замість парадигми уніфікації (*CLP – constraints logic programming*). Однак реалізації ефективні, у більшості випадків, лише для числових обмежень. Особливістю цих проектів є використання розширених джерел пошуку, введення символічних змінних, а також новітніх евристик і стратегій пошуку. Слід також звернути увагу на подібні реалізації з класичними імперативними інструментами [40, 41]. Так як одним з найважливіших застосувань *CSP / CLP* є технологія планування завдань. Поки що існують лише окремі класи диз'юнктивних і кумулятивних обмежень [42–44].

У запропонованому дослідженні ми прагнемо побудувати інтелектуальну систему, яка підтримує розв'язання технологічних задач, засновану на головному елементі – відборі помилок [45–49]. Основна ідея підходу полягає в тому, що об'єкт прийняття рішень допускає помилки, виконуючи технологічні завдання та вирішуючи технологічні проблеми. Досвід показує, що помилки будуть траплятися навіть тоді, коли правильні способи дій і рішень були вивчені на попередніх сеансах роботи з системою. Згідно з цією теорією, джерелом помилок є об'єкт прийняття рішень, який припускається помилкою, оскільки отримані знання мають декларативний характер, а практичне розв'язання технологічних задач і проблем потребує процедурних (алгоритмічних) знань.

Виклад основного матеріалу

Очікувані класи технологічних проблем зі структурою оцінювання у формі *CSP* можуть бути визначені та формально описані у вигляді оцінок з *ваговими коефіцієнтами* (шляхом обчислення сум вагових коефіцієнтів для наборів, систем та ієрархій і, відповідно, визначення максимального та мінімального обсягів цього); у вигляді оцінок з коефіцієнтами ймовірності, які, відповідно, визначають ймовірність відповідності обмеження реальній технологічній задачі, оскільки, згідно з початковим припущенням, система фіксує порушення та задовольняє лише відповідних обмежень. Формальна структура оцінки з можливими коефіцієнтами дозволила б оцінити можливу потребу задовольнити або порушити набір (*систему, ієрархію*) накладених обмежень при побудові завдань у процесі пошуку рішення технологічної задачі (проблеми). Основною складовою слід вважати пошук оптимального призначення із введеною функцією мінімізації на загальній множині значень переваг множини порушених обмежень з функцією максимізації сумарних значень переваг множини задовольняючих обмежень процесу пошуку рішення. Введення переваг у цьому випадку виконує ранжування багатьох відповідних обмежень у порядку важливості. Нарешті, інтерпретація обмеження у формі правил надто можлива у формі значення релевантності в умові задоволення та оцінки можливих компараторів

домену для випадку технологічних проблем, що дозволило б сформувати формальну структуру для побудови рішення для введеної ієрархії обмежень.

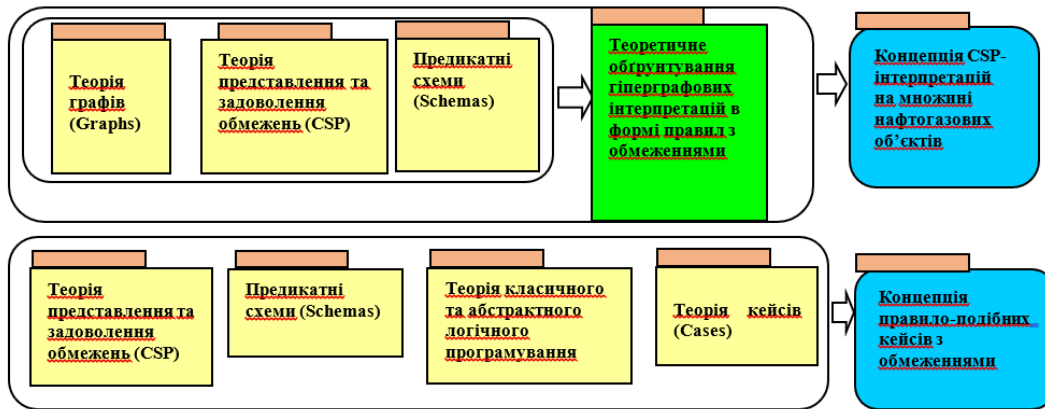


Рис. 1. Класифікаційна структура інстанціації *рушія обмежень*

Загалом існує три форми міркування:

- 1) *дедукція* - вид аналітичного процесу, заснований на застосуванні загальних правил до деяких часткових випадків, результатом якого є процес виведення (*логічного висновку*);
- 2) *індукція* - синтетичне міркування, що дозволяє вивести правило з заданого випадку і результату;
- 3) *абдукція* - вид синтетичного виведення (логічного висновку) на випадок наявного правила і результату.

Абдукція – це часткова форма міркування як форма синтетичного міркування, яке виводить випадок із правила та результату. У своїй найпростішій формі, у формі на основі обмежень, абдукцію можна описати за правилом:

$$\frac{sc, rc \Rightarrow sc}{rc}, \tag{1}$$

де *sc* - умова задоволення, *rc* - умова релевантності. Тобто, з неявного факту та спостереження, ми можемо зробити висновок. Абдукцію можна розглядати як апробацію процесу прийняття гіпотези, як пояснення спостережуваних фактів відповідно до відомих правил. Очевидно, що це слабкий вид висновку, оскільки ми не можемо сказати, що віримо в істинність пояснення, а лише в те, що воно може бути істинним. Таким чином, абдукція є розрахунком певних пояснень для спостережень і є прикладом немонотонних міркувань, оскільки пояснення, що відповідає одному стану бази знань, може стати несумісним з отриманням якоїсь нової інформації. Абдуктивне міркування характеризується існуванням кількох пояснень, і вибір переважного пояснення є важливою складовою цієї проблеми.

Дослідження дедукції виконується в контексті логічного програмування. Структура абдуктивного логічного програмування позначається як триплет

$$\langle CLP, Hyp^{set}, C^{LF} \rangle \tag{2}$$

що складається з логічної програми *CLP*, набору основних вихідних атомів *Hyp^{set}* і набору логічних формул, які називаються обмеженнями цілісності. Атоми з *Hyp^{set}* – можливі абдуктивні гіпотези, які можна розглядати як припущення, спрямовані на пояснення наведених спостережень у контексті *CLP* за умови, що ці припущення є сумісними (непослідовними) з обмеженою цілісністю в *C^{LF}* (*Constraints imposed on logical formulas*). У багатьох випадках зручно визначити *Hyp^{set}* як набір вихідних предикатних символів з деяким запланованим значенням, таким чином, що кожен базовий атом, чий предикатний символ знаходиться в *Hyp^{set}* є можливою гіпотезою.

Слід також зазначити, що такі обчислювальні процедури складаються з двох етапів: абдуктивної фази та фази узгодженості. На абдуктивній фазі обчислені процедури в основному виконують кроки SLD-резолюції та виконують збір абдуктивних гіпотез, необхідних для пояснення встановлених початкових цілей. У кожному випадку, при виборі нової гіпотези для виконання міркувань, відповідна процедура перевіряє її узгодженість (непослідовність) щодо відповідних обмежень цілісності (у цьому випадку кожна така процедура включає принаймні один вихід, отриманий у кожному обмеженні цілісності), який відкриває, відповідно, новий етап узгодженості (неузгодженості). У загальному випадку, неважко помітити, що фаза консистентності (неконсистентності) може вимагати, у свою чергу, побудови нової абдуктивної фази і т.д. Основною метою запропонованого підходу є обґрунтування того факту, що технологія абдукції може бути ефективно використана для завдань вилучення даних

Структурна ідентифікація задачі включає:

Формування релевантних баз знань $\{KB_1, KB_2, \dots, KB_n\}$, які відображають нелінійний взаємозв'язок вхідних параметрів (керованих – $\{tcp_1, tcp_2, \dots, tcp_{n1}\}$; некерованих - $\{ucp_1, ucp_2, \dots, ucp_{n2}\}$; збурюючих - $\{dcp_1, dcp_2, \dots, dcp_{n3}\}$) з вихідними параметрами $\{ocp_1, ocp_2, \dots, ocp_{n4}\}$, що дозволяє визначити та

утримувати вибраний режим буріння нафтогазових свердловин (*оптимальний, раціональний, спеціальний*).

Параметрична ідентифікація задачі включатиме:

Визначення залежності множин вхідних та вихідних параметрів шляхом знаходження таких входжень релевантної бази знань KB для яких вектори інстанціації параметрів $v = \{v_1, v_2, \dots, v_{n5}\}$ при ініціалізованих векторах вагових коефіцієнтів $CF = \{CF_1, CF_2, \dots, CF_{n6}\}$ правил бази знань $Rules^{set}: [rc \rightarrow sc]$ мінімізують очікувані відхилення результатів кейс-базованого резонінгу (*case based reasoning*)

$$\min\{\inf(Case^{CBR})\}Rules^{set} \quad (3)$$

від промислових регламентів процесу буріння нафтогазових свердловин.

Основна ідея полягає в побудові зв'язків між методами індукції як типом машинного навчання та методами абдукції. Щоб побудувати початкове рішення, необхідно розглянути можливості загального призначення абдуктивного фреймворку з точки зору побудови рішень для абдуктивних запитів і вивчення відомих систем, які використовують у своїх міркуваннях методи абдукції та індукції. *ACLP(абдуктивне логічне програмування в обмеженнях)* це система, яка поєднує в собі методи абдуктивного міркування та задоволення обмежень шляхом інтеграції фреймворку *абдуктивного логічного програмування (ALP)* із фреймворком програмування *логіки обмежень (CLP)*. Відповідно, структура ACLP була створена для вирішення проблеми надання високорівневих декларативних програмних рішень або областей моделювання для проблем штучного інтелекту, які в той же час мають прийнятний рівень обчислювальної ефективності. У ACLP роль абдуктивного міркування полягає в тому, щоб забезпечити автоматичне скорочення (звуження) представлень проблем високого рівня та цілей верхнього рівня відповідно до обчислювальних проблем нижчого рівня для незалежних форм загального представлення проблеми. Використання методів логічного програмування в обмеженнях є фундаментальним рішенням для підвищення ефективності обчислювального процесу абдуктивного висновку, оскільки воно застосовується на високому рівні представлення проблеми, що розглядається. Таким чином, цей підхід забезпечує зв'язок між властивостями домену задач високого рівня та предметно-незалежними методами розв'язання.

Загальна обчислювальна модель для ACLP складається із взаємодії між гіпотезами та генеруванням обмежень через абдуктивний висновок із перевіркою узгодженості для припущень висновку та задоволення обмежень для їхніх відповідно згенерованих рівнів. Інтеграція абдуктивного міркування з пошуком обмежень у ACLP є гібридизацією засобів вирішення обмежень що лише вирішує остаточний набір обмежень, створених процедурою абдуктивного зменшення, і також динамічно впливає на процес абдуктивного вирішення. Це дозволяє завчасно відрізати абдуктивні скорочення шляхом встановлення нових відповідних обмежень CLP для проєктованих абдуктивних рішень.

На цьому рівні система реалізована як відповідний метаінтерпретатор, який є надбудовою для відповідної мови CLP. Наприклад, той, який використовує програму вирішення обмежень CLP для керування обмеженнями над кінцевими доменами. Отримана структура системи буде загальною, а її реалізація може бути подібною до інших існуючих обмежень. Концепція ACLP застосовується до різних типів проблем. Зокрема, початкові програми були зосереджені на питаннях планування. Таким чином, структура ACLP також використовується для впровадження систем, які можуть виконувати завдання неповного розпізнавання, вирішувати неузгодженості в завданнях, що збігаються, ідентифікувати причини невідповідностей і оновлювати дизайн, які можуть відновити узгодженість специфікацій та інтеграцію інформації в цілому. Тут ACLP використовується як базова структура для розробки інформаційних посилань для семантичної інтеграції інформації над джерелами, створеними з веб-сторінок. Хоча більшість із цих додатків не використовуються широко, за винятком додатків для планування, цей клас рішень корисний як загальні методологічні вказівки щодо розробки абдуктивних додатків. Цінність цього класу рішень, наприклад, програм планування, полягає у створенні високоякісних експертних рішень, які можна порівняти з рішеннями, створеними вручну фахівцями з великим досвідом у цьому класі проблем. Незважаючи на велику кількість додатків у сфері абдукції та їх потенційних застосувань, загальна основа логіки абдукції зазвичай зосереджується на важливості репрезентації, тоді як у загальному випадку поглиблене вивчення проблем обчислювальної ефективності для задач практичного розміру та складності є слабо можливим для абдуктивних каркасів.

Під AF-системою можна розуміти базований на Prolog абдуктивний фреймворк, орієнтований на дослідження масштабованості. Його отримують в основному шляхом злиття ідей і концепцій, які слідує відомих процедурам абдукції. На найзагальнішому рівні це дозволяє нам сформулювати правила висновку у формі переписаних правил для станів. Відповідно, вихідні правила розширюються правилами, які можуть обробляти вирази кінцевої області для обмежень. На внутрішньому рівні, для рівня реалізації, системи AF-використовують аналогічні схеми з вирішувачами обмежень для кінцевих областей. Тобто вони оцінюють всю детерміновану інформацію перед тим, як робити недетермінований вибір. Відповідно, коли всі недетерміновані варіанти успішно оцінені і всі набори обмежень узгоджуються, тоді рішення вважається знайденим.

Важливо, щоб набори обмежень, вбудовані в режим виконання, активно брали участь у процесі пошуку. Це означає, що поточний стан набору обмежень може ініціювати нові виходи або ініціювати процедуру зворотного відстеження. Цей рівень рішення досягається шляхом впровадження для кожного обмеження домену набору міркувань над набором виразів обмежень Гербранда та термінів кінцевої області, які дозволяють контролювати стан виразу щодо поточного набору обмежень. Така поведінка в системах

автофокусування приймає форму повторного об'єднання, оскільки система використовує попередньо визначений модуль для виведення на кілька умов. Відповідно, система AF також використовує окремий розв'язувач Гербранда, який також підтримує процес об'єднання. Таким чином, AF-системи дозволяють включити окремий рівень на верхньому рівні розв'язувача обмежень для кінцевих областей, який забезпечить функціональність, подібну до загальних виразів для кінцевих областей.

Імовірнісна абдукція Горна є комбінацією формальної логіки та ймовірнісних уявлень. Загальна ідея полягає в тому, щоб побудувати набір незалежних виборів за допомогою логічної програми, яка визначає послідовність виборів. Формальна логіка, яка використовується в цьому випадку, базується на так званих «слабких» представленнях, які не містять диз'юнкції, і всі значення, пов'язані з невизначеністю, вбудовані в процедуру підтримки користувача. У той же час ніякі ймовірнісні залежності не накладаються на вибір, зроблений користувачем.

Цей підхід формує просту структуру для абдукції на основі тверджень Горна з ймовірностями, пов'язаними з гіпотезами. Така структура включає припущення щодо базових правил і припущення щодо незалежності багатьох гіпотез. Таким чином, можна показати, що кожен тип ймовірнісного знання, представленого в мережі дискретних байєсівських суджень, також може бути представлений у цій структурі. Таким чином, основний внесок буде полягати у пошуку зв'язків між логічними та ймовірнісними концепціями побудови міркувань. Все це забезпечить ефективну мову репрезентації, яка створить компроміс між евристичною адекватністю, а саме між інтуїтивним і теоретично обґрунтованим доведенням. Це також показує, як байєсівські мережі можна розширити за межі пропозиційної мови і як мови, яка використовує лише безумовні незалежні гіпотези, що може представляти довільне ймовірнісне знання. Це також аргументує загальний стан, який пояснює доцільність побудови нових гіпотез для пояснення залежностей замість розгляду залежностей на рівні мови.

Системи, засновані на ймовірнісному представленні Горна, не виконують явних операцій з обмеженнями цілісності. Відповідно кожен вид їх реалізації базується на засобах Прологу. Суть реалізації полягає в організації пошуку за методом меж і гілок, де в кожен момент часу розглядається кожне часткове пояснення з найменшим коефіцієнтом цінності і найвищою ймовірністю.

Крім того, такий спосіб реалізації дозволяє підтримувати пріоритетну чергу для наборів гіпотез, яка може бути розширена до рівня набору пояснень, включаючи підрівень часткових пояснень. У будь-який момент часу набір усіх пояснень розглядається як набір пояснень, які вже були згенеровані з додаванням набору пояснень, які можуть бути згенеровані з доступних часткових пояснень у пріоритетній черзі. Абдукція з коефіцієнтами вартості є однією з важливих проблем у побудові міркувань в умовах невизначеності. Суть цього підходу полягає в тому, щоб знайти краще пояснення для багатьох фактів шляхом пошуку доказів з мінімальною цінністю (вагою) для таких фактів. Тому також використовується термін абдукція з вагами, що вважається особливо корисним застосуванням, в контексті інтерпретації природної мови. Як і в інших застосуваннях та інтерпретаціях теорії абдукції, головною проблемою для побудови застосування є пошук на основі даних, що описують спостереження або події, відповідного набору гіпотез, які найкраще пояснюють узгодженість доступного набору вхідних даних.

Для загальної задачі абдукції з коефіцієнтами спостерігається збільшення складності її вирішення в гіршому випадку. Метод пошуку абдуктивних доказів мінімальної вартості за допомогою методів лінійного програмування, заснований на перетворенні вихідної задачі в лінійну задачу задоволення обмежень. Водночас за певних умов абдукцію з коефіцієнтами вартості можна вважати такою, що має поліноміальний розв'язок. Для цього випадку використовується набір достатніх умов існування поліноміального розв'язку абдукції з коефіцієнтами вартості. Існують інші методи лінійного програмування, які можуть шукати відповідні неоптимальні рішення. Цикл зворотного зв'язку в стилі Прологу з доданою операцією розкладання на множники та відсутністю обмежень упорядкування означає, що кожен, а не лише крайній лівий оператор, що може бути розв'язаний, здатний генерувати всі можливі пояснення, узгоджені з базою знань. Це означає, що кожне очікуване можливе пояснення, узгоджене з базою знань, класифікується за допомогою пояснення, яке може бути згенероване в процедурах ланцюга зворотного зв'язку та факторизації. Задовільним вважається той випадок, коли процедура гарантовано не породжує жодних пояснень, які не узгоджуються з базою знань.

У загальному випадку, узгодженість пояснення з базою знань необхідно перевіряти поза абдуктивною системою міркувань. Слід також зазначити, що не всі суперечливі пояснення генеруються системою. Навпаки, система може генерувати лише ті пояснення, які припускають існування літералів, які можуть бути досягнуті, відповідно, з початкових формул за допомогою ланцюга зворотного зв'язку. Таким чином, проблема визначеності узгодженості (неузгодженості) в загальному випадку нерозв'язна, хоча є розв'язні випадки, і ряд пояснень можна швидко відкинути як несумісні з початковою базою знань. Зокрема, у більшості випадків припущення може бути відхилено, якщо воно порушує обмеження на тип або порядок. Тому обмеження є особливо ефективними для усунення суперечливих пояснень, особливо в інтерпретаціях природною мовою.

У загальному контексті логічного програмування було створено низку структур для об'єднання індуктивних і дедуктивних представлень. Частина з них була виконана, а частина залишилася нереалізованою. Специфікації існуючих середовищ також можуть містити певні типи синтаксичних зміщень, а також існуючі обмеження цілісності. У загальному випадку така система є абстрактним середовищем для експериментування з різними методами висновків і способами побудови міркувань. Крім того, відмінною рисою таких систем є

простота комбінування різноманітних методів, що дозволяє інтегрувати абдукцію та індукцію за загальними правилами.

Загалом, використання абдуктивних технік підвищує ефективність процедур і алгоритмів вилучення даних. Такі реалізації класифікуються на верхньому рівні абдуктивної структури зі значеннями вартості, що дозволяє витягувати та інтегрувати значущі знання, отримані з необроблених даних. Абдуктивна структура створює нові абдуктивні теорії або обмеження цілісності як тип індукції за допомогою індуктивного логічного програмування з заданою початковою базою знань. У проектованому підході більш доцільно використовувати алгоритми інтелектуального аналізу даних, засновані на частому входженні наборів елементів для виведення нових фактів, зокрема, у вигляді раніше невідомих розширених знань (наприклад, за відсутності початкових значень даних). Ці висновки пізніше використовуються для покращення результатів попереднього вилучення даних, наприклад, у задачі класифікації екземплярів.

Ідея використання кластеризації полягає в тому, щоб знайти подібності між правилами та продемонструвати відмінності між правилами в одному кластері шляхом абдукції. Основна складність цієї задачі полягає у визначенні метрик, які можуть порівнювати різні правила. Після введення оптимальної метрики ми можемо отримати за допомогою кластеризації кілька наборів правил. Фіксуючи набори та порівнюючи схожі правила, тобто ті, які мають подібні висновки з різними заголовками, ми можемо дедуктивно вивести причину існування різних заголовків для правил. Таким чином, проблема полягає в тому, щоб знайти рішення наступних підпроблем: для заданого набору прикладів з правильною класифікацією кожного прикладу нам потрібно вибрати значення атрибутів, необхідних для внесення змін, щоб отримати різні класифікації.

Таким чином, з правильним вибором метрики та відповідного асоційованого алгоритму кластеризації разом із правильним абдуктивним формулюванням проблеми стає набагато можливим отримати рішення. Зокрема, одним із підходів до цієї проблеми після того, як її зазвичай кластеризували, має бути автоматичне додавання вихідних значень кожної змінної.

Проблема класифікації є однією з найважливіших проблем у сфері машинного навчання, яка полягає, як ми бачили раніше, у передбаченні класу об'єктів із заданими характеристиками і набором прикладів правильних і неправильних класифікацій для інших відомих об'єктів. Тобто при такому підході ми отримуємо суміш описового та прогнозного підходів до вирішення проблеми. Сутність машинного навчання зводиться до ряду алгоритмів, здатних якісно класифікувати нові об'єкти відповідно до заданих прикладів. Однак екземпляри та нові об'єкти можна описати лише слабкими мовами, які зазвичай базуються на ідеях атрибутів, у яких кожен об'єкт описується як такий, що складається з атрибутів та їхніх значень, які відрізняються від значень класу. Для того, щоб розширити можливості мови до рівня представлення гіпотез і прикладів першого порядку, важливо визначити, як процес навчання в логіці першого порядку узагальнює навчання типу атрибут-значення. Сьогодні більшість систем цього класу використовують логіку першого порядку для визначення структури об'єкта. У запропонованому випадку об'єкти можна розглядати як кортежі приписуваних значень. Відповідно, у мовах першого порядку також можливий опис об'єктів складними типами на верхньому рівні множин і списків.

Більшість досліджень у цій галузі пов'язані з проблемою індуктивного логічного програмування, тому більшість цих результатів можна адаптувати до проблеми вбудовування їх у абдуктивну структуру. На практиці можливо розділити ймовірнісне значення об'єкта на відповідні ймовірності його пропозиційних атрибутів, що дозволяє оцінити появу об'єкта в навчальних наборах замість появи його атрибутів. Використання методів логічного програмування для індуктивних задач дозволило нам досягти верхньої межі посилення описового компонента проблеми класифікації, використовуючи підмножину логіки першого порядку як новий опис відповідних гіпотез. Однак такий підхід став результатом значного збільшення пошукового простору. Стверджується, що навчання на основі атрибутів у більшості випадків дозволяє досягти кращих результатів по відношенню до реляційних аналогів, навіть якщо останні базуються на більш інформативних початкових знаннях. Основна причина полягає в тому, що атрибут не навчання працює в набагато меншому просторі гіпотез, що дозволяє нам досліджувати цей простір більш ретельно. Друга важлива причина полягає в тому, що позиційне навчання враховує відповідну додаткову інформацію за допомогою імовірнісного аналізу гіпотез щодо навчального набору.

Таким чином, основним напрямком досліджень є використання інструментів абдуктивного міркування для вилучення даних. Необхідно дослідити, як класифікацію можна інтерпретувати як одне із завдань абдуктивного логічного програмування, яке дозволяє використовувати обмеження домену, встановлені користувачем. Рішення з моделями класифікації на основі дерева інтерпретуються абдуктивним способом, використовуючи обмеження та покращуючи продуктивність за відсутності конкретних даних. Необхідно також розширити загальну структуру для розгляду ймовірнісної інформації, заснованої на цій основі, і похідних формальних теорій. Абдуктивна структура, заснована на витратах, також підходить для програм інтелектуального аналізу даних, що також покращить загальну якість очікуваних результатів.

Це дослідження базується на результатах, отриманих за допомогою абдуктивних міркувань у сфері вилучення даних. Існуючі класифікатори, навіть якщо вони засновані на деревах рішень і індуктивних засобах, заснованих на класичних алгоритмах, дозволяють отримати в якості відправної точки набір правил для класифікації нових заданих прикладів. Таким чином, більшість існуючих класифікаторів на основі правил виходять з припущення, що на момент класифікації ми маємо всю інформацію про нові наведені приклади. Імовірнісні підходи дозволяють заснованим на правилах класифікаторам отримати найбільш ймовірний клас за відсутності атрибута в навчальному наборі. У той же час такий тип припущень у більшості випадків іноді

призводить до неправильної класифікації. Таким чином, використання абдуктивного підходу є особливо корисним у виборі правила класифікації, яке слід застосувати, якщо новий приклад із відсутньою інформацією потребує класифікації, яку можна виконати з використанням знань предметної області.

Необхідно показати, як будувати пояснення до наведених класифікацій шляхом обчислення відсутніх значень атрибутів у процесі застосування абдукції. Крім того, слід довести обґрунтованість вибору способу представлення результатів алгоритму вилучення даних у вигляді дерева рішень або дерева класифікації. Також важливим є спосіб представлення додаткових знань предметної області у вигляді обмежень і спосіб вибору техніки міркувань у формі абдукції, що може значно покращити поведінку витягнутої моделі класифікації. Крім того, розроблений метод дозволяє обробляти додаткові знання предметної області у вигляді обмежень цілісності. Також доцільно використовувати механізм ідентифікації асоціативних правил за повної відсутності зовнішнього знання. Ми можемо розширити абдуктивну структуру, щоб представити імовірнісні знання як оцінку вартості гіпотез. У базі знань правила класифікації отримують на основі шляхів у дереві рішень шляхом їх злиття з асоціативними правилами, отриманими з наборів даних; результати класифікації в експериментах поєднуються з виразними поясненнями вибору користувача, зробленого під час класифікації. Ці правила називають кореляцією. Важливо адаптувати існуючі процедури, які можуть знайти правила кореляції, до великих баз даних.

Проблеми класифікації з точки зору абдуктивних методів можна розглядати як альтернативний процес і, водночас, як еквівалентний спосіб виконання класифікації на основі дерева рішень. Слід зазначити, що абдуктивне міркування, необхідне в такому альтернативному представленні, є досить обмеженим і не повністю використовує потенційну силу абдуктивного міркування в значенні логічного програмування. Зокрема, реалізація відповідних перетворень вимагає використання попередньо визначених канонічних наборів обмежень цілісності, в яких побудова пояснень дозволяє уникнути включення різних значень для того самого атрибута. Вважається, що ефективним способом використання абдуктивних міркувань є додавання предметно-специфічних знань для підвищення ефективності проблем класифікації. Таким чином, предметно-специфічні знання можна враховувати при виконанні класифікації багатьма способами, зокрема шляхом виділення класифікації або надання переваги певному типу класифікації над іншими категоріями. Також додавання предметно-специфічних знань до стандартних алгоритмів класифікації на основі дерева рішень не завжди забезпечить пряму та просту реалізацію, і, отже, може вимагати значних модифікацій базових алгоритмів. Однак абдуктивні фреймворки та їх існуючі реалізації зазвичай уже містять механізми, які можна безпосередньо використовувати для представлення предметно-специфічних знань і керування ними, тобто знань, представлених у певній області.

У загальному випадку ми пропонуємо використовувати обмеження цілісності в абдуктивній структурі як спосіб вираження предметно-специфічних знань. Як показують існуючі додатки, обмеження цілісності можна використовувати для додавання знань про значення різних атрибутів, включаючи той, якого немає в початковому дереві рішень. Таким чином, ця функція може допомогти підвищити ефективність завдань класифікації, оскільки обмеження цілісності можуть додати знання, які мають відношення до атрибутів, які вже є в дереві рішень, але не представлені явно. У багатьох випадках дерева рішень можуть використовувати ймовірнісні вимірювання на рівні результату тестування пар *атрибут/значення*. Іншими словами, кожна гілка дерева має мітку у вигляді значення, відповідного атрибуту, що позначає батьківський вузол, а також мітку у вигляді ймовірнісного значення, яке відображає ймовірність того, що результат спостереження присвоїть це значення для заданого атрибута. Цей тип ймовірнісної інформації корисний під час класифікації нових екземплярів із відсутніми атрибутами. Навіть у цьому випадку в задачах класифікації можна враховувати додаткові знання предметної області, щоб динамічно отримувати найкращі значення ймовірності можливих класифікацій нових екземплярів із недостатньою специфікацією через брак відповідної інформації. Наведена вище техніка потребує спеціальних зовнішніх знань про домен, який ми використовуємо як обмеження цілісності в системі. У цьому дослідженні ми розглядаємо предметні знання як правила, отримані від експерта. Як показано вище, чудовий спосіб отримати правила в домені — отримати їх із вхідних даних. Використання асоціативних правил як активних правил є можливим і простим у фреймворках абдукції загального призначення та дозволяє побудувати повністю автоматизований фреймворк шляхом поєднання двох найважливіших парадигм *вилучення даних* у формі класифікації на основі дерев рішень і на основі парадигми асоціативних правил.

Тому в окрему задачу можна виділити побудову методів абдукції та методів вилучення даних на основі асоціативних правил як обмежень цілісності за рахунок використання фактичних наборів даних, у нашому випадку *даних процесу буріння*. Задача знаходження обмежень цілісності може бути вирішена шляхом інтерактивного аналізу певних асоціативних правил, отриманих на основі апріорного алгоритму. Оскільки існує певний баланс між негативними та позитивними значеннями атрибутів, отримані правила мають високі значення довіри та підтримки. Але для покращення їх практичного застосування необхідно використовувати окремі метрики. Отримані асоціативні правила дозволяють нам представити фактичні зв'язки між атрибутами.

Тестування результатів проблем класифікації з використанням абдуктивної структури, яка наслідуює свою функціональність з дерева рішень як теорії та асоціативних правил щодо обмежень цілісності, доводить здатність системи виводити правильні інстанції класу, навіть якщо відсутні реальні значення даних. Абдуктивний висновок із обмеженнями можна розглядати як спосіб інтерпретації для виконання усікань у дереві рішень, хоча в деяких випадках абдуктивний висновок може бути більш потужним, ніж усікання.

Основне завдання полягає в тому, щоб продемонструвати спосіб поєднання результатів роботи алгоритмів класифікації при вилученні даних. Вони базовані на дереві рішень з додатковими знаннями предметної області, які походять від асоціативної моделі і, відповідно, базується на асоціативних правилах. Ці правила цікаві з точки зору отримання вхідних даних для абдуктивного середовища, здатного класифікувати, а також визначення відповідних міркувань, які використовуються для побудови класифікацій (зокрема, під час прогнозування відсутніх значень). У рамках цього підходу отримані нові ознаки, які представлені на рівні використаної в даному випадку нової абдуктивної схеми, а саме прийнятні цілі та припущення, що виконуються на різних рівнях конкретності, де надмірність припущень слід враховувати, зокрема, при побудові доказів.

Проведений аналіз показує, що абдукція з коефіцієнтами вартості є однією з найкращих структур, які задовольняють вищезазначеним вимогам. Крім того, для абдуктивної структури, базованої на логіці предикатів, яка використовується в аналізі, вагові значення, пов'язані з кожним похідним твердженням і правилами в початкових знаннях, представляють пов'язані міри невизначеності. Необхідність розширити базову структуру шляхом включення вартісних (вагових) коефіцієнтів приймається на основі міркувань щодо введення кількісної міри для вибору серед найкращих можливих рішень, у тому числі абдуктивних. Слід зазначити, що класифікація, побудована на цьому підході, може мати більше одного пояснення. І друга причина полягає в тому, що база знань використовує імовірнісну інформацію, щоб відхилити деякі малоімовірні докази. І, нарешті, процедури, які використовуються для аналізу простору пошуку, є процедурами повного пошуку з можливими скороченнями, які виконуються за допомогою базової структури обмеження цілісності. Іншим типом обмежень, які повинні міститися в кожному реалізованому рішенні, є перевірене виконання етапів обробки даних.

Враховуючи наявність великих масивів даних, які легко збираються та зберігаються за допомогою відповідних інформаційних технологій, завдання вилучення даних стає особливо важливим. Тому вилучення даних вважається одним із найважливіших кроків у процесі відкриття та вилучення знань. На сьогоднішній день отримано низку алгоритмів і методик у різних предметних областях, а також відповідні середовища, які реалізують технології інтелектуального аналізу даних. Такі середовища, відповідно, підтримують низку кроків, які безпосередньо передують процесу інтелектуального аналізу даних у формі очищення даних, а також процесів, які слідує за застосуванням алгоритмів інтелектуального аналізу даних і методів та засобів візуалізації. Важливо також забезпечити концептуальний контекст, у якому результати процедури вилучення даних можуть бути оброблені для отримання відповідей на завдання більш високого рівня складності, ніж ті, що безпосередньо надаються переобчисленими моделями вилучення даних. Наприклад, вилучення асоціативних правил із набору транзакцій дозволяє відповісти на низку основних запитань, а також вирішити більш цікаву проблему, відповідаючи на питання, які безпосередньо не представлені в даних. Це вимагає використання, наприклад, асоціативних правил, заснованих на наявних даних і структурі прогнозованих і очікуваних рішень.

Таким чином, ряд дослідників виділяють результати застосування алгоритмів інтелектуального аналізу даних як вхідних даних відповідних середовищ, які можуть покращити якість таких даних і ефективно використовувати їх, наприклад, для логічних міркувань і експертних суджень на основі результатів вилучення даних для задоволення початкових потреб користувача в частині автоматизації обробки даних. Як відомо, індукція є одним із методів узагальнення. Більшість алгоритмів інтелектуального аналізу даних засновані, відповідно, на індуктивних методах, які полягають у певному способі виведення інформації, узагальненої з вхідних даних. Однак дуже часто використання індуктивних методів на масивах необроблених даних недостатньо для генерації відповідей на запити користувачів. Це, звичайно, зумовлює необхідність додаткової обробки даних за допомогою дедуктивних методів.

Щоб отримати відповіді на запити високого рівня, потрібне середовище, здатне підтримувати процес машинних міркувань, оскільки в більшості випадків пошук рішення таких запитів можна розглядати як аналіз багатьох гіпотез, які відповідають певним обмеженням. Метод абдукції особливо ефективний у задачах класифікації та пояснення результатів застосування технології вилучення даних. Зокрема, можна ввести правила класифікації, отримані з даних та інформації, отриманих від експертів у предметній області. Також можна використовувати асоціативні правила з введеними коефіцієнтами достовірності, що дозволяє підвищити загальну якість результатів і рівень автоматизації процесу без безпосередньої необхідності в попередніх знаннях і досвіді в предметній області.

Інтелектуальний аналіз даних можна визначити як процес пошуку кореляцій або закономірностей між різними полями у великих базах даних, переважно реляційних. Відповідно, інтелектуальний аналіз даних є важливим кроком у процесі вилучення знань, який передбачає використання інтелектуальних методів для вилучення шаблонів даних із великих наборів даних. Зокрема, найважливішими моделями для інформативного опису даних, розробленими в цій галузі, є проблеми класифікації, які передбачають мітки категорій для заданих прикладів та асоціативні правила для пошуку значущих залежностей серед наборів атрибутів. І, нарешті, метод кластеризації полягає в пошуку оптимальних розділів багатьох елементів розділення наборів даних.

Класифікація даних вимагає двох послідовних кроків. Перший полягає в тому, щоб побудувати модель, яка описує заданий набір прикладів, додаючи мітку класу до кожного елемента. По-друге, розглянути використання конкретної моделі для класифікації нових прикладів для передбачення мітки категорії. Використовуються різні моделі для розрахунку задач класифікації, але найбільший інтерес представляє саме дерево рішень.

Дерево рішень - це деревовидна структура, в якій кожен внутрішній вузол представляє певний тип тестування атрибутів, кожна гілка, відповідно, представляє результат тестування, а листові вузли - класи. Індукція дерева рішень полягає в побудові такого дерева з навчальних наборів прикладів, щоб його можна було використовувати для класифікації нових прикладів для заданих значень їхніх атрибутів шляхом переходу від кореня до листків. Завдяки цій структурі можна трансформувати дерева рішень у відповідні правила класифікації, які можна легко вставити в середовище побудови суджень у формі логічних висновків. Слід зазначити, що ряд засобів машинного навчання містять генератор для побудови наборів правил класу. Все це дозволяє в деяких випадках краще класифікувати нові приклади за наявності зовнішньої інформації, такої як домен знання, що стосується певної предметної області.

Асоціативні правила корисні для визначення кореляції між атрибутами великих наборів даних. Відповідно, асоціативні правила демонструють умови щодо значення атрибута, який зустрічається з певною частотою в даному наборі даних. У свою чергу, асоціативні правила надають інформацію такого типу у формі тверджень «якщо-то». Ці правила перераховуються на основі даних i , на відміну від правил «якщо-тоді» в математичній логіці, асоціативні правила мають виражений імовірнісний характер. А саме, фактично, крім попередньої частини *if* і наступної частини *then*, асоціативне правило також має два коефіцієнти, які виражають ступінь невизначеності, пов'язаної з цим правилом. Таким чином, в асоціативному аналізі антецеденти та консеквенти — це набори подій, які не перетинаються, тобто не мають спільних елементів. Відповідно, перший коефіцієнт називається коефіцієнтом підтримки правила. Це ряд транзакцій, які містять усі елементи в попередніх і наступних частинах правила. У більшості випадків це виражається у відсотках від загальної кількості записів у базі даних. Другий коефіцієнт вважається коефіцієнтом впевненості для правила. *Коефіцієнт довіри* — це відношення кількості транзакцій, які містять усі елементи, як у послідовних частинах, так і в попередніх, що виражає реальний рівень підтримки до кількості транзакцій. Коефіцієнт підтримки представляється як ймовірність того, що випадково вибрана транзакція з бази даних міститиме всі записи в антецеденті та консеквенті, і ймовірність, відповідно, того, що випадково вибрана транзакція міститиме всі елементи в консеквенті, за умови, що всі транзакції містять введення в антецеденті.

Таким чином, очікуваний рівень достовірності можна визначити як кількість транзакцій, що містять перевірені наслідки, поділену на загальну кількість транзакцій. Завдання виділення асоціативних правил для заданих мінімальних порогів підтримки та довіри можна розділити на підпроблеми. По-перше, це пошук усіх входжень (наборів входжень), у яких підтримка транзакцій перевищує мінімальне значення підтримки. Відповідно, підтримкою для набору входжень є кількість транзакцій, які містять заданий набір входжень. Набір випадків із опорним значенням, що перевищує або дорівнює пороговому значенню, розглядається як набір частих випадків. Ця підзадача перенаправляється на виконання алгоритму типу APRIORI. По-друге, це створення всіх асоціативних правил з мінімальною підтримкою та рівнем достовірності з пулу всіх великих наборів подій.

Вилучення асоціативних правил може бути використано для вирішення проблем класифікації. Технологія кластеризації даних передбачає розбиття даних на групи схожих об'єктів. Зрозуміло, що представлення даних невеликою кількістю кластерів призводить до певної неточності, але дозволяє досягти певного рівня спрощення загальної проблеми. Таким чином, завдання кластеризації полягає в структуруванні заданого набору некласифікованих екземплярів для нового пулу екземплярів шляхом створення концепцій, заснованих на подібності, які можна знайти спочатку в наборі навчальних даних.

На відміну від *асоціативних правил* і *класифікаторів*, у цьому випадку зазвичай немає визначеного результату чи атрибута, який можна передбачити. Це означає, що процес пошуку кластерів слід розглядати як процес неконтрольованого навчання. У цьому випадку основною відмінністю від контрольованого навчання є відсутність цільового предикату, який розділяє екземпляри навчального набору на категорії. Відповідно, категорії формує сам об'єкт навчання.

Таким чином, задачу кластеризації можна формалізувати наступним чином.

Дано: набір некласифікованих екземплярів мови з накладеними обмеженнями.

Знайти: набір понять, що охоплює всі наведені приклади, щоб:

1. Подібність між прикладами однакових понять була максимально збільшена.
2. Подібність між прикладами різних концепцій зведена до мінімуму.
3. Усі накладені обмеження задовольнялися принаймні частково.

Технології кластеризації є предметом активних досліджень у ряді областей, таких як статистика, розпізнавання образів і машинне навчання. Технологія інтелектуального аналізу даних загалом ускладнює інструменти кластеризації, а саме використання великих наборів даних із багатьма атрибутами різних типів. Все це накладає особливі обчислювальні вимоги для використання відповідних алгоритмів кластеризації. У деяких програмах деякі форми кластеризації є вирішальними. Основна перевага полягає в тому, що кластеризація об'єктів за подібністю відкриває нові можливості для їх застосування. Ряд алгоритмів для пошуку частих входжень наборів елементів засновані на порівняльній генерації можливих кандидатів для алгоритму APRIORI. Одноранговий підхід призначений для збереження простору пошуку достатньо малий, щоб поміститися в основну пам'ять. Ця стратегія також вимагає кількох проходів через кожен набір даних.

Техніка розділення вимагає двох проходів через базу даних. На першому етапі набір даних ділиться на кілька частин, досить малих, щоб поміститися у виділений обсяг пам'яті. На наступному кроці кожен розділ уточнюється за допомогою порівняльного алгоритму, а потім результати кожного розділу об'єднуються. Ця

процедура дозволяє побудувати надмножину рішень. Наступним кроком є видалення хибно позитивних елементів із цієї надмножини. Однак є проблема: якщо набір даних досить великий, то результуюча розбивка може виявитися занадто малою для нього. Це, в свою чергу, призведе до формування надвеликої надмножини, що істотно знизить загальну ефективність алгоритму.

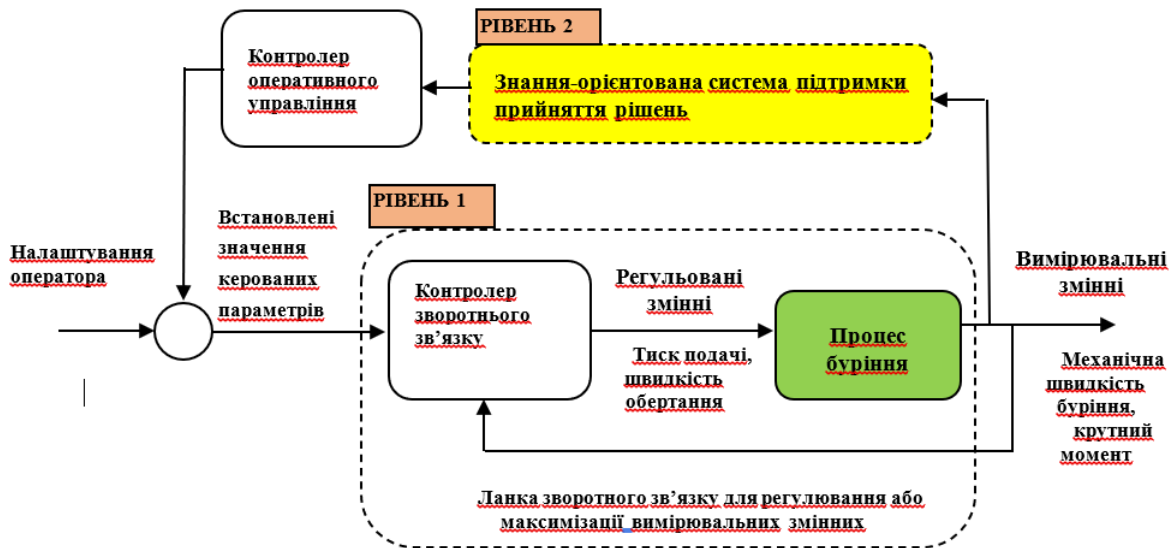


Рис. 2. Структурна схема знання-орієнтованої підтримки прийняття рішень

Іншим важливим підходом до зменшення кількості проходів через набір даних є підхід, заснований на оцінці малих випадкових вибірок (кортежів) успішних наборів. Відповідно, створюється набір шаблонів, які, ймовірно, є загальними для всього набору даних, і на наступному кроці перевіряється точна частота їх появи в решті набору даних. Якщо під час генерації записів-кандидатів стався збій, надається механізм, який обчислює решту частих шаблонів під час другого проходу. Відповідно, знизивши поріг підтримки, можна знизити ймовірність відмови. Але для низьких значень ймовірності це значно збільшить кількість записів кандидатів. Крім того, якщо ми маємо справу з великими наборами даних, ймовірно, що невелика вибірка не буде достатньо репрезентативною для всього набору даних, і це означатиме, що існує висока ймовірність невдачі. У цьому випадку в основній пам'яті буде занадто багато записів-кандидатів для перевірки другого проходу. Для цього буде потрібно більше двох кроків. Було протестовано відповідні алгоритми, які показали високе споживання пам'яті навіть для середніх наборів даних. Для запропонованого вдосконалення базових алгоритмів емпірично виправдано зменшити необхідний обсяг пам'яті в очікуваному найкращому випадку. Відповідно, вимоги до пам'яті залежать від розміру набору даних. Цей факт робить ці алгоритми практично непридатними, якщо набори даних досить великі. Додатковим доказом обмежень масштабованості для існуючих алгоритмів для вилучення частих наборів входжень є кілька відомих алгоритмів, які були реалізовані та незалежно протестовані. Результати показали, що жоден із існуючих алгоритмів не підходить для масштабування надвеликих наборів даних із занадто великою кількістю транзакцій.

Розробка алгоритму для отримання частих наборів даних у розрідженому пулі наборів даних вимагає невеликого обсягу пам'яті та зберігає невелику кількість переходів у вхідних масивах. Він долає проблему обробки великої кількості транзакцій, зменшуючи простір пошуку, зберігаючи дійсність встановлених значень пулу. Основним завданням для дослідження є пошук наборів даних заданого розміру безпосередньо без створення менших наборів даних, як це робиться в порівняльних підходах. Таким чином, вилучення всіх частих наборів даних зменшить використання пам'яті та збільшить ефективність використання часу. З іншого боку, виконання вилучення максимального набору даних може бути недостатнім для побудови відповіді на запит, оскільки фактично максимальний набір елементів не дозволяє обчислити точне значення коефіцієнта підтримки для вузького (меншого) набору даних. Однак для великого набору даних з великою кількістю малих транзакцій можна розрахувати часті набори даних, хоча більшість доступних алгоритмів не дають позитивного результату через надмірні вимоги до ресурсів пам'яті.

Перш за все, слід відмітити, що проблему пошуку частих наборів даних можна перетворити на проблему пошуку частих символів у великому потоці символів у заданому алфавіті, відомому як проблема агрегованих маркерів або проблема аналізу гарячих списків. Використання доступних на даний момент алгоритмів для вирішення проблеми агрегатів вирішить початкову проблему частих наборів даних лише двома послідовними проходами по набору даних із додатковим підготовчим кроком, спрямованим на обчислення певних статистичних значень у наборі даних. Щоб вирішити цю проблему, нам потрібно показати, що для розріджених наборів даних, тобто наборів даних з невеликою кількістю елементів на транзакцію відносно загальної кількості можливих елементів, необхідний обсяг основної пам'яті, необхідний для розробленого алгоритму, є достатньо низьким і не залежить від числа елементів і розміру набору даних. Таким чином, при пошуку асоціативних правил нам необхідно визначити часті набори ключових елементів. При використанні стандартних алгоритмів порівняння його необхідно виконувати перед проходженням набору даних.

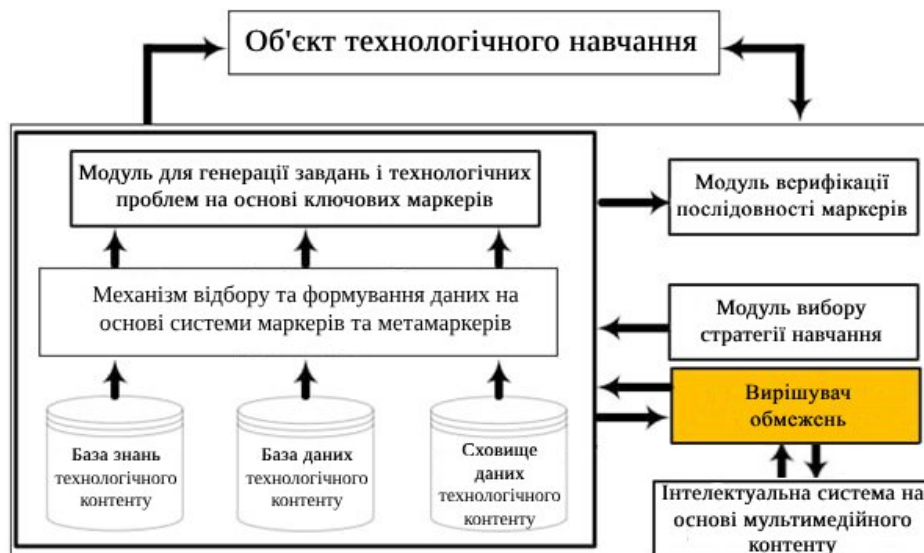


Рис. 3. Структура типової інтелектуальної системи на основі рушія обмежень

Використовуючи запропонований підхід, можна реалізувати два екземпляри алгоритму паралельно, що вимагає загалом до трьох проходів набору даних. Таким чином, основним завданням буде розробка алгоритму, який вимагатиме обмеженого обсягу пам'яті для розріджених наборів даних при збереженні невеликої постійної кількості проходів над вхідним набором даних. Крім того, можна створити розширення цього алгоритму, яке працюватиме з заданими потоками даних. У цьому випадку головним результатом буде розробка алгоритму, який при низькому споживанні пам'яті зможе витягувати часті набори елементів над підмножинами потоку з розміром, пропорційним довжині потоку зчитування.

Розроблений підхід дозволяє обчислювати часті набори даних заданого розміру безпосередньо, тобто без обчислення менших наборів даних, шляхом двох проходів через набір даних. Необхідний обсяг пам'яті відомий заздалегідь і є прийнятним за початкової гіпотези, що вказаний набір даних є розрідженим. Запропоноване рішення базується на підході до вирішення проблеми сукупних маркерів, який пропонує простий алгоритм пошуку всіх запитів із частотою, більшою або рівною заданому пороговому значенню, із заданого потоку запитів (сукупних маркерів), використовуючи пам'ять із заданий бал і виконання двох проходів через потік. Очікувані результати базуються на методі звуження задачі обчислення частих наборів даних до задачі знаходження маркерів агрегатного класу для заданої технологічної задачі.

Висновки

В даному дослідженні показано, як абдуктивні міркування можна використовувати в контексті проблем класифікації та для підвищення загальної ефективності, коли система працює з неповними даними та знаннями зовнішньої області. Запропонований підхід був застосований шляхом поєднання різних парадигм вилучення даних, таких як класифікація, асоціативні правила та кластеризація, а також використання абдуктивної структури з обмеженнями. Результати показують, що абдуктивну структуру можна розглядати як рівень, здатний отримувати знання від користувача та відповідні запити мови програмування високого рівня, а потім окреслювати рішення. Відправною точкою дослідження було поєднання інтелектуального аналізу даних із системою логічного програмування, що дало можливість використовувати знання предметної області в процесі вилучення та використання знань, що в цілому забезпечить результати для покращення якості процедури дерева класифікації на рівні простого обходу та застосування загальної основи відповідно на основі абдуктивного підходу. Основна гіпотеза полягає в тому, що застосування класифікаційного дерева до нових спостережень можна вважати методом прямого розрахунку, як тільки дерево буде представлено набором правил очевидним чином. Все це відкриває нові можливості, оскільки абдуктивне міркування може мати досить складну форму, зокрема, воно може враховувати декілька типів знань одночасно. У цьому контексті вилучення реляційних даних, яке є вилученням даних на описовій мові першого порядку, можна вважати ефективним поєднанням логічного програмування та відносно невеликого простору пошуку за допомогою використання імовірнісних значень. Таким чином, використання індуктивної ймовірнісної структури може бути більш ефективним для знань предметної області, описаних засобами, відмінними від логіки першого порядку. Сконструйовані представлення містять асоціативні правила як обмеження цілісності в межах домену і як правила бази знань для абдуктивних доказів. Звичайна структура в аналізі, заснована на асоціативних правилах для визначення коефіцієнта підтримки або коефіцієнта довіри, не дає хороших результатів у всіх випадках, якщо не виконується точне встановлення порогових значень. У запропонованому підході очікуваним результатом є використання показників, які встановляють точну кореляцію між атрибутами. Цей тип кореляції важливий у сфері абдуктивного міркування, оскільки він може краще виражати зв'язки між причинно-наслідковими подіями.

Література

1. Bulatov, A. Krokhin, B. Larose, Dualities for Constraint Satisfaction Problems. Complexity of Constraints, LNCS 5250. 2008. Pp. 93-124.
2. S.H. Nasserini, E. Behmanesh, F. Taleshian, M. Abdolalipoor, Taghi Nezhad N.A. Fully fuzzy linear programming with inequality constraints. International Journal of Industrial Mathematics. Fall 2013. Vol. 5. Number 4. P. 309-316.
3. J. Mula, R. Poler, J.P. Garcia-Sabater, Material Requirement Planning with fuzzy constraints and fuzzy coefficients. Fuzzy Sets and Systems. 2007. Volume 158. Issue 7. P. 783-793.
4. M. Jiménez, M. Arenas, A. Bilbao, V.M. Rodríguez, Linear programming with fuzzy parameters: An interactive method resolution. European Journal of Operational Research. 2007. Volume 177. Issue 3. P. 1599-1609.
5. P. Hofstedt, P. Pepper, Integration of declarative and constraint programming. Theory and Practice of Logic Programming. 2007. Vol. 7. Issue 1-2. P. 93-121.
6. F. Kuipers, T. Korkmaz, M. Krunz, P. Mieghem Van. Performance evaluation of constraint-based path selection algorithms. Network. Sept.-Oct. 2004. Vol. 18. No. 5. P. 16-23.
7. X. Vu, B. O'Sullivan, Semiring-Based Constraint Acquisition. 19th IEEE International Conference on Tools with Artificial Intelligence (Patras, Greece, 2007). ICTAI' 2007. P. 251-258.
8. M. Sachenbacher, B. Williams, Diagnosis as Semiring-based Constraint Optimization. Proceedings of the 16th European Conference on Artificial Intelligence (Valencia, Spain, August 22-27, 2004). ECAI' 2004).
9. S. Bistarelli, U. Montanari, F. Rossi Semiring-based constraint logic programming: syntax and semantics. ACM Transactions on Programming Languages and Systems. January 2001. Vol. 23. No. 1. P. 1-29.
10. Y.-M. Wang, K.-S. Chin, Fuzzy analytic hierarchy process: A logarithmic fuzzy preference programming methodology. International Journal of Approximate Reasoning. 2011. Volume 52. Issue 4. P. 541-553.
11. Y.-Y. Chen, J.T. Lin, Hierarchical multi-constraint production planning problem using linear programming and heuristics. Journal of the Chinese Institute of Industrial Engineers. 2008. Vol. 25. No. 5. P. 347-357.
12. O. Sundström, D. Ambühl, L. Guzzella, On Implementation of Dynamic Programming for Optimal Control Problems with Final State Constraints. Oil & Gas Science and Technology. 2010. Vol. 65. No. 1. P. 91-102.
13. M. Borges, A. Filieri, M. d'Amorim, C.S., Păsăreanu, W. Visser Compositional solution space quantification for probabilistic software analysis. 35th ACM SIGPLAN Conference on Programming Language Design and Implementation, June 2014. PLDI' 14. P. 123-132.
14. L. De Raedt, G. Tias, N. Siegfried, Constraint programming for itemset mining. 14th ACM SIGKDD international conference on Knowledge discovery and data mining, (Las Vegas, Nevada, USA, August 2008). KDD' 2008. P. 204-212.
15. Bessiere, Constraint Propagation. Foundations of Artificial Intelligence. 2006. Vol. 2. Pp. 29-83.
16. E. Hebrard, B. Hnich, B. O'Sullivan, T. Walsh, Finding Diverse and Similar Solutions in Constraint Programming. The Twentieth National Conference on Artificial Intelligence and the Seventeenth Innovative Applications of Artificial Intelligence Conference, (Pittsburgh, Pennsylvania, USA, July 9-13, 2005). 2005.
17. T. Crain, V. Gramoli, M. Raynal A Contention-Friendly Binary Search. Euro-Par 2013 Parallel Processing. Euro-Par 2013. Lecture Notes in Computer Science. 2013. Vol 8097. Springer, Berlin, Heidelberg Tree.
18. P.-E. Hladik, H. Cambazard, A.-M. Déplanche, N. Jussien, Solving a real-time allocation problem with constraint programming. Journal of Systems and Software. 2018. Volume 81. Issue 1. P. 132-149.
19. K. Apt, E. Monfroy, Constraint programming viewed as rule-based programming. Theory and Practice of Logic Programming. 2001. Vol.1. Issue 6. P. 713-750.
20. Li Zhu, A Decision Method for Over-constraint of Parametric Model.
21. R. Jagadeesan, W. Marrero, C. Pitcher, V. Saraswat, Timed constraint programming: a declarative approach to usage control. 7th International ACM SIGPLAN Conference on Principles and Practice of Declarative Programming, (Lisbon, Portugal, July 11-13 2005). PPDP' 2005.
22. M. Gelfond, V.S. Mellarkod, Y. Zhang, Systems integrating answer set programming and constraint programming. Annals of Mathematics and Artificial Intelligence. August 2008. Vol. 53. Issue 1-4. P. 251-287.
23. F. Rossi, K.B. Venable, T. Walsh, Preferences in Constraint Satisfaction and Optimization. AI Magazine. Vol. 29. No. 4. P. 58-68.
24. L. Thiele, K. Miettinen, P.J. Korhonen, J. Molina, A Preference-Based Evolutionary Algorithm for Multi-Objective Optimization. Evolutionary Computation. 2009. Vol. 17. No. 3. P. 411-236.
25. U. Junker, Preferred Explanations and Relaxations for Over-Constrained Problems. P. 167-172.
26. L. De Raedt, K. Kersting, Probabilistic Inductive Logic Programming. Lecture Notes in Computer Science. 2008. Vol 4911. P. 1-27.
27. Yan Hong, Yu Zhenxin, T.C. Edwin Cheng, A strategic model for supply chain design with logical constraints: formulation and solution. Computers & Operations Research. 2003. Volume 30. Issue 14. P. 2135-2155.

28. J.N. Hooker, Logic, Optimization, and Constraint Programming. *INFORMS Journal on Computing*. 2002. Vol. 14. No. 4. P. 293-420.
29. O. Schimpf, K. Shen, ECLiPSe – From LP to CLP. *Theory and Practice of Logic Programming*. Vol. 12. Special Issue 1-2: Prolog Systems. 2012. Pp. 127-156.
30. T. D. S. Swift, D. S. XSB Warren, Extending Prolog with Tabled Logic Programming. *Theory and Practice of Logic Programming*. Vol.12. Special Issue 1-2: Prolog Systems. January 2012. Pp. 157–187.
31. H. Zhu, D. Liu, S. Zhang, S. Teng, Y. Zhu, Solving the Group Multirole Assignment Problem by Improving the ILOG Approach. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*. Vol. 47. Issue: 12. Dec. 2017. Pp. 3418 – 3424.
32. M. Lienhardt, I. Lanese, C. A. Mezzina, J.-B. Stefani, Reversible Abstract Machine and Its Space Overhead. *International Conference on Formal Techniques for Distributed Systems. Formal Techniques for Distributed Systems*. Vol. 7273. 2012. Pp. 1-17.
33. M. EL H. CHARAF, M. BENATTOU, S. JESS AGENT AZZOUZIA, Based Architecture for Testing Distributed Systems. *JOURNAL OF INFORMATION SCIENCE AND ENGINEERING*. 2014. Pp. 1619-1634.
34. S. Heinz, J. Schulz, Explanations for the Cumulative Constraint: An Experimental Study. *Experimental Algorithms*. 2011. Vol 6630. P. 400-409.
35. M. Fischetti, D. Salvagnin, An In-Out Approach to Disjunctive Optimization. *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*. 2010. Vol 6140. P. 136-140.
 - A. Schutt, T.Feydy, P.J., Stuckey, Explaining the cumulative propagator. *Constraints*. 2011. Vol. 16. P. 250-282.
36. S. Rach, S. Ufer, A. Heinze, Learning from errors: effects of teachers training on students' attitudes towards and their individual use of errors. *PNA*. 2013. Vol. 8. No. 1. P. 21-30.
37. M. Piteira, C. Costa, Learning computer programming: study of difficulties in learning programming. *International Conference on Information Systems and Design of Communication*, July 2013. P. 75-80.
38. F.L. Lewis, D. Vrabie Reinforcement learning and adaptive dynamic programming for feedback control. *Circuits and Systems Magazine*. 2009. Vol. 9. No. 3. P. 32-50.
39. F.Rahmani, K. Leifels, Abductive Grounded Theory: a worked example of a study in construction management. *Construction Management and Economics*. 2018. Vol. 36. Issue 10. P. 565-583.
40. K. Philipsen, Theory Building: Using Abductive Search Strategies. *Collaborative Research Design*. 2018. P. 45-71.
41. S. Timmermans ,Abductive analysis. *Kwalon*. 2016. Vol. 21. P. 66-67.
42. Awuzie, P. McDermott, An abductive approach to qualitative built environment research: A viable system methodological exposé. *Qualitative Research Journal*. Vol. 17. Issue 4. P. 356-372.
43. F. Soler-Toscano, D. Fernandez-Duque, A. Nepomuceno-Fernandez, A modal framework for modelling abductive reasoning. *Logic Journal of the IGPL*. Apr. 2012. Vol. 20. No. 2. P. 438-444.
 - A. Alrajeh, A. Fearfull, E. Monk Qualitative Research Process Using Abductive Approach. 2012.
44. L. Caroprese, E. Zumpano Indefinite abductive explanations. 2019. P. 233-254.
45. M. Komosinski, A. Kups, M. Urbanski, Multi-criteria Evaluation of Abductive Hypotheses: Towards Efficient Optimization in Proof Theory. *18th International Conference on Soft Computing*, January 2012. P. 320-325.
46. [Towards_efficient_optimization_in_proof_theory/links/02e7e51d593c9c9a0a000000/Multi-criteria-evaluation-of-abductive-hypotheses-Towards-efficient-optimization-in-proof-theory.pdf](#). (Date of access: 18.02.2021).
47. Melnyk V. D. *Intellimedia automated decision support system for drilling oil and gas wells*. Dissertation On the manuscript rights for obtaining the scientific degree of Candidate of Technical Sciences in the specialty 05.13.07 - "Automation of control processes". Ivano-Frankivsk National Technical University of Oil and Gas, Ministry of Education and Science of Ukraine, Ivano-Frankivsk, 2021. 289 p.