

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ТЕХНОЛОГІЙ ТА
ДИЗАЙНУ

КАФЕДРА ЕКОНОМІЧНОЇ КІБЕРНЕТИКИ

ШТУЧНИЙ ІНТЕЛЕКТ
ОПОРНИЙ КОНСПЕКТ ЛЕКЦІЙ

*для студентів денної та заочної форми навчання
галузі знань 0305 Економіка підприємництва
напряму підготовки 6.030502 Економічна кібернетика*

Київ – 2013

УДК 004.853(07)

Штучний інтелект: опорний конспект лекцій (спеціальність – “Економічна кібернетика”) денної та заочної форми навчання галузі знань 0305 Економіка підприємництва, напряму підготовки 6.030502 Економічна кібернетика / упор.: О. Ю. Чубукова, Н. О. Іванченко – К.: КНУТД, 2013. – 43 с.

Упорядники:

д.е.н, проф. Чубукова О. Ю.

к.е.н, доц. Іванченко Н. О.

Затверджено на засіданні кафедри

Економічної кібернетики

Протокол № 2 від 18.09.13

Література

1. Люгер Дж. Ф. Искусственный интеллект. — М.: Изд. дом «Вильямс», 2005. — 864 с.
2. Бондарев В. Н., Аде Ф. Г. Штучний інтелект. Навчальний посібник для вузів. — Севастополь: Вид-во СевНТУ, 2002. — 615 с.
3. Адаменко А., Кучуков А. Логическое программирование и Visual Prolog. — СПб.: БХВ. — Петербург, 2003. — 992 с.
4. Марселлус Д. Программирование экспертных систем на Турбо-Прологе. — М.: Финансы и статистика, 1994. — 256 с.
5. Братко И. Программирование на языке Пролог для систем искусственного интеллекта. — М.: Мир, 1990. — 560 с.
6. Тейз А., Грибомон П., Луи Ж. Логический подход к искусственному интеллекту: от классической логики к логическому программированию. — М.: Мир, 1990. — 432 с.

1. Місце та значення дисципліни

”Штучний інтелект”

Дисципліна «Штучний інтелект» відіграє фундаментальну роль у підготовці фахівців у галузі інформатики й обчислювальної техніки. Різні навчальні курси з проблематики штучного інтелекту (ШІ) включено до навчальних планів багатьох університетів. Асоціаціями ACM, AIS і AITP розроблено програму підготовки фахівців у галузі інформаційних систем, яку рекомендовано для міжнародного використання. Програмою передбачено в рамках дисципліни «Штучний інтелект» розгляд таких тем: історія, основні поняття і напрями розвитку; подання завдань і простору пошуку рішень; основні стратегії управління висновком (пошук у глибину і в ширину, прямий і зворотний висновок); евристичний пошук; представлення знань; прикладні експертні системи й оболонки; нечітка логіка і нечіткий висновок; машинне навчання; розпізнавання образів; нейронні мережі; обробка природної мови; розпізнавання мови і машинний зір.

Мета видання конспекту лекцій — забезпечити систематичний розгляд всього спектра тем, рекомендованих для вивчення програмою підготовки фахівців у галузі економічної кібернетики, включаючи й опис загальних підходів та принципів ШІ, і алгоритмічні основи.

Для опанування даної дисципліни потрібні знання в галузі програмування і ЕОМ, вищої математики, теорії інформації та математичної статистики.

2. Розширена програма курсу

Тема 1. ІСТОРИЧНИЙ ТА ФІЛОСОФСЬКИЙ АСПЕКТИ ПРОБЛЕМИ ШТУЧНОГО ІНТЕЛЕКТУ

Перелік знань та навичок

Після опанування теми студент має знати:

- історичні етапи систем штучного інтелекту;
- соціально-філософські наслідки створення та впровадження в життя систем штучного інтелекту;

- переваги та заперечення тесту Тьюринга і фактичний діалог;
- сучасні напрями дослідження у СШІ та їх класифікацію;
- узагальнену структуру інформаційної системи інтелектуального робота;

- три закони роботехніки Айзека Азімова.

Має **вміти**:

- пояснити визначення предмета теорії штучного інтелекту та системи штучного інтелекту (СШІ);

- навести приклади з розвитку цивілізації інформаційних революцій;

- пояснити спроможність моделювання мислення людини;

- пояснити місце інтелектуальної системи у СШІ;

- які є підходи в моделюванні інтелектуальної діяльності людини?

- описати та використати процедуру тесту Алана Тьюринга;

- класифікувати напрямки досліджень у галузі штучного інтелекту за Хантом;

- визначати сфери застосування СШІ;

- скласти загальну характеристику символного та конекціоністського підходів до створення систем штучного інтелекту.

Штучний інтелект — галузь наукового знання, яка об'єднує велике число напрямів, принципів, займається дослідженням і закономірностей розумової діяльності, і моделюванням завдань, які традиційно відносять до інтелектуальних;

область застосування ШІ це: доведення теорем, аналіз ситуацій; прийняття рішень, розпізнавання образів, логічне мислення, розуміння нової інформації, навчання й самонавчання, планування цілеспрямованих дій, адаптивне програмування, обробка даних на природній мові, нейромережі, програмні агенти, робототехніка;

тест Тьюринга не втратив своєї фундаментальності й актуальності, мало того набув нового звучання у зв'язку з виникненням глобальних мереж;

майбутнє ШІ є створення роботехніки та штучного життя.

Термін «штучний інтелект» — Artificial Intelligence (AI) — був запропонований в 1956 р. на семінарі, присвяченому рішенням логічних задач, з аналогічною назвою в Дартсмудському коледжі (США).

Термін інтелект (intelligence) походить від латинського intellectus, що означає розум, глузд, розумові здібності людини. Відповідно штучний інтелект — ШІ (AI) звичайно тлумачиться як властивість автоматичних систем брати на себе окремі функції інтелекту людини, наприклад, обирати й ухвалювати оптимальні рішення на основі раніше одержаного досвіду й раціонального аналізу зовнішніх дій.

В історії людства можна виділити кілька інформаційних революцій, які були переламними моментами в розвитку цивілізації.

Перша була пов'язана з появою мови. Мова дала можливість передавати знання від одного індивідуума до другого і, отже, зберігати його з покоління в покоління.

Друга інформаційна революція була пов'язана з появою писемності. Писемність дозволила обмінюватися знаннями індивідуумам без безпосереднього контакту і, таким чином істотно збільшити доступність і надійність збереження знань.

Третя революція — поява книгодрукування ще збільшило доступ до знань і зробило можливим їх масове поширення і збереження.

Четверта революція була пов'язана з появою електрозв'язку електричними засобами запису знань (інформації).

П'ята революція, яку ми переживаємо зараз, пов'язана з появою масових ЕОМ, об'єднаних в мережі, і які мають потужні засоби для зберігання, накопичення й використання знань. В межах цієї інформаційної революції можна виділити також ряд етапів, останнім з яких є поява систем штучного інтелекту, або систем, заснованих на знаннях. Загалом п'ята революція дала змогу зберігати й мати швидкий доступ практично до необмеженого об'єму знань. Крім того, завдяки системі штучного інтелекту з'явилася можливість

вперше пов'язати безпосередньо знання з матеріальним виробництвом (або в більш загальному випадку — з навколишнім матеріальним світом), виключивши людину як проміжну ланку. Мало того, системи самі можуть виробляти нові знання. Для цього служать засоби видобування (або придбання) знань, наприклад з баз даних шляхом виявлення закономірностей і формулювання цих закономірностей у вигляді баз знань. У комп'ютерному світі набули поширення дослідження й розробки в галузі так званих технологій «data mining» — видобування корисних даних (знань) з великої їх кількості, зокрема з неструктурованих даних.

Сучасна інформатика багато в чому зобов'язана дослідженням у галузі штучного інтелекту. Наприклад, багато розділів дослідження операцій з'явилося з розроблених методів обмеження перебирання варіантів при вирішенні завдань ШІ. На термінологію й методи розробки компіляторів вплинули дослідження в галузі машинного перекладу. Засоби розпізнавання й синтезу мови зараз вже є невід'ємною часткою деяких спеціалізованих інформаційних систем і претендують на широке використання в операційних системах. Системи розпізнавання тексту є звичайною часткою офісних програмних систем (друкарський текст) або кишенькових комп'ютерів перекладачів. Об'єктно-орієнтоване програмування виросло з представлення знань у вигляді фреймів, введених американським ученим Мінським. Нейромережеві технології і технології експертних систем успішно застосовують у системах економічного аналізу і прогнозування. Приклади можна було б продовжити.

Терміни та поняття до теми

Штучний інтелект (ШІ) — галузь комп'ютерної науки, що займається автоматизацією розумної поведінки.

СШІ — система штучного інтелекту.

Artificial Intelligence (AI) — штучний інтелект був запропонований в 1956 р. на семінарі в Дортсмудському коледжі (США).

Перцептрон — нейрона модель зорового сприйняття й розпізнавання Ф. Розенблатта.

Айзек Азімов сформулював три принципи створення робототехніки.

Тест Алана Тьюринга запропоновано 1950 року в статті «Обчислювальні машини і розум».

Питання для самоконтролю

1. Дайте пояснення терміну інтелект (intelligence).
2. Які були переломні моментами в розвитку цивілізації, які можна виділити інформаційні революції в історії людства?
3. Наведіть приклади евристичного методу завдань.
4. Поясніть спроможність або не спроможність моделювання мислення людини.
5. Які основні підходи історично склалися в моделюванні ШІ?
6. Дайте пояснення терміну «інтелектуальні системи» та їхнього місця у США.
7. Які соціальні результати інтелектуалізації комп'ютерних технологій?
8. Які є підходи в моделюванні інтелектуальної діяльності людини?
9. Які є заперечення з приводу можливості моделювання життєвих процесів психіки і розуму?
10. Наведіть визначення предмета теорії штучного інтелекту, яке було введено видатним дослідником у галузі штучного інтелекту Марвіном Мінським.
11. Опишіть процедуру тесту Алана Тьюринга.
12. Які були заперечення проти тесту Алана Тьюрінга?
13. Який сенс в інтелектуалізації комп'ютерних технологій?
14. Назвіть декілька відомих вам інтелектуальних задач; опишіть, у чому вони подібні, і в чому відмінні.
15. Дайте пояснення за класифікацією Ханта напрямів досліджень у галузі штучного інтелекту.

16. У чому полягає основна проблема під час автоматизації логічних побудов?

17. Чи є логічне мислення людини бездоганним?

18. Чи можна дати повне й загальноприйнятне визначення інтелекту взагалі і штучного інтелекту зокрема?

19. Сфери застосування ШІ.

20. Охарактеризуйте біологічний та прагматичний напрями досліджень у галузі штучного інтелекту.

21. Дайте загальну характеристику символного та конекціоністського підходів до створення систем штучного інтелекту.

22. Три закони роботехніки Айзека Азімова.

23. Які функції спілкування з людьми має Asimo зразка 2007 року?

Завдання для індивідуальної роботи, обов'язкові та додаткові практичні завдання

1. Охарактеризуйте відомі методи оцінки інтелектуальних здібностей. Що між ними спільного та в чому полягає відмінність?

2. У чому полягають основні проблеми застосування інтелектуальних тестів для оцінки штучного інтелекту?

3. Чи зводиться процес отримання нових знань, зокрема процес виведення нових фактів з уже існуючих, до суто логічного виведення?

4. Охарактеризуйте роль інтуїції в розумовій діяльності людини.

5. Охарактеризуйте основні завдання, які повинна вміти виконувати інтелектуальна система.

6. Поясніть, яким чином сприйняття зовнішньої ситуації може впливати на внутрішній стан і знання інтелектуальної системи. Наведіть декілька прикладів.

7. Охарактеризуйте позитивні та негативні соціальні наслідки інтелектуалізації комп'ютерних технологій.

8. Наведіть відомі вам приклади практичних досягнень у створенні систем штучного інтелекту.

9. У чому полягають основні проблеми застосування інтелектуальних тестів для оцінки штучного інтелекту?

10. Узагальнена структура інформаційної системи інтелектуального робота.

Тема 2. ЗАГАЛЬНА ХАРАКТЕРИСТИКА МОДЕЛЕЙ ПОДАННЯ ЗНАНЬ

Перелік знань та навичок

Після опанування теми студент **має знати:**

- визначення, класифікацію та властивості знань;
- сучасні моделей подання знань;
- логічну форму моделей знань та опис формальної системи;
- числення висловів;
- числення предикатів;
- побудову прямого та зворотного висновку в продукційних системах;
- асоціативну модель знань;
- структуру семантичної мережі та фреймову модель знань.

Має **вміти:**

- побудувати ППФ;
- написати двозначну логічну функцію — предикат;
- визначити безліч базових елементів (алфавіт) числення предикатів;
- видобувати знання за допомогою правил-продукцій;
- побудувати асоціативну, семантичну, фреймову моделі знань.

Знання є сукупністю відомостей про об'єкти, їх суттєві властивості і відносини, що їх зв'язують, процеси, що протікають, а також методи аналізу ситуацій, які виникають і способи розв'язання асоційованих з ними проблем.

Враховуючи широчінь подібного тлумачення знань, коротко розглянемо інші тлумачення цієї базової категорії.

Поняття «знання» можна вважати одним із ключових з погляду і теорії штучного інтелекту, і гносеології. Саме філософія намагається дати повну картину, повне пояснення природи того або іншого поняття.

Проблема подання знань виникла як одна з проблем штучного інтелекту. Вона пов'язана з переходом досліджень у цій галузі в деяку нову фазу. Йдеться про створення практично корисних систем (перш за все так званих експертних систем), використовуваних у медицині, геології, хімії. Створення подібних систем вимагає інтенсивних зусиль з формалізації знання, накопиченого у відповідній науці.

Поширеним аспектом класифікації знань є їх поділ на декларативні (факти, відомості описового характеру), процедурні (інформація про способи вирішення типових завдань) і метазнання.

Джерелами знань є стислі й повні описи суті. Всіх їх можуть використовувати при формалізації знань, тобто подання їх з використанням певної формальної моделі знань, прийнятної для апаратно-програмних реалізацій.

Будемо казати, що знання про відповідні факти або поняття поле знань зберігає вербально, якщо використання фіксованого набору процедур, що задані над полем знань, дає змогу за оптимальний час отримати вербальне формулювання деякого факту або вербальний опис поняття.

Людина в своїй діяльності комбінує дедуктивні висновки з рефлекторними рішеннями. Таке комбінування повинно бути характерним і для розвинених інтелектуальних систем. Наприклад, можна організувати паралельне функціонування механізмів дедуктивного виведення і конекціоністських механізмів.

У БЗ прийнято виділяти екстенціональну та інтенціональну частини. Екстенціональною частиною бази знань називаємо сукупність усіх явних фактів. Інтенціональною частиною бази знань називаємо сукупність усіх

правил виведення та процедур, за допомогою яких з наявних фактів можна виводити нові твердження.

Структурне подання знань характеризує відносини фактів, або об'єктів. Структура знань може змінюватися, за рахунок чого вони конкретизуються при описі заданої проблемної галузі. У динамічних БЗ структура знань може змінюватися еволюційно або адаптивно.

Концептуальні властивості знань: внутрішня інтерпретація; структурованість; зв'язність; активність знань; функціональна цілісність; ситуативність, незалежність, спільність; однорідність.

Подання знань — це спосіб формального подання знань про наочну галузь у комп'ютерні формі, що інтерпретується.

Моделлю знань називають фіксовану систему формалізмів (понять і правил), відповідно до яких інтелектуальна система подає знання в своїй пам'яті та здійснює операції над ними.

Ключові вимоги до моделей знань:

- 1) спільність (універсальність);
- 2) «психологічна» наочність подання знань;
- 3) однорідність;
- 4) реалізація в моделі властивості активності знань;
- 5) відвертість БЗ.

Логічні моделі реалізуються засобами логіки предикатів. У цьому випадку знання про наочну галузь подаються у вигляді сукупності логічних формул.

Алфавіт числення предикатів утворюється за рахунок додавання до алфавіту числення висловів символів предикатів, предметних змінних і констант, а також кванторів спільності й існування.

Елементарне твердження складається з предиката і поєднаних з ним термів. Складні твердження будуються з елементарних за допомогою логічних зв'язок.

Прямий висновок починається з приведення початкових даних розв'язуваної задачі, які фіксуються у вигляді фактів у робочій пам'яті системи. Правила, використані до початкових даних, забезпечують генерацію нових фактів, що додаються в робочу пам'ять. Процес триває, поки не буде отримано цільове збагачення робочої пам'яті.

Окрім прямого висновку, в продукційних системах широко застосовується і зворотний висновок, тобто висновок, керований цільовими умовами. Такий висновок починається з цільового твердження, яке фіксується в робочій пам'яті.

Безумовно, можливі й змішані стратегії висновку, коли об'єднуються обидва види пошуку — пошук у ширину і пошук у глибину, а також різні евристичні стратегії, оскільки ці стратегії в продукційних системах пов'язані з правилами вирішення конфліктів,

Існують стратегії керування розв'язанням конфліктів, мета яких — уникнути експоненційного зростання.

В основі асоціативної моделі знань лежить асоціативна логіка, нейродинаміка і когнітологія. Загальним для асоціативного підходу є подання знань у вигляді асоціативної мережі вузлових елементів, що мають конекції між собою відповідно до розв'язуваної задачі.

Семантичні мережі є найбільш широким класом мережевих моделей, у яких поєднані різні типи зв'язків. Часткові випадки семантичних мереж відповідають спеціальним типам зв'язків. Виділяють, зокрема, класифікаційні мережі, функціональні мережі та сценарії.

Фрейм — це мінімальний опис деякої сутності, такий, що подальше скорочення опису призводить до втрати цієї сутності. Фрейм будь-якого поняття може бути утворений шляхом об'єднання всіх бінарних фактів, пов'язаних з цим поняттям.

Терміни та поняття до теми

ПРГ — предметна або проблемна галузь формалізації класу типових проблем, яка дає змогу казати про доцільність виділення знань у базу знань інформації (БЗ);

ЕС — експертна система;

СУБД — система управління (керування) базами даних;

СУБЗ — система управління базою знань;

ІС — інформаційна система;

ФС — формальна система — це спосіб формального подачі знань ПРГ в комп'ютерний формі;

ППФ — правильно побудована формула;

предикат — неоднорідна двозначна логічна функція від будь-якого числа аргументів;

ОП — обчислення предикатів;

дедуктивний висновок, тобто висновок за якого із заданої системи посилянь за допомогою фіксованого набору правил формуються власні висновки;

атоми — елементарні формули;

modus ponens і generalization — основні правила висновку в логіці предикатів;

предикат — логічна функція, яка приймає одне з двох можливих значень — «істина» або «хибність»;

фраза Хорна — диз'юнкція довільної кількості атомарних формул; з яких позитивною є не більш ніж одна;

продукція — правила висновку з логічних моделей.

Питання для самоконтролю

1. Охарактеризуйте поняття «знання».
2. Які види знань зазвичай виділяють?
3. У чому полягає відмінність між декларативними і процедурними знаннями?

4. Що таке евристики? Для чого вони використовуються?
5. Що належить до концептуальних властивостей знань? Поясніть кожну властивість.
6. Що таке розмиті квантифікатори?
7. Що становлять опозиційні шкали?
8. У яких типових ситуаціях має виявлятися активність знань?
9. Що розуміють під моделлю знань? Яка її роль?
10. Які існують класи моделей знань?
11. Дайте характеристику логічних моделей знань.
12. Що служить центральною ланкою продукційної моделі знань?
13. Чим відрізняються прямий і зворотний висновки в продукційній моделі знань?
14. Охарактеризуйте фреймову модель знань.
15. Охарактеризуйте мережеві моделі знань.
16. Порівняйте об'єктно-орієнтовані моделі знань MDA і ODP.
17. Сформулюйте основні вимоги до моделей знань.
18. Охарактеризуйте вербально-дедуктивний підхід до опису поняття «знання». Чи є він єдино можливим?
19. Що означають поняття екстенціональної та інтенціональної частин бази знань?
20. Наведіть визначення експертної системи.
21. Що таке псевдофізичні логіки?
22. Охарактеризуйте поняття агрегації.
23. опишіть поняття «клас» та «екземпляр класу».
24. Охарактеризуйте принцип логічного виведення за успадкуванням.
25. Чим відрізняються відношення «екземпляр—клас» та «підклас—клас».
26. Охарактеризуйте «підклас—клас» як відношення часткового порядку.
27. Охарактеризуйте проблему винятків. Як вона пов'язана з монотонністю логічного виведення?

28. Перелічіть галузі знань, що застосовують для діалогових експертних систем.
29. Яким чином можна розкласти деякий складний предикат на бінарні предикати? Наведіть приклади.
30. Яким чином можна перетворити унарний предикат на бінарний? Наведіть приклади.
31. Охарактеризуйте поняття «об'єкт-атрибут-значення».
32. Охарактеризуйте визначення фрейму як структури даних для опису певного поняття.
33. Що означає «мінімальність» опису у визначенні фрейму, що було дано Мінським?
34. Яким чином можна описати об'єкт на основі фреймової моделі?
35. Яким чином описи конкретних об'єктів утворюються з описів відповідних фреймів?
36. Опишіть схему поповнення первинних описів на основі фреймових моделей.
37. Що таке приєднана процедура? Що таке процедура «демон»?
38. Що таке мережі подібностей та відмінностей? Опишіть їх можливі застосування?
39. У чому полягає зв'язок між семантичними мережами і фреймами?
40. Що таке сценарій?
41. Опишіть деяку послідовність подій з деякої предметної галузі у вигляді сценарію, що залежить від ролей та змінних.
42. Поясніть зв'язок між фреймовими моделями та об'єктно-орієнтованим моделюванням і програмуванням.
43. Перелічіть та охарактеризуйте основні властивості об'єктів та класів.
44. Дайте загальну характеристику мови UML.
45. Що таке онтологія процесу (ієрархія понять та атрибутний склад)?

Завдання для індивідуальної роботи, обов'язкові та додаткові практичні завдання

1. Охарактеризуйте основні проблеми, які потрібно розв'язувати при розробці експертних систем і баз знань. Наведіть приклади.
2. Чому знання експерта важко формалізувати, тобто подати ці знання у вигляді, придатному для занесення до бази знань?
3. Наведіть приклад бази знань з довільної предметної галузі, що включає до свого складу факти та правила.
4. Наведіть власний приклад бази даних, яку можна розкласти на інтенціональну та екстенціональну частини.
5. Чому фреймові структури відіграють важливу роль у розумінні, зокрема, текстів?
6. Яким чином відношення узагальнення, асоціації та залежності можуть бути реалізовані в конкретних мовах програмування?

Тема 3. ЗАГАЛЬНА ХАРАКТЕРИСТИКА ДЕДУКТИВНОГО ТА ІНДУКТИВНОГО ВИСНОВКУ

Перелік знань та навичок

Після опанування теми студент має знати:

- використання дедуктивних та індуктивних схем висновку в системах штучного інтелекту;
- дедуктивний висновок методом Ербрана;
- використання законів де Моргана;
- правила побудови пропозицій-диз'юнктивів Хорна;
- методи написання програми семантичної мережі та створення реляційної бази знань мовою Prolog;
- методи планування переміщень робота та алгоритм переміщення мурахи у просторі.

Має **вміти**:

- побудувати універсум Ербрана з константних термів;
- написати мовою Prolog конкретний список, що складається з елементів списку;
- написати мовою Prolog спеціальні засоби для організації баз фактів;
- написати програму семантичної мережі на прикладі сімейних відносин мовою Prolog;
- написати діалогову програму для реляційної бази знань, яка здійснює автоматичну вибірку та демонструє інтерфейс розмовної мови;
- використовувати дистрибутивний закон для приведення виразу до кон'юнктивної нормальної форми.

Розглянуто загальні процедури, що дають змогу автоматизувати дедуктивний висновок у численні предикатів, і передусім процедуру резолюції.

Індуктивні схеми висновку є елементом підсистем штучного інтелекту, що навчаються, оскільки під час навчання із сукупності фактів формуються шляхом узагальнення нові поняття і правила.

Індукцією називають висновок, який становить знання про клас предметів, отриманий у результаті дослідження окремих представників цього класу.

Повною індукцією називають загальний висновок, який виводиться на підставі вивчення всіх наявних фактів.

Універсум Ербрана будується з константних термів.

Приведення формул множини $\Phi_0 \cup \overline{F_s}$ до виду пропозицій виконують у процесі вказаних тотожних перетворень: виключення знаків імплікації й еквіваленції; зменшення сфери дії операції заперечення; стандартизація змінних; виключення кванторів існування; переміщення кванторів спільності; приведення до кон'юнктивної нормальної форми; виключення кванторів спільності; виключення кон'юнкцій.

Терміни та поняття до теми

НЗУ — найбільш загальний уніфікатор;

універсум Ербрана — становить таку сферу інтерпретації H , що якщо не існує інтерпретації множини пропозицій S в цій сфері, то її не існує взагалі;

літерал — елементарна формула або її заперечення;

пропозиція, або клозом (від англ. clause) — диз'юнкція літералів;

фундаментальний диз'юнкт, який не містить змінних.

Питання для самоконтролю

1. Використання дедуктивних та індуктивних схем висновку в системах штучного інтелекту.
2. Наведіть приклади дедуктивного висновку методом Ербрана.
3. Наведіть приклади елементарних формул або їх заперечень, пропозицій, клозів.
4. Наведіть приклади використання законів де Моргана.
5. Які принципи стратегії пошуку?
6. Як пояснити поняття диз'юнкт Хорна?
7. Поясніть принцип резолюції та процедуру висновку.
8. Який висновок називають індуктивним та дедуктивним, за яким принципом руху думки вони будуються?
9. Яка мета узагальнення як процесу отримання знань?

Завдання для індивідуальної роботи, обов'язкові та додаткові практичні завдання

1. Індуктивні схеми висновку.
2. Метод Ербрана.
3. Диз'юнкція літералів.
4. Закони де Моргана.
5. Стратегії пошуку.
6. Процедура виключення кванторів існування.

7. Використання дистрибутивного закону для приведення до кон'юнктивної нормальної форми.
8. Універсум Ербрана та константні терми.
9. Диз'юнкт Хорна та пропозиції.

Тема 4. ВИСНОВКИ В УМОВАХ НЕНАДІЙНИХ АБО НЕПОВНИХ ЗНАНЬ

Перелік знань та навичок

Після опонування теми студент **має знати:**

- види невизначеності;
- Баєсовський підхід до достовірності кожного з фактів бази знань;
- метод коефіцієнтів упевненості, застосований в експертній системі “Super Finance”;
- положення теорії Демпстера—Шефера (ТДШ);
- положення теорії нечітких, або розпливчастих, множин Лотфі Заде.

Має **вміти:**

- розраховувати коефіцієнти упевненості в експертних системах;
- застосовувати правила нечіткої логіки й функцій належності при моделюванні та створенні програмного забезпечення, яке використовує нечіткі або розпливчасті значення;
- визначити складність систем, які використовують нечітку логіку.

Теорія числення, предикатів дає змогу одержувати правильні висновки, спираючись на загальнозначущі твердження і обґрунтовані правила висновку

У багатьох випадках є потреба робити висновок в умовах, коли початкові дані не є абсолютно точними й достовірними, а правила висновку мають евристичний характер і ненадійні.

Невизначеність може бути пов'язана або з неповнотою знань, або з їх неоднозначністю.

Часто всі види неоднозначності не виділяють в окремі групи, а розглядають у межах одного терміна “ненадійні знання”. Основними поняттями, що використовують під час побудови моделей висновку на таких знаннях, є поняття достовірності.

Неповнота знань виникає, коли зібрано не всю інформацію, необхідну для побудови висновків

Коли істинність висловів не може бути встановлена з абсолютною достовірністю — це є неоднозначність.

Фізична невизначеність може бути пов’язана з випадковістю подій, ситуацій, станів об’єкта або неточністю подання даних.

Лінгвістична невизначеність пов’язана з використанням природної мови для подання знань.

Терміни та поняття до теми

“Mycin”, “Prospector”, “Super Finance”, “Emycin” — експертні системи;

Fuzzy logic — нечітка логіка;

JTMS (justification based TMS) — система підтримки значень істинності, що використовує обґрунтування;

ATMS (assumption based TMS) — система підтримки значень істинності, що використовує припущення.

СПЗІ — системи підтримки значень істинності.

Питання для самоконтролю

1. Охарактеризуйте поняття невизначеності.
2. Дайте визначення баєсовському підходу ступеня достовірності
3. Охарактеризуйте теорему гіпотез Баєса.
4. Яка теорія використовується для формалізації нечітких знань, що характеризуються невизначеністю?

5. Що таке лінгвістична змінна, значеннями якої є слова і фрази природної мови (терми)?
6. Поясніть формальну систему немонотонного висновку, розширивши логіку числення предикатів.
7. Дайте визначення нечіткої множини. Наведіть кілька власних прикладів.
8. Дайте визначення рівності та включення нечітких множин.
9. Охарактеризуйте завдання, пов'язані з нечітким логічним виведенням.
10. Опишіть метод простого підставлення нечіткого значення, який може бути застосований у разі, якщо в умові вочевидь задається ступінь належності.
11. Що таке нечітке відношення?

Завдання для індивідуальної роботи, обов'язкові та додаткові практичні завдання

1. Метод коефіцієнтів упевненості.
2. Підхід, прийнятий в теорії Демпстера—Шефера.
3. Шанси — це інша шкала для представлення вірогідності.
4. Види невизначеності.
5. Баєсовський підхід як ступінь достовірності кожного з фактів бази знань.
6. Теорема гіпотез Баєса.
7. Відношення правдоподібності.
8. Метод коефіцієнтів упевненості, застосований в експертній системі “Super Finance”.
9. Підхід, прийнятий у теорії Демпстера-Шефера, та відмінність від баєсовського підходу й методу коефіцієнтів упевненості.
10. Застосування теорії нечітких, або розпливчастих множин.
11. Опишіть функції пакету Matlab для створення нейро-нечітких мереж.

Тема 5. КОНЕКЦІОНІСТСЬКИЙ ПІДХІД ЯК СПРОБА МОДЕЛЮВАННЯ ЛЮДСЬКОГО МОЗКУ

Перелік знань та навичок

Після опанування теми студент має знати:

- Визначення штучної нейронної мережі (ШНМ).
- Історичне систематичне вивчення ШНМ.
- Характерні риси штучних нейромереж як універсального інструменту

для розв'язання задач

- Конекціоністський підхід у побудові систем штучного інтелекту.

- Стандарти класифікації нейромереж.

- Структура одношарової нейронної мережі.

- Структура багатшарового персептрона, або багатшарової нейронної мережі

- Структура двошарової мережі зворотного поширення.

- Нейронні мережі Хопфілда, Хеммінга, Больцмана

- Процедури навчання мережі.

Має вміти:

- Вибирати функції активації персептрона.

- Будувати схему нейрокомп'ютера — програмно-технічної системи, яка реалізує деяку формальну модель природної мережі нейронів.

- Вибирати кількість нейронів і шарів.

- Описати алгоритм навчання одношарового дискретного персептрона.

- Будувати БШП та вибрати його параметри.

Першим систематично досліджували штучні нейронні мережі Маккалокк і Піттс 1943 р.

Штучна нейронна мережа — система, яка складається із певної кількості пристроїв, так званих штучних нейронів, з'єднаних між собою.

Нейрон є основним будівельним блоком нервової системи. Він є клітиною, подібною до всіх інших клітин тіла.

Конекціоністський підхід до побудови систем штучного інтелекту розвинувся на протигагу символному. В основі конекціоністського підходу лежить спроба безпосереднього моделювання розумової діяльності людського мозку.

Мислення людини є переважно асоціативним, і в основу роботи систем, які моделюють мислення людини, може бути покладений принцип асоціації. Під асоціацією розуміємо певний ступінь подібності або наявність певних спільних рис між тими чи іншими явищами й поняттями.

На цей час не існує єдиної стандартної класифікації НМ, тому використовується класифікація за деякими базовими характеристиками.

Немає суворо визначеної процедури для вибору кількості нейронів і кількості шарів у мережі. Що більша кількість нейронів і шарів, то ширше можливість мережі, тим повільніше вона навчається і працює і тим більше нелінійної може бути залежність вхід-вихід.

Одна з найважливіших властивостей НМ — здатність до самоорганізації й самоадаптації з метою поліпшення якості функціонування.

Обравши одну з моделей ШНМ, створивши мережу й виконавши алгоритм навчання, ми можемо навчити мережу розв'язувати задачу. Після навчання мережа стає нечутливою до малих змін вхідних сигналів і дає правильний результат на виході.

Якщо пред'явити мережі кілька змінених варіантів вхідного образу, то мережа сама може створити на виході ідеальний відтворений образ, з яким вона ніколи не зустрічалася.

Терміни та поняття до теми

Штучний нейрон — ШН.

Штучні нейронні мережі (ШНМ) — сукупність моделей біологічних нейронних мереж, набір математичних й алгоритмічних методів для розв’язання широкого кола задач.

Дендріти є структурою, що рясно гілкується від тіла клітини. На дендритах розташовуються синаптичні з’єднання, які отримують сигнали від інших аксонів.

Штучний нейрон імітує в першому наближенні властивості біологічного нейрона.

Активаційна функція застосовується для тих моделей мереж, де не потрібно послідовне з’єднання шарів нейронів один за одним.

БНМ — багат шарові нейромережі.

Навчання — обравши одну з моделей ШНМ, створивши мережу і виконавши алгоритм навчання, ми можемо навчити мережу розв’язанню задач.

Здібність узагальнення — після навчання мережа стає нечутливою до малих змін вхідних сигналів і дає правильний результат на виході.

Здібність абстрагування — якщо надати мережі кілька змінених варіантів вхідного образу, мережа сама може створити на виході ідеальний образ, з яким вона раніше не зустрічалася.

BrainMaker, OWL, «The AI Trilogy», Neural10, NeuroPro — програмні пакети, які моделюють роботу нейронних мереж на універсальних комп’ютерах.

Питання для самоконтролю

1. Поясніть структуру штучної нейронної мережі як набір математичних і алгоритмічних методів для розв’язання широкого кола задач.
2. Покажіть характерні риси штучних нейромереж як універсального інструменту для розв’язання задач
3. Що таке конекціоністський підхід у побудові систем штучного інтелекту та чим відрізняється від символічного?
4. Які системи (та багато їм подібних) отримали назву персептронів?

5. Поясніть принцип асоціації в мисленні людини.
6. Вибір функції активації визначається:
7. Назвіть стандартну класифікацію НМ.
8. Яка схема одношарової нейронної мережі?

Завдання для індивідуальної роботи, обов'язкові та додаткові практичні завдання

1. Що є характерним для сучасних моделей знань?
2. Моделювання розумової діяльності людського мозку.
3. Конекціоністський підхід як спроба безпосереднього моделювання розумової діяльності людського мозку.
4. Асоціативне мислення людини.
5. Нейрокомп'ютер як програмно-технічна система.
6. Двошарова мережа зворотного поширення.
7. Багатошаровий персептрон або багатошарова нейронна мережа.
8. Приклади навчання одношарового дискретного персептрона.
9. Радіально-базисні нейронні мережі.
10. Алгоритм функціонування мережі Хеммінга.
11. Структурна схема мережі Хопфілда.
12. Больцмановське навчання — стохастичний метод, що безпосередньо застосовують для навчання штучних нейронних мереж.

Тема 6. НАВЧАННЯ НЕЙРОННОЇ МЕРЕЖІ

Перелік знань та навичок

Після опанування теми студент **має знати:**

- активаційні функції порогового елемента;
- основний принцип навчання модельних нейронів;
- класифікацію навчальних алгоритмів;
- алгоритм навчання Кохонена, Коші, Хебба;
- особливості алгоритмів навчання персептрона.

Має **вміти**:

- побудувати алгоритм навчання нейронної мережі за допомогою процедури зворотного поширення

- моделювати нейронні мережі за допомогою пакета Matlab.

Штучна нейронна мережа навчається за допомогою певного процесу, що модифікує її ваги.

Алгоритм називають алгоритмом з учителем, якщо під час навчання відомі і вхідні, і вихідні вектора мережі. Разом їх називають навчальною парою.

Алгоритм належить до навчання без учителя, якщо відомі тільки вхідні вектора, і на їх основі мережа вчиться давати якнайкращі значення виходів.

Зворотне поширення — це метод для навчання багат шарових штучних нейронних мереж.

Терміни та поняття до теми

Параліч мережі — в певних умовах мережа може під час навчання потрапити в такий стан, коли модифікація вагів не веде до справжніх змін мережі.

Backpropagation algorithm — алгоритм зворотного поширення.

Neurocontroller — нейтроконтролер, зазвичай називають комп'ютерні системи, побудовані на основі ШНМ.

Запитання для самоконтролю

1. Опишіть модель нейрона як пороговий елемент.
2. Що таке активаційна функція порогового елемента?
3. У чому полягає основний принцип навчання модельних нейронів?
4. Опишіть процедуру Уїдроу для навчання модельних нейронів.
5. Які функції, крім порогових, можуть використовуватися як активаційні?

6. Що таке сигмоїдальна функція? Наведіть приклад.
7. Опишіть узагальнену дельта-процедуру для сигмоїдальної активаційної функції.
8. У чому полягають основні принципи використання методу зворотного поширення помилки для навчання нейромереж?
9. Як визначити апріорну інформацію про навчальну вибірку?
10. Як впливає задана кількість циклів навчання на точність навчання?
11. Як впливає задана точність навчання на тривалість навчання?
12. Які вимоги потрібно висувати до навчальної вибірки, та як це вплине на процес навчання?

Завдання для індивідуальної роботи, обов'язкові та додаткові практичні завдання

1. Параліч мереж.
2. Метод сигнального навчання Хебба.
3. Метод диференціального навчання Хебба.
4. Алгоритм навчання Кохонена.
5. Алгоритм навчання Коші Хебба.
6. Комбінування зворотного поширення з навчанням Коші.
7. Процедура зворотного поширення.
8. Пастки локальних мінімумів.
9. Моделювання нейронних мереж за допомогою пакета Matlab.

Тема 7. ОСНОВНІ ПОНЯТТЯ ТЕОРІЇ РОЗПІЗНАВАННЯ ОБРАЗІВ

Перелік знань та навичок

Після опанування теми студент має знати:

- класифікацію систем розпізнавання;
- деякі практичні галузі застосування систем розпізнавання;
- класи та їх властивості;
- метричні методи, засновані на кількісній оцінці цієї близькості;

- гіпотезу компактності;
- лінійну роздільність класів;
- типову схему розпізнавання в просторі ознак.

Має **вміти**

- розраховувати відстань між об'єктами різними методами:
- відстань Хеммінга (відстань за Манхеттенем, метрика міських кварталів);
- Евклідову відстань;
- відстань Мінковського (узагальнену відстань);
- діагностичну міру відстані;
- відстань у неізотропному просторі ознак;
- узагальнену відстань у просторі ознак;
- відстань у нелінійному просторі;
- узагальнену (зважену) відстань Махалонобіаса;
- відстань Камберра;
- відстань Чебишева;
- кореляційний метод знаходження відстані;
- кутову відстань.

Розпізнавання образів — процес, за якого на підставі численних характеристик (ознак) деякого об'єкту визначаються одна або кілька найбільш суттєвих, але недоступних для безпосереднього визначення його характеристик, зокрема, його приналежність до певного класу об'єктів.

Класифікація систем розпізнавання ґрунтується на застосуванні принципу властивості інформації, яку використовують у процесі розпізнавання.

Системи розпізнавання можна поділити на прості й складні залежно від того, фізично однорідна або неоднорідна інформація використовується для опису об'єктів, які потрібно розпізнати чи мають вони ознаки, тієї мови, якою здійснено опис алфавіту класів, чи мають вони одну або різну фізичну природу.

Об'єктом теорії розпізнавання зазвичай називають будь-яку сутність, що існує або могла б існувати в реальному світі, а також будь-яке явище або процес.

Основна відмінність між різними групами методів розпізнавання полягає в тому, яким чином формуються модельні описи об'єктів.

У більшості методів розпізнавання робиться природне припущення, що зображення об'єктів одного класу (образу) ближчі одне одному, аніж зображення різних класів.

Метричні методи засновані на кількісній оцінці цієї близькості. Як зображення об'єкта приймається точка в просторі ознак, ступенем близькості вважається відстань між точками.

Терміни та поняття до теми

Завдання ідентифікації полягає в тому, щоб вирізнити певний конкретний об'єкт серед йому подібних.

Віднесення об'єкта до певного класу, наприклад, завдання розпізнавання літер або прийняття рішення про наявність дефекту у деякій технічній деталі.

Кластерний аналіз полягає в поділі заданого набору об'єктів на класи, які подібні між собою за тим чи іншим критерієм.

Класом у теорії розпізнавання образів зазвичай називають сукупність об'єктів, які мають якісь спільні ознаки.

Метрика — правило, за допомогою якого вводиться відстань між елементами простору.

Гіпотеза компактності — відповідно до якої класові відповідає компактна множина точок у деякому просторі ознак.

Запитання для самоконтролю

1. Охарактеризуйте класифікацію систем розпізнавання
2. Які вам відомі приклади практичного застосування систем розпізнавання?

3. Метричні методи, засновані на кількісній оцінці цієї близькості.
4. Наведіть типову схему розпізнавання в просторі ознак.
5. Чи можна для будь-якої предметної галузі використовувати розпізнавання в просторі ознак? Наведіть приклади.
6. Охарактеризуйте дискримінантні методи розпізнавання.
7. Що таке простір ознак?
8. Перерахуйте відомі вам типи ознак.
9. Які ви знаєте ступені близькості між об'єктами, якщо ознаки є кількісними?
10. Наведіть формулу евклідової відстані між векторами.
11. Що таке зважені евклідові відстані? В якому випадку їх доцільно використовувати?
12. Охарактеризуйте гіпотезу компактності.
13. Опишіть типову схему розпізнавання в просторі ознак.
14. Охарактеризуйте баєсівські методи розпізнавання образів.
15. Сформулюйте метод найближчого сусіда.
16. Чи вимагає метод найближчого сусіда попереднього вироблення вирішального правила?
17. Охарактеризуйте основні переваги й недоліки методу найближчого сусіда.
18. Опишіть стиснуте правило найближчого сусіда. З чим пов'язана доцільність його застосування?

Теми для рефератів та обговорення

1. У чому полягають основні проблеми під час створенні штучних систем розпізнавання?
2. Опишіть відомі вам постановки завдання класифікації.
3. Перерахуйте відомі вам практичні застосування методів класифікації.
4. Наведіть ключову парадигму теорії розпізнавання.
5. Що таке клас у теорії розпізнавання?

6. Що таке інваріанти класу?
7. Що таке інформативні ознаки?
8. Перерахуйте основні властивості класів.
9. Охарактеризуйте можливі напрями розвитку діалогових людинно-машинних систем та систем автоматизованого перекладу.
10. Що змінюється при переході від нижчого рівня розуміння до вищого?
11. Наведіть приклади речень, які можна перекласти з однієї мови на іншу без розуміння змісту цих речень.

Додаток

Алгоритми переміщення мурахи

У цьому додатку розглядається цікавий алгоритм, що базується на використанні кількох агентів, за допомогою яких можна розв'язати найрізноманітніші задачі. *Алгоритми мурахи* (Ant algorithms), або оптимізація за принципом мурашиної колонії (ця назва була вигадана винахідником алгоритму Марко Доріго (Marco Dorigo)), мають специфічні властивості, притаманні мурахам, і використовують їх для орієнтації у фізичному просторі. Природа пропонує різні методики для оптимізації деяких процесів. Алгоритми мурахи особливо цікаві тому, що їх можна використовувати для розв'язання не лише статичних, а й динамічних проблем, наприклад, проблем маршрутизації в сітках, що змінюються.

Мурахи хоча й сліпі, вони вміють переміщатися в складних місцевостях, знаходити харчі на великій відстані від мурашника та успішно повертатися додому. Виділяючи ферменти під час переміщення, мурахи змінюють навколишнє середовище, забезпечують комунікацію, а також шукають і знаходять зворотний шлях до мурашника.

Найдивовижніше в цьому процесі — це те, що мурахи вміють знаходити найоптимальніший шлях між мурашником та навколишніми пунктами. Що ближча зовнішня точка до мурашника, то більше разів до неї переміщалися мурахи. Що стосується віддаленішого пункту, то до нього мурахи дістаються рідше, тому дорогою до нього вони застосовують сильніші ферменти. Що вища

концентрація ферментів на шляху, то більшу перевагу віддають йому мурахи порівняно з іншими доступними. Так мурашина «логіка» дає змогу обирати коротший шлях між кінцевими пунктами.

Алгоритми мурахи цікаві, оскільки мають ряд специфічних якостей, притаманних самим мурахам. Мурахи легко йдуть на компроміс і співпрацю один з одним заради досягнення спільної мети. Це має прояв у тому, що змодельовані мурахи розв'язують проблему та допомагають у подальшій її оптимізації.

Розглянемо приклад, який наведено на рис. 1. Дві мурахи з мурашника мають дістатися їжі, яка знаходиться за перешкодою. Під час переміщення кожна мураха виділяє небагато ферменту, використовуючи його як маркер.

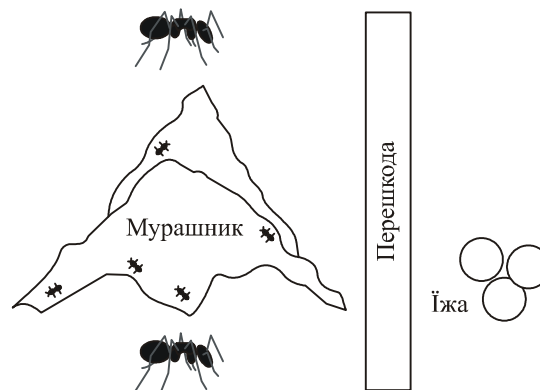


Рис.1. Початкова конфігурація (t_0)

За інших рівних умов кожна мураха обирає свій шлях. Перша обирає верхній шлях, а друга — нижній. Оскільки нижній шлях у два рази коротший за верхній, друга мураха досягне цілі за час t_1 . Перша мураха в цей момент подолає лише половину шляху (рис. 2).

Коли мураха дістається їжі, вона бере один з об'єктів і повертається до мурашника тим самим шляхом. За час t_2 друга мураха повернулася до мурашника з їжею, а перша тільки дістається їжі (рис. 2).

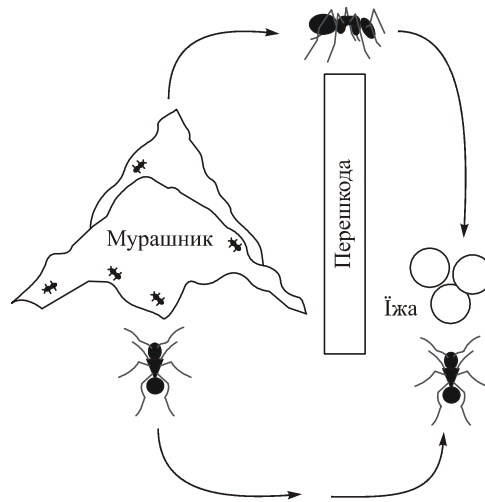
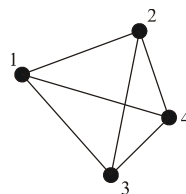


Рис. 2. Минув один період часу (T_1)

Згадайте, що під час переміщення кожної мурахи на дорозі залишається трохи ферменту. Для першої мурахи за час $t_0 - t_2$ шлях був вкритий ферментом тільки один раз. Водночас друга мураха вкрила шлях ферментом двічі. За час t_4 перша мураха повернулася до мурашника, а друга встигла ще раз сходити до їжі й повернутися. При цьому концентрація ферменту на нижній дорозі буде в два рази вища, ніж на верхній. Тому перша мураха наступного разу обере нижній шлях, оскільки там концентрація ферменту вища.

У цьому й полягає базова ідея алгоритму мурахи — оптимізація шляхом непрямого зв'язку між автономними агентами.

Припустимо, що навколишнє середовище для мурахи презентує закриту двовимірну мережу. Згадайте, мережа — це група вузлів, що з'єднуються за допомогою меж. Кожна межа має вагу, яку ми позначимо як відстань між двома вузлами, з'єднаними нею. Граф двоспрямований, тому мураха може подорожувати ним в будь-якому напрямку (рис. 3).



Граф з вершинами $V = \{1,2,3,4\}$

Межі $E = \{\{1,2\}, \{1,4\}, \{1,3\}, \{2,3\}, \{2,4\}, \{3,4\}\}$

Рис. 3. Граф двоспрямований

Мураха

Мураха — це програмний агент, котрий є членом великої колонії і використовується для розв'язання будь-якої проблеми. Мураха доповнюється набором простих правил, котрі дають змогу йому обирати шлях у графі. Він підтримує *список табу* (tabu list), тобто список вузлів, котрі він вже відвідав. Таким чином, мураха має проходити через кожен вузол лише один раз. Шлях між двома вузлами графа, за котрим мураха відвідав кожен вузол тільки один раз, називають *шляхом Гамільтона* (Hamiltonian path), за ім'ям математика сера Вільяма Гамільтона (Sir William Hamilton).

Вузли в списку «поточної подорожі» розташовуються в тому порядку, у котрому мураха відвідував їх. Пізніше список використовується для визначення довжини шляху між вузлами.

Справжній мураха під час переміщення шляхом лишатиме за собою фермент. В алгоритмі мурахи агент лишає фермент на межах мережі після завершення подорожі. Про те, як це відбувається, розповідається у розділі «Подорожі мурахи»

Після створення популяція мурах нарівно розподіляється вузлами мережі. Потрібний рівний поділ мурах між вузлами, щоб усі вузли мали однакові шанси стати відправною точкою. Якщо всі мурахи почнуть рух з однієї точки, це буде означати, що дана точка є оптимальною для старту, а насправді, ми цього не знаємо.

Рух мурахи

Рух мурахи засновується на одному і дуже простому ймовірнісному рівнянні. Якщо мураха ще не закінчила шлях (path), тобто не відвідала всі вузли мережі, для визначення наступної межі шляху використовується рівняння:

$$P = \frac{\tau(r,u)^\alpha \times \eta(r,u)^\beta}{\sum_k \tau(r,u)^\alpha \times \eta(r,u)^\beta}.$$

Тут $\tau(r,u)$ — інтенсивність фрагмента на межі між вузлами r та u , $\tau(r,u)$ — функція котра представляє виміри зворотної відстані для межі, α — вага ферменту, а β — коефіцієнт евристики. Параметри α і β визначають відносну значущість двох параметрів, а також їх вплив на рівняння. Згадайте, що мураха подорожує тільки вузлами, котрі ще не були відвідані (як вказано списком табу). Тому ймовірність розраховується тільки для меж, котрі ведуть до ще не відвіданих вузлів. Змінна k є представником меж, котрі ще не були відвідані.

Подорож мурахи

Шлях, який пройшла мураха, відображається, коли мураха відвідає усі вузли діаграми. Зверніть увагу, що цикли заборонені, оскільки до алгоритму включено список табу. Після завершення довжина шляху може бути підрахована — вона дорівнює сумі всіх меж, якими мандрував мураха. Рівняння показує кількість ферменту, що був залишений на кожній межі шляху для мурахи k . Змінна Q є константою:

$$\Delta\tau_{ij}^k(t) = \frac{Q}{L^k(t)}$$

Результат рівняння засобом вимірювання шляху — короткий шлях характеризується високою концентрацією ферменту, а довший шлях — нижчою. Потім отриманий результат використовується, щоб збільшити кількість ферменту вздовж кожної з меж пройденого мурахою шляху:

$$\tau_{ij}(t) = \Delta\tau_{ij}(t) + (\tau_{ij}^k(t) \times \rho) .$$

Зверніть увагу, що це рівняння застосовується до всього шляху, при цьому кожна межа помічається ферментом пропорційно довжині шляху. Тому слід дочекатися, поки мураха закінчить подорож, і тільки після цього оновити рівні ферменту, в іншому разі справжня довжина шляху залишиться невідомою. Константа ρ має значення між 0 і 1.

Випаровування ферменту

На початку шляху в кожній межі є шанс бути обраною. Щоб з часом віддалити межі, що входять у найгірші шляхи в мережі, до всіх меж застосовується процедура *випаровування ферменту* (Pheromone evaporation). Використовуючи константу ρ , ми отримуємо рівняння:

$$\tau_{ij}(t) = \tau_{ij}(t) \times (1 - \rho).$$

Тому для випаровування ферменту використовується зворотний коефіцієнт оновлення шляху.

Після того, як шлях мурахи завершено, межі оновлено відповідно до довжини шляху і відбулося випаровування ферменту на всіх межах, алгоритм запускається вдруге. Список табу очищується, і довжина шляху обнулюється. Мурахам дозволяється переміщуватися мережею, здійснюючи вибір межі. Цей процес може виконуватися для постійної кількості шляхів або до того моменту, коли упродовж декількох запусків не було помічено повторних змін. Потім визначається кращий шлях, що і становить рішення.

Приклад ітерації

Давайте розберемо функціонування алгоритму на простому прикладі, щоб побачити, як працюють рівняння. Згадайте простий сценарій з двома мурахами, які обирають два різних шляхи для досягнення однієї цілі. На рис. 4 наведено цей приклад з двома межами між двома вузлами (v_0 і v_1). Кожна межа ініціалізується та має однакові шанси на те, щоб бути обраною.

Дві мурахи знаходяться у вузлі v_0 і позначаються як A_0 і A_1 . Оскільки ймовірність вибору будь-якого шляху однакова, в цьому циклі ми проігноруємо керування вибору шляху. На рис. 4 кожна мураха обирає свій шлях (мураха A_0 йде верхнім шляхом, мураха A_1 — нижнім).

Мураха A_0 зробила 20 кроків, а мураха A_1 — лише 10. Розраховуємо кількість ферменту, який має бути нанесений. Рівень ферменту Q /подолана відстань для A_0 — 0,4, для A_1 — 0,7.

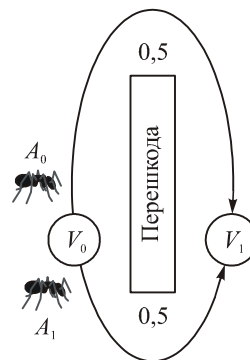


Рис. 4. Навчальна конфігурація проблеми

Роботу алгоритму можна змінити, перерахувавши його параметри (наприклад ρ , α чи β), наприклад придбавши більшу вагу ферменту чи відстані між вузлами. Далі за допомогою рівняння розраховується кількість ферменту, яка буде застосована. Для мурахи A_0 результат такий: $\Delta\tau^1 = 0,1 + (0,5 \times 0,6) = 0,4$, а для мурахи A_1 — $\Delta\tau^2 = 0,1 + (1 \times 0,6) = 0,7$.

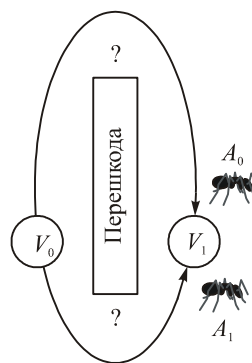


Рис. 5. Шлях мурахи закінчено

Для мурахи A_1 результат становить: $\Delta\tau^2 = 0,1 + (1,0 \times 0,6) = 0,7$. Далі визначається, яка частина ферменту випарується і, відповідно, скільки залишиться. Результати (для кожного шляху):

$$\tau^0 = 0,4 \times (1,0 - 0,6) = 0,16$$

$$\tau^1 = 0,7 \times (1,0 - 0,6) = 0,28.$$

Ці значення становлять нову кількість ферменту для кожного шляху (верхнього і нижнього, відповідно). Тепер перемістимо мурах зворотно у вузол

v_0 і використаємо імовірне рівняння вибору шляху, щоб визначити, який шлях мають обрати мурахи. Імовірність того, що мураха обере верхній шлях, представлений кількістю фермента 0,16, становить:

$$(0,16)^{3,0} \times (0,5)^{1,0} / ((0,16)^{3,0} \times (0,5)^{1,0}) + ((0,28)^{3,0} \times (1,0)^{1,0}) = \\ = 0,002048 / 0,024 = P(0,085).$$

Імовірність того, що мураха обере нижній шлях (який представлений кількістю ферменту 0,28), становить:

$$(0,28)^{3,0} \times (1,0)^{1,0} / ((0,16)^{3,0} \times (0,5)^{1,0}) + ((0,28)^{3,0} \times (1,0)^{1,0}) = \\ = 0,021952 / 0,024 = P(0,915).$$

При порівнянні двох імовірностей обидві мурахи оберуть нижній шлях, який є найбільш оптимальним.

Приклад задачі

Як приклад розглянемо задачу комівояжера (Traveling Salesman Problem — TSP). Вона полягає в тому, щоб знайти найкоротший шлях між містами, за якого кожне місто відвідуватиметься лише один раз. Тобто треба знайти найкоротший Гамільтонів шлях у графі, де вузлами виступають міста, а межами — шляхи, що їх з'єднують. Математики вперше вивчили задачу TSP у 1930-ті рр., зокрема, нею займався Карл Менгер (Karl Menger) у Відні. Слід зазначити, що на Сході задачі досліджував ще у 19 ст. ірландський математик сер Уільям Ромен Гамільтон (Sir William Hamilton).

У наступному розділі розглядається початковий код програми, яка використовується для розв'язання задачі, і наведено варіанти розв'язання.

Початковий код

Лістинги ілюструють алгоритм мурахи, який використовується для пошуку оптимального розв'язання задачі комівояжера.

Спочатку вивчимо структуру даних і для міст, і для агентів (мурах), які ними мандруватимуть. Лістинг включає типи даних і символні константи, як використовують для подання міст і мурах.

Програмный код мовою С++:

Файл Common.h:

```
#ifndef __COMMON_H
#define __COMMON_H
#include <math.h>
#include <stdlib.h>
#define MAX_CITIES 53
#define MAX_ANTS 20
#define MAX_DISTANCE 100
#define MAX_TOUR (MAX_CITIES * MAX_DISTANCE)
typedef struct {
    int x;
    int y;
} cityType;
typedef struct {
    int curCity;
    int nextCity;
    unsigned char tabu[MAX_CITIES];
    int pathIndex;
    unsigned char path[MAX_CITIES];
    double tourLength;
} antType;
#define getSRand() ((float)rand() / (float)RAND_MAX)
#define getRand(x) (int)((x) * getSRand())
#define ALPHA 1.0
#define BETA 5.0
#define RHO 0.5
#define QVAL 100
#define MAX_TOURS 500
#define MAX_TIME (MAX_TOURS * MAX_CITIES)
#define INIT_PHEROMONE (1.0 / MAX_CITIES)
#endif
```

Файл Ant.cpp:

```
#include <vcl.h>
#pragma hdrstop
#include "Ant.h"
#include <stdio.h>
#include <conio.h>
#include <time.h>
#include <assert.h>
#include "Common.h"
#pragma package(smart_init)
#pragma link "Chart"
#pragma link "Series"
#pragma link "TeEngine"
#pragma link "TeeProcs"
#pragma resource "*.dfm"
TMainForm *MainForm;
cityType cities[MAX_CITIES];
antType ants[MAX_ANTS];
double distance[MAX_CITIES][MAX_CITIES];
double pheromone[MAX_CITIES][MAX_CITIES];
double best=(double)MAX_TOUR;
int bestIndex;
void init( void )
{ int from, to, ant;
  for (from = 0; from < MAX_CITIES; from++) {
    cities[from].x = getRand( MAX_DISTANCE );
    cities[from].y = getRand( MAX_DISTANCE );
    for (to = 0; to < MAX_CITIES; to++) {
      distance[from][to] = 0.0;
      pheromone[from][to] = INIT_PHEROMONE; } }
  for (from = 0; from < MAX_CITIES; from++) {
    for (to = 0; to < MAX_CITIES; to++) {
      if ((to != from) && (distance[from][to] == 0.0)) {
        int xd = abs(cities[from].x — cities[to].x);
        int yd = abs(cities[from].y — cities[to].y);
        distance[from][to] = sqrt((xd * xd) + (yd * yd));
        distance[to][from] = distance[from][to]; } } }
  to = 0;
  for (ant = 0; ant < MAX_ANTS; ant++) {
    if (to == MAX_CITIES) to = 0;
```

```

ants[ant].curCity = to++;
for (from = 0; from < MAX_CITIES; from++) {
ants[ant].tabu[from] = 0;
ants[ant].path[from] = -1; }
ants[ant].pathIndex = 1;
ants[ant].path[0] = ants[ant].curCity;
ants[ant].nextCity = -1;
ants[ant].tourLength = 0.0;
ants[ant].tabu[ants[ant].curCity] = 1; } }
void restartAnts( void )
{ int ant, i, to=0;
for (ant = 0; ant < MAX_ANTS; ant++) {
if (ants[ant].tourLength < best) {
best = ants[ant].tourLength;
bestIndex = ant; }
ants[ant].nextCity = -1;
ants[ant].tourLength = 0.0;
for (i = 0; i < MAX_CITIES; i++) {
ants[ant].tabu[i] = 0;
ants[ant].path[i] = -1; }
if (to == MAX_CITIES) to = 0;
ants[ant].curCity = to++;
ants[ant].pathIndex = 1;
ants[ant].path[0] = ants[ant].curCity;
ants[ant].tabu[ants[ant].curCity] = 1; } }
double antProduct( int from, int to )
{ return ((pow(pheromone[from][to], ALPHA) *
pow((1.0 / distance[from][to]), BETA))); }
int selectNextCity( int ant )
{ int from, to;
double denom=0.0;
from = ants[ant].curCity;
for (to = 0; to < MAX_CITIES; to++) {
if (ants[ant].tabu[to] == 0) {
denom += antProduct(from, to); } }
assert(denom != 0.0);
do { double p;
to++;
if (to >= MAX_CITIES) to = 0;
if (ants[ant].tabu[to] == 0) {
p = antProduct(from, to)/denom;
if (getSRand() < p ) break; } } while (1);
return to; }
int simulateAnts( void )
{ int moving = 0, k;
for (k=0; k < MAX_ANTS; k++) {
if (ants[k].pathIndex < MAX_CITIES) {
ants[k].nextCity = selectNextCity(k);
ants[k].tabu[ants[k].nextCity] = 1;
ants[k].path[ants[k].pathIndex++] = ants[k].nextCity;
ants[k].tourLength += distance[ants[k].curCity][ants[k].nextCity];
if (ants[k].pathIndex == MAX_CITIES) {
ants[k].tourLength += distance[ants[k].path[MAX_CITIES-1]][ants[k].path[0]]; }
ants[k].curCity = ants[k].nextCity;
moving++; } }
return moving; }
void updateTrails( void )
{ int from, to, i, ant;
for (from=0; from < MAX_CITIES; from++) {
for (to = 0; to < MAX_CITIES; to++) {
if (from != to) {
pheromone[from][to] *= (1.0 - RHO);
if (pheromone[from][to] < 0.0) pheromone[from][to] = INIT_PHEROMONE;
} } }
for (ant=0; ant < MAX_ANTS; ant++) {
for (i=0; i < MAX_CITIES; i++) {
if (i < MAX_CITIES-1) {
from = ants[ant].path[i];
to = ants[ant].path[i+1]; } else {
from = ants[ant].path[i];
to = ants[ant].path[0]; }
pheromone[from][to] += (QVAL / ants[ant].tourLength);
pheromone[to][from] = pheromone[from][to]; } }
for (from=0; from < MAX_CITIES; from++) {
for (to=0; to < MAX_CITIES; to++) {

```



```

        pheromone[from][to] *= RHO; } }
void emitDataFile( int ant )
{ int city; FILE *fp;
  fp = fopen("cities.dat", "w");
  for (city = 0 ; city < MAX_CITIES ; city++) {
    fprintf(fp, "%d %d\n", cities[city].x, cities[city].y); }
  fclose(fp);
  fp = fopen("solution.dat", "w");
  for (city = 0 ; city < MAX_CITIES ; city++) {
    fprintf(fp, "%d %d\n",
      cities[ ants[ant].path[city] ].x,
      cities[ ants[ant].path[city] ].y ); }
  fprintf(fp, "%d %d\n",
    cities[ ants[ant].path[0] ].x,
    cities[ ants[ant].path[0] ].y );
  fclose(fp); }
void emitTable(void)
{ int from, to;
  for (from = 0 ; from < MAX_CITIES ; from++) {
    for (to = 0 ; to < MAX_CITIES ; to++) {
      printf("%5.2g ", pheromone[from][to]); }
    printf("\n"); }
  printf("\n"); }
void __fastcall TMainForm::FormCreate(TObject *Sender)
{ XYGraph->Series[0]->Clear();
  PathMemo->Lines->Clear();
  TimePathProgressbar->Position=0; }
void __fastcall TMainForm::CalculatePathButtonClick(TObject *Sender)
{ int curTime = 0, city;
  char pathtime [50], maxcities [50];
  srand(time(NULL));
  init();
  TimePathProgressbar->Position=0;
  PathMemo->Lines->Clear();
  while (curTime++ < MAX_TIME) {
    if (simulateAnts() == 0) {
      updateTrails();
      if (curTime != MAX_TIME) restartAnts();
      sprintf(pathtime, "Час %d (%g)\n", curTime, best);
      PathMemo->Lines->Add(pathtime);
      TimePathProgressbar->Position++; } }
  sprintf(pathtime, "Найкращий шлях — %g", best);
  sprintf(maxcities, "Кількість міст — %d", MAX_CITIES);
  PathMemo->Lines->Add("\n\n");
  PathMemo->Lines->Add(maxcities);
  PathMemo->Lines->Add("\n");
  PathMemo->Lines->Add(pathtime);
  XYGraph->Series[0]->Clear();
  for (city = 0; city < MAX_CITIES; city++) {
    XYGraph->Series[0]->Add(cities[ants[bestIndex].path[city]].x,
      cities[ants[bestIndex].path[city]].y); }
  XYGraph->Series[0]->Add(cities[ants[bestIndex].path[0]].x,
    cities[ants[bestIndex].path[0]].y);
  emitDataFile(bestIndex); }

```

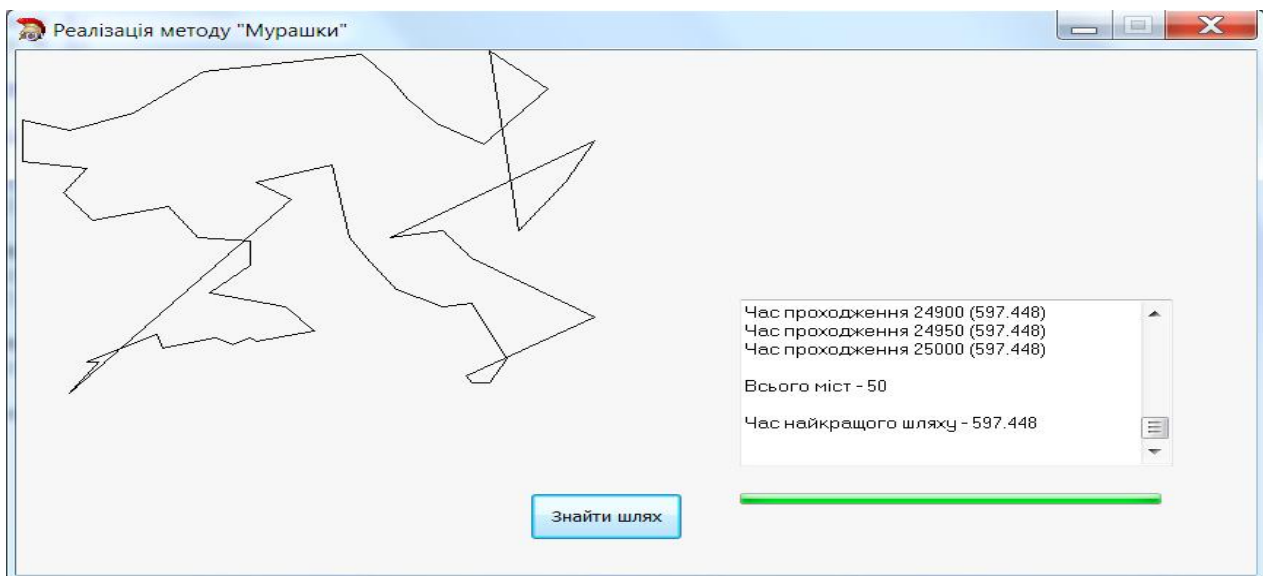


Рис. 6. Результати роботи програми

Зміни параметрів алгоритму

Марко Доріго (винахідник оптимізації за принципом мурашиної колонії) у статті «Система мурах: оптимізація за допомогою колонії агентів, які співпрацюють» (The Ant System: Optimization by a Colony of Cooperating Agents) пропонує дуже цікаву дискусію за параметрами алгоритму. Параметр α асоціюється з кількістю ферменту, а параметр β — з видимістю (довжиною перешкоди). Що більше значення параметра, то важливіший він для вірогідного рівняння, котре застосовується при виборі межі. Зверніть увагу, що в одному випадку значущість параметрів однакова. В усіх інших випадках видимість важливіша при виборі шляху.

Параметр ρ становить коефіцієнт, який застосовується до ферменту, що розпилюється на шлях, а $(1,0 - \rho)$ становить коефіцієнт випаровування для наявного ферменту. Були проведені тести при $\rho > 0,5$, та всі вони показали цікаві результати. При встановленні значення $\rho < 0,5$ результати були незадовільні.

В першу чергу цей параметр визначає концентрацію ферменту, який зберігається на межах.

Кількість мурах впливала на якість отриманих рішень. Хоча збільшення кількості мурах може здатися хорошою ідеєю, найкращий результат досягається в тому разі, якщо кількість мурах дорівнює кількості міст.

Алгоритм мурахи застосовується для вирішення інших завдяки, наприклад, прокладання маршрутів для автомобілів, роботів, розрахунку кольорів для графіків і маршрутизації у мережах. Детальніше способи вживання алгоритму мурахи викладено у книзі Марко Доріго «Алгоритми мурахи для абстрактної оптимізації» (Ant Algorithms for Discrete Optimization).

Алгоритм мурахи моделює поведінку мурах в їхньому природному середовищі, щоб визначити оптимальний шлях у просторі (за графом або мережею). Ця технологія розглядалась як засіб розширення завдання комівояжера (TSP). Крім того, були описані можливості зміни параметрів алгоритму і надані комбінації параметрів, котрі демонструють хороші результати.