

Міністерство освіти і науки України
Запорізький національний технічний університет

МЕТОДИЧНІ ВКАЗІВКИ
до виконання лабораторних робіт
з дисципліни

“ІНТЕЛЕКТУАЛЬНІ АГЕНТИ”

для студентів спеціальності 8.080403
“Програмне забезпечення автоматизованих систем”
напряму підготовки 6.050103 “Програмна інженерія”
(усіх форм навчання)

2007

Методичні вказівки до виконання лабораторних робіт з дисципліни “Інтелектуальні агенти” для студентів спеціальності 8.080403 “Програмне забезпечення автоматизованих систем” напряму підготовки 6.050103 “Програмна інженерія” (усіх форм навчання) / С.О. Субботін, А.О. Олійник, О.О. Олійник. – Запоріжжя: ЗНТУ, 2007. – 30 с.

*Видання підготовлено за Міжнародним проектом
“Європейсько-український ступінь магістра з програмного забезпечення”
(JEP 26182 2006) за програмою “TEMPUS” Європейської комісії.*

Автори: Сергій Олександрович Субботін,
Андрій Олександрович Олійник,
Олексій Олександрович Олійник

Рецензент: Г.В. Табунщик, к.т.н., доцент

Відповідальний
за випуск: А.В. Притула, к.т.н., доцент

Затверджено
вченою радою інституту
інформатики та радіоелектроніки

Протокол № 3
від 02.10.2007 р.

Затверджено
на засіданні кафедри
“Програмні засоби”

Протокол № 1
від 28.09.2007 р.

ЗМІСТ

Вступ.....	4
1 Лабораторна робота № 1 Метод мурашиних колоній	5
1.1 Мета роботи	5
1.2 Основні теоретичні відомості.....	5
1.2.1 Основні поняття	5
1.2.2 Послідовність виконання методу	6
1.3 Завдання до роботи	8
1.4 Зміст звіту	9
1.5 Контрольні питання	10
2 Лабораторна робота № 2 Моделювання природних систем	11
2.1 Мета роботи	11
2.2 Основні теоретичні відомості.....	11
2.2.1 Поняття штучного життя	11
2.2.2 Моделювання харчових ланцюгів	11
2.2.2.1 Опис моделі.....	12
2.2.2.2 Навколишнє середовище	12
2.2.2.3 Анатомія агенту	13
2.2.2.4 Енергія та метаболізм	16
2.2.2.5 Відтворення.....	16
2.2.2.6 Смерть	17
2.3 Завдання до роботи	17
2.4 Зміст звіту	19
2.5 Контрольні питання	19
3 Лабораторна робота № 3 Розробка Web-агента	20
3.1 Мета роботи	20
3.2 Основні теоретичні відомості.....	20
3.2.1 Основні поняття про програмні агенти	20
3.2.2 Властивості програмних агентів.....	20
3.2.3 Класифікація агентів	22
3.2.4 Розумні агенти.....	23
3.2.5 Web-агент	23
3.3 Завдання до роботи	26
3.4 Зміст звіту	29
3.5 Контрольні питання	29
Література	30

ВСТУП

Дане видання призначене для вивчення та практичного освоєння студентами усіх форм навчання основ інтелектуальних агентів.

Відповідно до графіка студенти перед виконанням лабораторної або самостійної роботи повинні ознайомитися з конспектом лекцій та рекомендованою літературою.

Для одержання заліку по кожній роботі студент здає викладачу цілком оформлений звіт, а також 3,5-дюймову дискету у форматі MS – DOS / Windows, перевірену на відсутність вірусів, з текстами розроблених програм, файлами програм, що виконуються, файлами даних і текстом звіту.

Звіт має містити:

- титульний аркуш;
- мету, варіант і завдання роботи;
- лаконічний опис теоретичних відомостей;
- текст програми, що обов'язково містить коментарі;
- вхідні та вихідні дані програми;
- змістовний аналіз отриманих результатів та висновки.

Звіт виконують на білому папері формату А4 (210 x 297 мм). Текст розміщують тільки з однієї сторони листа. Поля сторінки з усіх боків – 20 мм. Аркуші скріплюють за допомогою канцелярських скріпок. Для набору тексту звіту використовують редактор MS Word 97: шрифт Times New Roman, 12 пунктів. Міжрядковий інтервал: полуторний – для тексту звіту, одинарний – для листингів програм, таблиць і роздруківок даних.

Під час співбесіди студент повинний виявити знання про мету роботи, по теоретичному матеріалу, про методи виконання кожного етапу роботи, по змісту основних розділів розробленого звіту з демонстрацією результатів на конкретних прикладах. Студент повинний вміти правильно аналізувати отримані результати. Для самоперевірки при підготовці до виконання і здачі роботи студент повинний відповісти на контрольні питання, приведені наприкінці опису відповідної роботи. Загальний залік студент одержує після виконання і здачі останньої роботи.

1 ЛАБОРАТОРНА РОБОТА № 1

МЕТОД МУРАШИНИХ КОЛОНІЙ

1.1 Мета роботи

1.1.1 Вивчити метод мурашиних колоній.

1.1.2 Навчитися розв'язувати задачі комбінаторної оптимізації за допомогою методу мурашиних колоній.

1.2 Основні теоретичні відомості

1.2.1 Основні поняття

Мурашині алгоритми серйозно досліджуються європейськими вченими із середини 1990-х років. На сьогоднішній день вже отримано гарні результати для оптимізації таких складних комбінаторних задач, як задача комівояжера, задача оптимізації маршрутів вантажівок, задача розфарбування графа, квадратична задача про призначення, задача оптимізації сіткових графіків, задача календарного планування і т. ін. Особливо ефективні мурашині алгоритми при динамічній оптимізації процесів у розподілених нестационарних системах, наприклад, трафіків у телекомунікаційних мережах.

Метод мурашиних колоній заснований на взаємодії декількох *мурах* (програмних агентів, що є членами великої колонії) і використовується для вирішення різних оптимізаційних задач. Агенти спільно вирішують проблему й допомагають іншим агентам у подальшій оптимізації розв'язку.

Методи мурахи (Ant algorithms), або оптимізація за принципом мурашиної колонії, мають специфічні особливості, властиві мурахам, і використовують їх для орієнтації у фізичному просторі. Природа пропонує різні методики для оптимізації деяких процесів. Методи мурашиних колоній особливо цікаві тому, що їх можна використовувати для вирішення не тільки статичних, але й динамічних задач, наприклад, задач маршрутизації в мінливих мережах.

Базова ідея методу мурашиних колоній полягає у вирішенні оптимізаційної задачі шляхом застосування непрямого зв'язку між автономними агентами.

У методі мурашиних колоній передбачається, що навколишнє середовище являє собою закриту двовимірну мережу – групу вузлів, з'єднаних за допомогою граней. Кожна грань має вагу, що позначається як відстань між двома вузлами, з'єднаними нею. Граф – двонаправлений, тому агент може подорожувати по грані в будь-якому напрямку.

Агент забезпечується набором простих правил, які дозволяють йому вибрати шлях у графі. Він підтримує список табу *tList* – список вузлів, які він вже відвідав. Таким чином, агент повинен проходити через кожний вузол тільки один раз.

Вузли в списку “поточної подорожі” *tList* розташовуються в тому порядку, у якому агент відвідував їх. Пізніше список використовується для визначення довжини шляху між вузлами.

Справжня мураха під час переміщення по шляху залишає за собою деяку кількість *феромону*. У методі мурашиних колоній агент залишає феромон на гранях мережі після завершення подорожі.

1.2.2 Послідовність виконання методу

Метод мурашиних колоній виконується в наступній послідовності кроків.

Крок 1. Задати параметри методу: α – коефіцієнт, що визначає відносну значимість шляху (кількість феромона на шляху); β – параметр, що означає пріоритет відстані над кількістю феромона; ρ – коефіцієнт кількості феромона, що агент залишає на шляху, де $(1-\rho)$ показує коефіцієнт випару феромона на шляху після його завершення; Q – константу, що відноситься до кількості феромона, яку було залишено на шляху.

Крок 2. Ініціалізація методу. Створення популяції агентів. Після створення популяція агентів нарівно розподіляється по вузлах мережі. Необхідно рівномірний розподіл агентів між вузлами, щоб всі вузли мали однакові шанси стати відправною точкою. Якщо всі агенти почнуть рух з однієї точки, це буде означати, що дана точка вважається оптимальною для старту, а насправді вона такою може і не бути.

Крок 3. Рух агентів. Якщо агент ще не закінчив шлях, тобто не відвідав всі вузли мережі, для визначення наступної грані шляху використовується формула:

$$P = \frac{\tau_{ru}(t)^\alpha \cdot \eta_{ru}(t)^\beta}{\sum_{k \in J} \tau_{ru}(t)^\alpha \cdot \eta_{ru}(t)^\beta},$$

де J – множина граней, ще не відвіданих агентом; $\tau_{ru}(t)$ – інтенсивність феромона на грані між вузлами r і u , які утворюють k -у грань, у момент часу t ; $\eta_{ru}(t)$ – функція, що представляє собою зворотну величину відстані грані.

Агент подорожує тільки по вузлах, які ще не були відвідані ним, тобто по вузлах, яких не має у списку табу $tList$. Тому ймовірність розраховується тільки для граней, які ведуть до ще не відвіданих вузлів.

Крок 4. Пройдений агентом шлях відображається, коли агент відвідає всі вузли мережі. Цикли заборонені, оскільки в метод включений список табу $tList$. Після завершення може бути розрахована *довжина шляху*. Вона дорівнює сумі всіх граней, по яких подорожував агент. *Кількість феромону*, що було залишено на кожній грані шляху i -го агента, визначається за формулою:

$$\Delta\tau_{ru}^i(t) = \frac{Q}{L^i(t)},$$

де $L^i(t)$ – довжина шляху i -го агента.

Результат є засобом вимірювання шляху: короткий шлях характеризується високою концентрацією феромону, а більш довгий шлях – більш низкою. Потім отриманий результат використовується для збільшення кількості феромону уздовж кожної грані пройденого i -им агентом шляху:

$$\tau_{ru}(t+1) = \tau_{ru}(t) + (\Delta\tau_{ru}^i(t) \cdot \rho),$$

де r, u – вузли, що утворюють грані, які відвідав i -ий агент.

Дана формула застосовується до всього шляху, при цьому кожна грань позначається феромоном пропорційно довжині шляху. Тому варто дочекатися, поки агент закінчить подорож і тільки потім оновити рівні феромону, у протилежному випадку дійсна довжина шляху залишиться невідомою. Константа ρ приймає значення між 0 і 1.

$$\tau_{ru}(t) = \tau_{ru}(t) \cdot (1 - \rho).$$

Тому для випаровування феромону використовується зворотний коефіцієнт відновлення шляху $(1 - \rho)$.

Крок 5. Перевірка на досягнення оптимального результату. Перевірка може виконуватися для постійної кількості шляхів або до моменту, коли протягом декількох запусків не було отримано повторних змін у виборі найкращого шляху. Якщо перевірка дала позитивний результат, то відбувається закінчення роботи методу (перехід до кроку 7), у противному випадку – перехід до кроку 6.

Крок 6. Повторний запуск. Після того як шлях агента завершений, грані оновлені відповідно до довжини шляху, й відбулося випаровування феромону на всіх гранях, метод виконується повторно. Список табу очищується, і довжина шляху онулюється. Після цього виконується до кроку 3.

Крок 7. Кінець. Визначається кращий шлях, що і є розв'язком.

1.3 Завдання до роботи

1.3.1 Ознайомитися з літературою та основними теоретичними відомостями за темою роботи.

1.3.2 Розробити за допомогою пакету Matlab програмне забезпечення, що реалізує метод мурашиних колоній. При програмній реалізації методу дотримуватися вимог, наведених в п.1.3.2.1 – 1.3.2.4.

1.3.2.1. Для гнучкості програмного забезпечення всі параметри методу (наприклад, кількість агентів, коефіцієнт випаровування і т. ін.) зберігати в змінних або константах.

1.3.2.2. Для реалізації агентів, що моделюють поведінку мурах, використовувати структуру, що має наступні поля:

- поточна позиція, в якій знаходиться агент;
- наступна позиція;
- список табу, в якому зберігаються ті пункти, в яких вже побував агент;
- кількість пунктів, що вже відвідав агент;
- масив подорожі, де зберігається послідовність пунктів, в яких побував агент;
- довжина шляху, що пройшов агент. Розраховується після закінчення подорожі.

1.3.2.3. Виділити основні етапи методу в окремі функції, що реалізуються у відповідних .m-файлах:

- ініціалізація методу;
- моделювання переміщення агентів;
- вибір наступного пункту;
- оновлення шляхів;
- перезавантаження агентів.

Також необхідно реалізувати окрему функцію розрахунку довжини шляху.

1.3.2.4. Передбачити можливість наглядного (графічного) відображення змін у поточних результатах роботи методу.

1.3.3 Виконати тестування розробленого програмного забезпечення за допомогою вирішення конкретних прикладів задачі комівояжера. Задачі (не менше трьох) для виконання тестування програми сформулювати самостійно. Вибір тестових задач обґрунтувати.

1.3.4 Порівняти одержані результати вирішення різних прикладів задачі комівояжера. Результати порівняльного аналізу звести до таблиці, попередньо розробивши систему критеріїв порівняння результатів вирішення задачі комівояжера.

1.3.5 Оформити звіт з роботи.

1.3.6 Відповісти на контрольні питання.

1.4 Зміст звіту

1.4.1 Тема та мета роботи.

1.4.2 Короткі теоретичні відомості.

1.4.3 Текст розробленого програмного забезпечення з коментарями.

1.4.4 Результати роботи програмного забезпечення (таблиця порівняльного аналізу та графіки з поясненнями).

1.4.5 Висновки, що містять відповіді на контрольні запитання, а також відображують результати виконання роботи та їх критичний аналіз.

1.5 Контрольні питання

1.5.1 В чому полягає основна ідея методу мурашиних колоній?

1.5.2 Для розв'язання яких задач використовується метод мурашиних колоній?

1.5.3 В якій умовах особливо ефективно використання мурашиних алгоритмів?

1.5.4 Порівняйте метод мурашиних колоній з іншими оптимізаційними методами.

1.5.5 Що являє собою навколишнє середовище в методі мурашиних колоній?

1.5.6 З якою метою використовується список табу?

1.5.7 Яким чином розташовуються вузли в списку “поточної подорожі”?

1.5.8 Що таке феромон? Яке його призначення в методі мурашиних колоній?

1.5.9 Проаналізуйте послідовність виконання методу мурашиних колоній.

1.5.10 Як розраховується кількість феромону, що було залишено на кожній грані шляху i -го агенту?

1.5.11 Поясніть особливості модифікації методу мурашиних колоній при вирішенні таких задач комбінаторної оптимізації, як задача комівояжера, задача оптимізації маршрутів вантажівок, задача розфарбування графа, квадратична задача про призначення, задача оптимізації сіткових графіків, задача календарного планування.

1.5.12 Проаналізуйте засоби пакету Matlab, які були використані при розробці програмного забезпечення, що реалізує метод мурашиних колоній.

2 ЛАБОРАТОРНА РОБОТА № 2 МОДЕЛЮВАННЯ ПРИРОДНИХ СИСТЕМ

2.1 Мета роботи

2.1.1 Навчитися моделювати природні системи.

2.1.2 Вивчити методи, які застосовуються до моделювання природних систем.

2.2 Основні теоретичні відомості

2.2.1 Поняття штучного життя

Штучне життя (Artificial life) – поняття, що ввів Кріс Лангтон для позначення множини комп'ютерних механізмів, котрі використовуються для моделювання природних систем. Штучне життя використовується для моделювання процесів у економіці, поведінки тварин та комах, а також для взаємодії різноманітних об'єктів.

Штучне життя являє собою цілу науку з безліччю аспектів. До штучного життя відноситься синтетична наука про поведінку, що представляє собою підхід до вивчення поведінки тварин, при якому прості синтетичні організми певним чином діють у синтетичному світі.

Штучне життя може бути описане як теорія й практика моделювання біологічних систем. Розробники, які ведуть дослідження в даній сфері, сподіваються, що шляхом моделювання біологічних систем можна бути зрозуміти, чому і як вони працюють. За допомогою моделей дослідники можуть управляти створеним середовищем, перевіряти різні гіпотези й спостерігати, як системи й середовище реагують на зміни.

2.2.2 Моделювання харчових ланцюгів

Харчовий ланцюг описує ієрархію живих організмів в екосистемі. Найпростішим прикладом харчового ланцюгу може бути абстрактний ланцюг, що складається із трьох особин (рис. 2.1).

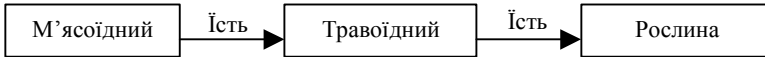


Рисунок 2.1 – Простий харчовий ланцюг

В нижній частині харчового ланцюгу перебувають рослини. Вони одержують енергію з навколишнього середовища (дощу, ґрунту й сонця). Наступний рівень займають травоїдні тварини – для виживання вони поїдають рослини. На верхньому щаблі перебувають хижаки. У цій моделі хижаки поїдають травоїдних тварин, щоб вижити.

Щоб змодельувати простий харчовий ланцюг, необхідно визначити деякі параметри: навколишнє середовище (фізичний простір, у якому взаємодіють агенти), самих агентів (а також їхнє сприйняття й поведінки в середовищі) і групу правил, які визначають, як і коли відбувається взаємодія.

2.2.2.1 Опис моделі

Модель простого харчового ланцюгу складається із середовища й трьох типів особин. Рослини являють собою нерухоме джерело їжі для травоїдних тварин. Травоїдні тварини є мігруючими агентами, які певним чином сприймають навколишнє середовище і їдять рослини. Іншими мігруючими агентами в середовищі є хижаки, що поїдають травоїдних тварин. Хижаки можуть їсти тільки травоїдних, а травоїдні можуть їсти тільки рослини. Якщо який-небудь агент живе в середовищі певний час і не одержує їжі, він гине від голоду. Коли агент поглинає достатню кількість їжі, він може розмножуватися. Таким чином, у середовищі створюється новий агент певного типу. Відбувається еволюція, при якій мутує мозок агенту (проста нейронна мережа).

Важливо відзначити, що агенти спочатку не знають, як потрібно виживати в середовищі. Вони не знають, що поїдання їжі дозволить їм прожити довше. Також вони не знають, що повинні уникати тих, хто їй їсть. Агенти повинні освоїти всі ці знання за допомогою еволюції.

2.2.2.2 Навколишнє середовище

Агенти живуть у світі, побудованому за принципом сітки, грані якої з'єднані за аналогією з тороїдом. Якщо агент переміщається за грань у певному напрямку, він з'являється на іншій стороні (рис. 2.2).

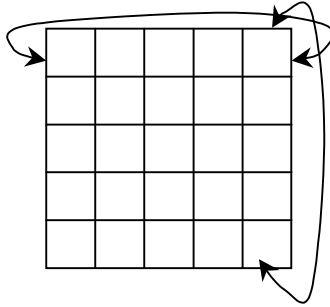


Рисунок 2.2 – Світ у вигляді сітки, побудованої за принципом тороїду

Рослини займають унікальні чарунки в середовищі, однак декілька агентів можуть займати одну чарунку (трав'ядна тварина й/або хижак).

2.2.2.3 Анатомія агенту

Агент є генетичною особою. Він може бути тільки певного типу (трав'ядним або хижак), але метод вивчення навколишнього середовища та порядок дій для всіх агентів однакові (рис. 2.3). Агента можна розглядати як просту систему з набором входів (його відчуттям світу), реакцій на навколишній світ (його мозком) та дій.

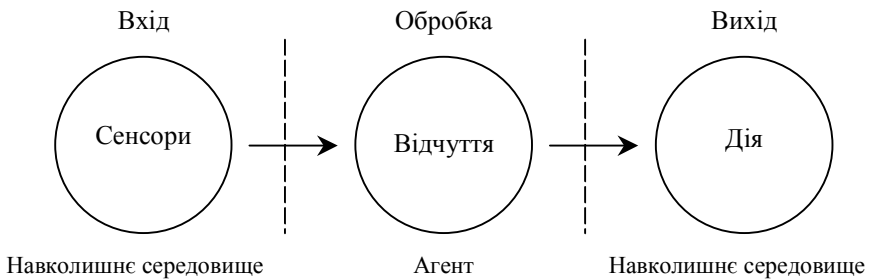


Рисунок 2.3 – Модель систем з агентами

Як відображено на рис. 2.3, агент складається із трьох окремих частин. Це сенсори, відчуття (визначення того, яку дію вибрати) і дія. Модель агента реагує на навколишнє середовище. Агенти не можуть

планувати й навчатися. Навіть у такій простій моделі навчання відбувається за принципом, що називається еволюцією Ламарка. При відтворенні характеристики батька передаються потомству.

Сенсори

Агенти можуть відчувати, що відбувається навколо них у середовищі. Однак агент не бачить все середовище, він реагує тільки на групу чарунок навколо нього (рис. 2.4). Локальне середовище, що може відчувати агент, розділене на чотири окремі області. Найближча до агента область називається *областю близькості*, саме в цій області агент може виконувати певні дії (наприклад, з'їсти інший об'єкт). Область поперед агента (5 чарунок) називається *фронтом*.

	Фронт	Фронт	Фронт	Фронт	Фронт	
	Ліворуч	Близькість	Близькість	Близькість	Праворуч	
	Ліворуч	Близькість	Агент	Близькість	Праворуч	

Рисунок 2.4 – Область відчуття агента. Агент “дивиться” на північ

Агент може визначати вид об'єктів у полі зору. Тому для чотирьох областей пропонуються три числа, які дозволяють ідентифікувати типи наявних об'єктів (рослини, трав'яні й хижакі), тобто всього дванадцять входів.

Активатори

Агент може виконувати обмежене число дій у середовищі: переміститися на одну чарунку (у заданому напрямку), повернутися ліворуч або праворуч або з'їсти об'єкт, що перебуває в області близькос-

ті. Дія, яку виконує агент, визначається його мозком при оцінці входів, отриманих на рівні сенсорів.

Мозок агента

Мозок агента може бути однією з численних комп'ютерних конструкцій. Існуючі симуляції штучного життя використовують принцип кінцевих автоматів, системи класифікації або нейронні мережі. Щоб зберегти аналогію з біологічною мотивацією, при моделюванні харчових ланцюгів використовується проста нейронна мережа, побудована за принципом “переможець отримує все”. Входи такої мережі u_i інтерпретуються наступним чином:

- u_0 – травоїдне на передньому плані;
- u_1 – хижак на передньому плані;
- u_2 – рослина на передньому плані;
- u_3 – травоїдне ліворуч;
- u_4 – хижак ліворуч;
- u_5 – рослина ліворуч;
- u_6 – травоїдне праворуч;
- u_7 – хижак праворуч;
- u_8 – рослина праворуч;
- u_9 – близькість травоїдного;
- u_{10} – близькість хижака;
- u_{11} – близькість рослини.

Така нейронна мережа має 4 виходи o_j :

- o_0 – повернути ліворуч;
- o_1 – повернути праворуч;
- o_2 – переміститися;
- o_3 – з'їсти.

Входи сенсорів відображають кількість агентів, які перебувають у полі зору в певній області. Після того як всі входи були отримані із середовища, програма “просуває” їх через мережу до виходів. Це виконується за допомогою формули:

$$o_j = b_j + \sum_{i=0}^n u_i w_{ij}$$

Тобто, для кожного виходу (o_j) мережі підсумовуються результати входів (u_i), які множаться на ваги з'єднань від входів до виходів (w_{ij}). Також додається зсув для виходу. У результаті на виходах мережі буде отриманий набір значень, які потім використовуються елементом дії агента.

Початкові ваги нейронної мережі агента вибираються випадковим чином. У результаті відтворення ваги повинні бути настроєні для виживання в середовищі.

Вибір дії агенту

Агент може виконувати одну дію із чотирьох можливих, що відповідають виходам нейронної мережі. Процес вибору дії полягає в пошуку вихідної чарунки з найбільшим значенням і виконанні відповідної дії. Це і є принцип “переможець отримує все” стосовно до мережі. Після виконання дії навколишнє середовище змінюється (якщо на нього впливали), і процес продовжується.

2.2.2.4 Енергія та метаболізм

Щоб вижити в навколишньому середовищі, агентам потрібна адекватна енергія. Якщо внутрішня енергія агента стає рівною нулю, агент вмирає. Агенти створюють енергію, з'їдаючи інші об'єкти в середовищі. Агент може з'їсти тільки той об'єкт, що допускається харчовим ланцюгом. Хижаки можуть їсти тільки травоядних, а травоядні – тільки рослини. Агенти також мають метаболізм, тобто коефіцієнтом поглинання енергії, що дозволяє їм зберігати життя. За кожну одиницю часу хижаки поглинають одну одиницю енергії, а травоядні – дві одиниці. Це означає, що для збереження життя травоядним потрібно з'їдати у два рази більше їжі, ніж хижакам. Хоча хижакам не потрібно так багато їжі, їм ще потрібно її знайти. Травоядні мають перевагу, що полягає в тому, що їхня їжа не переміщується по середовищу. Проте їм однаково потрібно відшукати свою їжу.

2.2.2.5 Відтворення

Якщо агент поглинає достатню кількість їжі, щоб досягти показника 90% від максимального рівня енергії, він допускається до участі у відтворенні. Відтворення дозволяє агентам, які змогли вижити в навколишньому середовищі, створити потомство (природний відбір). При створенні потомства агенти змінюють ваги своїх нейронних мереж за допомогою випадкової мутації. Навчання в середовищі недовступно, однак те, що агент може відтворювати себе, означає, що його нейронна мережа буде передана його дитині. Це повторює принцип

еволюції Ламарка, оскільки характеристики агента передаються його потомству (дитина успадковує нейронну мережу свого батька).

Відтворення має наслідок, який полягає в тому, що: батько й дитина розділяють наявну енергію батька (енергія батька ділиться навпіл). При цьому агент не зможе безупинно відтворювати себе.

2.2.2.6 Смерть

Агент може вмерти двома способами: він не може знайти їжу й помирає від голоду або його з'їдає інший агент, що стоїть вище в харчовому ланцюгу. У кожному разі мертвий агент видаляється з моделі.

2.3 Завдання до роботи

2.3.1 Ознайомитися з літературою та основними теоретичними відомостями за темою роботи.

2.3.2 Розробити за допомогою пакету Matlab програмне забезпечення, що моделює штучне життя (модель харчового ланцюгу). При розробці програмного забезпечення врахувати твердження, наведені в п.2.3.2.1–2.3.2.5.

2.3.2.1 Штучний світ моделюється за допомогою сітки розмірності $N \times N$.

2.3.2.2 Існує всього 3 типи агентів:

1) рослини (при цьому в процесі моделювання, коли одну рослину з'їдають, має моделюватися поява нової рослини);

2) травоядні: вони мають становити максимально 50% від максимального можливого значення агентів і не можуть бути в кількості, меншій за 25% від максимального можливого значення агентів. При цьому травоядні можуть їсти рослини, в той же час вони є їжею для 3-го типу агентів – м'ясоїдних;

3) м'ясоїдні: обмеження на їхню кількість такі ж, як і для травоядних.

2.3.2.3 Агенти, що моделюють травоядних та м'ясоїдних, мають бути реалізовані у вигляді структури, що містить наступні поля:

- тип;
- енергія;
- вік – кількість ітерацій, на протязі яких жив агент;

- покоління: спочатку встановлюється в 1. Збільшується, якщо агент є нащадком іншого (тобто збільшується в процесі розмноження);

- координата X;

- координата Y;

- напрям погляду агенту: може бути північним, східним, західним або південним;

- входи нейромережі: на входи поступають кількості інших агентів відповідно до їх положення відносно даного агента. Тобто на вхід 1 поступає кількість трав'янистих на передньому плані, на вхід 2 – кількість м'ясоїдних на передньому плані, на вхід 3 – кількість рослин на передньому плані, на входи 4-6 – кількості ті ж агентів, що і для входів 1-3, але вже ліворуч, 7-9 – праворуч, 10-12 – у безпосередній близькості;

- ваги нейромережі, які вибираються випадковим чином один раз при ініціалізації моделювання;

- зміщення для нейромережі – вибираються випадковим чином при ініціалізації;

- 4 виходи нейромережі, значення яких розраховуються за наведеною в основних теоретичних відомостях формулою. Значення виходів відповідають можливим діям агенту: повернути ліворуч (тобто лише змінити напрям погляду), повернути праворуч, рух вперед відповідно до напрямку, з'їсти іншого агента. При цьому, якщо було прийнято рішення з'їсти іншого агента, то обов'язково хтось буде з'їденим, оскільки дане рішення може бути помилковим. Обирається та дія, значення якої є найбільшим.

2.3.2.4 При моделюванні штучного життя вважати, що при поїданні рослини отримується енергія, вдвічі менша за поїдання трав'янистого;

2.3.2.5 Зберігати наступну статистику:

- кількість агентів відповідних типів на даній ітерації;

- максимальний вік за весь час моделювання агентів відповідних типів;

- кількість розмножень агентів для кожного типу.

2.3.3 Обґрунтовано сформувати набір даних для тестування розробленого програмного забезпечення. Виконати тестування. Результати тестування оформити в вигляді таблиць та графіків.

2.3.4 Оформити звіт з роботи.

2.3.5 Відповісти на контрольні питання.

2.4 Зміст звіту

2.4.1 Тема та мета роботи.

2.4.2 Короткі теоретичні відомості.

2.4.3 Текст розробленого програмного забезпечення з коментарями.

2.4.4 Обґрунтовано обраний набір даних для обробки (якщо він великий – навести фрагмент).

2.4.5 Результати роботи програмного забезпечення (таблиця порівняльного аналізу з поясненнями, графіки).

2.4.6 Висновки, що містять відповіді на контрольні запитання, а також відображують результати виконання роботи та їх критичний аналіз.

2.5 Контрольні питання

2.5.1 Що таке штучне життя? З якою метою ведуться дослідження в галузі штучного життя?

2.5.2 Яким чином пов'язані штучне життя та синтетична наука про поведінку?

2.5.3 В чому полягає сутність моделювання харчових ланцюгів?

2.5.4 Проаналізуйте параметри, які визначаються при моделюванні харчових ланцюгів.

2.5.5 Дайте визначення понять: область близькості агенту, фронт агенту.

2.5.6 З якою метою використовуються сенсори та активатори?

2.5.7 Які математичні засоби можуть бути використані для моделювання мозку агента?

2.5.8 Які дії може виконувати агент?

2.5.9 Що таке енергія та метаболізм?

2.5.10 При виконанні якої умови агент допускається до відтворення?

2.5.11 Якими способами агент може загинути?

2.5.12 Проаналізуйте засоби пакету Matlab, які були використані при моделюванні харчового ланцюгу.

3 ЛАБОРАТОРНА РОБОТА № 3 РОЗРОБКА WEB-АГЕНТА

3.1 Мета роботи

3.1.1 Вивчити основні принципи роботи програмних агентів.

3.1.2 Навчитися створювати Web-агента, який взаємодіє із певними службами в середовищі Internet.

3.2 Основні теоретичні відомості

3.2.1 Основні поняття про програмні агенти

Агент – це апаратна або програмна сутність, здатна діяти в інтересах досягнення цілей, поставлених перед нею користувачем.

В межах мультиагентної парадигми програмні агенти являють собою автономні компоненти, що діють від імені користувача.

Агентів також називають розумними агентами (Intelligent agent), тому що розумність є ключовим чинником при їхньому створенні. Хоча агенти можуть приймати різні форми, але вони всі реалізують якісь корисні функції за людину. Агент може одержати завдання, яке необхідно виконати, і приймати потрібні рішення в процесі комунікації з іншими агентами (можливо, навіть із агентами, що представляють інтереси інших користувачів).

3.2.2 Властивості програмних агентів

Агенти можуть мати одну або декілька властивостей, наведених в табл. 3.1.

Однією з основних властивостей, які асоціюються із програмними агентами, є автономність (Autonomy). Агент вважається автономним, якщо він здатний діяти без прямого управління людиною. Поняття автономності включає також наявність в агента декількох цілей, яких він може досягти. Наявність задач також сприяє тому, що агент одержує можливість планувати.

Таблиця 3.1 – Властивості програмних агентів

Властивість	Визначення
Автономність	Можливість діяти незалежно від користувача
Адаптивність	Здатність до навчання під час роботи
Комунікативність	Здатність до комунікації з користувачем або іншими агентами
Здатність до співробітництва	Працює з іншими агентами для досягнення мети
Персоніфікованість	Поводиться природно (проявляє емоції)
Мобільність	Можливість переміщатися по навколишньому середовищу

Агент є адаптивним (Adaptivity), якщо він може змінювати своє поведіння на підставі досвіду. Він повинен учитися й робити висновки залежно від стану навколишнього середовища й своїх знань.

Ще однією важливою характеристикою агентів є здатність до комунікації (Communicative). Агент повинен уміти спілкуватися з користувачем, щоб визначати свої завдання або ідентифікувати початкову інформацію. Крім того, агентові необхідно спілкуватися з навколишнім середовищем. Наприклад, пошуковий агент, що знаходить цікаві Web-сторінки, повинен уміти спілкуватися за допомогою протоколу HTTP, щоб підключатися до серверів і одержувати від них інформацію (шукані сторінки).

Агент здатний до співробітництва (Collaborative), якщо він може спілкуватися з іншими агентами для вирішення своїх завдань. Наприклад, агенти, які діють від імені користувачів на аукціонах, можуть брати участь в укладанні спільних угод, орієнтуючись на інформацію, отриману від користувачів. Подібні агенти звичайно входять у системи з більшою кількістю агентів, тому що для співробітництва потрібна безліч агентів.

Для деяких агентів найбільш істотною є здатність до персоналізації (Personality). Програми з їхнім застосуванням звичайно використовуються в розважальних цілях. Тому, якщо персоналізація агента буде змінювати його залежно від переданої інформації, користувач зможе зрозуміти те, що агент намагається повідомити. Наприклад, коли оповідач про що-небудь розповідає й вираз його особи при цьому

не змінюється, слухачеві може бути важко його зрозуміти. Інформація, закодована в людських почуттях, дозволяє одержати додаткові відомості, тому можливість персоналізації може бути важливою для агентів певного класу.

Деякі агенти мають мобільність (Mobility), або можливість переміщення по навколишньому середовищу. Ця властивість є дуже корисною при створенні певних програм. Властивість мобільності використовується у програмних агентах, що одержують інформацію з віддаленої бази даних. При звичайних умовах хост приймає інформацію від бази даних, збирає потрібні відомості, а потім фільтрує їх за допомогою критеріїв, заданих користувачем. При одержанні мобільності агент може бути спрямований у віддалену базу даних, щоб автоматично відтфільтрувати результати, а потім повернути тільки ті, які необхідні користувачеві. Це дозволяє як заощадити на пропускну здатності каналу, так і спростити архітектуру, особливо в тому випадку, якщо два хости можуть бути відз'єднані один від одного.

Агент необов'язково повинен мати всі властивості. У більшості випадків використовуються всього дві.

3.2.3 Класифікація агентів

Незалежні агенти можна класифікувати наступним чином.

1. Агенти штучного життя.
2. Програмні агенти.
 - 2.1. Агенти для завдань.
 - 2.2. Агенти розваг.
 - 2.3. Віруси.
3. Інші агенти.

Агент, використання якого залежить від завдання, застосовується для вирішення певної проблеми. Прикладом може служити пошуковий агент у мережі Internet. Крім пошуку за заданими критеріями, агент також стежить за тим, які зі знайдених посилань відкриє користувач. Це дозволяє створити шкалу цінностей пошуку, що сприяє кращому сортуванню результатів, а також прискорює одержання користувачем потрібного результату. Пошуковий агент може працювати у фоновому режимі, повторюючи пошук, щоб знайти й надати користувачеві нову інформацію.

Пошукові агенти володіють не тільки автономністю, але й адаптивністю, оскільки вони вивчають найцікавіші для користувача теми, ґрунтуючись на його реакції (тобто посиланнях, які відкриває користувач).

Іншим прикладом може служити агент, що використовується для роботи на аукціоні. Він здатний діяти від імені користувача, купуючи товари в мережі Internet по найнижчій ціні. Він може робити це в співробітництві з іншими агентами. Знайшовши агентів, які намагаються купити схожі товари, можна зробити спільну оптову покупку групи товарів за зниженою ціною.

Розважальні агенти корисні для взаємодії у віртуальному світі або подання персонажа в якості інтерфейсу користувача.

Вірус по-іншому можна назвати небезпечним мобільним програмним агентом. Хоча вірус не обов'язково повинен бути адаптивним, він є автономним і мобільним. Замість того щоб використовувати спеціальний протокол для переміщення по мережі, вірус застосовує стандартні протоколи й найчастіше протокол передачі пошти.

3.2.4 Розумні агенти

Хоча зробити агента розумним досить складно, існує безліч методів, які можна використовувати для того, щоб додати агентів здатність до прийняття розумних рішень. До таких методів належать:

- нейронні мережі;
- системи, засновані на знаннях;
- дерева рішачих правил;
- еволюційні методи;
- нечітка логіка;
- алгоритми кластеризації.

3.2.5 Web-агент

Завдання Web-агенту полягає в тому, щоб взаємодіяти з певними службами у мережі Internet (наприклад, із службами новин) і збирати інформацію на підставі критеріїв, заданих користувачем. Після збирання інформації Web-агент надає її користувачеві в тому порядку, який найбільш відповідає запиту (починаючи з даних, що мають найвищий рейтинг за умовами пошуку).

Web-агент використовує стандартні протоколи й передає користувачеві інформацію через звичайний браузер. Для спілкування з користувачем агент містить HTTP-сервер, для зв'язку із зовнішніми серверами є HTTP-клієнтом, а для одержання новин функціонує як клієнт протоколу NTTP (рис. 3.1).

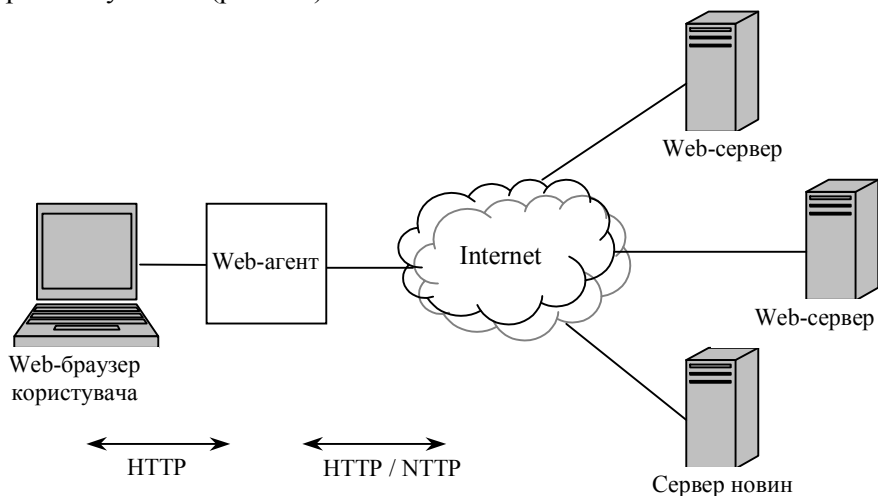


Рисунок 3.1 – Архітектура Web-агента

Web-агент буде відігравати роль додатка, що здійснює взаємодію між користувачем і мережею Internet при читанні новин. Користувач змінює параметри агента за допомогою простого файлу конфігурації. У ньому користувач указує Web-сторінки для моніторингу (для видачі повідомлення у випадку їхньої зміни) і групи новин для перегляду й визначає специфічні критерії пошуку. Коли агент знаходить об'єкти, які відповідають заданому критерію, він зберігає їх і видає користувачеві у вигляді Web-сторінки, посилання на які відсортовані в потрібному порядку. Користувач може клацнути по посиланню на певний Web-сайт або рядку новин і перейти до потрібної статті.

Завдання агента полягає в тому, щоб спростити читання новин, відфільтрувати непотрібну інформацію й представити новини в порядку, при якому найцікавіші дані розташовані на першому місці.

Web-агент є дуже простим і не має здатності до навчання. Він володіє двома властивостями: автономністю (здатний працювати у фоновому режимі без прямого керування з боку користувача) і здатністю до комунікації із зовнішніми серверами при зборі інформації.

Сенсорами агента є стандартні протоколи, які дозволяють йому збирати потрібну інформацію за заданими критеріями пошуку. Для передачі результатів пошуку користувачеві використовується протокол HTTP. Потім користувач підключається до Web-агента, як до будь-якого іншого Web-сервера.

Для отримання інформації з середовища Internet Web-агент виступає в якості HTTP-клієнта, який реалізує набір інтерфейсів для взаємодії з Internet серверами, наприклад, із серверами новин. Ці інтерфейси дозволяють додатку з'єднуватись із сервером новин, задавати групи новин за певними критеріями, переходити до заголовку статті, зчитувати всю статтю, аналізувати повідомлення, пропускати поточну новину та переривати зв'язок із сервером новин.

Протокол HTTP є інтерактивним протоколом, який повністю базується на ASCII-командах. Нижче наведений приклад роботи із HTTP-сервером новин yellow.geeks.org за допомогою Telnet. Дані, що вводить користувач, виділені жирним текстом.

```
root@plato /root]# telnet yellow.geeks.org 119
S: 201 plato.mtjones.com DNEWS Version 5.5dl, SO, posting OK
C: list
S: 215 list.of newsgroups follows
S: control 2 3 y
S: control.cancel 2 3 y
S: my.group 10 3 y
S: new.group 6 3 y
S: .
C: group my.group
S: 211 8 3 10 my.group selected
C: article 3
S: 220 3 <3C36AF8E. 1BD3047E@mtjones.com> article retrieved
S: Message-ID: <3C36AF8E. 1BD3047E@mtjones.com>
S: Date: Sat, 05 Jan 2002 00:47:27 -0700
S: From: "M. Tim Jones" <mtj@mtjones.com>
S: X-Mailer: Mozilla 4.74 [en] (Win98; U)
S: X-Accept-Language: en
S: MIME-Version: 1.0
S: Newsgroups: my.group
S: Subject: this is my post
```

```

S: Content-Type: text/plain; charset=us-ascii
S: Content-Transfer-Encoding: 7bit
S: NNTP-Posting-Host: sartre.mtjones.com
S: X-Trace: plato.mtjones.com 1010328764 sartre.mtjones.com
(6 Jan 2002 07:52:44 -0700)
S: Lines: 6
S: Path: plato.mtjones.com
S: Xref: plato.mtjones.com my.group:3
S:
S:
S: Hello
S:
S: This is my post.
S:
S:
S: .
C: date
S: 111 20020112122419
C: quit
S: 205 closing connection - goodbye!

```

У наведеному прикладі показано, як з'єднання з NNTP-сервером створюється за допомогою Telnet-клієнта. Сервер новин відповідає вітанням, у якому вказується тип сервера й інша подібна інформація. Тепер через з'єднання можна давати команди. Існує два базових типи відповідей, які можна чекати від сервера – з одного рядка й з декількох рядків. Щоб ідентифікувати кінець відповіді, протокол NNTP, як і протокол SMTP, застосовує символ “.” у рядку, (наприклад, відповідь на команди `list` і `article`).

3.3 Завдання до роботи

3.3.1 Ознайомитися з основними теоретичними відомостями за темою роботи.

3.3.2 Розробити програмного Web-агента, який взаємодіє із певними службами в Internet та збирає інформацію на основі критеріїв, що задані користувачем.

Програмний Web-агент має відповідати вимогам, наведеним в п. 3.3.2.1 – 3.3.2.5.

3.3.2.1 За допомогою протоколу HTTP Web-агент має перевіряти зміни заданих web-сайтів, на яких відображається потрібна інформація

(наприклад, новини). При цьому зв'язок з сайтами організується за допомогою інтерфейсу сокетів. Протокол мережевого рівня – IP, протокол транспортного рівня – TCP, протокол прикладного рівня – HTTP (порт 80). Перевірку на зміну сайтів з новинами рекомендовано виконувати шляхом перевірки розміру файлу, що пропонується скачати.

3.3.2.2 За допомогою протоколу HTTP Web-агент має відбирати на вказаних серверах потрібну інформацію (наприклад, на сайті новин, – статті), яка відповідає критеріям, що задані користувачем. Критеріями можуть бути: група новин та окремі слова, що мають зустрічатися в новині. Зв'язок з сервером новин має бути організований за допомогою інтерфейсу сокетів. Протокол мережевого рівня – IP, протокол транспортного рівня – TCP, протокол прикладного рівня – HTTP (порт 119). При цьому врахувати наступні особливості серверів новин:

- після підключення до серверу має бути повернуто повідомлення, що вміщує код 201 у випадку вдалого підключення;
- для встановлення заданої групи новин необхідно відправити команду: `group назва_групи`. При цьому в результаті успішності має бути повернуто повідомлення, що вміщує наступну послідовність: код 211, загальну кількість повідомлень даної групи, номер першого та останнього повідомлення;
- для завантаження заголовку повідомлення необхідно відправити команду: `head номер_повідомлення`. При цьому, якщо все було виконано успішно, то повертається код 221 та заголовок повідомлення;
- для завантаження усього повідомлення необхідно відправити серверу команду: `article номер_повідомлення`. Якщо повідомлення успішно передано, то повертається код 220 та саме повідомлення;
- для ідентифікації кінця відповіді сервера використовується рядок з символом “.”

3.3.2.3 Web-агент має надавати користувачу Web-інтерфейс. При цьому мають бути передбачені наступні можливості:

- можливість перегляду поточної конфігурації (тобто конфігурацію можна переглянути, прописавши в web-браузері посилання наступного типу: `http://192.168.115.2:8080/conf.html` – в випадку, коли web-агент працює на комп'ютері з ip-адресою 192.168.115.2 на порті 8080);
- можливість перегляду списку http-сайтів та http-серверів новин, на яких відбулися зміни (наприклад, `http://192.168.115.2:8080/index.html`). Тобто має бути відображена інформація в вигляді двох таблиць: таблиця http-сайтів, що зміни-

лися, та таблиця повідомлень на серверах новин. При цьому для цих повідомлень має бути відображена група, до якої вони відносяться, а також задані слова, що були знайдені в цих повідомленнях. Послідовність відображення новин має відповідати їхній відповідності критеріям користувача;

- можливість перегляду повідомлення з серверу новин у web-браузері користувача;

- надати можливість користувачу вказати http-сервери, що було переглянуто, тобто які вже не треба виводити у таблиці до їх наступної зміни.

3.3.2.4 Робота зі збору інформації про стан серверів та робота із користувачем мають не заважати одна одній.

3.3.2.5 Конфігурація роботи Web-агента має розміщуватися в окремому конфігураційному файлі, в якому можна вказати наступні параметри:

- http-сайти з новинами;
- сервери новин;
- групи новин та відповідні їм слова, що мають знаходитися в середині повідомлень.

3.3.3 Виконати тестування розробленого програмного агента.

3.3.4 З метою створення підсистеми автоматичного поповнення бази знань інтегрувати створеного програмного агента до інтелектуальної системи підтримки прийняття рішень, розробленої під час виконання лабораторної роботи з дисципліни “Обробка та менеджмент знань”. При необхідності з метою забезпечення інтеграції розробити додаткове програмне, інформаційне, лінгвістичне, математичне забезпечення.

3.3.5 Виконати тестування розробленого програмного забезпечення (інтелектуальна система підтримки прийняття рішень з інтегрованим Web-агентом) шляхом звертання за допомогою Web-агенту до заданих користувачем Web-сторінок із динамічно змінюваною інформацією з метою поповнення фактів бази знань.

3.3.6 Проаналізувати роботу створеного програмного агента. Виявити його переваги та недоліки.

3.3.7 Оформити звіт з роботи.

3.3.8 Відповісти на контрольні питання.

3.4 Зміст звіту

3.4.1 Тема та мета роботи.

3.4.2 Короткі теоретичні відомості.

3.4.3 Текст програми (із коментарями), що реалізує роботу Web-агенту.

3.4.4 Результати тестування Web-агенту.

3.4.5 Копії екранних форм, що реалізують взаємодію користувача із програмним агентом як складової частини інтелектуальної системи підтримки прийняття рішень.

3.4.6 Результати тестування Web-агенту як складової частини інтелектуальної системи підтримки прийняття рішень (отримані із Internet дані, поповнені факти бази знань тощо).

3.4.7 Висновки, що містять відповіді на контрольні запитання, а також відображують результати виконання роботи та їх критичний аналіз.

3.5 Контрольні питання

3.5.1 Що являє собою програмний агент?

3.5.2 Поясніть та проаналізуйте властивості агентів: автономність, адаптивність, комунікативність, здатність до співробітництва, персоніфікованість, мобільність.

3.5.3 З якою метою застосовуються агенти, використання яких залежить від завдань?

3.5.4 Що таке розважальні агенти?

3.5.5 Проаналізуйте комп'ютерні віруси як програмні агенти.

3.5.6 Які методи використовуються для надання інтелектуальності програмним агентам?

3.5.7 Які протоколи використовує Web-агент?

3.5.8 Виконайте аналіз архітектури Web-агенту?

3.5.9 Якими властивостями володіє Web-агент?

3.5.10 Проаналізуйте протокол, який використовується для передачі користувачу результатів пошуку за допомогою Web-агенту.

3.5.11 Наведіть і поясніть схему потоків усередині Web-агенту?

3.5.12 Проаналізуйте основні функції обраного інструментарію розробки Web-агенту: внутрішня структура, параметри, основні змінні, їх призначення та використання.

ЛІТЕРАТУРА

1. Bullnheimer B., Hartl R.F., Strauss C. Applying the ant system to the vehicle routing problem // *Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization* / In I. H. Osman S. Vo., S. Martello and C. Roucairol, editors. – Boston: Kluwer, 1998. – P. 109–120.
2. Costa D., Hertz A. Ants can colour graphs // *Journal of the Operational Research Society*. – 1997. – №48. – P. 295–305.
3. Dorigo M. *Optimization, Learning and Natural Algorithms*. – Milano: Politecnico di Milano, 1992. – 140 p.
4. Dorigo M., Gambardella L.M. Ant colonies for the traveling salesman problem // *BioSystems*. – 1997. – №43. – P. 73–81.
5. Gambardella L.M., Dorigo M. HAS-SOP: An hybrid ant system for the sequential ordering problem. – Lugano: CH, 1997. – P. 237–255.
6. Gambardella L.M., Taillard E., Agazzi G. Macs-vrptw: A multiple ant colony system for vehicle routing problems with time windows // *New Methods in Optimisation* / In D. Corne, M. Dorigo, and F. Glover, editors. – McGraw-Hill, 1999. – P. 63–79.
7. Maniezzo V. Exact and approximate nondeterministic tree-search procedures for the quadratic assignment problem. – Bologna: Università di Bologna, 1998 – 102 p.
8. Michel R., Middendorf M. An ACO algorithm for the shortest common supersequence problem // *New Methods in Optimisation* / Eds.: D. Corne, M. Dorigo, F. Glover. – McGraw-Hill, 1999. – P. 525–537.
9. Subbotin S., Oleynik A. Modifications of Ant Colony Optimization Method for Feature Selection // *The experience of designing and application of CAD systems in Microelectronics: Proceedings of the IX International Conference CADSM-2007 (20–24 February 2007)*. – Lviv: Publishing house of Lviv Polytechnic, 2007. – P. 493–494.
10. Джонс М.Т. Программирование искусственного интеллекта в приложениях. – М.: ДМК Пресс, 2004. – 312 с.
11. Олейник Ал.А. Сравнительный анализ методов оптимизации на основе метода муравьиных колоний // *Комп'ютерне моделювання та інтелектуальні системи: Збірник наукових праць* / За ред. Д.М. Пізи, С.О. Субботіна. – Запоріжжя: ЗНТУ, 2007. – С. 147–159.