

Лабораторна робота No 2

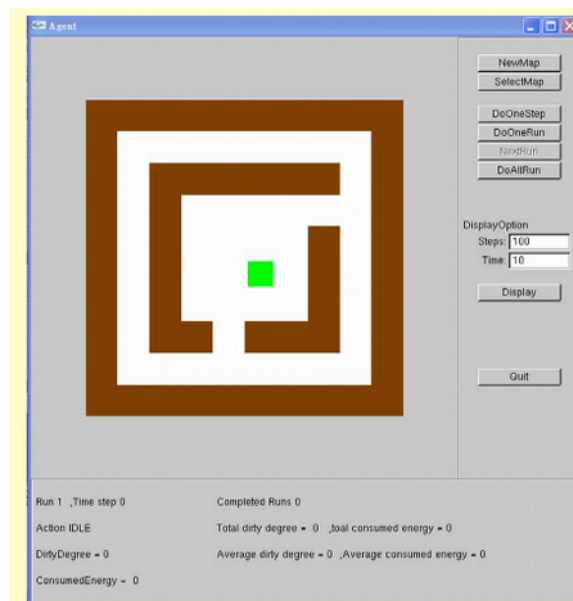
2.1 Інтелектуальні агенти

2.1.1 Ціль.

Розглянути та дослідити різні типи раціональних агентів. Порівняти ефективність їх поведінки.

В лабораторній роботі йдеться про «світ пилососу» - віртуальний світ, в якому існує один штучний агент – пилосос (див. тему 1 для опису поведінки агента-пилососа). Його задача – мінімізувати кількість бруду в заданому просторі.

В якості середовища моделювання використати проект «Vacuum Cleaner World» (<http://web.ntnu.edu.tw/~tcchiang/ai/Vacuum%20Cleaner%20World.htm>). Цей проект реалізований на мові C++ та реалізує середовище задачі і поведінку простого агента. Матеріал на англійській мові.



2.1.2 Завдання.

1) Необхідно розробити програми раціональних агентів двох типів: простого рефлексного агента та рефлексного агента, який заснований на моделі. Для створення власного агента потрібно модифікувати файли agent.cpp та agent.h оригінального проекту. Агент з моделлю середовища може отримувати інформацію про стан середовища через клас Environment (файли environment.cpp/.h)

2) Створити декілька власних карт середовища (на основі прикладу agent.map)

3) Провести серію експериментів для визначення ефективності роботи програм агентів.

Порівняти два створених агенти між собою, а також з оригінальним агентом (який йде в дистрибутиві проекту «Vacuum Cleaner World») та діє випадковим чином. Порівняння зробити для всіх створених карт, а також оригінальної карти agent.map. Вхідним параметром для експериментів вважати час моделювання (наприклад: 100, 500, 1000, 2000, 5000, 10000). За проведеними серіями необхідно визначити:

- середній ступінь бруду (average dirty degree);
- середню використану енергію (average consumed energy).

Результати експериментів представити у вигляді таблиць та графіків.

2.1.3 Звіт

За результатом виконання лабораторної роботи необхідно підготувати звіт. Звіт повинен містити:

1. Постановку задачі
2. Опис принципу роботи розроблених агентів
3. Результати проведення серії експериментів у вигляді таблиць та графіків
4. Аналіз отриманих результатів.
5. Висхідні коди програм агентів.

Звіт подається після здачі роботи в електронному вигляді.

2.1.4 Контрольні питання до цієї частини

Захист лабораторної роботи передбачає вміння давати відповіді на наступні питання:

1. Наведіть означення предметного середовища (PAES)? Які її складові частини?
2. Які є характеристики предметних середовищ?
3. Опишіть принципи роботи простих рефлексних агентів, рефлексних агентів на моделі, агентів на основі цілі та агентів на основі корисності.
4. Опишіть принцип, який покладений в основу агентів, що навчаються. Які складові частини їх архітектури?

2.2 Інформативний та неінформативний пошуки

2.2.1 Ціль.

Розглянути та дослідити алгоритми неінформативного та інформативного пошуку. Провести порівняльний аналіз ефективності використання алгоритмів.

2.2.2 Завдання.

1) Реалізувати програму, яка розв'язує поставлену задачу Task за допомогою алгоритму неінформативного пошуку AlgNoInf та алгоритму інформативного пошуку AlgInf, що використовує задану евристичну функцію Func.

Значення параметрів Task, AlgNoInf, AlgInf та Func визначаються за номером варіанту (див. нижче).

Програму реалізувати на довільній мові процедурного програмування (C, C++, C#, Java, ython, Matlab, PHP, ...) або функціонального (Lisp, Racket...).

Увага! Алгоритм неінформативного пошуку AlgNoInf реалізовується за принципом «AS IS», тобто так, як є, без додаткових модифікацій (таких як перевірка циклів, наприклад).

2) Провести серію експериментів для вивчення ефективності роботи алгоритмів. Кожний експеримент повинен відрізнятися початковим станом. Серія повинна містити не менше 20 експериментів для кожного алгоритму. За проведеними серіями необхідно визначити:

- середній час пошуку рішення у секундах
- середню кількість згенерованих станів під час пошуку
- середню кількість станів, що зберігаються в пам'яті під час роботи програми

Передбачити можливість обмеження виконання програми за часом (наприклад, 10 хвилин) та використання пам'яті (наприклад, 512 Мб)

2.2.3 Звіт

За результатом виконання лабораторної роботи необхідно підготувати звіт. Звіт повинен містити:

1. Постановку задачі
2. Формулювання проблеми та опис простору станів (див. лекцію 3).
3. Опис евристичної функції оцінки.
4. Результати проведення серії експериментів у вигляді таблиці.
5. Аналіз отриманих результатів (оптимальність, повнота, швидкість, використана пам'ять при роботі алгоритмів).

6. Висхідні коди алгоритмічної частини програм.

Звіт подається після здачі роботи в електронному вигляді.

2.2.4 Контрольні питання

Захист лабораторної роботи передбачає вміння давати відповіді на наступні питання:

1. Яка відмінність між алгоритмами інформативного та неінформативного пошуку?
2. Які алгоритми неінформативного пошуку є повними? Оптимальними?
3. Поясніть необхідність використання евристичних функцій при інформативному пошуку.
4. Якою повинна бути евристична функція для того, щоб алгоритми інформативного пошуку були оптимальними?
5. Які алгоритми інформативного пошуку є повними? Оптимальними? Чому?
6. Поясніть принцип роботи алгоритмів, які реалізовані у цьому етапі лабораторній роботі.

2.2.5 Варіанти

Варіант індивідуального завдання визначається за останніми двома цифрами залікової книжки.

Значення параметрів Task, AlgNoInf, AlgInf та Func для кожного варіанту за таблицею нижче. Використані позначення:

- 8-ферзів – задача розташування 8 ферзів на шахівниці. Зробити реалізацію задачі із повними станами, тобто у кожному стані (початковому включно) міститься всі 8 ферзів. Функція визначення наступника працює наступним чином: обирається один ферзь та переставляється у нову позицію (для неінформативного пошуку – це перша позиція за списком; у інформативному – краща за евристичною функцією).
- 8-puzzle - задача гри у 8. Початковий стан рекомендується отримувати шляхом рухів від цільового стану у зворотній бік (цим буде гарантуватись знаходження розв'язку та глибина дерева пошуку).
- LDFS - Пошук вглиб з обмеженням глибини
- BFS – Пошук вшир
- IDS – Пошук вглиб з ітеративним заглибленням
- A* – Пошук A*
- RBFS – Рекурсивний пошук за першим найкращим збігом

- F1 – кількість пар ферзів, які б'ють один одного з урахуванням видимості (ферзь А може стояти на одній лінії з ферзем В, проте між ними стоїть ферзь С; тому А не б'є В)
- F2 – кількість пар ферзів, які б'ють один одного без урахування видимості
- H1 – кількість фішок, які не стоять на своїх місцях
- H2 – Манхетенська відстань

NoNo	Task	AlgNoInf	AlgInf	Func
1	8-ферзів	LDFS	A*	F1
2	8-puzzle	LDFS	A*	H1
3	8-ферзів	LDFS	RBFS	F1
4	8-puzzle	LDFS	RBFS	H1
5	8-ферзів	BFS	A*	F1
6	8-puzzle	BFS	A*	H1
7	8-ферзів	BFS	RBFS	F1
8	8-puzzle	BFS	RBFS	H1
9	8-ферзів	IDS	A*	F1
10	8-puzzle	IDS	A*	H1
11	8-ферзів	IDS	RBFS	F1
12	8-puzzle	IDS	RBFS	H1
13	8-ферзів	LDFS	A*	F2
14	8-puzzle	LDFS	A*	H2
15	8-ферзів	LDFS	RBFS	F2
16	8-puzzle	LDFS	RBFS	H2
17	8-ферзів	BFS	A*	F2
18	8-puzzle	BFS	A*	H2
19	8-ферзів	BFS	RBFS	F2
20	8-puzzle	BFS	RBFS	H2
21	8-ферзів	IDS	A*	F2
22	8-puzzle	IDS	A*	H2
23	8-ферзів	IDS	RBFS	F2
24	8-puzzle	IDS	RBFS	H2

3.3 Локальний пошук

2.3.1 Ціль.

Розглянути та дослідити алгоритми локального пошуку для розв'язання задач оптимізації.

2.3.2 Завдання.

1) Реалізувати програму, яка розв'язує поставлену задачу Task за допомогою алгоритму локального пошуку AlgLocal для повного формулювання. Евристичну функцію обрати довільну для заданої задачі.

Значення параметрів Task, AlgLocal визначаються за номером варіанту (див. нижче).

Програму реалізувати на довільній мові процедурного програмування (C, C++, C#, Java, Python, Matlab, PHP, ...)

2) Провести 20 експериментів роботи алгоритму (для цього вміти генерувати випадковим чином початковий стан), які відрізняються характеристиками роботи алгоритму. За результатами експериментів визначити:

- середню кількість етапів (кроків), які знадобилось для досягнення розв'язку
- середню кількість випадків, коли алгоритм потрапляв в глухий кут (не міг знайти оптимальний розв'язок) – якщо таке можливе
- середній час роботи алгоритму в секундах

3) На додаткові бали. Порівняти результати розв'язання задачі за допомогою алгоритму локального пошуку із результатами з лабораторної роботи No2.

2.3.3 Звіт третього етапу ЛР

За результатом виконання лабораторної роботи необхідно підготувати звіт. Звіт повинен містити:

1. Постановку задачі
2. Формулювання проблеми та опис простору станів (див. Лекції 7-8).
3. Результати роботи алгоритмів (час, кількості етапів та повернень)
4. Висновки по роботі алгоритмів.
5. Вихідні коди алгоритмічної частини програм.

Звіт подається після здачі роботи в електронному вигляді.

2.3.4 Контрольні питання до етапу ЛР

Захист лабораторної роботи передбачає вміння давати відповіді на наступні питання:

1. Яка відмінність між алгоритмами звичайного пошуку та локального пошуку?
2. Поясніть в чому полягає складність роботи алгоритмів локального пошуку (пов'язана з ландшафтом станів).
3. Що таке задачі з обмеженнями (CSP)? Як вони формулюються?
4. Які складності виникають при розв'язанні задач у неперервних просторах станів?
5. Поясніть принцип роботи алгоритмів, які реалізовані у цьому етапі лабораторної роботи.

2.3.5 Варіанти

Варіант індивідуального завдання визначається за останніми двома цифрами залікової книжки.

Значення параметрів Task, AlgLocal, Evr1 та Constr для кожного варіанту за таблицею нижче. Використані позначення:

- 8-ферзів – задача розташування 8 ферзів на шахівниці. Зробити реалізацію задачі із повними станами, тобто у кожному стані (початковому включно) міститься всі 8 ферзів. Функція визначення наступника працює наступним чином: обирається один ферзь та переставляється у нову позицію (для неінформативного пошуку – це перша позиція за списком; у інформативному – краща за евристичною функцією).
- 8-puzzle - задача гри у 8. Початковий стан рекомендується отримувати шляхом рухів від цільового стану у зворотній бік (цим буде гарантуватись знаходження розв'язку та глибина дерева пошуку).
- HILL – Стохастичний пошук зі сходженням на вершину із використанням руху вбік (на 100 кроків) та випадковим перезапуском (кількість необхідних разів запуску визначити самостійно – див. лекцію 5).
- ANNEAL – Локальний пошук із симуляцією відпалу. Робоча характеристика – залежність температури T від часу роботи алгоритму t . Можна розглядати лінійну залежність: $T = 1000 - k \cdot t$, де k – змінний коефіцієнт.
- BEAM – Локальний променевий пошук. Робоча характеристика – кількість променів k . Експерименти проводи із кількістю променів від 2 до 21.
- TABU – Табу-пошук із короткотерміновою пам'яттю (табу-списком). Розмір списку та час перебування станів у списку визначати експериментальним шляхом

NoNo	Task	AlgLocal
1	8-ферзів	HILL
2	8-puzzle	HILL
3	8-ферзів	ANNEAL
4	8-puzzle	ANNEAL
5	8-ферзів	BEAM
6	8-puzzle	BEAM
7	8-ферзів	TABU
8	8-puzzle	TABU
9	8-ферзів	HILL
10	8-puzzle	HILL
11	8-ферзів	ANNEAL
12	8-puzzle	ANNEAL

Шкала оцінювання лабораторної №3

Кожний етап виконання лабораторної роботи

- Відповідь на контрольні питання: 1 бал

- Програмна реалізація алгоритмів: 5 балів
- Проведення серії експериментів та отримані висновки: 1 бали
- Наявність та інформативність звіту: 1 бал

Порівняння результатів 3.3 та 3.2 — 1.

В суммі можна набрати максимум 25 балів = $8 \cdot 3 + 1$.

Література

1. Електронний конспект (презентація).
2. Рассел, Норвіг «Штучний інтелект. Сучасний підхід» Глава 2-4.