

ĐẠI HỌC QUỐC GIA HÀ NỘI  
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN  
KHÓA VẬT LÝ



NHÓM 8

**XÂY DỰNG BOT CHƠI TRÒ CHƠI PACMAN**

Tiểu luận môn học

Học phần: Nhập môn Trí tuệ nhân tạo

(Lớp: K66 - Kỹ thuật Điện tử và Tin học)

**Hà Nội - 2024**

ĐẠI HỌC QUỐC GIA HÀ NỘI  
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN  
KHÓA VẬT LÝ



NHÓM 8

**XÂY DỰNG BOT CHƠI TRÒ CHƠI PACMAN**

Tiểu luận môn học

Học phần: Nhập môn Trí tuệ nhân tạo

(Lớp: K66 - Kỹ thuật Điện tử và Tin học)

**Giảng viên: PGS.TS Thân Quang Khoát**

**Hà Nội - 2024**

## **THÀNH VIÊN NHÓM 8**

1. Lê Trung Kiên - 21002214
2. Lê Sơn Tùng - 21002243
3. Phạm Thu Thủy - 21002240
4. Nguyễn Mỹ Anh - 21002187

## ***Lời cảm ơn***

Cuộc Cách mạng công nghiệp lần thứ 4 (CMCN 4.0) đã và đang diễn ra mạnh mẽ tại nhiều quốc gia, trong đó có Việt Nam. Có thể nói, việc tiếp cận và ứng dụng trí tuệ nhân tạo (AI) vào cuộc sống đang trở nên rất cấp thiết và đóng vai trò quan trọng trong sự phát triển của mỗi ngành, lĩnh vực và của mỗi quốc gia. Trong chương trình đại học, môn học "Nhập môn Trí tuệ nhân tạo" đã giúp chúng tôi có cái nhìn chi tiết hơn về khái niệm cũng như tính ứng dụng của lĩnh vực này vào trong đời sống.

Trong quá trình hoàn thành bài báo cáo này, đầu tiên xin cho phép nhóm được gửi lời cảm ơn đến thầy Thân Quang Khoát. Trong quá trình giảng dạy của môn học, thầy không chỉ là người truyền đạt kiến thức về môn học mà còn là người truyền cảm hứng với mỗi thành viên của nhóm về môn học này. Chúng tôi xin được cảm ơn anh Nguyễn Lâm Cường với những tiết bài tập được anh giảng dạy. Bên cạnh đó, chúng tôi xin được cảm ơn các thầy trong bộ môn Tin học Vật lý, các thầy đã luôn tạo điều kiện để chúng tôi có thể hoàn thành bài tập lớn này. Lời cuối cùng, xin được cảm ơn tất cả các thành viên trong nhóm đã không ngại khó khăn, luôn đoàn kết cùng hỗ trợ lẫn nhau để hoàn thành đề tài này.

# Mục lục

<b>Lời cảm ơn</b>	<b>ii</b>
<b>Danh sách hình vẽ</b>	<b>v</b>
<b>Danh sách tên viết tắt</b>	<b>vii</b>
<b>MỞ ĐẦU</b>	<b>1</b>
<b>1 TỔNG QUAN VỀ PACMAN GAME</b>	<b>2</b>
1.1 Nguyên tắc trò chơi . . . . .	2
1.2 Tác tử Pacman . . . . .	2
1.2.1 PEAS . . . . .	2
1.2.2 Môi trường hoạt động của tác tử . . . . .	2
1.2.3 Mục tiêu . . . . .	3
1.3 Tác tử ma . . . . .	3
1.3.1 PEAS . . . . .	3
1.3.2 Mục tiêu . . . . .	4
1.4 Một số tính năng của ứng dụng . . . . .	4
<b>2 TRIỂN KHAI BÀI TOÁN</b>	<b>5</b>
2.1 Thiết kế giao diện . . . . .	5
2.2 Thiết lập công cụ trò chơi . . . . .	5
2.3 Các thuật toán áp dụng . . . . .	5
2.3.1 Bài toán đơn tác tử . . . . .	5
DFS . . . . .	5
BFS . . . . .	6
UCS . . . . .	7
A* . . . . .	8
2.3.2 Bài toán đa tác tử . . . . .	8
Model-based Reflex Agent . . . . .	8
Minimax và Alpha-Beta Prunning . . . . .	10
Expectimax . . . . .	11
Tác tử ma "thông minh" . . . . .	12
<b>3 Kết quả</b>	<b>13</b>
3.1 Maze: Contest Classic . . . . .	13
3.2 Maze: Medium Classic . . . . .	15
	iii

3.3	Maze: Minimax Classic . . . . .	17
3.4	Maze: Open Classic . . . . .	19
3.5	Maze: Capsule Classic . . . . .	21
3.6	Maze: Small Classic . . . . .	23
3.7	Maze: Test Classic . . . . .	25
3.8	Maze: Trapped Classic . . . . .	27
3.9	Maze: Tricky Classic . . . . .	29
<b>4</b>	<b>Kết luận</b>	<b>31</b>
4.1	Đánh giá . . . . .	31
4.2	Khó khăn . . . . .	31
4.3	Hướng phát triển của đề tài . . . . .	32
4.4	Về bản thân . . . . .	32
	<b>Tài liệu tham khảo</b>	<b>33</b>
<b>A</b>	<b>Bảng số liệu chi tiết</b>	<b>34</b>
A.1	Tác tử Reflex . . . . .	34
A.2	Tác tử Minimax . . . . .	35
A.3	Tác Tử Alphabeta . . . . .	36
A.4	Tác tử Expectimax . . . . .	37
A.5	Bảng tổng hợp . . . . .	38

## Danh sách hình vẽ

2.1	Mô tả giải thuật DFS . . . . .	5
2.2	Minh họa giải thuật DFS . . . . .	6
2.3	Mô tả giải thuật BFS . . . . .	6
2.4	Minh họa giải thuật BFS . . . . .	7
2.5	Mô tả giải thuật UCS . . . . .	7
2.6	Mô tả Model-based Reflex Agent . . . . .	8
2.7	Mô tả thuật toán Minimax . . . . .	10
2.8	Mô tả thuật toán Alpha-beta . . . . .	11
2.9	Mô tả thuật toán Minimax . . . . .	11
3.1	Điểm trung bình trong map Contest Classic . . . . .	13
3.2	Tỉ lệ thắng trong map Contest Classic . . . . .	14
3.3	Thời gian trung bình trong map Contest Classic . . . . .	14
3.4	Điểm trung bình trong map Medium Classic . . . . .	15
3.5	Tỉ lệ thắng trong map Medium Classic . . . . .	15
3.6	Thời gian trung bình trong map Medium Classic . . . . .	16
3.7	Điểm trung bình trong map Minimax Classic . . . . .	17
3.8	Tỉ lệ thắng trong map Minimax Classic . . . . .	17
3.9	Thời gian trung bình trong map Minimax Classic . . . . .	18
3.10	Điểm trung bình trong map Open Classic . . . . .	19
3.11	Tỉ lệ thắng trong map Open Classic . . . . .	19
3.12	Thời gian trung bình trong map Open Classic . . . . .	20
3.13	Điểm trung bình trong map Capsule Classic . . . . .	21
3.14	Tỉ lệ thắng trong map Capsule Classic . . . . .	21
3.15	Thời gian trung bình trong map Capsule Classic . . . . .	22
3.16	Điểm trung bình trong map Small Classic . . . . .	23
3.17	Tỉ lệ thắng trong map Small Classic . . . . .	23
3.18	Thời gian trung bình trong map Small Classic . . . . .	24
3.19	Điểm trung bình trong map Test Classic . . . . .	25
3.20	Tỉ lệ thắng trong map Test Classic . . . . .	25
3.21	Thời gian trung bình trong map Test Classic . . . . .	26
3.22	Điểm trung bình trong map Trapped Classic . . . . .	27
3.23	Tỉ lệ thắng trong map Trapped Classic . . . . .	27
3.24	Thời gian trung bình trong map Trapped Classic . . . . .	28
3.25	Điểm trung bình trong map Tricky Classic . . . . .	29
3.26	Tỉ lệ thắng trong map Trickiest Classic . . . . .	29

3.27	Thời gian trung bình trong map Tricky Classic . . . . .	30
A.1	Bảng số liệu ma ngẫu nhiên Reflex . . . . .	34
A.2	Bảng số liệu ma thông minh Reflex . . . . .	34
A.3	Bảng số liệu Minimax ngẫu nhiên Minimax . . . . .	35
A.4	Bảng số liệu ma thông minh Minimax . . . . .	35
A.5	Bảng số liệu ma ngẫu nhiên Alphabeta . . . . .	36
A.6	Bảng số liệu ma thông minh Alphabeta . . . . .	36
A.7	Bảng số liệu ma ngẫu nhiên Expectimax . . . . .	37
A.8	Bảng số liệu ma thông minh Expectimax . . . . .	37
A.9	Bảng tổng hợp . . . . .	38



## **Danh sách tên viết tắt**

<b>DFS</b>	<b>Depth First Search</b>
<b>BFS</b>	<b>Breadth First Search</b>
<b>UCS</b>	<b>Uniform Cost Search</b>

## MỞ ĐẦU

Trò chơi Pac-Man là một trò chơi điện tử vừa mang tính thử thách vừa mang tính giải trí, là trọng tâm của một số nghiên cứu quan trọng về Trí tuệ nhân tạo. Mục tiêu của việc sử dụng trò chơi Pac-Man làm nơi thử nghiệm vì trò chơi Pac-Man cung cấp một nền tảng đủ phong phú và đầy thách thức để nghiên cứu AI trong trò chơi máy tính và nó đủ đơn giản để cho phép hiểu về các đặc trưng của nó.

Để đạt được những mục tiêu mong muốn này, chúng tôi đã sử dụng nhiều phương pháp tìm kiếm sáng suốt như DFS, BFS, UCS, A\*, Minimax, ExpectiMax, Reflex,... với các bản đồ khác nhau. Hơn nữa, chúng tôi đã sử dụng tính năng cắt tỉa Alpha Beta để giảm không gian tìm kiếm. Ngoài ra, chúng tôi đã kết hợp các tác nhân ma để tìm kiếm thông minh dựa trên nhiều biến số khác nhau như khoảng cách từ pacman, địa điểm ăn thức ăn, v.v. Cuối cùng, kết quả thử nghiệm từ trò chơi Pac-Man được trình bày để chứng minh tính hiệu quả của các thuật toán này.

# Chương 1 TỔNG QUAN VỀ PACMAN GAME

## 1.1 Nguyên tắc trò chơi

Pacman di chuyển trong một mê cung, ăn thức ăn và lẩn tránh kẻ thù là những con ma. Trong bài tập lớn lần này, nhóm sẽ xây dựng 2 tác tử:

- Pacman: có nhiệm vụ tìm kiếm thức ăn trong mê cung đồng thời phải tránh ma.
- Ma: có nhiệm vụ săn Pacman.

## 1.2 Tác tử Pacman

### 1.2.1 PEAS

Để thiết kế một tác tử thông minh (hợp lý), trước tiên cần phải xác định (thiết lập) các giá trị của các thành phần của PEAS:

- Đánh giá hiệu quả hoạt động (P): điểm trung bình, tỷ lệ thắng, thời gian tính toán.
- Môi trường xung quanh (E): mê cung chứa chấm thức ăn, viên năng lượng, ma.
- Bộ phận hành động (A): phím mũi tên (trong trường hợp người dùng).
- Bộ phận cảm biến (S): màn hình hiển thị.

### 1.2.2 Môi trường hoạt động của tác tử

- Observable: Fully - Pacman quan sát được các trạng thái đầy đủ của môi trường, vì trạng thái được duy trì.
- Deterministic: Strategic - Môi trường là xác định, ngoại trừ đối với các hành động của các tác tử khác. Hành động của tác tử mang tính chiến lược.
- Episodic: Sequential - Mỗi quyết định của tác tử phụ thuộc vào trạng thái trước đó.
- Observable: Static: Dynamic - Môi trường liên tục thay đổi, tức là tác tử di chuyển, giảm số lượng thức ăn, v.v.

- Deterministic: Discrete: Discrete - Tập các nhận thức và các hành động là hữu hạn, được định nghĩa phân biệt rõ ràng
- Agents: Multi-Agent - Đa tác tử

### **1.2.3 Mục tiêu**

Mục tiêu là xây dựng tác tử pacman có thể hành động hợp lý mà không cần sự điều khiển của người dùng. Tiếp theo, xác định mục tiêu cho Pacman đó là:

- Ăn hết các chấm thật nhanh.
- Nhận được càng nhiều điểm càng tốt.
- Không chết.

Một số hành động hợp lý cho pacman sẽ cho phép nó tối đa hóa các mục tiêu của mình bao gồm:

- Hướng tới thức ăn.
- Tránh ma
- Hướng tới viên sức mạnh

## **1.3 Tác tử ma**

### **1.3.1 PEAS**

Đánh giá hiệu quả hoạt động (P): ). Mục tiêu là để ngăn chặn Pac-Man và gây ra thất bại cho người chơi Môi trường xung quanh (E): lưới của các ô vuông trong trò chơi Pac-Man, bao gồm các viên bi năng lượng, các điểm thức ăn, và Pac-Man cùng với các con ma khác Bộ phận hành động (A): di chuyển trong các hướng khác nhau trên lưới của trò chơi Pac-Man. Bộ phận cảm biến (S): Cảm biến của ghost agent sẽ thu thập thông tin về môi trường bao gồm vị trí của Pac-Man.

### **1.3.2 Mục tiêu**

Tác tử ma di chuyển ngẫu nhiên (Random Agent): Di chuyển ngẫu nhiên trong lưới trò chơi (trong maps). Tác tử ma di chuyển thông minh (Smart Agent):

- Săn pacman trong trạng thái tấn công
- Tránh xa pacman trong trạng thái phòng thủ (lẩn trốn)

### **1.4 Một số tính năng của ứng dụng**

Giao diện GUI để lựa chọn các loại tác nhân Pac-man, tác tử ma và bố cục mê cung. Khả năng chơi nhiều lần để tạo ra kết quả tích lũy khi hoạt động của thuật toán. Trực quan hóa thuật toán trong game pac-man. Tính hiệu quả của thuật toán được thể hiện thông qua mô-đun tính điểm. Khả năng chuyển đổi nhiều thuật toán pac-man. /citefelstiner2019alpha

## Chương 2 TRIỂN KHAI BÀI TOÁN

### 2.1 Thiết kế giao diện

Giao diện GUI của dự án được thiết kế bằng công cụ thiết kế GUI Python có tên PyQt5. Trình thiết kế PyQt5 cho phép khả năng thiết kế GUI kéo và thả dễ dàng và cũng cho phép dễ dàng chuyển đổi tệp giao diện người dùng thành mã Python.

### 2.2 Thiết lập công cụ trò chơi

Trò chơi pacman đã được phát triển bằng thư viện tkinter để hiển thị tất cả các thực thể của thế giới pacman bao gồm viên thức ăn, viên sức mạnh, pacman, ghost, mê cung,...

### 2.3 Các thuật toán áp dụng

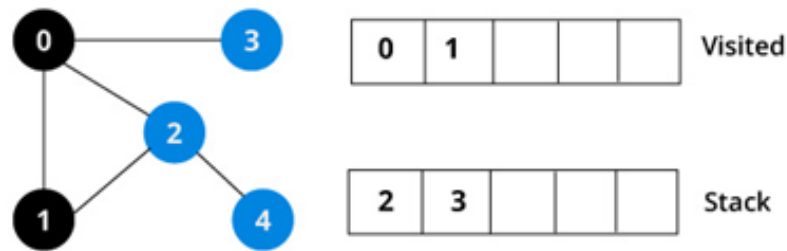
#### 2.3.1 Bài toán đơn tác tử

##### DFS

DFS tìm kiếm một cách sâu dọc qua các đường đi có thể. Khi thăm một ô, Pacman sẽ tiếp tục đi sâu vào đường đi hiện tại cho đến khi không còn đường đi nào hoặc gặp phải một ô không thể đi tiếp được.

```
DFS (N, A,  $n_0$ , ĐÍCH)
{
    fringe  $\leftarrow n_0$ ;
    closed  $\leftarrow \emptyset$ ;
    while (fringe  $\neq \emptyset$ ) do
    {
         $n \leftarrow \text{GET\_FIRST}(\text{fringe});$  // lấy phần tử đầu tiên của fringe
        closed  $\leftarrow \text{closed} \oplus n$ ;
        if ( $n \in \text{ĐÍCH}$ ) then return SOLUTION( $n$ );
        if ( $\Gamma(n) \neq \emptyset$ ) then  $\text{fringe} \leftarrow \Gamma(n) \oplus \text{fringe};$ 
    }
    return ("No solution");
}
```

Hình 2.1: Mô tả giải thuật DFS



Hình 2.2: Minh họa giải thuật DFS

Pacman sử dụng một cấu trúc dữ liệu là Stack để lưu trữ các ô cần thăm, đảm bảo rằng nó thăm các ô theo chiều sâu trước khi quay lại thăm các ô khác.

## BFS

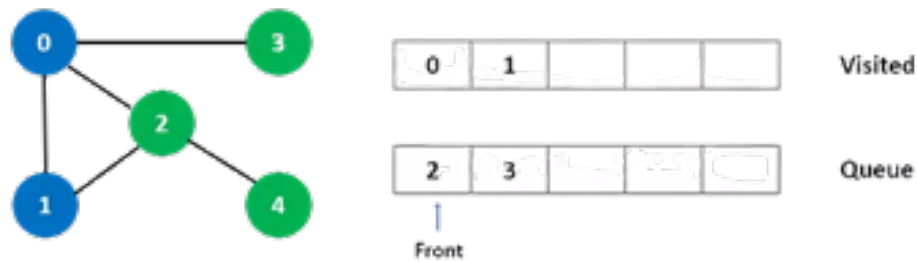
Trong trò chơi Pacman, thuật toán BFS giúp Pacman tính toán đường đi ngắn nhất đến các viên thức ăn hoặc đến các vị trí an toàn tránh hồn ma. Tìm đường đi đến viên thức ăn gần nhất: Mỗi khi Pacman ăn một viên thức ăn hoặc một viên thức ăn biến mất, BFS tính toán đường đi ngắn nhất từ vị trí hiện tại của Pacman đến các viên thức ăn còn lại. Sau đó, Pacman có thể di chuyển theo đường đi đó để tiếp tục ăn. Tránh hồn ma: Khi Pacman cần tránh hồn ma, BFS tính toán đường đi ngắn nhất từ vị trí hiện tại của Pacman đến các vị trí an toàn. Điều này giúp Pacman tìm ra các vị trí mà nó có thể đi để tránh hồn ma trong một khoảng thời gian ngắn.

```

BFS (N, A,  $n_0$ , ĐÍCH)
{
    fringe  $\leftarrow$   $n_0$ ;
    closed  $\leftarrow$   $\emptyset$ ;
    while (fringe  $\neq$   $\emptyset$ ) do
    {
         $n \leftarrow \text{GET\_FIRST}(\text{fringe});$            // lấy phần tử đầu tiên của fringe
        closed  $\leftarrow$  closed  $\oplus$  n;
        if ( $n \in \text{ĐÍCH}$ ) then return SOLUTION(n);
        if ( $\Gamma(n) \neq \emptyset$ ) then fringe  $\leftarrow$  fringe  $\oplus$   $\Gamma(n)$ ;
    }
    return ("No solution");
}

```

Hình 2.3: Mô tả giải thuật BFS



Hình 2.4: Minh họa giải thuật BFS

Khi thăm một ô, Pacman sẽ kiểm tra tất cả các ô kề của ô hiện tại và thêm chúng vào hàng đợi để thăm sau. Pacman sử dụng một cấu trúc dữ liệu là Queue để lưu trữ các ô cần thăm, đảm bảo rằng nó thăm các ô theo thứ tự từ gần đến xa.

## UCS

UCS (Uniform Cost Search) là một thuật toán tìm kiếm đường đi ngắn nhất trong các đồ thị có trọng số, tức là mỗi cạnh trong đồ thị có một trọng số khác nhau. Trong trò chơi Pacman, UCS được sử dụng để tính toán đường đi ngắn nhất từ Pacman đến các viên thức ăn hoặc để tránh hồn ma. Trọng số trong cây trò chơi thường là khoảng cách (khoảng cách Manhattan) hoặc chi phí di chuyển từ một ô đến ô khác (số bước di chuyển để làm trọng số).

```

UCS (N, A,  $n_0$ , ĐÍCH, c)
{
    fringe  $\leftarrow n_0$ ;
    closed  $\leftarrow \emptyset$ ;
    while (fringe  $\neq \emptyset$ ) do
    {
         $n \leftarrow \text{GET\_LOWEST\_COST}(\text{fringe})$ ; // lấy phần tử có chi phí // đường đi  $c(n)$  nhỏ nhất
        closed  $\leftarrow$  closed  $\oplus$  n;
        if (n  $\in$  ĐÍCH) then return SOLUTION(n);
        if ( $\Gamma(n) \neq \emptyset$ ) then fringe  $\leftarrow$  fringe  $\oplus$   $\Gamma(n)$ ;
    }
    return ("No solution");
}

```

Hình 2.5: Mô tả giải thuật UCS



## A\*

A\* là một thuật toán tìm kiếm thông minh kết hợp giữa BFS và heuristic để tìm kiếm đường đi tối ưu. Khi thăm một ô, Pacman tính toán giá trị  $f = g + h$ , trong đó  $g$  là chi phí thực hiện từ ô ban đầu đến ô hiện tại và  $h$  là giá trị ước lượng của chi phí từ ô hiện tại đến mục tiêu. Pacman sử dụng các giá trị  $f$  để quyết định xem ô nào sẽ thăm tiếp theo, ưu tiên các ô có giá trị  $f$  nhỏ nhất.

### 2.3.2 Bài toán đa tác tử

#### Model-based Reflex Agent

Đầu tiên, dự án liên quan đến việc Pacman trở thành một tác nhân phản xạ dựa trên mô hình, gợi ý hành động dựa trên trạng thái trước đó. Mã giả bên dưới mô tả về tác nhân phản xạ dựa trên mô hình.

```
function REFLEX-AGENT-WITH-STATE(percept) returns an action
  static:   state, a description of the current world state
             rules, a set of condition-action rules
             action, the most recent action, initially none

  state <-- UPDATE_INPUT(state, action, percept)
  rule <-- RULE_MATCH(state, rules)
  action <-- RULE_ACTION[rule]
  return action
```

Hình 2.6: Mô tả Model-based Reflex Agent

Để tác nhân quyết định hành động dựa trên trạng thái trò chơi hiện tại, nhóm đã sử dụng chức năng đánh giá. Hàm đánh giá được sử dụng để tính giá trị tiện ích cho trạng thái trò chơi. Hàm đánh giá được sử dụng trong dự án này bao gồm các tính năng sau: Việc thực hiện liên quan đến hàm số mũ cho khoảng cách ma. Khi những bóng ma đến gần pacman hơn, tiện ích sẽ giảm theo cấp số nhân, do đó pacman sẽ tránh được điều đó. Những con ma càng đuổi theo pacman càng muốn ăn đồ ăn. Hơn nữa, pacman sẽ tránh một con ma bị vô hiệu hóa, vì ăn nó sẽ tạo ra một con ma hoạt động ở trung tâm và mất điểm thưởng, vì vậy nếu được lựa chọn, nó sẽ có xu hướng để những con ma đó yên và tận dụng tối đa thời gian sợ hãi đó càng tốt. **Hàm đánh giá:** [1]

$f(n) = [\text{khoảng cách đến đồ ăn gần nhất}] + [\text{khoảng cách từ hồn ma đang hoạt động}] + [\text{khoảng cách từ hồn ma bị vô hiệu hóa}]$

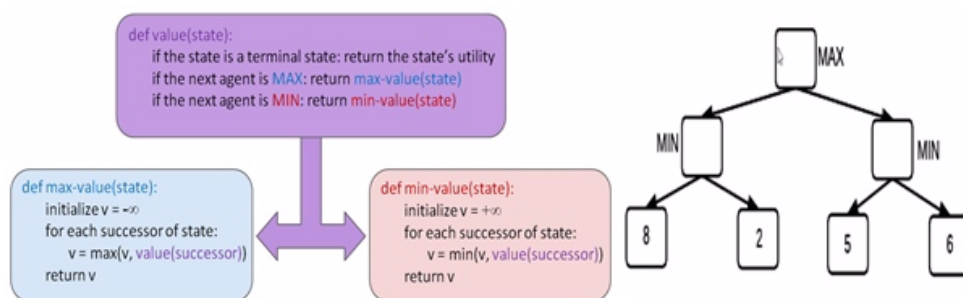
**Đặt vấn đề:** Vì nó dựa trên mô hình, do đó hàm đánh giá lấy các cặp trạng thái - hành động làm đầu vào để đánh giá các hành động từ trạng thái hiện tại. Tuy nhiên, các tác nhân phản xạ chọn hành động dựa trên nhận thức hiện tại về thế giới thay vì lập kế hoạch hoặc xem xét hậu quả trong tương lai. Đây là một quy tắc tốt trong thời gian ngắn để sống sót, nhưng có thể khiến pacman bị mắc kẹt trong một góc. Do đó, nhóm đã kết hợp sử dụng tác nhân dựa trên mục tiêu thay vì tác nhân phản xạ. Cần một mô hình bao gồm các yếu tố môi trường cần thiết để đo lường sự thành công. Ví dụ: nếu mục tiêu trong pacman là ăn các chấm thì cần phải có thước đo xem có bao nhiêu chấm trong môi trường. Nếu mục tiêu là ghi điểm thì cần phải đo điểm. Biết được bao nhiêu điểm được ghi cho một hành động nhất định, so với các hành động khác, cung cấp cho pacman thông tin để chọn một hành động. Trò chơi có đối thủ thường là những gì chúng ta nghĩ đến khi nghĩ đến trò chơi. Trong những trò chơi này, có một người chơi khác, trong trường hợp của dự án này là ma (ghost), có động cơ khác với pacman và do đó cần có thuật toán tìm kiếm xác định trạng thái tốt nhất từ một tập hợp các trạng thái có thể xem xét cả mục tiêu của bạn và hành động của đối thủ. Một vấn đề tìm kiếm điển hình phải bao gồm:

- States:  $S$  (start at  $s_0$ )
- Players:  $P=1\dots N$  (usually take turns)
- Actions:  $A$  (set of legal moves in state  $s$ )
- Transition Function:  $S \times A \rightarrow S$  (Successor functions that return a state based on a certain action)
- Players:  $P=1\dots N$  (usually take turns)
- Terminal Test:  $S \ni t, f$  (States at the end of the game. Win or lose!)
- Terminal Utilities:  $S \times P \rightarrow R$  (defines the numeric value for game that ends in terminal state  $s$  for player  $p$ )

Cây tìm kiếm đối nghịch được sử dụng trong dự án này bao gồm các trạng thái trò chơi, trong đó các nút trong cây được xác định bởi các hành động trong trò chơi và các chi tiết cụ thể của cây phụ thuộc vào các quy tắc của trò chơi. Giải pháp mà thuật toán cung cấp ánh xạ trạng thái tới hành động.

## Minimax và Alpha-Beta Pruning

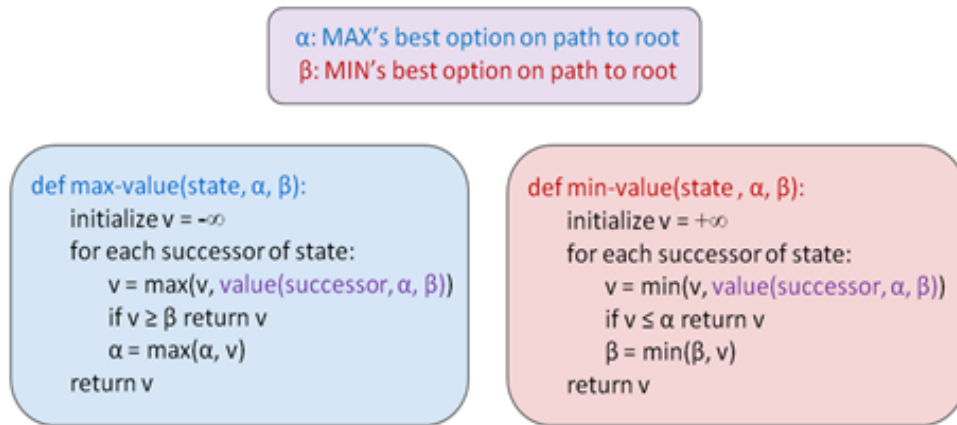
Trong tìm kiếm minimax, chúng ta bắt đầu với một cây trò chơi về tất cả các trạng thái trong tương lai có thể xảy ra. Ở phần dưới cùng của cây, hoặc phần dưới cùng đang được khám phá là kết quả của trò chơi và giá trị hàm tiện ích của kết quả đó. [2] Thuật toán minimax yêu cầu duyệt cây trò chơi một cách đệ quy. Bắt đầu từ gốc, MAX chọn hành động trả về giá trị tối đa trong số các giá trị tối thiểu của các phần tử con của nó. Thuật toán bắt đầu bằng lệnh gọi `minimax( )`, với nút để đánh giá, độ sâu hiện tại và liệu MAX hay MIN đưa ra quyết định làm đối số. Sơ đồ dưới đây mô tả thuật toán minimax:



Hình 2.7: Mô tả thuật toán Minimax

Trong bất kỳ cây tìm kiếm nào, có những nhánh trên cây không chứa chiến lược tối ưu cho trò chơi và có thể phát hiện những nhánh này và cắt tỉa chúng để không lãng phí thời gian tìm kiếm. Việc cắt bớt các nhánh vô dụng khỏi cây tìm kiếm làm cho minimax nhanh hơn nhiều, trong một số trường hợp, nhanh hơn theo cấp số nhân. Thuật toán Alpha-beta tìm ra giải pháp tránh các cây con không chứa giải pháp tối ưu. Việc cắt tỉa Alpha-beta đặt ra hai giới hạn cho các giá trị được quan sát trong quá trình tính toán. Sơ đồ dưới đây mô tả thuật toán: [3]

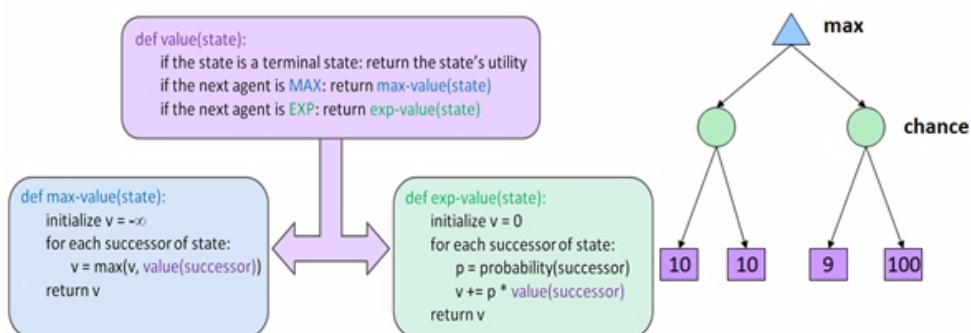
Trong một số trường hợp, có khả năng đối thủ (minimizer) có thể không tối ưu, do đó maximizer có thể tận dụng lợi thế của nó thay vì tìm kiếm giải pháp tối ưu.



Hình 2.8: Mô tả thuật toán Alpha-beta

## Expectimax

Thuật toán Expectimax, một biến thể của thuật toán Minimax, sử dụng cách tiếp cận tương tự. Các nút chance liên quan ở đây lấy mức trung bình của tất cả các giá trị hàm tiện ích có thể để đưa ra 'giá trị hàm tiện ích được mong đợi'. Sơ đồ dưới đây mô tả cho thuật toán: [4]



Hình 2.9: Mô tả thuật toán Minimax

Vấn đề là trong thực tế, chúng ta không thể tìm kiếm toàn bộ cây không gian của trò chơi vì nó đòi hỏi nhiều tính toán. Do đó giải pháp là tìm kiếm giới hạn độ sâu. Thay vào đó, nhóm em chỉ tìm kiếm ở độ sâu giới hạn trong cây và thay thế các terminal utilities bằng hàm đánh giá cho các vị trí non-terminal. Điều này có thể giúp chúng ta tiết kiệm chi phí tính toán khi đánh giá tất cả các trạng thái có thể. Tuy nhiên, các chức năng đánh giá luôn không hoàn hảo. Hàm đánh giá càng bị chôn sâu trong cây thì chất lượng của hàm đánh giá càng kém. Đây là một ví dụ quan trọng về sự cân bằng giữa độ phức tạp của các tính năng và độ phức tạp của tính toán. Trong dự án này, việc tìm kiếm được giới

hạn ở độ sâu cố định và khi đạt đến độ sâu đó, đường dẫn đến giá trị đánh giá cao nhất sẽ được chọn. Tìm kiếm sâu hơn sẽ mang lại kết quả tốt hơn nhưng cũng sẽ cần nhiều thời gian hơn. Hàm đánh giá được sử dụng ở đây cho các tác nhân dựa trên mục tiêu sẽ đánh giá các trạng thái thay vì hành động, như chúng ta đã làm đối với tác nhân phản xạ. Thay vì chỉ đánh giá trạng thái kế tiếp cho pacman , nhóm em đã đánh giá trạng thái kế tiếp cho tất cả các tác nhân có liên quan. Hơn nữa, không giống như các tác nhân phản xạ, chúng ta đang đánh giá các trạng thái. [5]

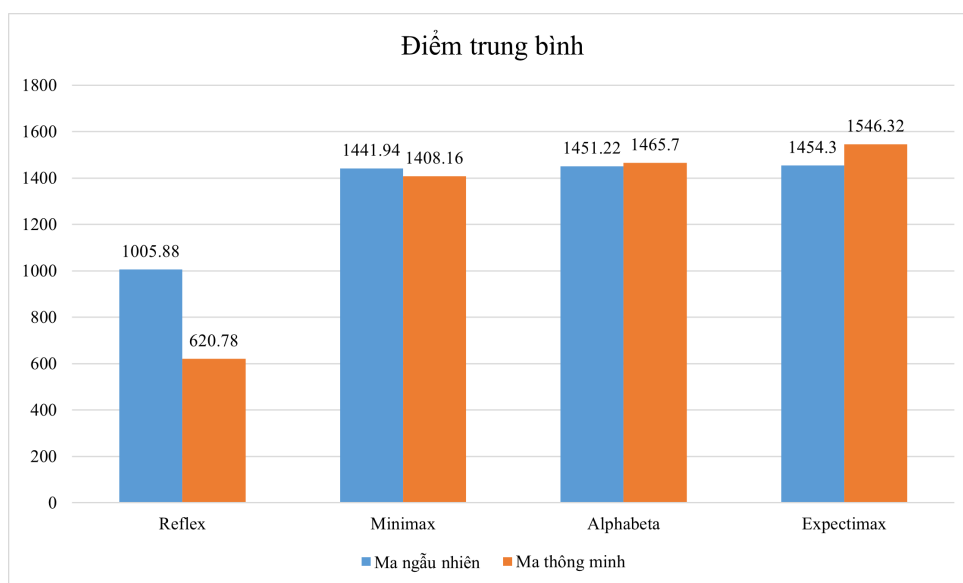
### **Tác tử ma "thông minh"**

Nhóm đã sử dụng hai loại tác tử ma trong dự án là ma di chuyển ngẫu nhiên (Random Agent) và ma di chuyển 'thông minh' (Smart Agent). Bên cạnh các con ma di chuyển ngẫu nhiên trong map, nhóm đã tạo một phiên bản thông minh hơn của con ma - chọn hành động của nó dựa trên khoảng cách với pacman . Trong trường hợp phòng thủ, nó sẽ chạy đến vị trí có khoảng cách xa nhất so với pacman , còn ở chế độ tấn công, nó sẽ tiến đến vị trí có khoảng cách tối thiểu với pacman . Hơn nữa, để đạt hiệu quả, nhóm đã kết hợp xác suất tấn công và phòng thủ của ma, để nó chọn hành động dựa trên tính chất xác suất của nó.

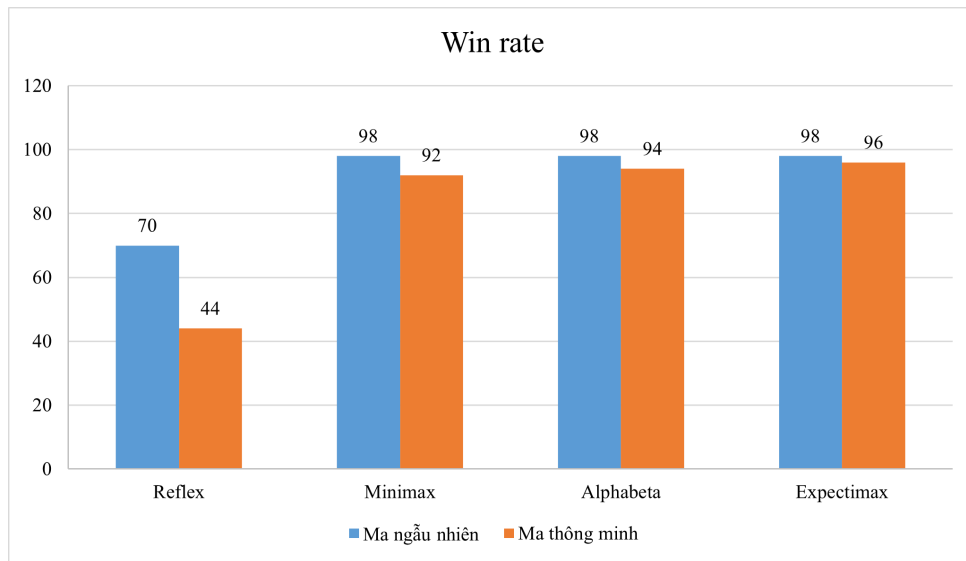
## Chương 3 Kết quả

Phần này nêu bật các kết quả thử nghiệm và những hiểu biết sâu sắc thu được từ chúng. Thiết lập thử nghiệm bao gồm chạy lặp lại 10 lần, sau đó ghi lại điểm trung bình, tỷ lệ thắng trung bình, thời gian thực hiện trung bình; chạy 10 lần như vậy trên mỗi tác tử (5 với ma ngẫu nhiên, 5 với ma thông minh). Hàm đánh giá là giống nhau đối với cả tác tử phản xạ cũng như tác tử dựa trên mục tiêu để tránh sự thiên vị trong đánh giá. Độ sâu cho các tác tử dựa trên mục tiêu đã bị giới hạn ở độ sâu = 2 vì độ sâu càng tăng khiến thời gian tính toán dài hơn nhiều. Các sơ đồ dưới đây tóm tắt tất cả các kết quả thử nghiệm cho tất cả các mô-cung.

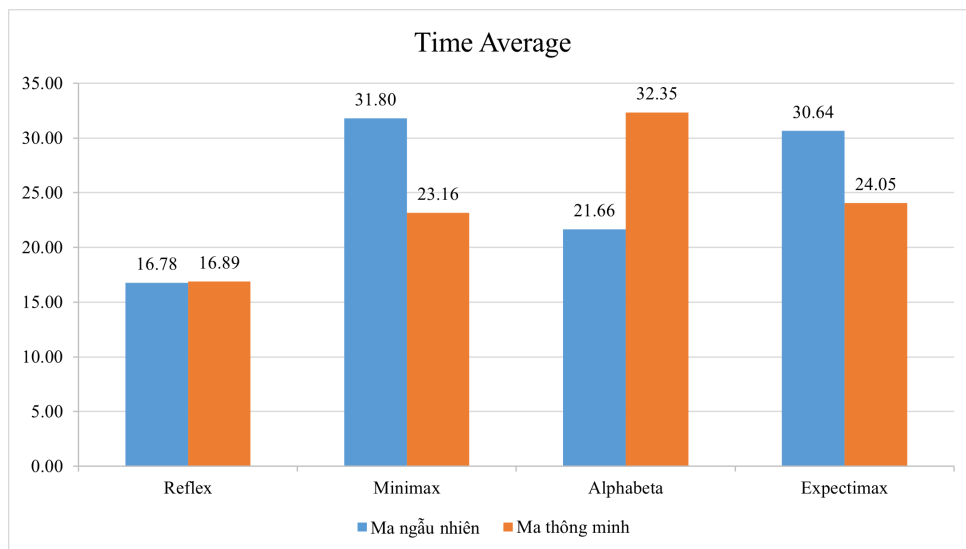
### 3.1 Maze: Contest Classic



Hình 3.1: Điểm trung bình trong map Contest Classic

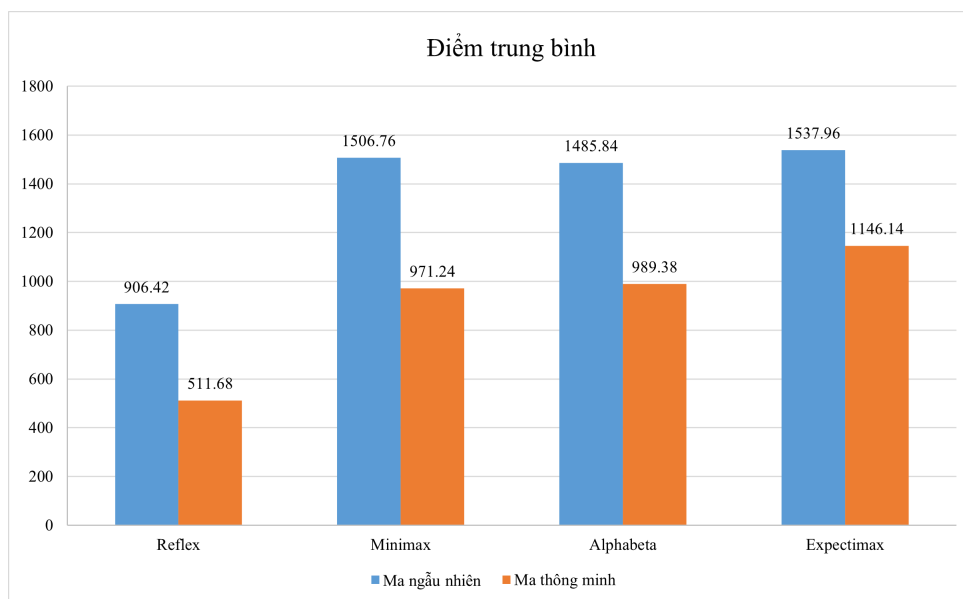


Hình 3.2: Tỷ lệ thắng trong map Contest Classic

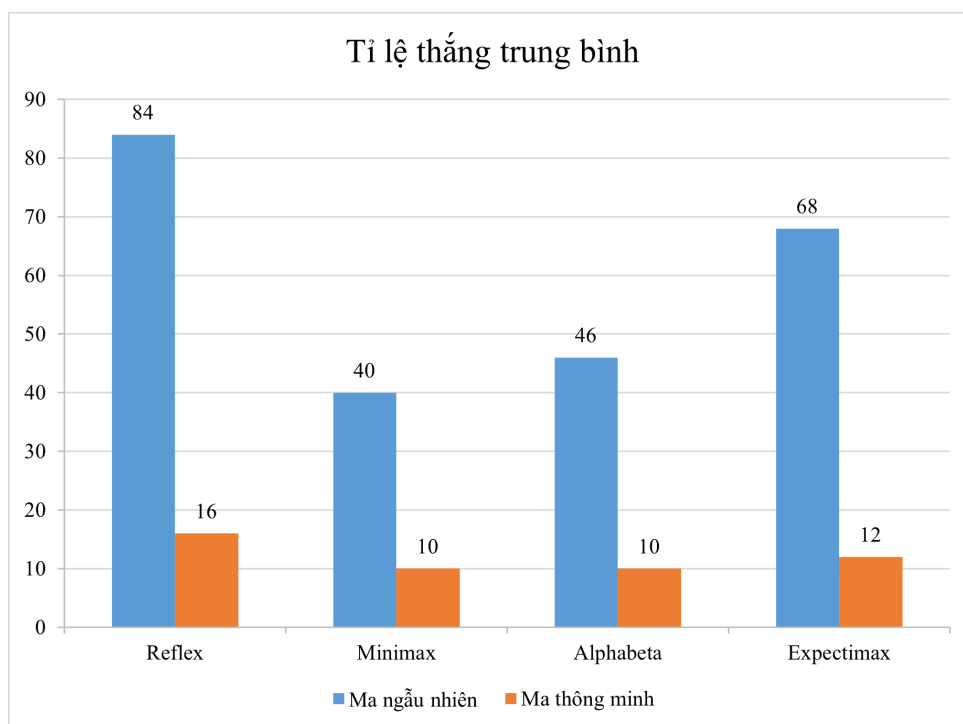


Hình 3.3: Thời gian trung bình trong map Contest Classic

### 3.2 Maze: Medium Classic

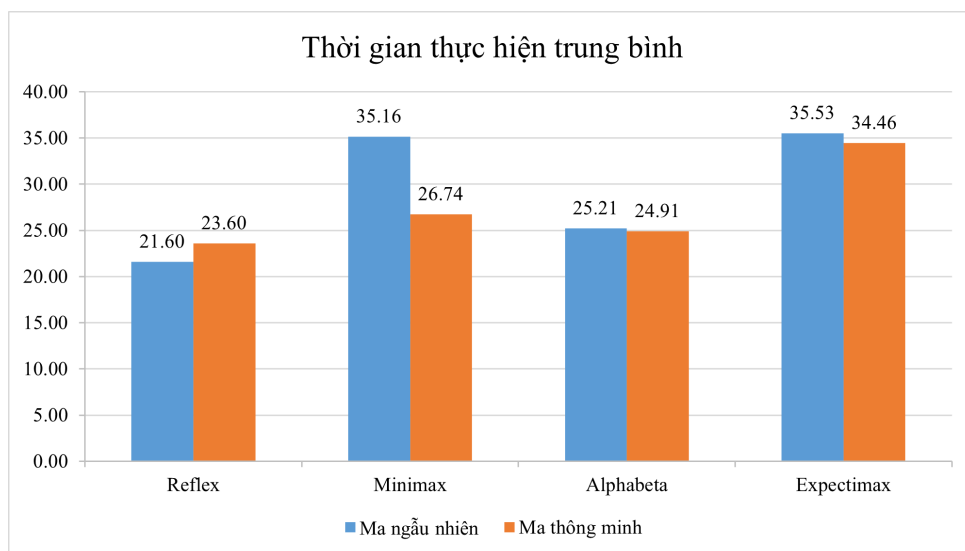


Hình 3.4: Điểm trung bình trong map Medium Classic



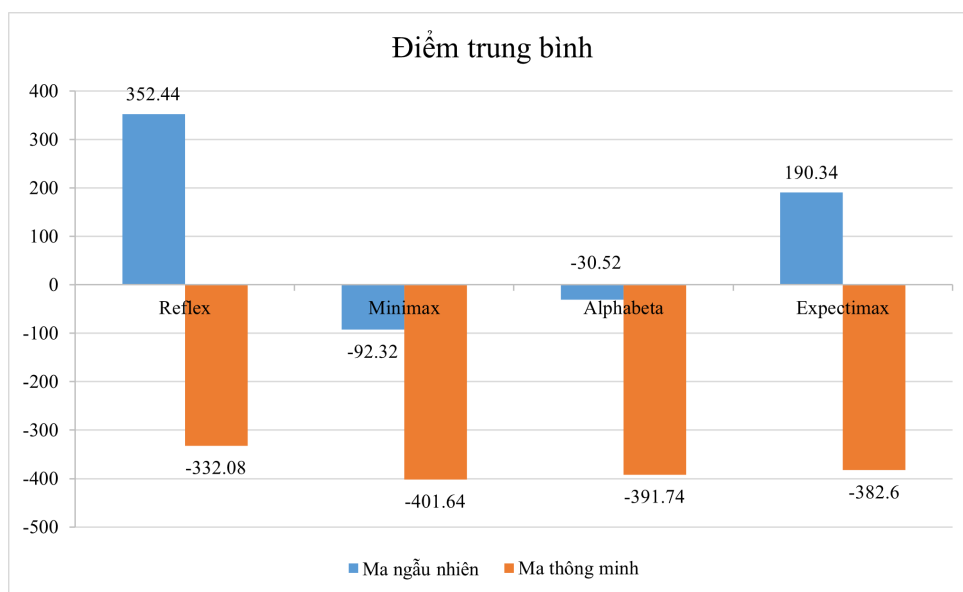
Hình 3.5: Tỉ lệ thắng trong map Medium Classic



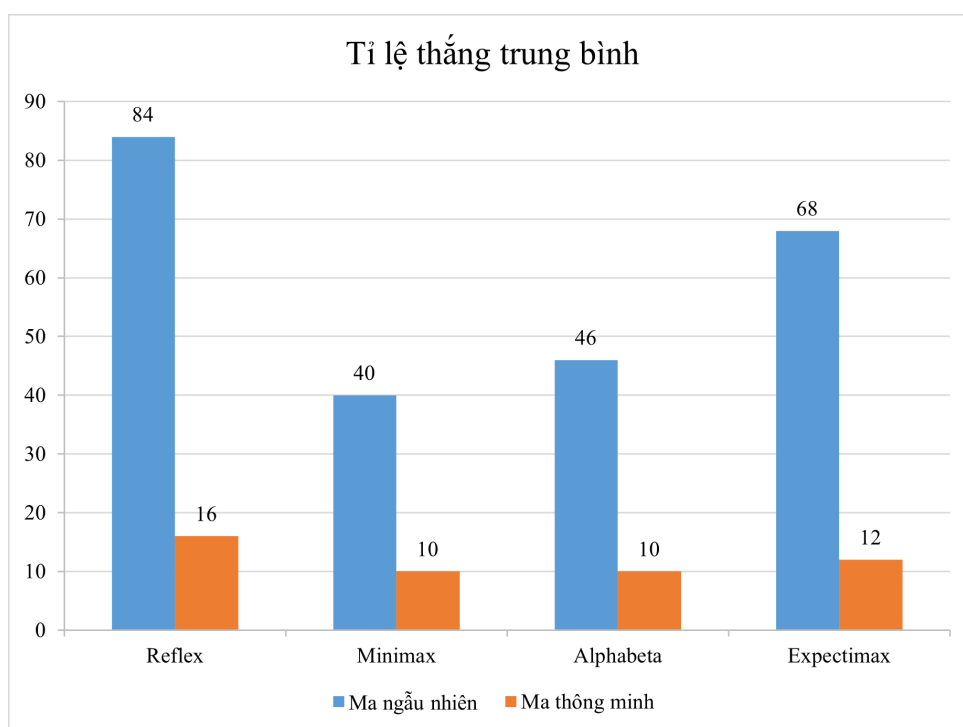


Hình 3.6: Thời gian trung bình trong map Medium Classic

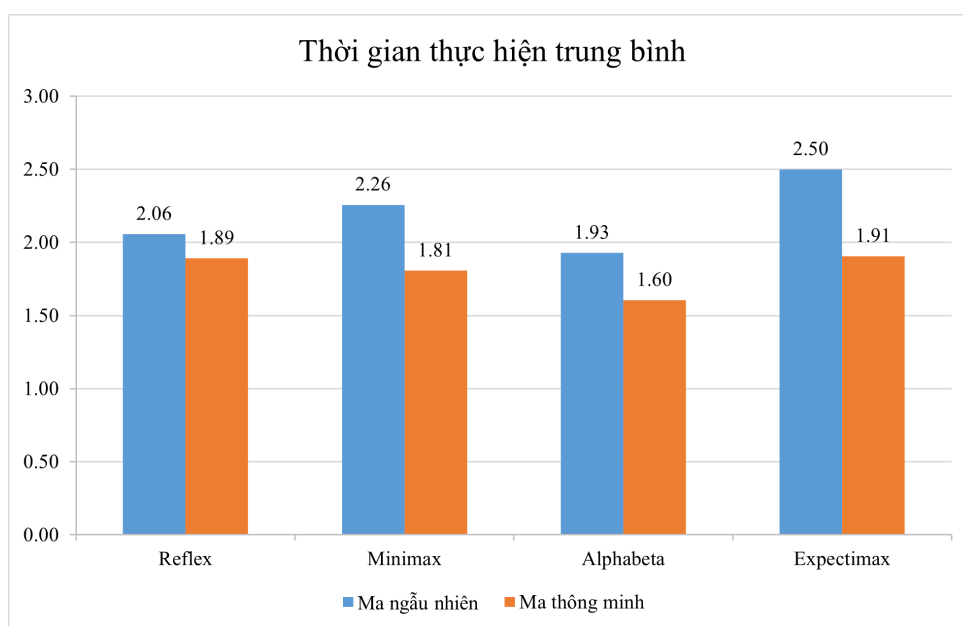
### 3.3 Maze: Minimax Classic



Hình 3.7: Điểm trung bình trong map Minimax Classic

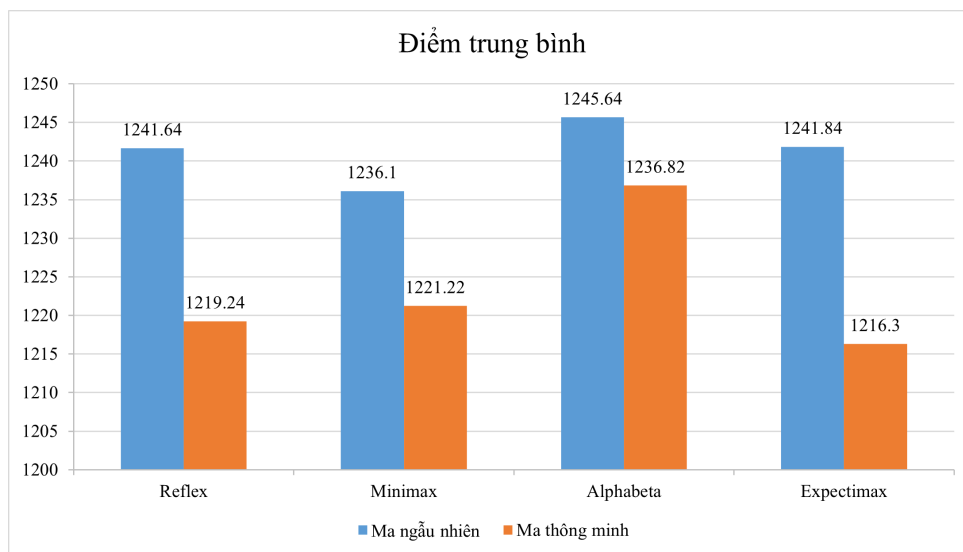


Hình 3.8: Tỉ lệ thắng trong map Minimax Classic

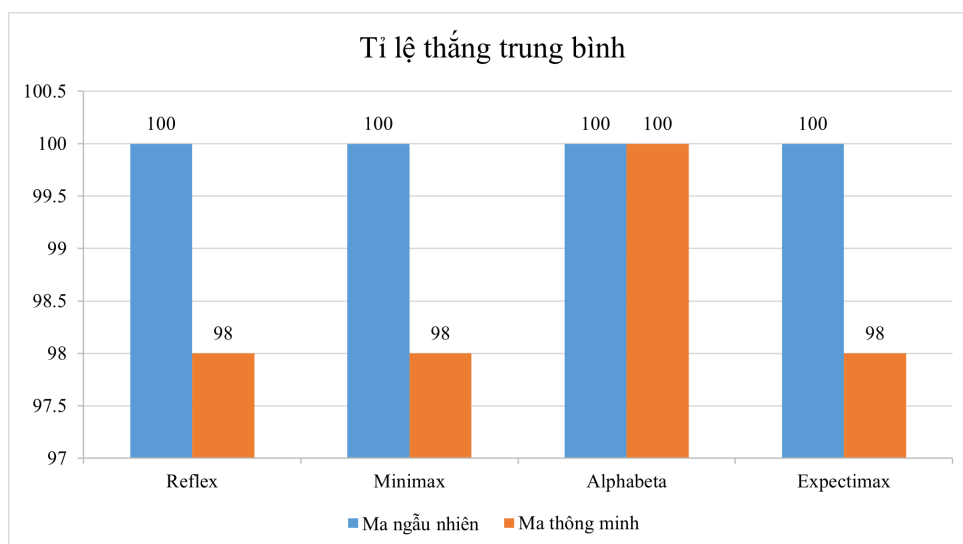


Hình 3.9: Thời gian trung bình trong map Minimax Classic

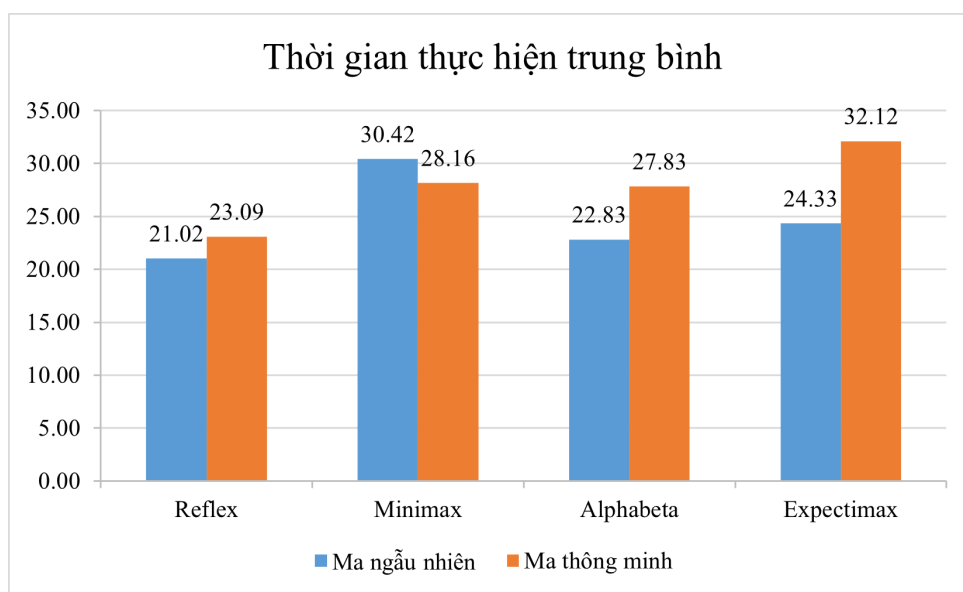
### 3.4 Maze: Open Classic



Hình 3.10: Điểm trung bình trong map Open Classic

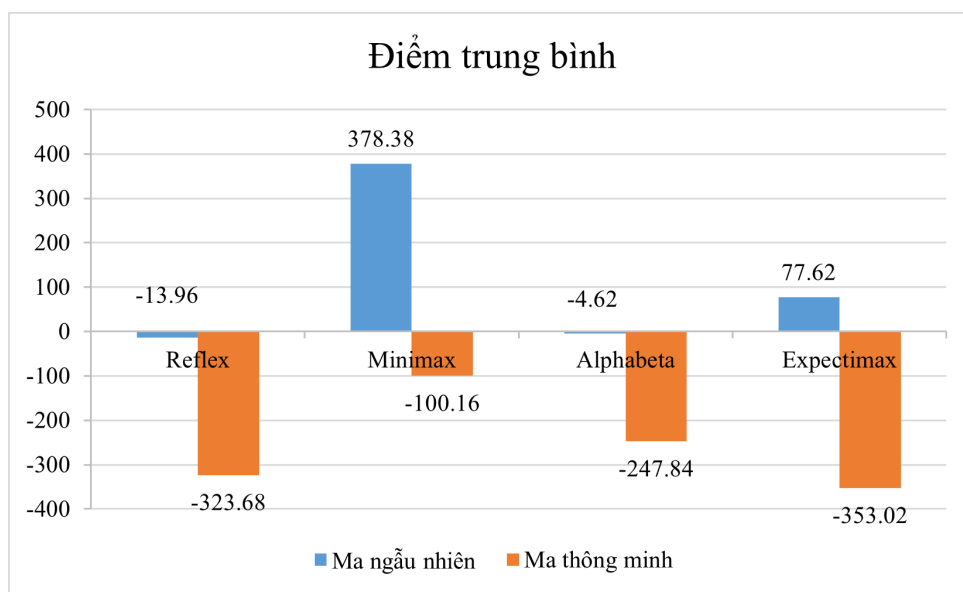


Hình 3.11: Tỉ lệ thắng trong map Open Classic

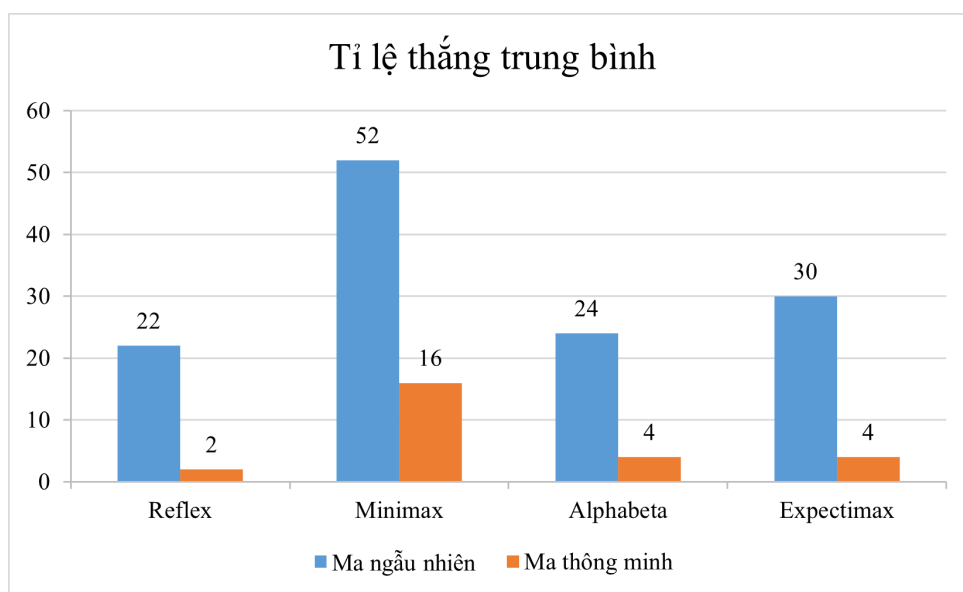


Hình 3.12: Thời gian trung bình trong map Open Classic

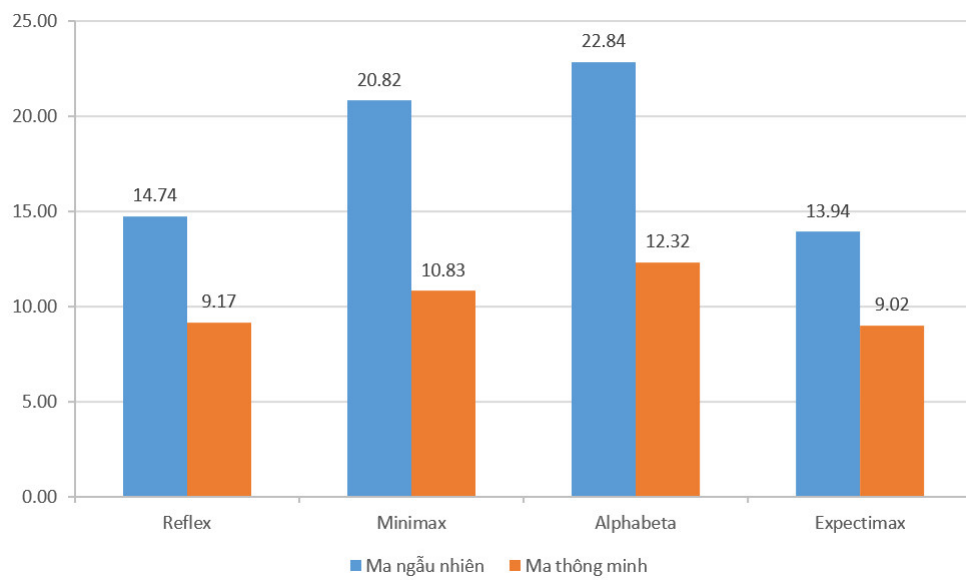
### 3.5 Maze: Capsule Classic



Hình 3.13: Điểm trung bình trong map Capsule Classic

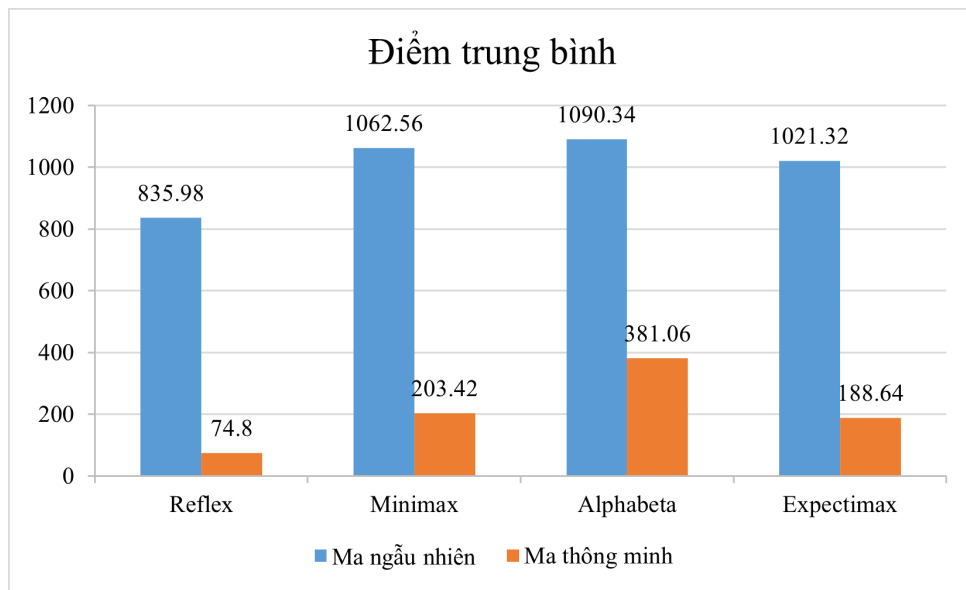


Hình 3.14: Tỉ lệ thắng trong map Capsule Classic

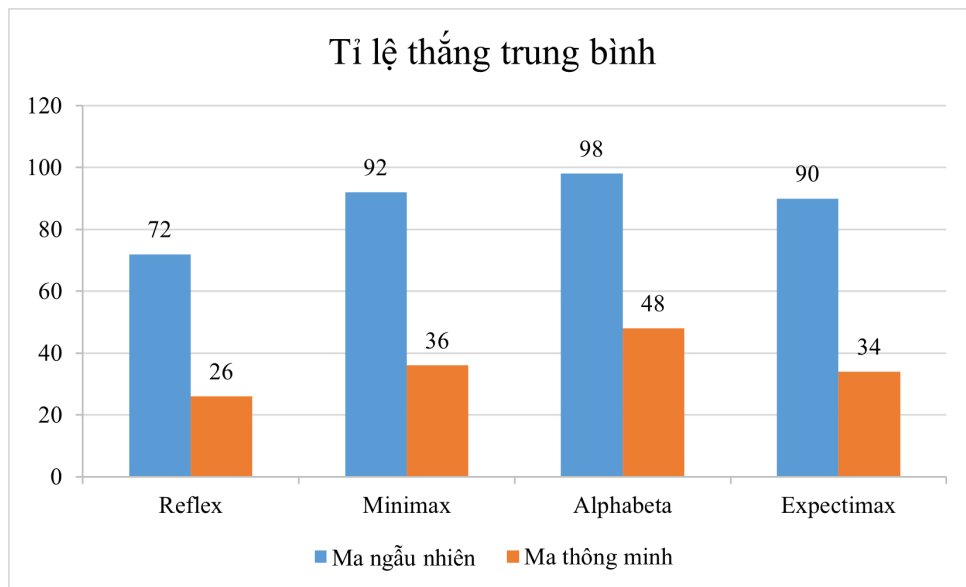


Hình 3.15: Thời gian trung bình trong map Capsule Classic

### 3.6 Maze: Small Classic

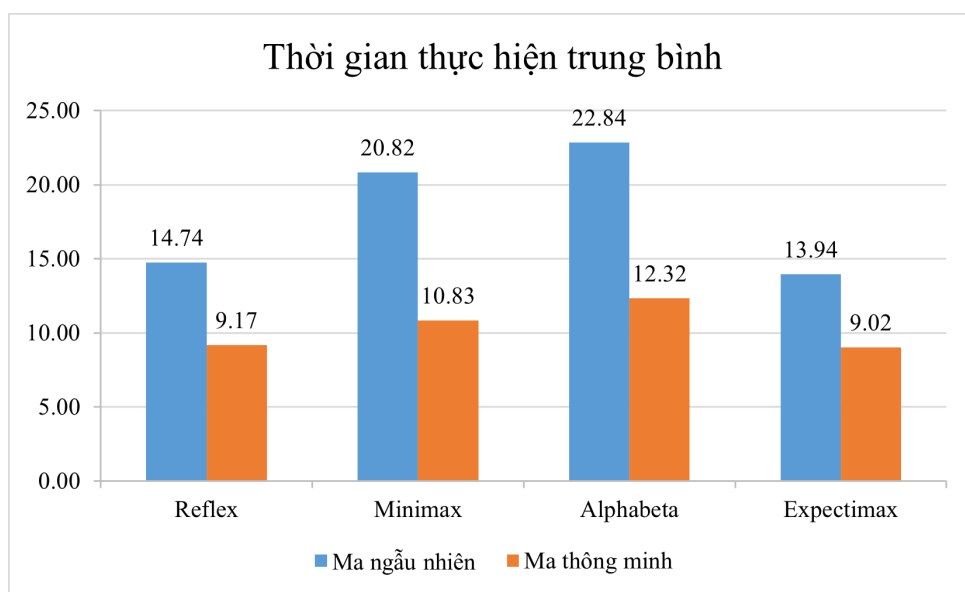


Hình 3.16: Điểm trung bình trong map Small Classic



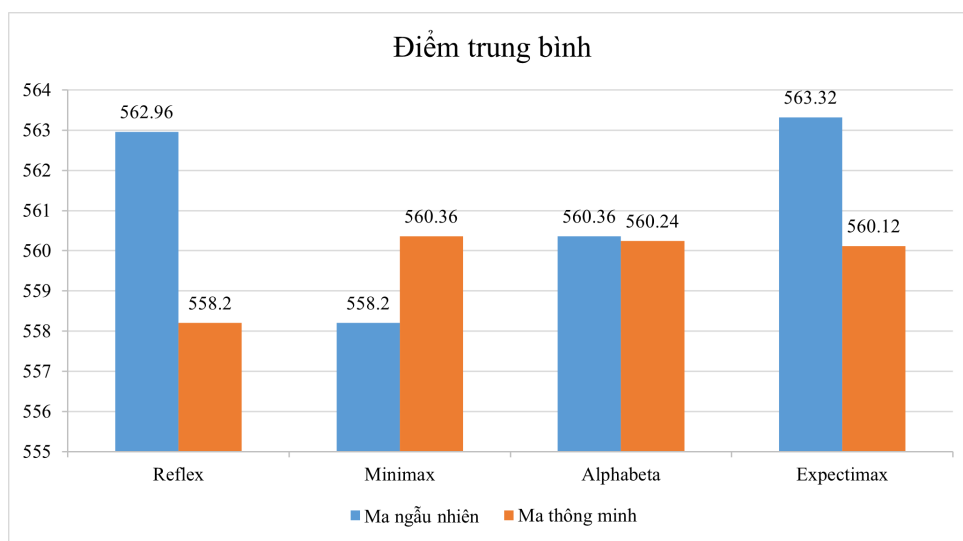
Hình 3.17: Tỉ lệ thắng trong map Small Classic



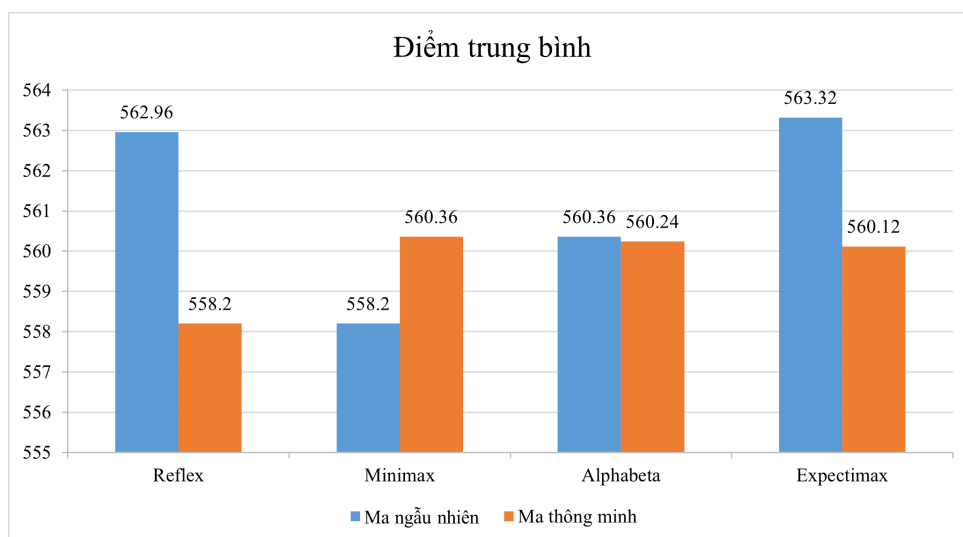


Hình 3.18: Thời gian trung bình trong map Small Classic

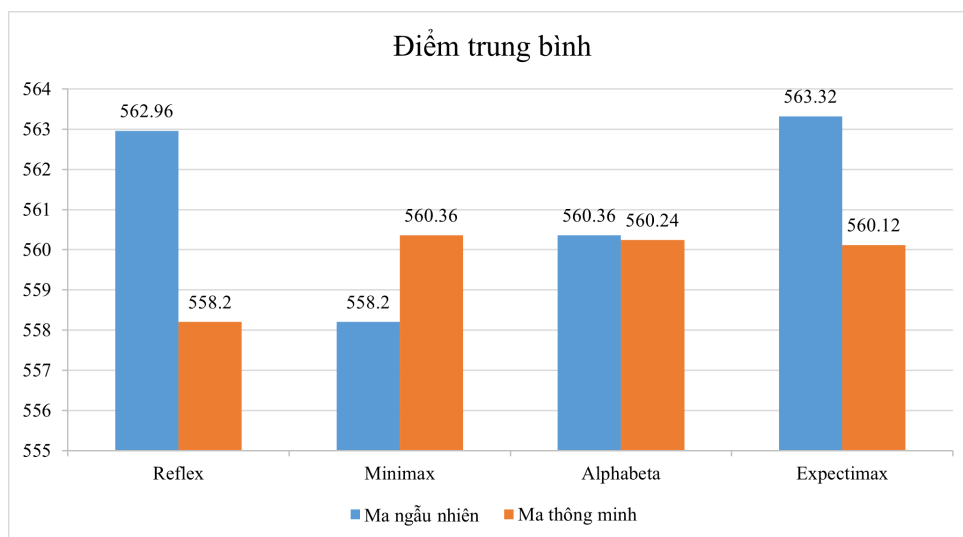
### 3.7 Maze: Test Classic



Hình 3.19: Điểm trung bình trong map Test Classic

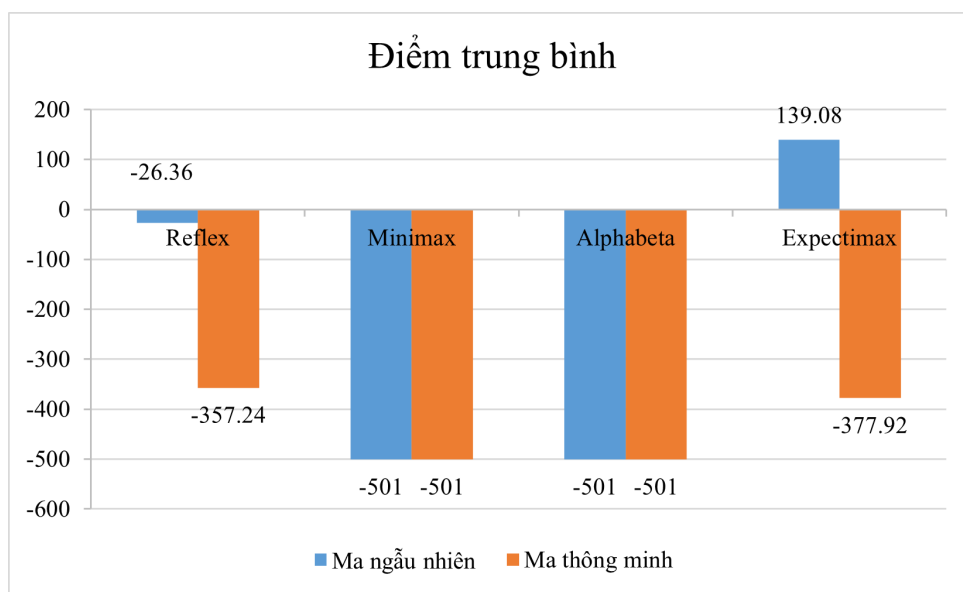


Hình 3.20: Tỷ lệ thắng trong map Test Classic

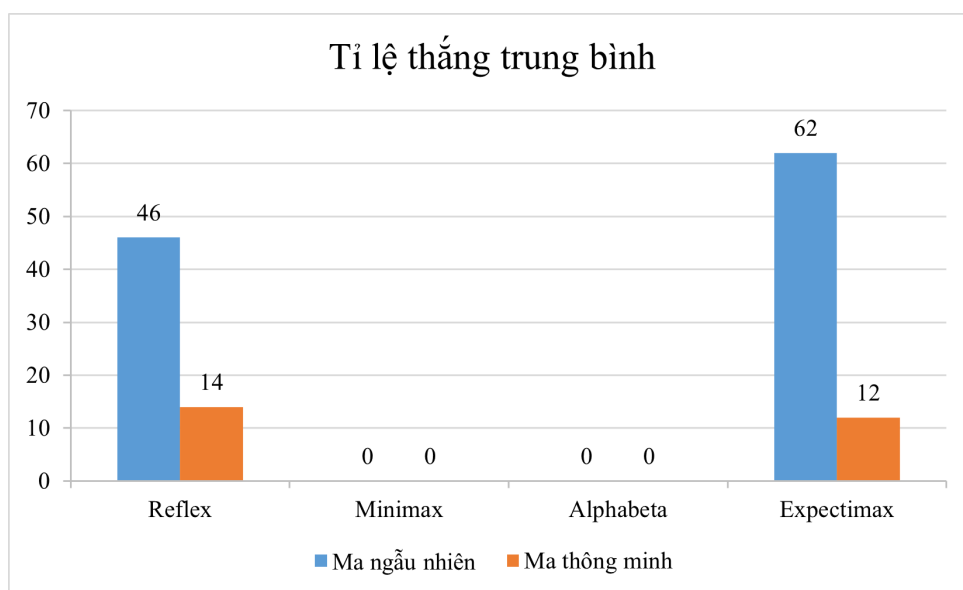


Hình 3.21: Thời gian trung bình trong map Test Classic

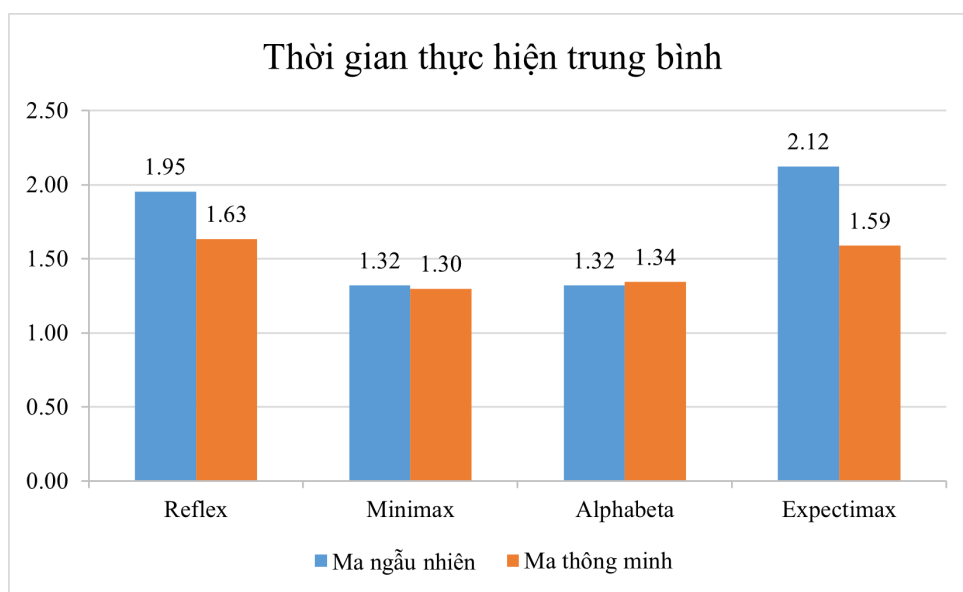
### 3.8 Maze: Trapped Classic



Hình 3.22: Điểm trung bình trong map Trapped Classic

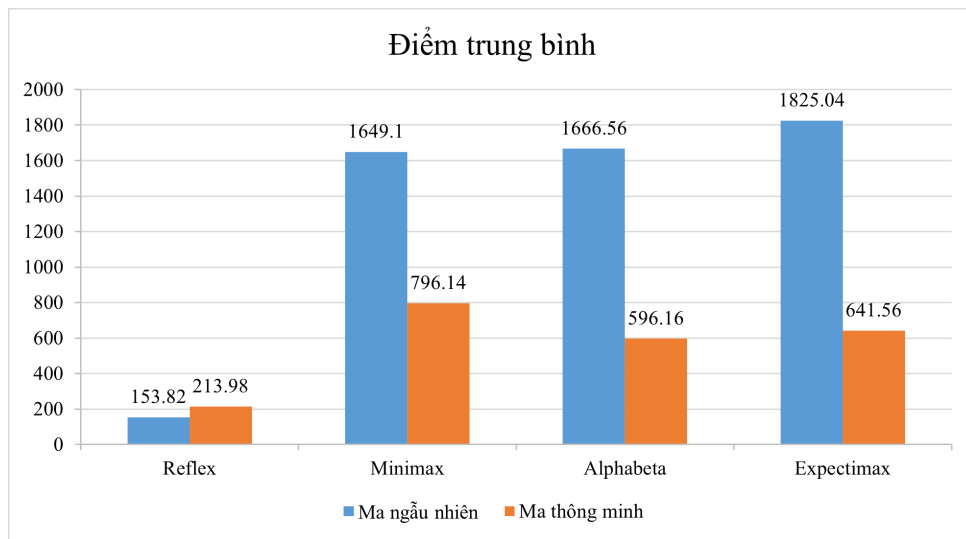


Hình 3.23: Tỉ lệ thắng trong map Trapped Classic

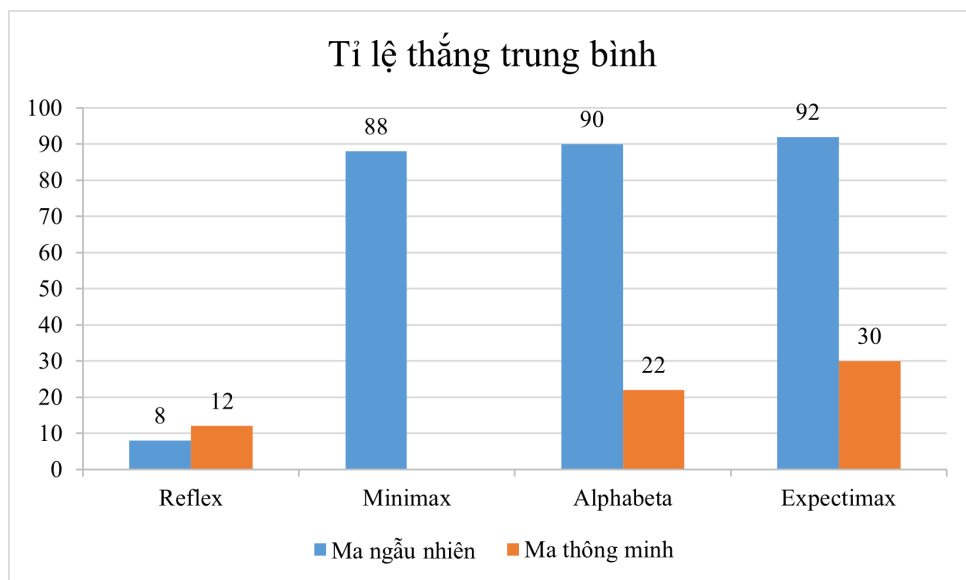


Hình 3.24: Thời gian trung bình trong map Trapped Classic

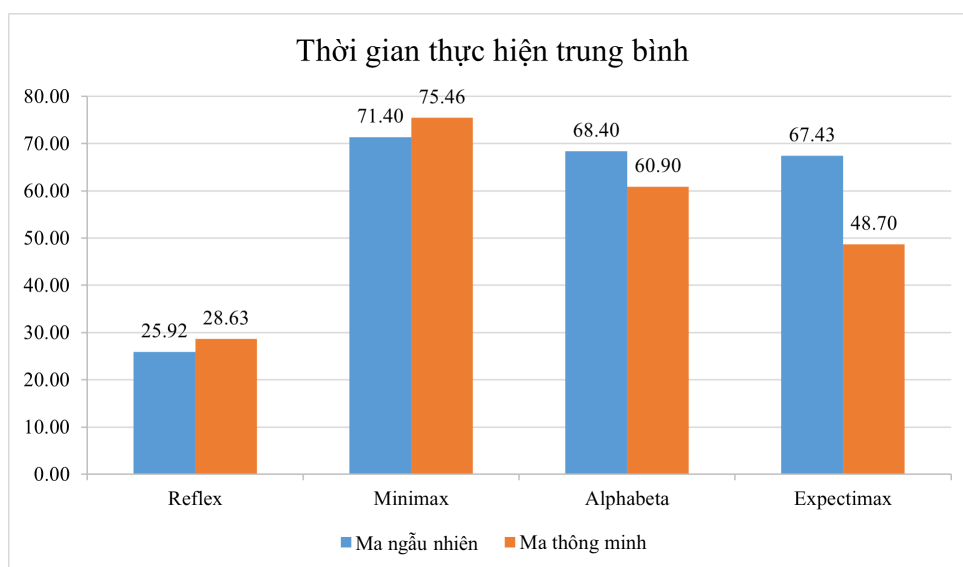
### 3.9 Maze: Tricky Classic



Hình 3.25: Điểm trung bình trong map Tricky Classic



Hình 3.26: Tỉ lệ thắng trong map Tricky Classic



Hình 3.27: Thời gian trung bình trong map Tricky Classic

## Chương 4 Kết luận

### 4.1 Đánh giá

Trong bài viết này, nhóm đã đề xuất một phương pháp chơi Pac-Man theo các thuật toán đã xác định bằng cách sử dụng trạng thái trò chơi và cây tìm kiếm để thực hiện các hành động bằng mô-đun cụ thể nhằm tối đa hóa điểm số của nó. Với bài toán Multi-agent, thuật toán được sử dụng có thể phát hiện ra sự kết hợp hành động để cho phép tác nhân theo đuổi nhiều mục tiêu. Hơn nữa, các quyết định phụ thuộc vào những quan sát hiện tại và trạng thái hành động trong tương lai. Chính sách của tác nhân được thể hiện dưới dạng danh sách các quy tắc nếu-thì có mức độ ưu tiên. Qua quan sát, nhóm kết luận rằng tác nhân tối tệ nhất là tác nhân phản xạ (Reflex Agent) vì nó chỉ sử dụng trạng thái trò chơi hiện tại. Chuyển sang thuật toán Minimax, thuật toán đánh giá các trạng thái thay vì hành động, như trường hợp của tác nhân phản xạ. Các tác nhân Minimax đánh giá các trạng thái trong tương lai trong khi các tác nhân phản xạ đánh giá các hành động từ trạng thái hiện tại, tuy nhiên, minimax là bộ nhớ và khả năng tính toán, do đó chỉ có thể di chuyển qua độ sâu 3 hoặc 4 trên máy tính hiện đại. Do đó, việc cắt tỉa Alpha beta có thể được sử dụng để loại bỏ trạng thái trò chơi không tối ưu nhằm giảm cây tìm kiếm nhằm cải thiện thời gian và độ sâu tìm kiếm. Minimax và alpha-beta hoạt động tốt, nhưng cả hai đều cho rằng đối thủ sẽ đưa ra quyết định tối ưu. Expectimax rất hữu ích cho việc mô hình hóa hành vi xác suất của các tác nhân có thể đưa ra những lựa chọn dưới mức tối ưu. Tất nhiên, các ma ngẫu nhiên không phải là tác nhân minimax tối ưu và do đó việc mô hình hóa chúng bằng tìm kiếm minimax có thể không phù hợp.

### 4.2 Khó khăn

Thiết kế Terminal hiển thị kết quả sau mỗi lượt chơi. Thiết kế một số map đặc biệt. Thuật toán với tác tử chưa tối ưu, mới chỉ dừng lại ở mức độ khảo sát.



### **4.3 Hướng phát triển của đề tài**

Dự kiến rằng công việc trong tương lai sẽ cần phải cải thiện hiệu suất (sự phù hợp) của tác tử. Tuy nhiên, cũng sẽ rất thú vị khi so sánh độ phức tạp (chiều, khả năng diễn giải, yêu cầu tính toán) của các phương pháp tiếp cận khác với cách tiếp cận dựa trên quy tắc được phát triển ở trên.

### **4.4 Về bản thân**

Trong quá trình tìm hiểu về đề tài báo cáo này, nhóm chúng tôi đã học được cách đặt vấn đề và giải quyết vấn đề, các phương pháp tìm đọc, chọn lọc tài liệu, nâng cao kỹ năng tự học và các kỹ năng cho bản thân. Cùng với đó là kỹ năng làm việc nhóm, các phương pháp trình bày một bài báo cáo đảm bảo tính thẩm mỹ, hợp quy chuẩn và khoa học, cùng nhau giải quyết vấn đề giúp nâng cao năng suất công việc.

## Tài liệu tham khảo

- [1] Moh Zikky. “Review of A\*(A star) navigation mesh pathfinding as the alternative of artificial intelligent for ghosts agent on the Pacman game”. In: *EMITTER International journal of engineering technology* 4.1 (2016), pp. 141–149.
- [2] Carl Felstiner. “Alpha-beta pruning”. In: *Whitman College* (2019).
- [3] Maharshi J Pathak, Ronit L Patel, and Sonal P Rami. “Comparative analysis of search algorithms”. In: *International Journal of Computer Applications* 179.50 (2018), pp. 40–43.
- [4] Pieter Abbeel John DeNero Dan Klein and many others. “The Pac-Man Projects”. In: (). URL: [http://ai.berkeley.edu/project\\_overview.html](http://ai.berkeley.edu/project_overview.html).
- [5] Pavlos Spanoudakis. “Berkeley-Pacman-Projects”. In: (). URL: <https://github.com/pspanoudakis/Berkeley-Pacman-Projects>.

Phụ lục A Bảng số liệu chi tiết

A.1 Tác tử Reflex

MA NGẪU NHIÊN									
Điểm trung bình									
	Contest Classic	Medium Classic	Minimax Classic	Open Classic	Capsule Classic	Small Classic	Test Classic	Trapped Classic	Tricky Classic
1	778.4	897.3	210.8	1242.7	125.2	948.7	563.4	15	264.3
2	669.8	709.1	311.1	1240	-77.5	1029.9	563.4	-88.4	190.5
3	843	1076.9	412.3	1242.9	-215	943.1	563	15	434.3
4	1395.5	747.2	313	1241.3	191.1	819.6	561.8	-88.4	-105.8
5	1342.7	1101.6	515	1241.3	-93.6	438.6	563.2	15	-14.2
Average	1005.88	906.42	352.44	1241.64	-13.96	835.98	562.96	-26.36	153.82
Tỉ lệ thắng trung bình									
1	60	60	70	100	30	80	100	50	20
2	50	40	80	100	20	80	100	40	0
3	60	60	90	100	10	80	100	50	20
4	90	50	80	100	40	70	100	40	0
5	90	70	100	100	10	50	100	50	0
Average	70	56	84	100	22	72	100	46	8
Thời gian thực hiện trung bình									
1	14.21	22.29	2.17	19.79	14.65	15.97	3.77	2.02	25.51
2	15.01	19.60	2.21	20.32	13.59	17.81	4.07	1.89	31.46
3	16.81	23.52	2.15	19.44	15.81	16.02	3.89	2.00	36.34
4	18.96	19.67	1.88	22.13	16.78	14.40	3.87	1.89	15.26
5	18.88	22.90	1.88	23.40	12.86	11.82	3.85	1.96	21.04
Average	16.78	21.60	2.06	21.02	14.74	15.20	3.89	1.95	25.92

Hình A.1: Bảng số liệu ma ngẫu nhiên Reflex

MA THÔNG MINH									
Điểm trung bình									
	Contest Classic	Medium Classic	Minimax Classic	Open Classic	Capsule Classic	Small Classic	Test Classic	Trapped Classic	Tricky Classic
1	419.4	1049.3	-291.6	1152.7	-364.9	-138.2	557.6	-502	96.5
2	800.6	288.3	-292.2	1239.9	-424.8	-161.8	558.8	-191.8	210.6
3	-2.2	65.7	-292	1235.8	-250.4	261.1	558.4	-295.2	370.1
4	1084.7	474.3	-291.6	1231.2	-330.1	241.1	558.4	-398.6	153
5	801.4	680.8	-493	1236.6	-248.2	171.8	557.8	-398.6	239.7
Average	620.78	511.68	-332.08	1219.24	-323.68	74.8	558.2	-357.24	213.98
Tỉ lệ thắng trung bình									
1	30	70	20	90	0	10	100	0	10
2	60	10	20	100	0	0	100	30	10
3	10	10	20	100	10	40	100	20	30
4	70	40	20	100	0	40	100	10	0
5	50	40	0	100	0	40	100	10	10
Average	44	34	16	98	2	26	100	14	12
Thời gian thực hiện trung bình									
1	14.84	28.29	1.86	23.25	8.73	7.78	4.83	1.49	20.80
2	20.76	23.49	2.02	22.23	8.69	9.95	4.89	1.79	27.88
3	9.00	17.72	1.94	23.26	9.06	11.92	4.80	1.70	30.29
4	21.45	22.18	1.82	24.04	9.50	11.62	4.86	1.62	36.26
5	18.38	26.30	1.80	22.67	9.88	10.96	4.92	1.57	27.94
Average	16.89	23.60	1.89	23.09	9.17	10.45	4.86	1.63	28.63

Hình A.2: Bảng số liệu ma thông minh Reflex

A.2 Tác tử Minimax

MA NGẪU NHIÊN									
Điểm trung bình									
	Contest Classic	Medium Classic	Minimax Classic	Open Classic	Capsule Classic	Small Classic	Test Classic	Trapped Classic	Tricky Classic
1	1364.8	1431.6	-91.7	1229.4	477.6	1187.1	562	-501	1721.7
2	1366	1456.7	-296.1	1237.6	459.3	1032.7	561.4	-501	1857
3	1407.6	1593.8	110.5	1229.5	308	1146.3	560.6	-501	1663.5
4	1589.3	1556.6	-292.6	1254.7	523	797.9	544.6	-501	1675.4
5	1482	1495.1	108.3	1229.3	124	1148.8	562.4	-501	1327.9
Average	1441.94	1506.76	-92.32	1236.1	378.38	1062.56	558.2	-501	1649.1
Tỉ lệ thắng trung bình									
1	100	90	40	100	60	100	100	0	90
2	100	100	20	100	60	90	100	0	100
3	90	100	60	100	40	100	100	0	90
4	100	100	20	100	60	70	100	0	90
5	100	100	60	100	40	100	100	0	70
Average	98	98	40	100	52	92	100	0	88
Thời gian thực hiện trung bình									
1	29.77	35.95	2.09	33.43	20.77	19.13	3.95	1.31	68.51
2	31.52	35.18	2.52	31.48	22.95	18.55	4.02	1.31	68.23
3	35.15	34.08	2.07	33.59	23.04	19.95	4.16	1.31	72.89
4	30.21	35.22	1.91	26.07	22.18	17.06	7.13	1.35	84.55
5	32.34	35.39	2.70	27.50	15.17	18.71	4.17	1.32	62.83
Average	31.80	35.16	2.26	30.42	20.82	18.68	4.69	1.32	71.40

Hình A.3: Bảng số liệu ma ngẫu nhiên Minimax

MA THÔNG MINH									
Điểm trung bình									
	Contest Classic	Medium Classic	Minimax Classic	Open Classic	Capsule Classic	Small Classic	Test Classic	Trapped Classic	Tricky Classic
1	1254.6	820.8	-492.4	1248.9	-340.4	236.6	559.8	-501	1173.2
2	1504.2	830.3	-391.6	1145.9	-81.3	268.6	560.8	-501	765.9
3	1543.5	852.1	-391.2	1248.8	0.9	76.8	560.8	-501	674.3
4	1327.2	1238.3	-190	1230.8	119.1	-3.2	560.8	-501	564.7
5	1411.3	1114.7	-543	1231.7	-199.1	438.3	559.6	-501	802.6
Average	1408.16	971.24	-401.64	1221.22	-100.16	203.42	560.36	-501	796.14
Tỉ lệ thắng trung bình									
1	80	40	0	100	0	40	100	0	50
2	100	40	10	90	20	40	100	0	40
3	100	50	10	100	20	30	100	0	40
4	90	80	30	100	30	20	100	0	10
5	90	70	0	100	10	50	100	0	30
Average	92	56	10	98	16	36	100	0	34
Thời gian thực hiện trung bình									
1	21.58	27.73	1.72	25.75	8.38	11.53	4.57	1.30	69.28
2	23.74	28.35	1.84	31.12	10.80	10.81	4.14	1.29	78.81
3	23.92	21.42	1.65	27.29	10.29	7.89	4.13	1.30	64.51
4	23.69	29.25	1.99	30.42	12.80	10.70	4.14	1.31	80.62
5	22.87	26.94	1.83	26.20	11.88	16.24	4.44	1.30	84.06
Average	23.16	26.74	1.81	28.16	10.83	11.43	4.28	1.30	75.46

Hình A.4: Bảng số liệu ma thông minh Minimax

### A.3 Tác Tử Alphabeta

	MA NGẪU NHIÊN									
	Điểm trung bình									
	Contest Classic	Medium Classic	Minimax Classic	Open Classic	Capsule Classic	Small Classic	Test Classic	Trapped Classic	Tricky Classic	
1	1443.2	1302.8	10.3	1224.7	164.1	1031.2	559.8	-501	1612.1	
2	1399.9	1478.3	-292.5	1270.9	162.8	1025.1	558.8	-501	1690	
3	1460.8	1654.8	-90.1	1226.6	-195.1	1085.2	561.8	-501	1586	
4	1505.6	1497	-92.1	1253.4	-83.9	1187.7	560.8	-501	1822.6	
5	1446.6	1496.3	311.8	1252.6	-71	1122.5	560.6	-501	1622.1	
Average	1451.22	1485.84	-30.52	1245.64	-4.62	1090.34	560.36	-501	1666.56	
	Tỉ lệ thắng trung bình									
1	100	80	50	100	40	90	100	0	90	
2	100	100	20	100	40	100	100	0	90	
3	90	100	40	100	10	100	100	0	80	
4	100	100	40	100	20	100	100	0	100	
5	100	100	80	100	10	100	100	0	90	
Average	98	96	46	100	24	98	100	0	90	
	Thời gian thực hiện trung bình									
1	25.97	24.30	1.86	23.91	18.78	24.42	4.33	1.33	84.81	
2	21.38	25.31	1.83	22.80	27.63	27.72	5.00	1.31	67.71	
3	20.41	25.66	1.74	23.36	22.14	26.14	4.75	1.32	78.53	
4	20.36	25.38	2.10	22.24	21.22	31.39	4.31	1.35	56.95	
5	20.19	25.40	2.11	21.83	24.42	24.83	4.45	1.30	53.98	
Average	21.66	25.21	1.93	22.83	22.84	26.90	4.57	1.32	68.40	

Hình A.5: Bảng số liệu ma ngẫu nhiên Alphabeta

	MA THÔNG MINH								
Điểm trung bình									
	Contest Classic	Medium Classic	Minimax Classic	Open Classic	Capsule Classic	Small Classic	Test Classic	Trapped Classic	Tricky Classic
1	1414.7	981	-391.6	1228.2	-200.3	253.9	561.8	-501	422.9
2	1519	917.1	-290.4	1231.7	-224.5	402.9	560.8	-501	1094
3	1448.1	627.3	-492.4	1249.4	-347.2	228.7	558.2	-501	360.6
4	1600.6	1224.6	-492.5	1228.7	-240.2	428.9	560	-501	754.6
5	1346.1	1196.9	-291.8	1246.1	-227	590.9	560.4	-501	348.7
Average	1465.7	989.38	-391.74	1236.82	-247.84	381.06	560.24	-501	596.16
Tỉ lệ thắng trung bình									
1	90	60	10	100	10	40	100	0	10
2	90	50	20	100	0	50	100	0	50
3	100	30	0	100	0	40	100	0	0
4	100	80	0	100	10	50	100	0	40
5	90	70	20	100	0	60	100	0	10
Average	94	58	10	100	4	48	100	0	22
Thời gian thực hiện trung bình									
1	30.53	25.32	1.55	27.74	12.80	12.03	4.16	1.35	61.75
2	34.61	22.59	1.55	27.63	13.22	13.54	4.55	1.34	81.78
3	32.00	21.52	1.56	27.26	8.93	12.06	4.84	1.34	53.31
4	33.18	26.54	1.55	27.90	12.02	14.49	4.54	1.34	60.22
5	31.42	28.57	1.81	28.63	14.62	16.58	4.76	1.34	47.47
Average	32.35	24.91	1.60	27.83	12.32	13.74	4.57	1.34	60.90

Hình A.6: Bảng số liệu ma thông minh Alphabeta

## A.4 Tác tử Expectimax

MA NGẪU NHIÊN									
Điểm trung bình									
	Contest Classic	Medium Classic	Minimax Classic	Open Classic	Capsule Classic	Small Classic	Test Classic	Trapped Classic	Tricky Classic
1	1430.7	1454.6	109.5	1234.2	397.3	850.6	563.6	15	1865.4
2	1428.3	1532.5	211	1254	211.5	1187.8	563.2	15	1796.4
3	1448.5	1572.6	209.9	1252.2	-147	1169.4	563.4	428.6	1826.2
4	1408.1	1613.6	109.5	1233.8	42.7	752.7	563	118.4	1789.5
5	1555.9	1516.5	311.8	1235	-116.4	1146.1	563.4	118.4	1847.7
Average	1454.3	1537.96	190.34	1241.84	77.62	1021.32	563.32	139.08	1825.04
Tỉ lệ thắng trung bình									
1	100	100	60	100	50	80	100	50	100
2	100	100	70	100	40	100	100	50	90
3	100	100	70	100	10	100	100	90	90
4	100	100	60	100	30	70	100	60	90
5	90	100	80	100	20	100	100	60	90
Average	98	100	68	100	30	90	100	62	92
Thời gian thực hiện trung bình									
1	29.44	35.79	2.21	25.10	20.71	16.83	3.76	1.94	97.29
2	30.70	35.17	2.23	24.16	13.05	19.04	3.75	1.93	70.89
3	30.25	35.37	2.76	24.26	12.43	20.20	4.07	2.41	59.73
4	31.88	35.59	2.66	24.94	10.90	14.93	3.95	2.17	52.16
5	30.94	35.74	2.63	23.19	12.61	19.37	3.86	2.16	57.06
Average	30.64	35.53	2.50	24.33	13.94	18.07	3.88	2.12	67.43

Hình A.7: Bảng số liệu ma ngẫu nhiên Expectimaxflex

MA THÔNG MINH									
Điểm trung bình									
	Contest Classic	Medium Classic	Minimax Classic	Open Classic	Capsule Classic	Small Classic	Test Classic	Trapped Classic	Tricky Classic
1	1686	1096.8	-494.5	1223.5	-160.7	328.9	560.8	-295.2	592.2
2	1336.4	993.2	-342.1	1225.1	-375.2	131.1	559	-398.6	638.6
3	1645.5	1214.8	-493	1242	-445.5	295.5	561.2	-398.6	784.8
4	1454.9	1132.4	-190.6	1249.5	-373.1	-63.5	559.4	-398.6	740.8
5	1608.8	1293.5	-392.8	1141.4	-410.6	251.2	560.2	-398.6	451.4
Average	1546.32	1146.14	-382.6	1216.3	-353.02	188.64	560.12	-377.92	641.56
Tỉ lệ thắng trung bình									
1	100	60	0	100	20	50	100	20	30
2	90	60	20	100	0	30	100	10	40
3	100	80	0	100	0	30	100	10	40
4	90	70	30	100	0	20	100	10	20
5	100	80	10	90	0	40	100	10	20
Average	96	70	12	98	4	34	100	12	30
Thời gian thực hiện trung bình									
1	21.83	35.00	2.03	32.67	10.65	12.23	4.14	1.65	38.41
2	23.06	32.07	2.00	32.09	10.31	11.52	4.43	1.55	38.47
3	21.54	36.30	1.78	33.19	5.50	12.96	4.05	1.54	41.19
4	22.78	34.91	1.79	30.79	9.31	8.56	4.50	1.56	50.22
5	31.06	34.00	1.93	31.85	9.30	12.11	4.68	1.64	75.21
Average	24.05	34.46	1.91	32.12	9.02	11.48	4.36	1.59	48.70

Hình A.8: Bảng số liệu ma thông minh Expectimax

A.5 Bảng tổng hợp

	agent	Điểm trung bình				Tỉ lệ thắng trung bình (%)				Thời gian thực hiện trung bình (s)			
		Reflex	Minimax	Alphabeta	Expectimax	Reflex	minimax	alphabeta	expectimax	Reflex	minimax	alphabeta	expectimax
Mã ngẫu nhiên	Contest Classic	1005.88	1441.94	1451.22	1454.3	70	98	98	98	16.78	31.80	21.66	30.64
	Medium Classic	906.42	1506.76	1485.84	1537.96	56	98	96	100	21.60	35.16	25.21	35.53
	Minimax Classic	352.44	-92.32	-30.52	190.34	84	40	46	68	2.06	2.26	1.93	2.50
	Open Classic	1241.64	1236.1	1245.64	1241.84	100	100	100	100	21.02	30.42	22.83	24.33
	Capsule Classic	-13.96	378.38	-4.62	77.62	22	52	24	30	14.74	20.82	22.84	13.94
	Small Classic	835.98	1062.56	1090.34	1021.32	72	92	98	90	15.20	18.68	26.90	18.07
	Test Classic	562.96	558.2	560.36	563.32	100	100	100	100	3.89	4.69	4.57	3.88
	Trapped Classic	-26.36	-501	-501	139.08	46	0	0	62	1.95	1.32	1.32	2.12
	Tricky Classic	153.82	1649.1	1666.56	1825.04	8	88	90	92	25.92	71.40	68.40	67.43
	Contest Classic	620.78	1408.16	1465.7	1546.32	44	92	94	96	16.89	23.16	32.35	24.05
Mã thông minh	Medium Classic	511.68	971.24	989.38	1146.14	34	56	58	70	23.60	26.74	24.91	34.46
	Minimax Classic	-332.08	-401.64	-391.74	-382.6	16	10	10	12	1.89	1.81	1.60	1.91
	Open Classic	1219.24	1221.22	1236.82	1216.3	98	98	100	98	23.09	28.16	27.83	32.12
	Capsule Classic	-323.68	-100.16	-247.84	-353.02	2	16	4	4	9.17	10.83	12.32	9.02
	Small Classic	74.8	203.42	381.06	188.64	26	36	48	34	10.45	11.43	13.74	11.48
	Test Classic	558.2	560.36	560.24	560.12	100	100	100	100	4.86	4.28	4.57	4.36
	Trapped Classic	-357.24	-501	-501	-377.92	14	0	0	12	1.63	1.30	1.34	1.59
	Tricky Classic	213.98	796.14	596.16	641.56	12		22	30	28.63	75.46	60.90	48.70
	Average	400.25	633.192222	614.033333	679.797778	50.2222222	63.2941176	60.4444444	66.4444444	13.52	22.21	20.85	20.34

Hình A.9: Bảng tổng hợp