

Assignment 2 - Evaluation functions for Minimax/AlphaBeta/Expectimax

CS106.N21.KHCL

Lê Gia Khang – 21522189

I. Giới thiệu

- Trò chơi Pacman: Người chơi điều khiển Pacman trong một mê cung và ăn các chấm thức ăn. Nếu người chơi ăn hết các chấm thì Pacman được đưa qua màn chơi mới. Có tối đa 4 con ma đi tự do trong mê cung và cố gắng bắt Pacman. Nếu để bị bắt, Pacman sẽ bị mất mạng. Trong mê cung sẽ có những chấm tròn to và phát sáng có tên là "capsule", nếu Pacman ăn được các chấm này sẽ giúp Pacman có khả năng ăn các kẻ địch trong một thời gian ngắn. Đây là một dạng bài toán Adversarial.
- Trong bài tập này, chúng ta sẽ thiết kế các agent cho Pacman. Các thuật toán được sử dụng là Minimax, AlphaBeta và Expectimax. Ngoài ra, chúng ta còn phải tự thiết kế một Evaluation Function để ước lượng giá trị của các trạng thái của Pacman.
- Mục tiêu của chúng ta là thiết kế sao cho Pacman ăn hết toàn bộ chấm thức ăn, đồng thời né tránh hoặc ăn capsule để ăn lại con ma và đạt số điểm cao nhất khi kết thúc trò chơi.

II. Các đặc trưng và Evaluation Function

- Mỗi hành động mà Pacman thực hiện sẽ ảnh hưởng đến điểm số và điểm kết thúc của trò chơi, do đó ta cần thiết kế hàm lượng giá dựa trên các kết quả mà hành động này mang lại.
- Điểm số tương ứng với hành động trong trò chơi:
 - **Thời gian theo giây:** -1 điểm.
 - **Chấm thức ăn:** 10 điểm.
 - **Chiến thắng:** 500 điểm.
 - **Ăn ma:** 200 điểm.
 - **Thua cuộc:** -500 điểm.
 - **Capsule:** 0 điểm.

- Hàm lượng giá cũ (scoreEvaluationFunction) chỉ sử dụng điểm số của trạng thái để ước lượng giá trị trạng thái.
- Các đặc trưng mà em sử dụng để thiết kế hàm lượng giá mới (betterEvaluationFunction) là:
 - **ghost_dir**: khoảng cách đến con ma gần nhất.
 - **closest_food**: khoảng cách đến chấm đồ ăn gần nhất.
 - **closest_capsule**: Khoảng cách đến capsule gần nhất.
 - **currentGameState.getScore()**: điểm số của trò chơi (trạng thái).
 - **no_food & no_capsule**: số thức ăn và capsule còn lại.
- Qua tìm hiểu trò chơi và chạy thử vô số lần, ta rút ra được một số kinh nghiệm:
 - **currentGameState.getScore()** là kết quả gần nhất với kết quả cuối cùng của trò chơi ta cần tối ưu, tức điểm hiện tại càng cao, điểm kết quả có khả năng sẽ cao. Thế nên ta cần tìm cách tăng điểm hiện tại.
 - **no_food** càng nhỏ thì ta gần thắng trò chơi. Ta cần khuyến khích giảm đặc trưng này.
 - **no_capsule** cũng tương tự **no_food**.
 - **closest_food** sẽ thay đổi dựa vào vị trí của Pacman và vị trí các chấm thức ăn hiện tại. Đặc trưng này càng nhỏ thì cho biết Pacman có thức ăn xung quanh ở rất gần.
 - **closest_capsule** cũng tương tự **closest_food**.

III. Thực nghiệm

- Kết quả được thử nghiệm trên 5 layout (capsuleClassic, contestClassic, mediumClassic, minimaxClassic và trappedClassic) với số depth là 2, được chạy với cả 3 thuật toán với lần lượt hàm lượng giá cũ và mới.
- Được thử nghiệm trên 2 loại ma là Random Ghost và Directional Ghost (thông minh hơn).
- Mỗi layout với hàm Eval đều được chạy 5 lần. Qua đó trích xuất win rate, thời gian chạy và điểm số trung bình để so sánh hiệu suất của các thuật toán và hàm Eval.
- Dưới đây là bảng tổng hợp kết quả chạy so sánh với 2 hàm Eval và 2 con ma khác nhau (Random và Directional).

Random Ghost									
betterEvaluationFunction	Minimax			AlphaBeta			Expectimax		
Layout	Average Score	Win count	Time	Average Score	Win count	Time	Average Score	Win count	Time
capsuleClassic	-472.6	0/5 (0.00)	3.20801973	-472.6	0/5 (0.00)	3.078659	26.8	1/5 (0.20)	6.302074
contestClassic	1445	3/5 (0.60)	21.0128562	1445	3/5 (0.60)	18.78514	1402.8	4/5 (0.80)	20.53814
mediumClassic	1196.8	4/5 (0.80)	26.029846	1196.8	4/5 (0.80)	23.57106	1553.8	5/5 (1.00)	22.53296
minimaxClassic	112	3/5 (0.60)	0.143581152	112	3/5 (0.60)	0.137425	313.4	4/5 (0.80)	0.156194
trappedClassic	118.4	3/5 (0.60)	0.05367804	118.4	3/5 (0.60)	0.054445	118.4	3/5 (0.60)	0.055846

scoreEvaluationFunction	Minimax			AlphaBeta			Expectimax		
Layout	Average Score	Win count	Time	Average Score	Win count	Time	Average Score	Win count	Time
capsuleClassic	-348.2	0/5 (0.00)	4.11093307	-348.2	0/5 (0.00)	3.944522	-354.2	0/5 (0.00)	4.488085
contestClassic	8.8	0/5 (0.00)	9.82456374	8.8	0/5 (0.00)	9.13435	397.2	1/5 (0.20)	18.23341
mediumClassic	-66.6	1/5 (0.20)	31.8358696	-66.6	1/5 (0.20)	30.39058	-388.2	2/5 (0.40)	58.60023
minimaxClassic	-92.8	2/5 (0.40)	0.20980978	-92.8	2/5 (0.40)	0.189866	108.6	3/5 (0.60)	0.224686
trappedClassic	118.4	3/5 (0.60)	0.05399561	118.4	3/5 (0.60)	0.053777	118.4	3/5 (0.60)	0.054998

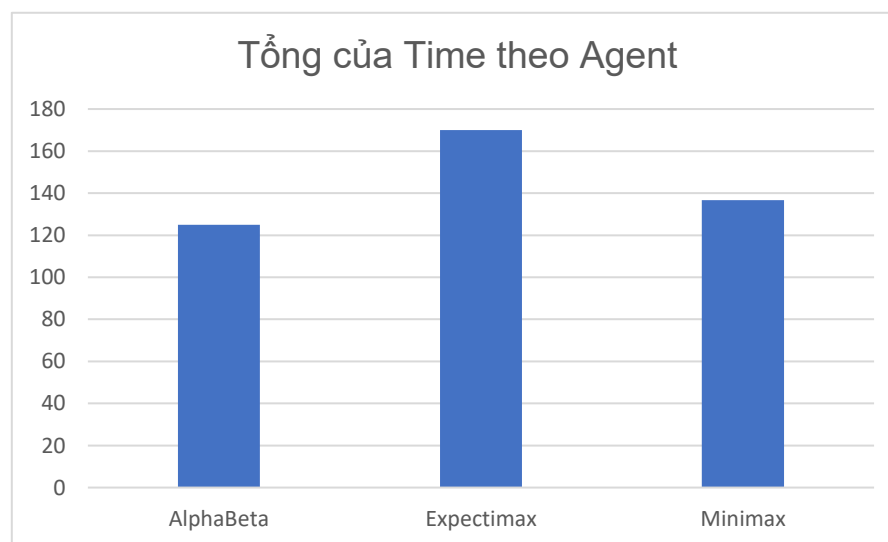
- Khi so 2 hàm lượng giá mới và cũ ở Random Ghost, ta có thể thấy sự lệch hẳn về phía hàm Eval mới. Tổng số trận thắng ở hàm eval mới là 44/75 ván trong khi đó ở hàm Eval cũ là 21/75, hơn hẳn gấp đôi.

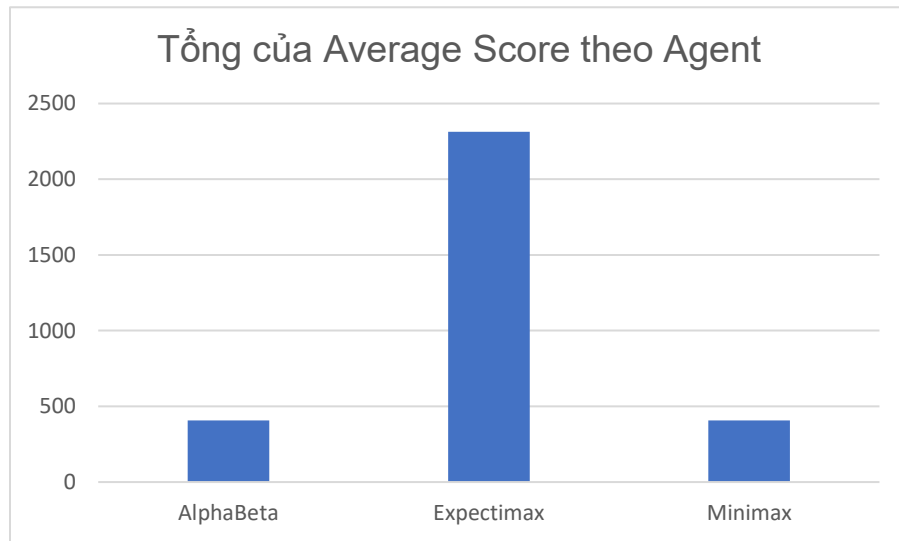
Directional Ghost									
betterEvaluationFunction	Minimax			AlphaBeta			Expectimax		
Layout	Average Score	Win count	Time	Average Score	Win count	Time	Average Score	Win count	Time
capsuleClassic	-307.8	0/5 (0.00)	2.54710484	-307.8	0/5 (0.00)	2.406948	-286.2	0/5 (0.00)	2.395412
contestClassic	42.4	0/5 (0.00)	13.7640052	42.4	0/5 (0.00)	11.84797	1.2	0/5 (0.00)	13.06993
mediumClassic	441.6	1/5 (0.20)	14.8010621	441.6	1/5 (0.20)	13.03539	689.8	2/5 (0.40)	14.16438
minimaxClassic	-492	0/5 (0.00)	0.11357427	-492	0/5 (0.00)	0.07795	-291.2	1/5 (0.20)	0.093256
trappedClassic	-295.2	1/5 (0.20)	0.04653263	-295.2	1/5 (0.20)	0.039289	-295.2	1/5 (0.20)	0.036641
scoreEvaluationFunction	Minimax			AlphaBeta			Expectimax		
Layout	Average Score	Win count	Time	Average Score	Win count	Time	Average Score	Win count	Time
capsuleClassic	-420.4	0/5 (0.00)	0.88449144	-420.4	0/5 (0.00)	0.821069	-420.4	0/5 (0.00)	0.887579
contestClassic	-258	0/5 (0.00)	2.89121604	-258	0/5 (0.00)	2.479766	-260.4	0/5 (0.00)	2.914431
mediumClassic	465.4	0/5 (0.00)	5.03221798	465.4	0/5 (0.00)	4.76262	-388.2	2/5 (0.40)	58.60023
minimaxClassic	-492	0/5 (0.00)	0.07272911	-492	0/5 (0.00)	0.068444	-291.2	1/5 (0.20)	0.087554
trappedClassic	-295.2	1/5 (0.20)	0.03400135	-295.2	1/5 (0.20)	0.032837	-295.2	1/5 (0.20)	0.035589

- Khi so 2 hàm lượng giá mới và cũ ở Directional Ghost, ta có thể thấy sự chênh lệch giữa 2 hàm eval là không đáng kể. Với 8/75 và 6/75 trận thắng cho Eval mới và cũ tương ứng.
- Trong 5 map ta thực nghiệm ở trên và dựa vào kết quả, có thể dễ dàng thấy được Expectimax trả về kết quả tương đối vượt trội hơn so với Minimax và AlphaBeta.
- Chiến lược ta xây dựng ban đầu là ưu tiên ăn thức ăn thay vì ưu tiên ăn ma. Bởi vì thế nên ở map capsuleClassic đã trả ra kết quả không mấy tốt đẹp, ở một map tương đối nhỏ và phụ thuộc vô chiến lược ăn ma thì hàm Eval mới đã không thể phát huy sức mạnh cả ở 3 thuật toán trên. Tuy vậy Expectimax vẫn có thể thắng đc 1/5 trận, là vì thuật toán lựa chọn xác suất để đưa ra hành động.
- Ở cả 3 map như contestClassic, mediumClassic hay minimaxClassic nhìn chung cả 3 thuật toán đều trả về kết quả tốt với hàm Eval mới. Nhưng có vẻ Expectimax nhỉnh hơn 2 thuật toán kia.
- Với depth = 2 thời gian xử lý của thuật toán AlphaBeta tốt hơn so với thuật toán Minimax ở các layer có độ khó cao hơn, nhờ vào kĩ thuật Pruning. Còn

ở Expectimax thì sẽ tốn nhiều thời gian để rình mồi trong những màn quá nhiều Capsule hoặc bị lưỡng lự

- Ở map trappedClassic chúng ta có kết quả tương đối giống nhau ở cả 3 thuật toán với hàm Eval mới. Một phần là vì ta chạy 5 lần để đánh giá, phần còn lại do đây là Random Ghost. Kết quả tương tự cũng xảy đến ở hàm Eval cũ.
- Còn hai thuật toán MinimaxAgent và AlphaBetaAgent gần như tương tự nhau, nguyên nhân là do 2 thuật toán này tương tự nhau, nhưng thời gian xử lý của AlphaBetaAgent sẽ tốt hơn MinimaxAgent do cơ chế pruning.
- Tuy nhiên, khi nhìn qua bảng kết quả ở Directional Ghost ta sẽ thấy điều khác biệt, số trận thắng thấp hơn hẳn so với Random Ghost. Nếu ta có một hàm lượng giá tốt hơn và được thử nghiệm trên nhiều map hơn thì có vẻ chúng ta sẽ có một kết quả khả quan hơn rất nhiều.
- Hàm Eval mới đã trả về kết quả cao hơn nhiều so với hàm Eval cũ, tuy nhiên như vậy vẫn là quá ít. Hàm Eval mới tương đối đơn giản, trong khi đó có những đặc trưng không tăng hay giảm một cách đơn điệu như khoảng cách tới chấm thức ăn (cụ thể nếu chấm thức ăn gần nhất quá gần, chấm gần nhì quá xa, dẫn đến điểm khuyến khích chấm gần nhì quá nhỏ so với điểm khuyến khích chấm gần nhất; sự chênh lệch quá lớn này sẽ làm Pacman chọn đứng im thay vì chọn hành động ăn chấm thức ăn). Qua đó có thể thấy ta cần 1 hàm Eval phức tạp hơn nhiều.
- Ta cần thêm đặc trưng về khoảng cách thực tế giữa Pacman tới ma hoặc tới chấm thức ăn, bởi vì khoảng cách Manhattan không phản ánh chính xác độ dài đường đi giữa chúng. Ta có thể chạy BFS hoặc Floyd-Warshall để có thể tìm được khoảng cách chính xác bên cạnh độ đo Manhattan.
- Dưới đây là bảng so sánh về thời gian chạy và kết quả trung bình.





IV. Tổng kết

- Trong các thuật toán, ta có thuật toán Minimax và AlphaBeta tuy là luôn có cùng kết quả thực nghiệm, nhưng ở AlphaBeta sẽ cho thời gian xử lý thấp hơn so với Minimax nếu các nhánh có thứ tự sắp xếp tốt hơn và độ sâu lớn hơn nhờ vào bước cải tiến Pruning so với minimax.
- Expectimax cho ra kết quả tốt hoặc tệ hơn tùy vào thiết kế màn chơi.
- Chiến lược đi săn thức ăn thay vì ăn ma có vẻ sẽ không hoạt động tốt ở những map như capsule hay những map khó nhằn hơn với nhiều ma hơn. Dù sao nếu muốn đạt điểm cao thì ăn ma cũng là một nguồn điểm đáng để cân nhắc.
- Hàm Eval tuy đã hoạt động tốt hơn hàm cũ, nhưng vẫn có những cải thiện cần được thực hiện như đã đề cập ở trên.
- Directional Ghost là một trở ngại lớn hơn nhiều so với Random Ghost.
- Đánh giá hiệu năng: Minimax < AlphaBeta < Expectimax