



VIETNAM NATIONAL UNIVERSITY HO CHI MINH CITY
UNIVERSITY OF INFORMATION TECHNOLOGY

CS106 KHTN

Báo cáo bài tập Multi Agent

Sinh Viên :
Đồng Quốc Thắng

Giảng viên :
Lương Ngọc Hoàng

Ngày 3 tháng 5 năm 2025

Mục lục

1	Ý tưởng và giải thích hàm evaluation	2
1.1	Code cho evaluation function	2
1.2	Ý tưởng	3
1.3	Cách thực hiện các ý tưởng và trọng số	3
2	Kết quả chạy thực nghiệm và nhận xét	5
2.1	Kết quả chạy	5
2.2	Nhận xét	6

Chương 1

Ý tưởng và giải thích hàm evaluation

Code của em cho bài tập này đã đính kèm cùng tệp zip hoặc thầy có thể xem ở:

https://github.com/LowTechTurtle/CS106-AI/tree/main/multi_agent/multiagent

1 Ván chơi demo ở map mediumClassic với depth = 2 và dùng betterEvaluationFunction có ở link sau: <https://drive.google.com/drive/u/1/folders/1Txu5E0kcx2ShZw0w-f-dA1xfVG8TzJ3W>

1.1 Code cho evaluation function

Để cho dễ đọc và theo dõi thì em để phần code của mình ở đây, trong phần code này có hàm closestItemDistance là một hàm sử dụng BFS để tính chính xác khoảng cách từ pacman đến thức ăn/capsule gần nhất(tùy theo mình pass list thức ăn hay list capsule vào function)

```
1 def betterEvaluationFunction(currentGameState):
2     infinity = float('inf')
3     position = currentGameState.getPacmanPosition()
4     score = currentGameState.getScore()
5     ghostStates = currentGameState.getGhostStates()
6     foodList = currentGameState.getFood().asList()
7     capsuleList = currentGameState.getCapsules()
8
9     for ghost in ghostStates:
10         d = manhattanDistance(position, ghost.getPosition())
11         if ghost.scaredTimer > 6 and d < 2:
12             return infinity
13         elif ghost.scaredTimer < 5 and d < 2:
14             return -infinity
15
16     # khoảng cách đến thức ăn gần nhất, tìm bằng BFS, có tính đến tường
17     cFD = closestItemDistance(currentGameState, foodList)
18     if cFD is not None and cFD == 0.0:
19         cFD += 1.0
20     if cFD is None:
```



```
21         cFD = 1.0
22         foodDistance = 1.0/cFD
23
24         # tương tự như thức ăn ở trên, tìm khoảng cách đến capsule gần nhất
25         cCD = closestItemDistance(currentGameState, capsuleList)
26         capsuleDistance = 0.0
27         if cCD is not None:
28             if cCD == 0.0:
29                 capsuleDistance = 1/(cCD+1)
30             else:
31                 capsuleDistance = 1/cCD
32
33         return 10.0*foodDistance + 5.0*score + 0.5*capsuleDistance
```

1.2 Ý tưởng

Về phần ý tưởng của evaluation function này, em có thực hiện những phần sau:

- Khi ở gần các con ma đang bị sợ, pacman sẽ đuổi theo ma để ăn ma => có thêm điểm thưởng lớn
- Khi các con ma không bị sợ, pacman sẽ cố gắng né ma
- Game score sẽ được tính vào evaluation function, việc này để giúp tránh việc pacman đứng im một chỗ trong nhiều lượt
- Có tính tới tường khi tính khoảng cách đến thức ăn/capsule gần nhất, cách này để giúp chọn thức ăn hợp lý hơn(ví dụ có thức ăn cách pacman 3 ô sang ngang và thức ăn cách pacman 2 ô ở chiều dọc, nhưng lại có tường khá dài ở giữa, thì giờ chấm thức ăn gần nhất là chấm đầu tiên chứ không phải là chấm thứ 2)

1.3 Cách thực hiện các ý tưởng và trọng số

Cách thực hiện ý tưởng rượt theo ma khá đơn giản, khi scaredTimer lớn hơn 6 và khoảng cách manhattan nhỏ hơn 2 thì sẽ ưu tiên đuổi theo ma(vì khoảng cách khá nhỏ và thời gian ma đang sợ còn khá lớn).

Khi ma đang đến gần($d < 2$), và thời gian bị sợ còn khá ít hoặc không sợ, pacman sẽ ưu tiên bỏ chạy khỏi ma, có một điều đáng để ý ở đây là em thiết kế pacman khá tham lam, evaluation function nếu không có ma ở gần thì pacman sẽ liên tục ăn các chấm thức ăn mà không quan tâm đến ma, còn nếu ma đến gần thì pacman sẽ ưu tiên bỏ chạy mà không quan tâm đến thức ăn(return -infinity)

Về việc thực hiện ý tưởng giúp pacman không đứng yên một chỗ(tính gamescore vào kết quả của hàm evaluation function), em chỉ đơn giản là sử dụng hàm có sẵn getScore của object currentGameState, sau đó nhân với một scalar, trong trường hợp này sau khi thử một số lần, em thấy chọn số 5 có vẻ cho kết quả tốt.

Về việc tính khoảng cách đến các chấm thức ăn và capsule, em sử dụng BFS để tìm chấm thức ăn/capsule gần nhất nếu có. Vì kết quả của việc đánh giá một trạng thái có



số càng lớn thì càng tốt, mà khoảng cách tới chấm thức ăn càng gần thì càng tốt, nên nó sẽ đồng biến với $1/D$ với D là khoảng cách tới chấm thức ăn.

Trong phần code, em có phòng việc $D = 0$, nên khi $D = 0$ thì em sẽ return 1(tức là trong code em viết $1/(D+1)$ để đọc code dễ hiểu ý định hơn)

Về cách chọn trọng số cho thức ăn và capsule, vì thức ăn có rất nhiều và chấm thức ăn gần nhất với pacman có thể là gần hơn capsule gần nhất với pacman rất nhiều nên pacman sẽ ưu tiên ăn các chấm thức ăn hơn rất nhiều, ở đây em chọn $10 * \text{khoảng cách đến chấm thức ăn}$, nhưng chỉ chọn $0.5 * \text{khoảng cách đến capsule}$, em thấy pacman thực hiện khá tốt với cặp trọng số này.

Chương 2

Kết quả chạy thực nghiệm và nhận xét

2.1 Kết quả chạy

Kết quả của dùng betterEvaluationFunction

Các số liệu trong bảng lần lượt là số điểm đạt được, thắng hay thua và thời gian chạy.

Các test trong bài đều được chạy với depth = 2, và frameTime = 0

	alphabeta	minimax	expectimax
medium classic(23521421)	1929, win, 2.38564395904541	1929, loss, 2.7298295497894287	1518, win, 2.8022260665893555
medium classic(23521422)	1727, win, 2.484457015991211	1727, win, 2.7661068439483643	1515, win, 2.744891405105591
medium classic(23521423)	1543, win, 2.294806957244873	1543, win, 2.4388861656188965	1543, win, 2.4530909061431885
medium classic(23521424)	1705, win, 2.7609803676605225	1705, win, 3.065098524093628	1698, win, 3.067396402359009
medium classic(23521425)	1903, win, 3.0104012489318848	1903, win, 3.486083507537842	1688, win, 3.3189778327941895
test classic(23521421)	561, win, 1.1613445281982422	561, win, 1.1750481128692627	559, win, 1.1876397132873535
test classic(23521422)	557, win, 1.1661787033081055	557, win, 1.2042896747589111	557, win, 1.2155406475067139
test classic(23521423)	549, win, 1.192296028137207	549, win, 1.2385618686676025	561, win, 1.1841650009155273
test classic(23521424)	-499, loss, 1.08493971824646	-499, loss, 1.101736307144165	-497, loss, 1.0960009098052979
test classic(23521425)	561, win, 1.167637825012207	561, win, 1.1829464435577393	561, win, 1.2005770206451416
power classic(23521421)	2865, win, 6.016572713851929	2865, win, 9.049036026000977	1862, win, 6.930486440658569
power classic(23521422)	1510, win, 4.949602365493774	1510, win, 6.735820531845093	1111, win, 6.159541606903076
power classic(23521423)	2268, win, 4.957549810409546	2268, win, 7.392526149749756	366, loss, 4.961523771286011
power classic(23521424)	2869, win, 5.193290710449219	2869, win, 6.814060211181641	1898, win, 6.321489095687866
power classic(23521425)	1502, win, 6.314326763153076	1502, win, 8.747381925582886	1534, win, 7.0340166091918945
tricky classic(23521421)	2442, win, 16.01389765739441	2442, win, 25.033050298690796	2582, win, 35.78552603721619
tricky classic(23521422)	1198, loss, 14.40521764755249	1198, loss, 26.08645486831665	1806, win, 24.582924365997314
tricky classic(23521423)	1650, win, 15.37104320526123	1650, win, 22.171001195907593	1638, win, 34.11615037918091
tricky classic(23521424)	1308, loss, 16.520543098449707	1308, loss, 29.444886207580566	2201, win, 22.631269931793213
tricky classic(23521425)	-322, loss, 4.487882614135742	-322, loss, 7.22793984413147	-362, loss, 8.005659818649292
contest classic(23521421)	1865, win, 2.5486466884613037	1865, win, 2.9023985862731934	1812, win, 3.6283786296844482
contest classic(23521422)	1493, win, 2.436218500137329	1493, win, 2.6590375900268555	2000, win, 4.106660842895508
contest classic(23521423)	1856, win, 2.6712465286254883	1856, win, 2.979978084564209	2210, win, 3.848423957824707
contest classic(23521424)	783, loss, 2.720353364944458	783, loss, 3.1099727153778076	1594, win, 3.894534111022949
contest classic(23521425)	1677, win, 2.477421283721924	1677, win, 2.7353813648223877	1820, win, 3.6596479415893555

Kết quả của dùng scoreEvaluationFunction



	alphabeta	minimax	expectimax
medium classic(23521421)	-140, loss, 5.925870180130005	-140, loss, 6.871183633804321	128, loss, 4.082847595214844
medium classic(23521422)	-161, loss, 1.5477473735809326	-161, loss, 1.6115975379943848	333, loss, 3.898566246032715
medium classic(23521423)	-422, loss, 7.549378395080566	-422, loss, 8.811150789260864	-198, loss, 3.217586040496826
medium classic(23521424)	41, loss, 3.47926926612854	41, loss, 3.914333435058594	-781, loss, 8.340965509414673
medium classic(23521425)	-665, loss, 10.390665531158447	-665, loss, 12.239197254180908	133, loss, 2.9846487045288086
test classic(23521421)	518, win, 1.2245090007781982	518, win, 1.2850453853607178	548, win, 1.1779060363769531
test classic(23521422)	infinite loop, ma không bắt pacman	infinite loop, ma không bắt pacman	562, win, 1.1192138195037842
test classic(23521423)	526, win, 1.177544355392456	526, win, 1.2178997993469238	560, win, 1.1254053115844727
test classic(23521424)	538, win, 1.1702117919921875	538, win, 1.2120156288146973	508, win, 1.2829253673553467
test classic(23521425)	552, win, 1.1235477924346924	552, win, 1.151916742324829	552, win, 1.1614325046539307
power classic(23521421)	2296, loss, 7.099449157714844	2296, loss, 12.00665545463562	-501, loss, 78.1387152671814
power classic(23521422)	1150, loss, 6.184284448623657	1150, loss, 10.409898042678833	1261, loss, 12.413796186447144
power classic(23521423)	981, loss, 5.20570969581604	981, loss, 8.322741508483887	-191, loss, 68.63315987586975
power classic(23521424)	208, loss, 2.663276433944702	208, loss, 3.758108139038086	397, loss, 11.135916233062744
power classic(23521425)	-250, loss, 4.524945020675659	-250, loss, 7.568579912185669	489, loss, 34.31569790840149
tricky classic(23521421)	238, loss, 12.425345182418823	238, loss, 22.92916774749756	-189, loss, 4.418254137039185
tricky classic(23521422)	1313, loss, 26.58048129081726	1313, loss, 55.599884271621704	1421, loss, 29.62616753578186
tricky classic(23521423)	1191, loss, 43.345176696777344	1191, loss, 83.50525879859924	453, loss, 11.0224928855896
tricky classic(23521424)	832, loss, 19.797443151474	832, loss, 35.90520691871643	-266, loss, 42.13588500022888
tricky classic(23521425)	-167, loss, 29.610733032226562	-167, loss, 56.42170715332031	-828, loss, 58.09081506729126
contest classic(23521421)	-284, loss, 2.681978464126587	-284, loss, 3.0070250034332275	261, loss, 4.678197383880615
contest classic(23521422)	-220, loss, 1.8217694759368896	-220, loss, 2.0845251083374023	513, loss, 3.1785266399383545
contest classic(23521423)	-156, loss, 1.5951766967773438	-156, loss, 1.731041431427002	-649, loss, 5.658922910690308
contest classic(23521424)	-256, loss, 2.4167416095733643	-256, loss, 2.7741916179656982	-679, loss, 8.282033205032349
contest classic(23521425)	359, loss, 4.073391437530518	359, loss, 5.061917781829834	-263, loss, 4.429476022720337

2.2 Nhận xét

- Giữa alpha beta pruning và minimax, ta có thể thấy số điểm đều là như nhau nhưng alpha beta pruning chạy nhanh hơn minimax khá nhiều, thời gian chạy của minimax khoảng 150% thời gian chạy của alpha beta pruning ở những test case dài. Việc này là do alpha beta pruning cơ bản là mini được tối ưu, bỏ đi những nhánh không có tác dụng để duyệt ít lại, không ảnh hưởng tới cách chọn bước đi.
- Giữa minimax và expectimax có thời gian chạy có thể xem là tương đương nhau, nhưng nhìn chung là expectimax có số điểm đạt được tốt hơn minimax, điều này là do trong bài này đối thủ không phải là hoàn hảo nên expectimax sẽ tốt hơn minimax
- expectimax có thời gian chạy dài hơn alpha beta pruning khá nhiều
- expectimax có số ván thắng nhiều hơn minimax/alphabeta, và số điểm nhìn chung là cao và ổn định hơn, vì đối thủ ở đây không phải là đối thủ hoàn hảo nên pacman dùng expectimax sẽ liều lĩnh hơn và khi chơi nhiều ván, pacman sẽ có số điểm trung bình và số ván thắng cao hơn.
- giữa betterEvaluationFunction và scoreEvaluationFunction có khác biệt lớn và tác động rất lớn đến kết quả: betterEvaluationFunction làm cho pacman liều lĩnh hơn và không bị đứng yên, nên thời gian chơi sẽ ngắn hơn rất nhiều. Ngoài ra, betterEvaluationFunction thực sự là tốt hơn rất nhiều, có thể giúp cho pacman né ma tốt hơn và ăn nhiều thức ăn hơn, dẫn tới số ván thắng rất nhiều. Hầu hết các ván chơi dùng betterEvaluationFunction đều thắng, còn hầu hết các ván chơi dùng scoreEvaluationFunction đều thua