

課題 : I235 6th Report

言語 : C++(Console Application)

氏名 : GAO, Yuwei

学生番号 : s1910092

提出日 : 2019/05/27

- ① ○×ゲーム（三目ならべ）を考える。先手は本質的には3通りあるが、それぞれ10000シミュレーションして、勝率を求めよ。（各シミュレーション中は着手は完全にランダムに選択し、勝てば1勝、引き分けは0.5勝と考える。）

0 --

-- -

- - -

WinRate=0.669

- 0 -

-- -

- - -

WinRate=0.6031

-- -

- 0 -

-- -

WinRate=0.7527

- ② 【発展】上記のアルゴリズムで動くプレイヤ同士を戦わせてみよ。シミュレーション数30のものと、1000のものを戦わせてみよ。20戦以上させて統計的評価を行うこと。各試合結果まではレポートに書かなくてよい。

Match-0:0| Match-1:0| Match-2:0| Match-3:1| Match-4:1| Match-5:0| Match-6:1| Match-7:0| Match-8:1| Match-9:0| Match-10:0| Match-11:0| Match-12:0| Match-13:0| Match-14:1|  
Match-15:0| Match-16:1| Match-17:1| Match-18:1| Match-19:0| Match-20:1| Match-21:1|  
Match-22:1| Match-23:0| Match-24:0| Match-25:1| Match-26:1| Match-27:0| Match-28:1|  
Match-29:0| Match-30:0| Match-31:0| Match-32:0| Match-33:0| Match-34:1| Match-35:0|  
Match-36:1| Match-37:1| Match-38:0| Match-39:1| Match-40:1| Match-41:0| Match-42:1|  
Match-43:1| Match-44:0| Match-45:1| Match-46:0| Match-47:0| Match-48:1| Match-49:0|  
Match-50:-1| Match-51:-1| Match-52:0| Match-53:0| Match-54:0| Match-55:-1| Match-56:0|  
Match-57:-1| Match-58:-1| Match-59:-1| Match-60:0| Match-61:-1| Match-62:0| Match-  
63:0| Match-64:-1| Match-65:0| Match-66:-1| Match-67:0| Match-68:-1| Match-69:-1|  
Match-70:0| Match-71:0| Match-72:0| Match-73:0| Match-74:0| Match-75:0| Match-76:0|  
Match-77:0| Match-78:0| Match-79:0| Match-80:0| Match-81:0| Match-82:0| Match-83:-1|  
Match-84:0| Match-85:0| Match-86:-1| Match-87:-1| Match-88:0| Match-89:0| Match-90:0|  
Match-91:0| Match-92:0| Match-93:0| Match-94:0| Match-95:-1| Match-96:-1| Match-97:-1|  
Match-98:0| Match-99:0|  
WinRate=0.53

シミュレーション 100 (O) VS シミュレーション 30 (X)

O 先手の 50 試合と X 先手の 50 試合（総 100 試合）を行った。

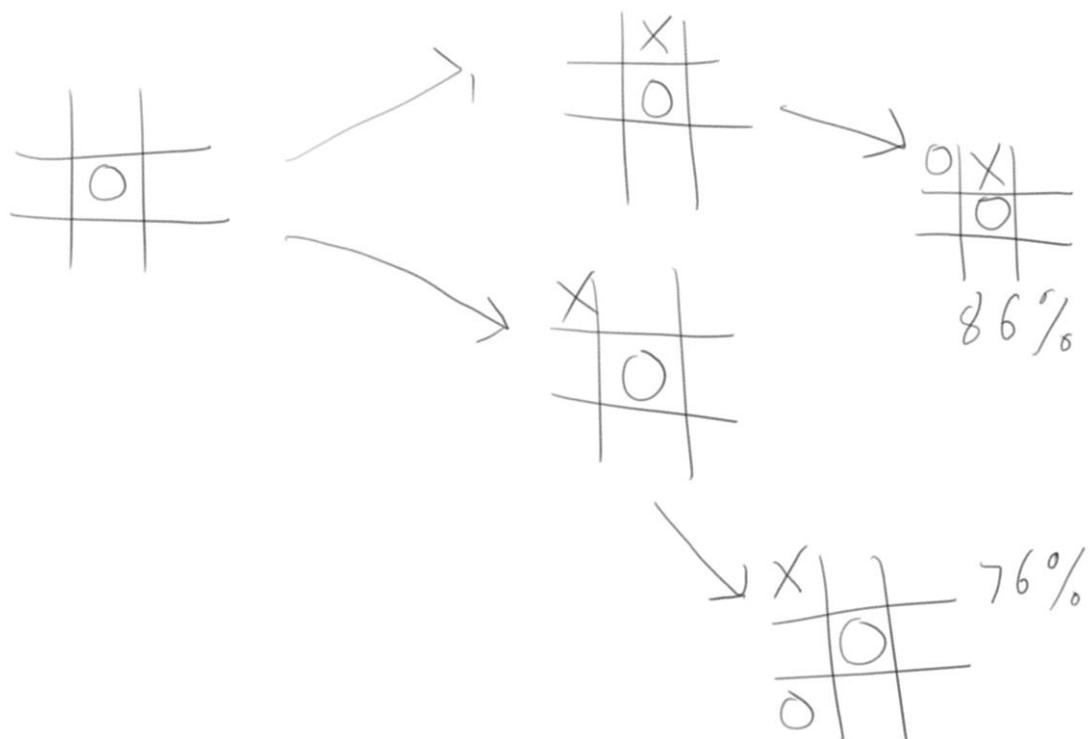
O 先手の 50 試合は、負けたことがない。

X 先手の 50 試合は、勝ったことがない、ただ引き分けの分がちょっと異なる。

$50 + 50 = 100$  試合を行って WinRate は 53 % だった。

- ③ 【発展】“シミュレーション中に、勝てる手があれば打ち、相手の勝ちを防ぐ手があれば打つ”ような「現実的なシミュレーション」を実装してみよ。そのようなプレイヤが強くなるか確認せよ。

先手の場合を考える（後手の場合のやり方もほぼ一緒、勝てないんだけど）：



つまりモンテカルロ法を通じて、各場面に対して勝率最大のを選択して、  
このような図を作ればよいと思う。

強さといえば、先手だったら負けないはずだ。

```
1 #include <iostream>
2 #include <vector>
3 #include <stdio.h>
4 #include <time.h>
5 #include <algorithm>
6 class Board
7 {
8 public:
9     int board[9] { 0, 0, 0, 0, 0, 0, 0, 0, 0 };
10    int whoseTurn = 1;
11    int turnSum = 0;
12    int winner = 0;
13    std::vector<int>* actionList;
14    Board()
15    {
16        actionList = new std::vector<int>();
17    }
18    Board(Board* b)
19    {
20        actionList = new std::vector<int>();
21        for (int i = 0; i < 9; i++)
22        {
23            board[i] = b->board[i];
24        }
25        whoseTurn = b->whoseTurn;
26        turnSum = b->turnSum;
27        winner = b->winner;
28        //actionList = *(b->actionList);
29    }
30    ~Board()
31    {
32        delete(actionList);
33    }
34    void refreshActionList()
35    {
36        actionList->clear();
37        for (int i = 0; i < 9; i++)
38        {
39            if (board[i] == 0)
40                actionList->push_back(i);
41        }
42    }
43    void doAction(int a)
44    {
45        board[a] = whoseTurn;
46        whoseTurn = -whoseTurn;
47        turnSum++;
48    }
49    int randomAction()
50    {
51        refreshActionList();
52        if (actionList->size() > 0)
53        {
54            int a = (*actionList)[rand() % (actionList->size())];
55            doAction(a);
56            return a;
57        }
58        else
59        {
60            return -1;
61        }
62    }
63    int isGoal()
64    {
```

```
65     int winner = 0;
66     int goalRule[][][3] { {0, 1, 2}, {3, 4, 5}, {6, 7, 8}, {0, 3, 6}, {1, 4, 7}, {2, 5, 8}, {0, 4, 8},  ↵
67     {2, 4, 6} };
68     for (auto& i : goalRule)
69     {
70         int sum = 0;
71         for (auto& j : i)
72         {
73             sum += board[j];
74         }
75         if (sum == 3)
76         {
77             winner = 1;
78             break;
79         }
80         else if (sum == -3)
81         {
82             winner = -1;
83             break;
84         }
85     }
86     this->winner = winner;
87     return winner;
88 }
89 void print()
90 {
91     int n = 0;
92     for (auto& i : board)
93     {
94         if (i == 0)
95         {
96             std::cout << "- ";
97         }
98         else if (i == 1)
99         {
100            std::cout << "O ";
101        }
102        else
103        {
104            std::cout << "X ";
105        }
106        n++;
107        if (n == 3)
108        {
109            n = 0;
110            std::cout << std::endl;
111        }
112    }
113 }
114 };
115 Board* simulation(std::vector<Board*> simulationList, int simulationNumber, int ↵
whoseBestAction, bool isShow)//モンテカルロシミュレーション
116 {
117     int maxWinCount = -1;
118     int minWinCount = -1;
119     Board* maxBoard = nullptr;//勝率最大手
120     Board* minBoard = nullptr;//勝率最小手
121     for (auto& i : simulationList)
122     {
123         if (isShow) i->print();
124         int winCount = 0;
125         for (int j = 0; j < simulationNumber; j++)
126         {
```

```
127     Board* b = new Board(i);
128     while (b->randomAction() != -1 && b->isGoal() == 0);
129     if (b->winner == 1)
130     {
131         winCount += 2;
132     }
133     else if (b->winner == 0)
134     {
135         winCount += 1;
136     }
137     delete(b);
138 }
139 if (maxWinCount == -1)
140 {
141     maxWinCount = winCount;
142     minWinCount = winCount;
143     maxBoard = i;
144     minBoard = i;
145 }
146 else
147 {
148     if (winCount > maxWinCount)
149     {
150         maxWinCount = winCount;
151         maxBoard = i;
152     }
153     if (winCount < minWinCount)
154     {
155         minWinCount = winCount;
156         minBoard = i;
157     }
158 }
159 float winRate = winCount / (float)(simulationNumber * 2); //勝率
160 if (isShow) std::cout << "WinRate=" << winRate << std::endl << std::endl;
161 }
162 if (whoseBestAction == 1) return maxBoard;
163 else return minBoard;
164 }
165 int main()
166 {
167     srand((unsigned int)time(NULL));
168     std::vector<Board*> simulationList;
169     Board* b0 = new Board(); //先手は本質的には3通りある
170     b0->doAction(0);
171     Board* b1 = new Board();
172     b1->doAction(1);
173     Board* b4 = new Board();
174     b4->doAction(4);
175     simulationList.push_back(b0);
176     simulationList.push_back(b1);
177     simulationList.push_back(b4);
178     simulation(simulationList, 10000, 1, true); //10000シミュレーションし
179     //以下は発展の部分
180     int winCount = 0;
181     for (int match = 0; match < 100; match++)
182     {
183         Board* b_ = new Board();
184         if (match >= 50) b_->whoseTurn = -1;
185         while (b_->isGoal() == 0 && b_->turnSum < 9)
186         {
187             b_->refreshActionList();
188             simulationList.clear();
189             for (auto& i : (*b_->actionList))
190             {
```

```

191         Board* b_next = new Board(b_);
192         b_next->doAction(i);
193         simulationList.push_back(b_next);
194     }
195     b_ = simulation(simulationList, b_->whoseTurn == 1 ? 1000 : 30, b_->whoseTurn, false); //自分のターンだったら勝率最大手を選択し、相手のターンだったら勝率最小手を選択すること
196 }
197 if (b_->winner == 1) { winCount += 2; }
198 else if (b_->winner == 0) { winCount += 1; }
199 std::cout << "Match-" << match << ":" << b_->winner << "| ";
200 //b_->print();
201 }
202 float winRate = winCount / (100.0f * 2.0f);
203 std::cout << std::endl << "WinRate=" << winRate << std::endl;
204 }
205 /*////////出力///////////
206 0 --
207 --- -
208 --- -
209 --- -
210
211 WinRate=0. 669
212
213 - 0 -
214 - - -
215 - - -
216
217 WinRate=0. 6031
218
219 - - -
220 - 0 -
221 - - -
222
223 WinRate=0. 7527
224
225 Match-0:0| Match-1:0| Match-2:0| Match-3:1| Match-4:1| Match-5:0| Match-6:1| →
   Match-7:0| Match-8:1| Match-9:0| Match-10:0| Match-11:0| Match-12:0| Match-13:0| →
   Match-14:1| Match-15:0| Match-16:1| Match-17:1| Match-18:1| Match-19:0| Match-20:1| →
   Match-21:1| Match-22:1| Match-23:0| Match-24:0| Match-25:1| Match-26:1| →
   Match-27:0| Match-28:1| Match-29:0| Match-30:0| Match-31:0| Match-32:0| Match-33:0| →
   Match-34:1| Match-35:0| Match-36:1| Match-37:1| Match-38:0| Match-39:1| →
   Match-40:1| Match-41:0| Match-42:1| Match-43:1| Match-44:0| Match-45:1| Match-46:0| →
   Match-47:0| Match-48:1| Match-49:0| Match-50:-1| Match-51:-1| Match-52:0| →
   Match-53:0| Match-54:0| Match-55:-1| Match-56:0| Match-57:-1| Match-58:-1| →
   Match-59:-1| Match-60:0| Match-61:-1| Match-62:0| Match-63:0| Match-64:-1| →
   Match-65:0| Match-66:-1| Match-67:0| Match-68:-1| Match-69:-1| Match-70:0| →
   Match-71:0| Match-72:0| Match-73:0| Match-74:0| Match-75:0| Match-76:0| Match-77:0| →
   Match-78:0| Match-79:0| Match-80:0| Match-81:0| Match-82:0| Match-83:-1| →
   Match-84:0| Match-85:0| Match-86:-1| Match-87:-1| Match-88:0| Match-89:0| →
   Match-90:0| Match-91:0| Match-92:0| Match-93:0| Match-94:0| Match-95:-1| →
   Match-96:-1| Match-97:-1| Match-98:0| Match-99:0|
226 WinRate=0. 53
227
228 /*///////////
229

```