

COMS W4701 HW2

Dawei He (dh3027)

February 24, 2022

Problem 1

(a)

According to the consistency property:

$$\begin{aligned} c(n, a, n') + h(n') &\geq h(n) \\ g(n) + c(n, a, n') + h(n') &\geq g(n) + h(n) \\ g(n') + h(n') &\geq g(n) + h(n) \\ f(n') + g(n') &\geq f(n) \end{aligned}$$

(b)

If $f(n) > f(n')$, the previous node, which is on the shortest path to such node, could prevent the algorithm to enter this shortest path. So the algorithm could enter another suboptimal path to reach this node, and then enter the optimal path to reach this node again.

(c)

$$\begin{aligned} 4 + h(C) &> h(A) + 1 > h(B) + 2 \\ \Rightarrow \\ h(A) - 1 &> h(B) > 0 \\ h(C) + 3 &> h(A) > h(B) + 1 \\ 10 \geq h(C) &> h(A) - 3 \\ 12 \geq h(S) &> 0 \\ h(G) = 0 \end{aligned}$$

(d)

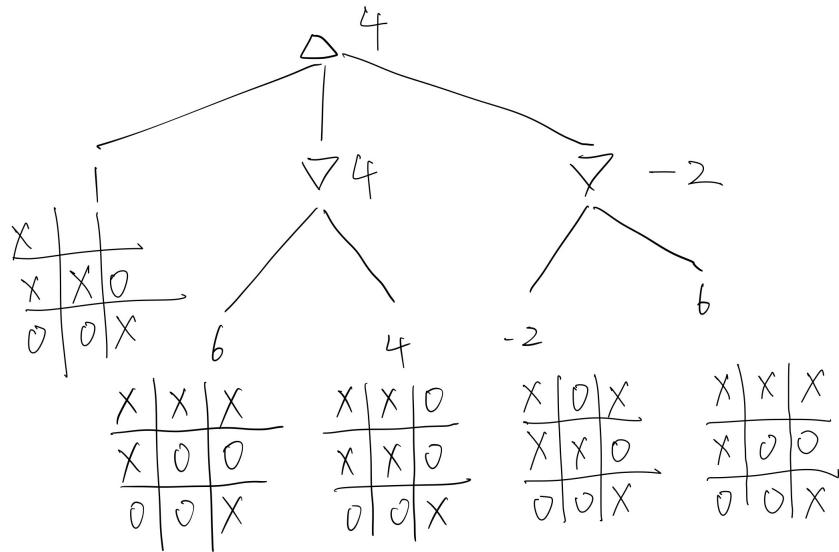
$$\begin{aligned} 1 + \alpha h(A) &> 4 + \alpha h(C) \\ 1 + \alpha h(A) &> 2 + \alpha h(B) \end{aligned}$$

Example:

$$\alpha = 4; h(S) = 11; h(A) = 11; h(B) = 10; h(C) = 10; h(G) = 0$$

Problem 2

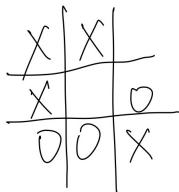
(a)



(b)

The minimax values of each node is shown in (a).

The best move for X to make is to place the block whose point is 5.



The expected score of the game is 4.

(c)

The order is the same shown in (a)

The α is 4, β is -2, the v is -2.

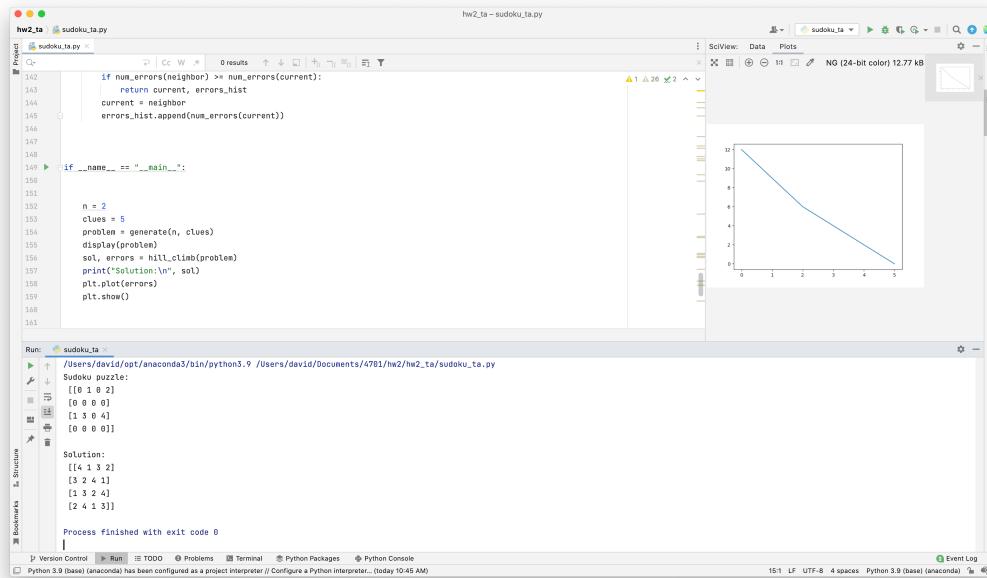
(d)

The value of chance nodes will become the expectation of their children's value instead of the minimum values. The new utility is 5. The best move is also to place the x to the block whose point is 5.

Problem 3

3.3

Solved puzzle:

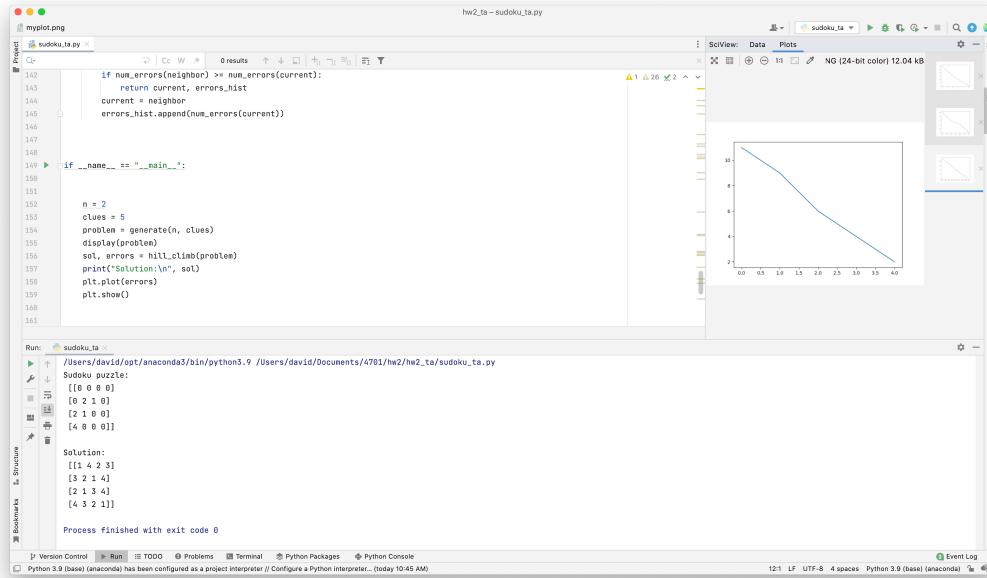


A screenshot of a Jupyter Notebook interface. The code cell contains Python code for solving a Sudoku puzzle using a hill-climbing algorithm. The output shows a solved 4x4 Sudoku grid:

```
[[4 1 3 2]
 [3 2 4 1]
 [1 3 2 4]
 [2 4 1 3]]
```

The plot pane shows a line graph of error counts over time, starting at 12 and decreasing to 0.

Unsolved puzzle:



A screenshot of a Jupyter Notebook interface. The code cell contains Python code for solving a Sudoku puzzle using a hill-climbing algorithm. The output shows an unsolved 4x4 Sudoku grid:

```
[[0 0 0 0]
 [0 2 1 0]
 [2 1 0 0]
 [4 0 0 0]]
```

The plot pane shows a line graph of error counts over time, starting at 12 and decreasing to approximately 2.

Report:

The screenshot shows the PyCharm IDE interface. The top navigation bar displays 'hw2_ta - sudoku_ta.py'. The main editor window contains the code for 'sudoku_ta.py'. The code performs a batch processing loop where it generates problems, solves them using hill-climbing, and tracks success and errors. It then calculates average success rate and average error across all final states. The run output window at the bottom shows the command run and the resulting output: 'Average success rate is: 0.59' and 'Average error over all final states is: 1.87'. The status bar at the bottom indicates Python 3.9 (base) (anaconda) has been configured.

```
batch = 100
final_errors = []
success_count = 0
for i in range(batch):
    n = 2
    clues = 5
    problem = generate(n, clues)
    sol, errors = hill_climb(problem)
    final_errors.append(errors[-1])
    if errors[-1] == []:
        success_count += 1

avg_success_rate = success_count/len(final_errors)
avg_final_error = sum(final_errors) / len(final_errors)
print('Average success rate is: ', avg_success_rate)
print('Average error over all final states is: ', avg_final_error)

if __name__ == "__main__":
    for i in range(batch):
        if errors[-1] == []:
            success_count += 1

Process finished with exit code 0
```

3.4

The sideways move with horizontal segment:

hw2_ta - sudoku_ta.py

```

148     sideways += 1
149     current = neighbor
150     errors_hist.append(num_errors(current))
151
152
153
154 if __name__ == "__main__":
155
156     n = 2
157     clues = 5
158     max_sideways = 5
159     problem = generate(n, clues)
160     display(problem)
161     sol, errors = hill_climb(problem, max_sideways)
162     print("Solution:\n", sol)
163     plt.plot(errors)
164     plt.show()
165
166
167 if __name__ == "__main__":

```

Run: sudoku_ta

```

/usr/local/bin/python3.9 /Users/david/opt/anaconda3/bin/python3.9 /Users/david/Documents/4701/hw2/hw2_ta/sudoku_ta.py
Sudoku puzzle:
[[0 4 0 0]
 [3 0 1 4]
 [0 8 4 0]
 [0 0 0 0]]
Solution:
[[1 2 3]
 [3 2 1 4]
 [2 3 4 1]
 [4 1 3 2]]

```

Process finished with exit code 0

Version Control Run TODO Problems Terminal Python Packages Python Console Event Log

Toggle transparency chessboard under image

159:33 LF UTF-8 4 spaces Python 3.9 (base) (anaconda)

Report:

hw2_ta - sudoku_ta.py

```

161     sideways = 0
162     # sol, errors = hill_climb(problem, max_sideways)
163     # print("Solution:\n", sol)
164     # plt.plot(errors)
165     # plt.show()
166
167
168     n = 2
169     clues = 5
170     max_sideways = 10
171     batch = 100
172     final_errors = []
173     success_count = 0
174     for i in range(batch):
175         problem = generate(n, clues)
176         sol, errors = hill_climb(problem, max_sideways)
177         final_errors.append(errors[-1])
178         if errors[-1] == 0:
179             success_count += 1
180
181
182 if __name__ == "__main__":

```

Run: sudoku_ta

```

/usr/local/bin/python3.9 /Users/david/opt/anaconda3/bin/python3.9 /Users/david/Documents/4701/hw2/hw2_ta/sudoku_ta.py
Average success rate is: 0.84
Average error over all final states is: 0.69

```

Process finished with exit code 0

Version Control Run TODO Problems Terminal Python Packages Python Console Event Log

Python 3.9 (base) (anaconda) has been configured ... (today 10:45 AM) 166:1 LF UTF-8 4 spaces Python 3.9 (base) (anaconda)

3.5

With sideways and restarts, the hill climbing method can successfully solve all $n=2$ and $c = 5$ puzzles.

The screenshot shows the PyCharm IDE interface with the following details:

- Project:** hw2_ta
- File:** sudoku_ta.py
- Code Content:**

```
n = 2
clues = 5
max_sideways = 10
max_restarts = 10
batch = 100
final_errors = []
success_count = 0
for i in range(batch):
    problem = generate(n, clues)
    sol, errors = hill_climb(problem, max_sideways, max_restarts)
    final_errors.append(errors[-1])
    if errors[-1] == 0:
        success_count += 1

avg_success_rate = success_count/len(final_errors)
avg_final_error = sum(final_errors) / len(final_errors)
print('Average success rate is: ', avg_success_rate)
print('Average error over all final states is: ', avg_final_error)
if __name__ == "__main__":
    pass
```
- Run:** sudoku_ta
- Output:**

```
Average success rate is: 1.0
Average error over all final states is: 0.0

Process finished with exit code 0
```
- Sidebar:** Structure, Bookmarks
- Bottom Navigation:** Version Control, Run, TODO, Problems, Terminal, Python Packages, Python Console, Event Log
- Status Bar:** Python 3.9 (base) (anaconda) has been configured... (today 10:45 AM) 182:16 LF UTF-8 4 spaces Python 3.9 (base) (anaconda)

Successfully solved standard sudoku problem:

The screenshot shows the PyCharm IDE interface with the following details:

- Project:** hw2_ta
- Code Editor:** sudoku_ta.py


```

152         current = neighbor
153     elif num_errors(neighbor) == num_errors(current) and
154         current == initialize(problem):
155         restarts += 1
156     elif restarts >= max_restarts:
157         return current, errors_hist
158     else:
159         current = neighbor
160
161     errors_hist.append(num_errors(current))
162
163
164 if __name__ == "__main__":
165
166     n = 3
167     clues = 40
168     max_sideways = 300
169     max_restarts = 30
170     problem = generate(n, clues)
171     display(problem)
172     sol, errors = hill_climb(problem, max_sideways, max_restarts)
173     print("Solution:\n", sol)
174     print("Last error: ", errors[-1])
175     plt.plot(errors)
176     plt.show()
177
178     #
179     # n = 2
180
181 if __name__ == "__main__"
      
```
- Run:** sudoku_ta


```

Sudoku puzzle:
[[0 0 0 5 0 6 7 3 2]
 [0 5 0 0 0 7 0 9 4 0]
 [0 2 3 0 0 4 0 0 5]
 [8 0 7 6 5 9 0 0 0]
 [5 6 0 3 0 0 8 7 0]
 [2 0 0 4 0 0 0 0 0]
 [3 0 5 7 4 2 6 0 9]
 [0 9 0 0 3 5 4 2 0]
 [4 7 0 0 6 0 0 0 0]]
      
```

Solution:

```

[[9 8 4 5 1 6 7 3 2]
 [1 5 6 2 7 3 9 4 8]
 [7 2 3 8 9 4 1 6 5]
 [8 4 7 6 5 9 2 1 3]
 [5 6 9 3 2 1 8 7 4]
 [2 3 1 4 8 7 5 9 6]
 [3 1 5 7 4 2 6 8 9]
 [6 9 8 1 3 5 4 2 7]
 [4 7 2 9 6 8 3 5 1]]
      
```

Last error: 0
- Bottom Status Bar:**
 - Version Control, Run, TODO, Problems, Terminal, Python Packages, Python Console
 - Python 3.9 (base) (anaconda) has been configured as a project interpreter // Configure a Python interpreter... (yesterday 10:45 AM)
 - 169:23 LF UTF-8 4 spaces Python 3.9 (base) (anaconda)

Problem4

4.3

Connect 3 on 4X4 board:

hw2_ta connectfour_ta.py

```

199
200     # Play a Connect-k game given grid size (m,n)
201     # Optional search depth parameters for player X and player O
202     state = np.full((m, n), ".")
203     print("Connect", k, "on a", m, "by", n, "board")
204     player = "X"
205
206     while state is not None:
207         print(np.matrix(state), "\n")
208         if player == "X":
209             value, state = alpha_beta_search(state, player, k, x_max_depth)
210             player = "O"
211         else:
212             value, state = alpha_beta_search(state, player, k, o_max_depth)
213             player = "X"
214
215         if value > 0:
216             print("X wins!")
217         elif value < 0:
218             print("O wins!")
219         else:
220             print("Draw!")
221
222     if __name__ == "__main__":
223         m, n, k = 4, 4, 3
224         game_loop(m, n, k)
225
226
Run: connectfour_ta
[[[., ., ., .], [., ., ., .], [., ., ., .], [., ., ., .]], [[[., ., ., .], [., ., ., .], [., ., ., .], [., ., ., .]], [[., ., ., .], [., ., ., .], [., ., ., .], [., ., ., .]], [[., ., ., .], [., ., ., .], [., ., ., .], [., ., ., .]]], [[., ., ., .], [., ., ., .], [., ., ., .], [., ., ., .]], [[., ., ., .], [., ., ., .], [., ., ., .], [., ., ., .]]]
X wins!
Process finished with exit code 0

```

Connect 4 on 4X4 board:

The screenshot shows the PyCharm IDE interface with the following details:

- Project:** hw2_ta
- Code Editor:** connectfour_ta.py


```

199
200:     # Play a Connect-k game given grid size (m,n)
201:     # Optional search depth parameters for player X and player O
202:     state = np.full((m, n), ".")
203:     print("Connect", k, "on a", m, "by", n, "board")
204:     player = "X"
205:
206:     while state is not None:
207:         print(np.matrix(state), "\n")
208:         if player == "X":
209:             value, state = alpha_beta_search(state, player, k, x_max_depth)
210:             player = "O"
211:         else:
212:             value, state = alpha_beta_search(state, player, k, o_max_depth)
213:             player = "X"
214:
215:         if value > 0:
216:             print("X wins!")
217:         elif value < 0:
218:             print("O wins!")
219:         else:
220:             print("Draw!")
221:
222:
223:     if __name__ == "__main__":
224:         m, n, k = 4, 4, 4
225:         game_loop(m, n, k)
226
      
```
- Run:** connectfour_ta


```

[['.', '.', 'X', 'X'],
 ['X', 'X', 'O', 'O'],
 ['.', 'O', 'X', 'X'],
 ['.', 'O', 'X', 'X']]
[[['.', '.', 'X', 'O'],
 ['.', '.', 'X', 'X'],
 ['X', 'X', 'O', 'O'],
 ['.', 'O', 'X', 'X']],
 [[['.', 'O', 'X', 'O'],
 ['.', 'X', 'X', 'X'],
 ['X', 'X', 'O', 'O'],
 ['.', 'O', 'X', 'X']],
 [[['.', 'O', 'X', 'O'],
 ['.', 'X', 'X', 'X'],
 ['X', 'X', 'O', 'O'],
 ['.', 'O', 'X', 'X']],
 Draw!
Process finished with exit code 0
      
```
- Bottom Status Bar:**
 - Version Control
 - Run
 - TODO
 - Problems
 - Debug
 - Terminal
 - Python Packages
 - Python Console
 - Python 3.9 (base) (anaconda) has been configured as a project interpreter // Configure a Python interpreter... (2/22/22, 10:45 AM)
 - 225:23 LF UTF-8 4 spaces Python 3.9 (base) (anaconda)
 - Event Log

Connect 4 on 6X7 board with both 5 max depth (draw):

Connect 4 on 6X7 board with both 6 max depth (O wins):

The screenshot shows the PyCharm IDE interface with the following details:

- Project:** hw2_ta
- Code Editor:** connectfour_ta.py


```

199     );
200     # Play a Connect-k game given grid size (m,n)
201     # Optional search depth parameters for player X and player O
202     state = np.full((m, n), ".")
203     print("Connect", k, "on a", m, "by", n, "board")
204     player = "X"
205
206     while state is not None:
207         print(np.matrix(state), "\n")
208         if player == "X":
209             value, state = alpha_beta_search(state, player, k, x_max_depth)
210             player = "O"
211         else:
212             value, state = alpha_beta_search(state, player, k, o_max_depth)
213             player = "X"
214
215         if value > 0:
216             print("X wins!")
217         elif value < 0:
218             print("O wins!")
219         else:
220             print("Draw!")
221
222     if __name__ == "__main__":
223         m, n, k = 6, 7, 4
224         game_loop(m, n, k, 6, 6)
225
226 if __name__ == "__main__"
      
```
- Run:** connectfour_ta


```

[[ ' ', ' ', ' ', ' ', ' ', ' ', ' ']
 [ ' ', ' ', ' ', ' ', ' ', ' ', ' ']
 [ ' ', ' ', ' ', ' ', ' ', ' ', ' ']
 [ ' ', ' ', ' ', ' ', ' ', ' ', ' ']
 [ ' ', ' ', ' ', ' ', ' ', ' ', ' ']
 [ ' ', ' ', ' ', ' ', ' ', ' ', ' ']

[[ ' ', ' ', ' ', ' ', ' ', ' ', ' ']
 [ ' ', ' ', ' ', ' ', ' ', ' ', ' ']
 [ ' ', ' ', ' ', ' ', ' ', ' ', ' ']
 [ ' ', ' ', ' ', ' ', ' ', ' ', ' ']
 [ ' ', ' ', ' ', ' ', ' ', ' ', ' ']
 [ ' ', ' ', ' ', ' ', ' ', ' ', ' ']

0 wins!
      
```
- Status Bar:** Process finished with exit code 0
- Bottom Bar:** Version Control, Run, TODO, Problems, Debug, Terminal, Python Packages, Python Console
- Event Log:** Python 3.9 (base) (anaconda) has been configured as a project interpreter // Configure a Python interpreter... (2/22/22, 10:45 AM)

Connect 4 on 6X7 board with 6 max depth for X and 5 max depth for O(X wins):

