CS441 Artificial Intelligence (Fall 2019)

Writeup--HW 1: Dinner Party

Weiwei Chen

weiwchen@pdx.edu

Task:

The task for the program is to carefully arrange a dinner table to get the best scores based on the given set of n people, the pair points, opposite points and the integer-valued preference function.

Strategy used:

In this program, I used local search algorithm to solve the problem. I initialized the seat states in order with the set of people. That is, if the given number of people is 4, then the 1st people seats at seat#1, the 2nd people seats at seat#2, the same thing with 3rd and 4th people. We can also tell that the1st people seats opposite to the 3rd people, 2nd is opposite to the 4th. The first two are hosts and the rest two are guests. The reason to initialize the seats like this is that the opposite points will be optimal since each host has a guest that is opposite to him/her.

The local search algorithm will help to swap the seat for the people with all other seats and keep the best record of the score and the related arrangement of the set of people. The detailed steps are as follows:

Step 1: Set the initial seats to be seated according to the people's order, that is,

if the seat number is i, then the people is the ith people in the seat

Step 2: Calculate the score for the table

Step 3: Swap the people for the first seat with its next seat

Step 4: Repeat Step 2, compare the scores, keep the record of the seats state with the higher score. Repeat Step 3, then swap with the next seat

Step 5: Repeat Step 4, until time(60 seconds) is over, return the highest score along with the seats information

Thoughts:

In this program I also tried a different swap strategy, that is, swap the position with the opposite people, but it didn't give me a better solution. I think the reason might be it overlapped with my best solution with the previous strategy.

My best solutions for the three input are 77, 476, and 106, which are not that high. I might also try to change the initial states to be: every other position can be hosts, and the others would be guests. In this way, not only opposite points will be optimized, but also the pair points will be the highest. I may also try to tune the swap strategy to direct the program to a better solution. For example, I may set some conditions to make sure that every host people seats beside a guest and also seats opposite to a guest after every swap to get a better pair points and opposite points as a better base. Based on this base, we can try different way of swaps to find an even better solution.

Setup and Build:

1) Install Python
2) $ git clone https://github.com/waiwaixiaochen/Dinner_table.git
3) Run the program: $ python dinner_party.py