CSL452– Artificial Intelligence
Lab 2

Due on 19/2/2016 11.55pm

Instructions: Upload to your moodle account one zip file containing the following. Please do not submit hardcopy of your solutions. In case moodle is not accessible email the zip file to the instructor at ckn@iitrpr.ac.in. Late submission is not allowed without prior approval of the instructor. You are expected to follow the honor code of the course while doing this homework.

1. **You are allowed to work in teams of size at most 2 for this programming lab**.

2. A neatly formatted PDF document with your answers for each of the questions in the homework. You can use latex, MS word or any other software to create the PDF.

3. Include a separate folder named as 'code' containing the scripts for the homework along with the necessary data files.

4. Include a README file explaining how to execute the scripts.

5. Name the ZIP file using the following convention rollnumber1rollnumber2hwnumber.zip

---

1. Coal Blocks Auction

In this problem you will be experimenting with different local search algorithms. You will take a complex problem, learn to formulate it as an AI single agent search problem and solve it using local search algorithms.

The Ministry for Power, Coal, New and Renewable Energy needs help with the auctioning of the coal blocks whose allotments were cancelled by the Supreme Court. Being the smartest AI student, the Ministry has approached you for writing software that will take as inputs the different bids and outputs an allotment that will maximize the government's revenue. The Ministry has decided to auction N coal blocks and has asked companies to bid for each block. However, the companies with aim of maximizing their profit submit bids that combine multiple blocks. The companies would like to have ownership of as many blocks in a state so as to minimize their costs for developing the infrastructure. For example, a company might bid 400 (in crores) for blocks 1, 3, 4, and 7, but bid 100 for only 1, 3 and 4.

Your software should tell the Ministry the bids that should be accepted. The Ministry has decided not to accept more than one bid from any company and not every company needs to win a bid. A company can submit any number of bids. It might also happen that some of the coal blocks do not get allotted at the end of the auction.

Formulate this problem as a state space search for a single agent and implement hill climbing with random restarts as your initial solution. You are welcome to add any novel element to the hill-climbing search. However, you are not allowed to use any integer linear programming or related algorithms to solve this problem. Include in your submission a brief description of your modification to hill climbing search.

Input Format:

Your code must be able to read inputs from a text file that will be provided as command argument. The input format is as follows

T
N
B
C
cid ncid
cid nbid bid-value block-id block id … block id
cid nbid bid-value block-id block-id … block-id

T refers to the wallclock time in minutes that will be given to your software to find a solution. *N* is the number of coal blocks; *C* is the number of companies; *B* is the number of bids. The bids of a single company are grouped together. The first line is the company id (*cid*) and the total number of bids made this company (*ncid*). Each line thereafter corresponds to a bid. Each bid starts with the company id, the number of coal blocks bid by the company in this bid (*nbid*), the bid-value, and the coal block ids. The bids are implicitly numbered. This is repeated for every company. For example,

3
7
2
6
0 3
0 4 400 0 2 3 6
0 3 100 0 2 3
0 3 50 2 3 4
1 3
1 3 300 1 4 5
1 2 200 1 5
1 3 40 1 3 4

Output Format:

The output of your code (the winning bids) should be written to a text file that will also be provide as a command argument in the following format:

revenue bid-id bid-id … bid-id
For the example problem, this would be
700 0 3

Program Execution Format:

We will be running automated scripts to check the output of your code. Hence we should be able to run your code as follows:
mysolution inputfile outputfile

Evaluation:

We will run your code multiple times till the time limit is met, and take the best solution amongst all the runs. You are provided with 10 sample problems. The final evaluation will be conducted on a similar set of problems. The points awarded will be your normalized performance score relative to other groups in the class. **To be fair to all, this part of the programming lab should be implemented in C.**

2. The Three Musketeers Game

Three musketeers is a strategy board game involving two players. Read more about the game here[1]. An interactive version of this game can be found in *p2\interactivegameplay* directory of the lab folder. Familiarize yourself with the game using this utility. In the interactive game play, you will be competing as the musketeer against the computer agent, which will play the role of the soldier. Execute *main.py* in the *p2\interactivegameplay* folder from command line to start the game. You will need Pygame library and Python 2.7 to run this application

The next step in this lab is to write the software for the computer agent musketeer, which will compete against the soldier agent. You will implement minimax search with alpha-beta pruning. Implement your code in the *controller2.py* script in the *p2\multiagent* folder. The input is the current board state. Use game search techniques to determine the best possible move for the musketeer using the input as the start state. The output of your function should be in the following format [[x1, y1], [x2, y2]]. This is interpreted as the musketeer at board position (x1, y1) is to be moved to (x2, y2).

Important: At the time of submission, change *controller2.py* to *rollNumber.py* and submit only this script. We will be using automated script evaluators to check your code.

Include in your submission a brief description about your strategy to implement the musketeer agent.

Evaluation:

We will run your code multiple times and take the best score amongst all the runs. The points awarded will be your normalized performance score relative to other groups in the class.

[1] https://en.wikipedia.org/wiki/Three_Musketeers