CSIS, BITS Pilani K. K. Birla Goa Campus

Artificial Intelligence (CS F407)

Programming Assignment 1
Total Marks: 15
Submission Deadline: 9 PM on 07/09/2024 (Saturday)

Each student must individually do this programming assignment. Your program must be written in Python and should run (without errors) on Python 3.8 or later.
Any form of plagiarism will result in 0 marks being awarded to everyone involved. There will be *no* differentiation between minor and major plagiarism.
Note that the deadline is **9 PM** and not midnight. Five marks per day will be deducted for submissions after the deadline. It will be your responsibility to submit the assignment well in advance and avoid unforeseen problems like power failures etc.

Question 1 (15 marks)

Set Covering Problem (SCP): You are given a randomly generated instance of a Set Covering Problem (U, S), where the universe set U always contains integers from 1 to 100. The collection S contains m subsets of universe set U. The problem is to find minimum number of subsets in S whose union is the universe set U. We say that the subsets in an optimal solution *cover* all the elements in the universe U.

You have to think and decide what state representation you must use and how to design a fitness function so that the Genetic algorithm will find a solution for the Set Covering Problem.

a) Implement the genetic algorithm for finding a state with the best fitness function value. First, implement the version of the genetic algorithm given in the textbook. Find how the algorithm performs when the number of subsets in collection S is 50, 150, 250 and 350. Note that the universe must always contain integers from 1 to 100, inclusive. (Use the *Create* function (see ROLLNO_NAME.py file) for generating a random SCP.) Plot the best fitness function value obtained for the genetic algorithm (with a population size 50 and for 50 generations) for SCP having the following number of subsets in collection S: 50, 150, 250 and 350. The values plotted in the graph for a collection size $|S|$ should be based on at least 10 randomly generated SCPs; plot the mean and standard deviation. Also, plot how the mean best fitness function value changes across the 50 generations for the different collection sizes $|S|$: 50, 150, 250 and 350. Include the plot in the report.

b) Think of ways in which the genetic algorithm can be improved. You can try varying the population size and number of generations as well. In the report, mention the different modifications that you have tried and what effect it had on the best state/solution that could be found and the number of generations it took to find the best state/solution. Include graphs in the report so that the report is easier to read. The graphs must be easy to understand and have proper labels and captions. Use 1.5 line spacing.

The improved algorithm that you submit for evaluation must terminate within 45 seconds. This means that you will have to check the time elapsed inside your program and terminate if the time elapsed crosses 45 seconds[1]. You can also terminate your algorithm if the best fitness function value does not change over several generations.

Use the *Create* function (see ROLLNO_NAME.py file) to generate SCP with different collection S size. You can vary the number of subsets $|S|$, but the universe set $|U|$ must be 100.

Instructions for submission

- You must submit a single program file with the name "ROLLXYZ_FIRSTNAME.py". Your program should include only the improved version of the genetic algorithm. The program that you submit must read the SCP problem from the "scp_test.json" file using the $ReadSetsFromJson()$ function (see ROLLNO_NAME.py file). As mentioned earlier, the program must terminate within 45 seconds. The output of the program that you submit should be as shown below:

```
Roll no : 2020H1030999G
Number of subsets in scp_test.json file : 150
Solution :
0:1, 1:0, 2:1, 3:1, 4:1, 5:1, 6:0, 7:0, 8:1, 9:1, 10:0, 11:0, 12:0, 13:0, 14:0, 15:0, 16:0, 17:1, 18:0, 19:1, 20:1, 21:0,
22:0, 23:0, 24:1, 25:1, 26:0, 27:0, 28:0, 29:0, 30:1, 31:1, 32:1, 33:1, 34:1, 35:1, 36:0, 37:0, 38:1, 39:1, 40:0, 41:1, 42
:1, 43:0, 44:0, 45:0, 46:1, 47:1, 48:1, 49:0, 50:0, 51:0, 52:0, 53:0, 54:0, 55:0, 56:0, 57:1, 58:1, 59:1, 60:1, 61:0, 62:0
, 63:0, 64:0, 65:1, 66:1, 67:0, 68:1, 69:0, 70:0, 71:0, 72:1, 73:1, 74:0, 75:1, 76:1, 77:0, 78:1, 79:0, 80:1, 81:1, 82:1,
83:0, 84:1, 85:1, 86:1, 87:0, 88:1, 89:0, 90:0, 91:0, 92:1, 93:0, 94:0, 95:1, 96:1, 97:1, 98:0, 99:1, 100:0, 101:0, 102:1,
 103:0, 104:0, 105:0, 106:1, 107:0, 108:0, 109:1, 110:1, 111:1, 112:0, 113:1, 114:1, 115:0, 116:1, 117:1, 118:1, 119:0, 12
0:0, 121:0, 122:1, 123:1, 124:1, 125:0, 126:1, 127:0, 128:0, 129:1, 130:0, 131:0, 132:0, 133:0, 134:1, 135:1, 136:1, 137:1
, 138:1, 139:1, 140:0, 141:0, 142:1, 143:1, 144:0, 145:0, 146:1, 147:0, 148:0, 149:0
Fitness value of best state : 90
Minimum number of subsets that can cover the Universe-set : 12
Time taken : 22.32 seconds
```

  The above screenshot may contain mistakes; the screenshot is to be used only to understand how the output should be formatted. The notation 17:1 denotes that the 17^{th} subset in the list of subsets (read from "scp_test.json" file) is present in the best solution found.

- Your report must be named "ROLLXYZ_FIRSTNAME.pdf".
- Please use only capital letters for both the file names. Eg. 2020H1030999G_ADARSH.py and 2020H1030999G_ADARSH.pdf.
- Submit **only** the two files mentioned above. **Don't** zip the files. The assignment submission will be through quanta.
- 7.5 marks will be for the report and 7.5 marks will be for the output of your submitted program.

[1]This can be easily done in Python using very few lines of code.