BRAC University

Department of Computer Science and Engineering

CSE 422: Artificial Intelligence (Section 08)

Quiz 02: Fall 2024 Time: 30 Minutes Marks: 10

Name:	ID:

1. For each of the following problems find out the size of the neighborhood and the search space. [2×2=4]

   (a) A satisfiability problem with 80 variables and 200 clauses. Neighborhood is single flip.

   (b) A graph vertex-coloring problem for a graph with 50 nodes, 250 edges and 6 colors. Use single flip neighborhood.

   SAT: Search space, 2^{80}, Neihghborhood, 80

   Vertex-Color: Search space, 6^{50}, Neihghborhood, 5×50

2. Two friends of yours are running Simulated Annealing Algorithm to solve N-queen problem. Both of them is cooling the temperature using a simple formula $T = \alpha T$. However, one is using $\alpha = 0.99$ another $\alpha = 0.8$. Which algorithm you think will perform better and why? [2] T=0.99 will be better as it will allow temperature to cool down slowly and thus more exploration.

3. Consider the following three algorithms: {Population Based Hill Climbing, Simulated Annealing, Genetic Algorithm}. Sort them according to both their intensification/exploitation and diversification/exploration features. Justify your answer. [2] Exploration wise: Genetive algorithm > Simulated Annealing > Population based hill climbing

4. Consider the local search algorithm *LocalSearch(maxIter,th))*. Suppose that you are using it to solve a graph-coloring problem on a graph with 100 vertices, with 20 colors. How will it perform compared to the greedy Hill Climbing algorithm? What values do you suggest for '*maxIter*' if you want the algorithm to work as good as a complete search algorithm? [2]

```
LocalSearch(maxIter,th)
  s=generateRandomly()
  if isGoal(s) return s
  it=0, nonImproving=0
  while(it++ < maxIter)
    s'=generateRandomNeighbor(s)
    if isGoal(s') return s
    if quality(s') better than quality(s)
      s=s', nonImproving=0
    else
      nonImproving++
    if nonImproving > th
      s=generateRandomly()
            nonImproving++
  return s
```

- Should perform better than greedy Hill Climbing
- MaxIter depends on problem size, as there are 100 vertices and 20 colors, it should be allowed to generate good numebr of neighbors, 100 *times* 20 could be a good solution. However, there is not guarantee.