1. Explain what is Gray coding and why it is preferred over binary coding.

2. Explain what is Elitism and why is it useful?

3. Explain the meaning of the following concepts: generational gap, non-overlapping and overlapping populations.

4. Discuss 3 limitations of Evolutionary Algorithms.

5. i) Compute the time complexity of tournament selection with size q = 10? 
   ii) Is it possible to do a parallel implementation of tournament selection?

6. Explain, and give the pros and cons of static, dynamic, and adaptive penalty functions.

7. Evolutionary search, as opposed to blind random search, have biases. What do you think the term “search bias” means?

8. Comment on the weaknesses of the repair approach for constraint handling.

9. Is it true that by using a mutation scheme that combines Gaussian and Cauchy mutations then we can find the global optimum of any continuous-valued function?

10. Fitness sharing and niching
    i) What is fitness sharing and how is it useful?
    ii) What problems may arise with the method of explicit fitness sharing?
    iii) How is the shared fitness computed in implicit fitness sharing?

11. Comment on the analogy of Evolutionary Algorithms (EA) with Darwin's ideas on biological evolution. The latter has no specific problem solving scope, so what makes EA suitable for problem solving?

12. Given N chromosomes (individuals) in a population and their fitnesses f1,f2,...,fN, describe the probability of selecting the i-th chromosome (1<= i <= N) using the roulette-wheel selection.

13. Roulette-wheel selection may lead to premature convergence due to the "super-individuals" problem. Explain what this problem is.

14. What other selection scheme you know of that suffers less (or not at all) from the "super-individuals" problem?

15. What quantities should you provide when reporting the results obtained by an evolutionary optimisation method? Give examples.
ANSWERS

1. Hamming cliffs can be avoided by Gray coding. The idea is to use a smarter coding scheme to avoid the Hamming Cliff. If we have a smooth function to optimize (in case we know it) it could be a good idea to have smooth trajectories in phase space. It is possible to use Gray coding to represent a number as a bit string. Successive numbers in the Gray code only differ by one bit. The algorithm can achieve a smooth jump if it mutate the correct bit (single point mutation). E.g:

0: 000
1: 001
2: 011
3: 010
4: 110
5: 111
6: 101

2. It is useful to maintain a sub-population of the best individuals. Elitism takes the best individuals and passes them directly to the next generation without any modification. If it is not used then you can lose the best individual already found.

3. Generation gap is the amount of overlap between parents and offspring. Non-overlapping Populations: Parents and offspring never compete, the entire population is replaced. Overlapping Populations: They compete for survival.

4. Parameters need to be tuned, such as the population size, crossover and mutation probabilities. These parameters are problem dependent. EAs require more computation than other optimisation methods. It is not guaranteed that you find the optimal solution, even though the best solution found could be close to optimum. Every time you run the algorithm, it could give a different solution. There is not a general algorithm that is superior for all objective functions / solution spaces (related with the NFL theorem). Thus you need to choose one algorithm to solve a particular task (there is an advantage if you have previous domain knowledge of the problem). Even EAs could be trapped in a local minimum. If the algorithm converges we can't know if the algorithm fell in a local minimum or not.

5. i) The complexity of tournament selection for q=n is O(n), as follows: The algorithm takes a constant number of individuals picked in a random way, each comparison takes a constant time, and n comparisons are required to create the new generation. (Put 10 for n to answer the question.) ii) Yes, tournament selection can be implemented in a parallel fashion, the selection in each sample of q individuals can run in parallel.

6. The penalty function approach converts a constrained problem into an unconstrained one by introducing a penalty function into the objective function. - Static penalty functions: It is fixed at the programming stage and it does not change during evolution. One disadvantage is that it require much prior knowledge of the problem at hand to come up with the right amount of penalty for the fitness function. - Dynamic penalty functions: This kind of function changes as the generations advance, namely the strength of penalty increases over time. In this way the Dynamic function forces the population back into a feasible area. One problem is that it is possible that the algorithm does not find the global optimum if the Dynamic penalty function forces back the population too fast -- then the solution could be stagnated in a local optimum. One possible solution is to introduce prior knowledge of the problem at hand, when such knowledge is available. - Adaptive penalty functions: The function changes as the generations advance (increase or
decrease the strength)
Decrease: If the best members of the population are feasible
Increase: On the other hand the strength is increased to force the population towards a feasible area in the search space

7. The search bias explains how some offspring tend to be more likely generated than others. The search bias depends on the representation and the search operators (crossover and mutation - e.g. 1-bit-flip vs. k-bit-flip).

8. Repair algorithms let us map infeasible individuals into feasible ones. The main weakness here is that for a particular problem, you need to design a repair algorithm - there is no general equation that could be applied. Depending on the problem at hand, the Repair algorithm could be as difficult as to solve the original problem.

9. No. There are no guarantees to find the global optimum and the probability of finding the global optimum depends on a many factors.

10. i) Fitness sharing is a method of lowering the fitness in densely populated regions of the search space. This encourages the population to spread out, i.e. encourages diversity and a better exploration of the search space.
ii) In explicit fitness sharing, the individuals may end up staying around, rather than converge to the optima. Fitness scaling is often used to get round of this problem. However there is a tradeoff between the scaling factor (if set too large then super-individuals result).
iii) In implicit fitness sharing, the shared fitness is computed by dividing the payoff among the equal-best solutions in a random sub-population. Only the equal bests get payoff.

11. The implicit assumption in EA is that high quality parent candidate solutions from different regions of the search space, when combined together via crossover. And subjected to mutation, can sometimes produce high quality offspring candidate solutions.

12. \[ p_i = \frac{f_i}{\sum_{j=1}^{N} f_j} \]

13. The problem of super-individuals occurs when an individual is significantly fitter than the others. It occurs because the sampling with replacement will create too many copies of that individual and the population becomes dominated by these copies, which then leads to loss of genetic diversity. This is a problem especially in the beginning of the evolutionary process, as it leads to premature convergence to a local optimum.

14. Tournament selection, or using fitness-scaling before roulette-wheel selection. Rank-based selection does not have this problem.

15. Statistics computed from several repeated independent runs on the same problem should be reported. E.g. the best fitness value found, averaged over several independent runs, the avg computation time, and the std of these best fitnesses, Another example (when it applies) is the global hit rate, and another is the average and std of the number of fitness evaluations taken for the best fitness to reach a certain pre-defined value.