# NOTE: This is gathered from previous mock exams. Answers are my approximate answers, not official. They might be wrong or incomplete!!! Schema Theorem and Intro

1. Discuss the 5 general principles of a Darwinian system. Illustrate the application of these principles in designing an evolutionary algorithm to solve evo checkers.
2. Discuss the 5 general principles of a Darwinian system. Illustrate the application of these principles in designing an evolutionary algorithm to solve the evo checkers problem.
3. Discuss the 5 general principles of a Darwinian system. Illustrate the application of these principles in designing an evolutionary algorithm to solve the evo checkers problem.  
   **ANSWER**: (Not in cheat sheet, include)  
   EvoCheckers: game board as fitness function. Build a NN at for each game play.
   1. Information structure: binary strings, rv-vectors, graphs, etc.
      1. Evo: neural networks
   2. Copies: selection algorithm (proportionate, truncation, tournament)
      1. Evo: Tournament selection size 5. Select the best half.
   3. Variation: Mutation/crossover operators (uniform, 2-point CX)
      1. Self-adaptive mutation. First mutate Stdev for gaussian distribution, then use this Stdev to mutate NN weights.
   4. Competition: Fitness based/fixed size
      1. Evo: Fixed population size in each generation.
   5. Inheritance: Parent and offspring must be fitness-wise correlated. Schema theorem, building blocks (important partial solutions), disruption, etc.
      1. Evo: Offsprings are NNs functionally like parents.
4. How does proportional selection work? What is the probability that an individual is selected? Give 2 disadvantages of this method and illustrate them with a numerical example.  
   Selects and copies individuals based on their relative fitness to population.  
   A math equations and formulas

   AI-generated content may be incorrect.

Disadvantages: 1- Too much selection pressure if single individuals have much higher fitness value than rest of the population. Fit individuals dominate the population.   
2- Loss of selection pressure when all fitness values converge to similar values. Selection becomes near random, not much improvement.  
Numerical example: A population with 5 individuals. Total fitness 100. A table with 3 columns: Fitness val, Probability, Expected Copies. Rows are individuals.

1. What are schemata? How is schema fitness defined? Formulate the schema theorem and explain what it means. Explain the nature of GA (Genetic Algorithm) convergence using the “building block hypothesis.”
2. What are schemata? How is schema fitness defined? Formulate the schema theorem and explain its meaning. Explain the nature of GA convergence using the “building block hypothesis.”  
   **ANSWER:**Schemata are similarity subsets: #11##0 o(h) = 3, len(h)=4.  
   o(h): schema order 🡪 number of fixed positions in the schema.  
   len(h): schema length 🡪 length between leftmost and rightmost fixed position.  
   m(h,t): number of schema members in generation t.  
   f(h,t): schema fitness 🡪 average fitness of schema members in generation t.  
   Schema Theorem: Low order, high fitness schemata gains exponentially increasing trials (building blocks). This is near-optimal solution. Happens implicitly parallel, multiple schemata processed. Enough samples present statistically reliable information, need a population large enough so that schemata can survive disruption by variation operators. Building block hypothesis says, building blocks (low-order, fit schemata) can be juxtaposed to form a near optimal solution. This means building blocks must be mixed before convergence. Needs representation and variation operators suitable to problem structure, that minimize disruption.
3. Selection algorithms can be categorized based on several criteria. Explain what these criteria mean and illustrate them using a specific selection algorithm: • generational vs. steady-state • selection by reproduction vs. replacement • non-elitism vs. elitism  
   **Answer:**Generational algorithms replace whole population: Select parents 🡪 create offsprings 🡪 replace whole population. Steady-state algorithms replace only few individuals. Elitism ensures best individuals survive to next generation 🡪 Fast convergence, but risk of premature convergence, if pressure too high. Non-elitism: No guarantee best individuals get into new generation 🡪 Slow convergence, more diversity, more exploration.  
   Example: Tournament selection with population size 6, tournament size 2:  
   - Generational + Reproduction + Non-elitist: Select parents using tournament, reproduce 6 children, replace population.  
   - Steady-state + Replacement + Elitist: Select 2 parents using tournament. Create 1 offspring, replace worst individual in the population, if offspring is not the worst.
4. We want to optimize a Trap Function T F with 20 blocks of length k = 3, thus string length ` = 60. The genetic algorithm we use is the same as in the first practical assignment (= family competition, no use of mutation) with a population size of 500.

A math problem with numbers and symbols

AI-generated content may be incorrect.

Do you expect the GA to find the global optimum when we use Uniform Crossover ? Explain your answer !  
**ANSWER:**Compare possible schemata, F(11\*) vs. F(00\*) and F(1\*\*) vs. F(0\*\*).  
F(11\*) = (4+0)/2 = 2 vs. F(00\*) = (2+1)/2 =1.5 . Do the same for the other schemata. The stronger schemata have always higher fitness than weaker. Thus, yes I expect it to converge to global optimum, since the given population seems large enough. According to the schema theorem, low order high performance schemata increase in proportion in population, when the population is large enough to survive disruption.

# Representation – Permutation Problems

1. Name three recombination operators specifically designed for permutation problems. Compare them with each other. What are their strengths and weaknesses?  
   **Answer**:  
   - Order crossover: select 2 random points. Copy subsequence from p1. Fill rest from p2 while skipping already included elements. Preserves relative order but ignores absolute order and adjacency.  
   - Cycle Crossover: Identify cycles between parents (positions with matching values). Copy all values for cycle positions from parent 1, fill the rest from parent 2. Favors absolute positioning over relative order or adjacency.  
   - Edge recombination: build an edge map from 2 parents. Juxtapose edges iteratively. Respects adjacency. Good for TSP.
2. What is Edge Recombination? How is it implemented? Illustrate with an example. Why is this operator better for the TSP (Traveling Salesman Problem) than other permutation crossover methods?  
   **ANSWER**:  
   Edge recombination is a crossover operator for permutation problems. There are 2 parent tours [ABCDEF] and [BDCAEF]. Build an edge map. 1-Choose initial city. 2-remove current from edge map. 3-If current has remaining edges go to 4, else go to 5. 4- choose current city edge with fewest remaining edges. 5-if still remaining cities, choose one with fewest remaining edges. See slide 18 and 19.  
   For TSP best, because in TSP adjacency of elements are important.
3. What is the fitness correlation coefficient ρOP? What can it be used for? How is ρOP calculated for mutation and crossover operators respectively?  
   **Answer:**  
   pOP is the measure of inheritance for a variation operator. It measure how much a child has similar characteristics with the parent after the operator is applied. It can be used to determine the performance of a variation operator. It is calculated as follows: generate n sets of k parents. Apply operator to create children. Compute fitness of all individuals. Compute covariance of parents and children.  
   pOP = cov(Fp, Fc)/Stdev(Fp)xStdev(Fc). Covariance is [-1,1]. If 0, means child has inherited nothing.

# Neural Network Redundancy (Representation)

1. Suppose you want to optimize neural networks using a genetic algorithm. You have decided to use 2-point crossover but are still uncertain between two possible genotype representations: G1 and G2. However, optimizing neural networks with genetic algorithms is very computationally intensive, and you cannot thoroughly test both representations and then choose the best one. How can you proceed to make this choice in a well-founded way without actually performing the optimization experiments? Provide a detailed description of your solution. (Hint: think about the principle of fitness correlation coefficients.)
2. We compared the redundant and non-redundant genetic representation of multi-layer perceptrons by computing the crossover correlation coefficient ρx. (a) How is ρx computed ? (b) For the single hidden layer neural network (2-15-1) the redundant representation had ρx = 0.456 while for the non-redundant ρx = 0.892. What can you conclude from these numbers ?  
   **ANSWER for both:**I can calculate px correlation coefficient for redundant and non-redundant networks. Generate random 2500 pairs for each group and recombine. Compute the correlation coefficient same as pOP with formula above. Higher px means parents transmits more information to children.

# Combinatorial Optimization Problems:

1. Explain how the Greedy Partitioning Crossover (GPX) is applied to graph coloring problems. Why is this crossover better suited for graph coloring than an assignment based crossover.   
   **ANSWER**:  
   The solutions are represented as partitions, sets of vertices. The partitions are disjoint sets. GPX eliminates the vertices, which have the most conflicting edges (eliminates the ones with lowest cost function value). Assignment crossover just swaps vertices randomly, which does not respect the important partial solutions. GPX applies partitioning based variation, that respects the building blocks.  
     
   **ANSWER 2 For Knapsack**:  
   GLS for Knapsack: 1-Parent pair selected at random. 2-Generate 1 offspring by uniform crossover + Local Search. 3- Replace the worst solution with child if child is better. No duplicates allowed. Why UX is good? It protects important part of the solution from disruption. Items with high profit/weight ratio are kept, while low profit/weight ratio are discarded. This turns blind knapsack into a quasi-non blind knapsack problem.

# Adaptive Pursuit

1. Consider the adaptive pursuit algorithm and the probability matching algorithm for allocating an operator to the current state of our search process. Assume we have 4 possible operators {a1, a2, a3, a4} to choose from. The current probability vector is P (t) = [0.1; 0.2; 0.3; 0.4] and the current reward estimate is Q(t) = [5; 10; 15; 20]. After applying the currently most likely operator we receive a reward of 40. Calculate for both algorithms the updated values P (t + 1) and Q(t + 1) when α = 0.5, β = 0.75, and Pmax = 0.85.

**ANSWER**: Write both formulas in cheat sheet, easy calculation.

# Model Building GAs

1. Both the COMIT and LTGA (linkage tree GA) algorithms are Model Building EAs that build a tree using mutual information. (a) What exactly do both trees compute/represent ? (b) How are both trees constructed ?  
   **ANSWER:**   
   They model important dependencies between problem variables. These dependencies are built upon mutual information between variables (or sets of variables). Both algorithms build hierarchical structures (trees) to represent the dependencies.   
   COMIT considers only bivariate (pairwise) dependencies. First it computes a fully connected weighted graph between problem variables. Weights are mutual information between 2 nodes. For computing this graph, it picks a random variable from solution and adds the variable with max. mutual information as children, this is done iteratively for each node. Then, it computes the maximum spanning tree (that is a tree that connects every node with fewest number of connections while maximizing the sum of weights).  
   LTGA starts from a univariate structure (where each node is a variable) and builds a linkage tree bottom-up while increasing the group size (e.g. bottom level is univariate, one level up is bivariate). Group the variables together at each level and this group will be the parent. LTGA learns a new tree for each solution and use the clusters (nodes) for crossover.

# Facet Models

1. How do you define the selection intensity of a selection algorithm? What is the selection intensity of roulette-wheel selection (i.e., proportionate selection), and tournament selection with a tournament size of 2? What can you deduce from these values?
2. How do you define the selection intensity of a selection algorithm? What is the selection intensity of roulette-wheel selection (i.e., proportionate selection), and tournament selection with tournament size 2? What can be deduced from these values?  
   🡪 Copy the formulas from slide to cheat sheet.   
   Proportionate selection: pressure drops near the end, as # of 1s increase in population. Slow convergence. Good diversity, because unfit individuals have also chance to reproduce. Contribution of individuals: proportional to their fitness.   
   Truncation selection: Fitness increase is proportional to Stdev. Pressure is constant based on r%. Convergence at a constant rate. Fast convergence because it goes for the best. Risk of premature convergence because of loss of diversity, can favor good individuals too much. As r increase, pressure decrease.  
   Tournament selection: Tournament size s. Selection intensity = expected value of the best ranked individual from a sample of s individuals. Calculated with order statistics: I(t) = Us:s. Adjustable pressure, as s increase, pressure increase. Good parallelization (for holding tournaments). Slower convergence than truncation, because truncation goes for the best. Con: average individuals can survive.
3. Selection intensity: • How is the Selection Intensity defined ? • We have shown that for proportionate selection I(t) = σ(t) f(t) . Explain how this can be used to derive the difference equation p(t + 1) − p(t) = 1 l (1 − p(t)) that models the population convergence for the counting ones problem.  
   🡪 Copy formulas to cheat sheet. Plug-in basic equations and derive from there.
4. Suppose we use a GA to optimize the number of 1-bits in a binary string. We use tournament selection and uniform crossover. How does the number of function evaluations, the population size, and the number of generations to convergence scale with increasing string length?
5. Assume we have a population of 1000 cows whose milk production is normally distributed with mean value 10 litres and standard deviation 2 litres. Calculate the expected mean value of the milk production of the 500 cows that are selected from the original group according to their milk production using the following selection algorithms (note: higher milk production is better):
6. standard proportionate selection
7. tournament selection with tournament size 2
8. truncation selection with threshold 50%  
   🡪 This is calculated using the I(t) = (mean(Ts) – mean(t)) / Stdev(t). What is asked: mean(Ts). For proportionate: I = st(t)/f(t). 2/10 = 0.2 🡪 0.2 = (f(Ts)-10, f(Ts))/2 = 10.4. For the rest selection methods, you need the table of constants 🡪 copy to cheat sheet.
9. In the lectures notes on population sizing it is stated that P [SelErr] = Φ(−1√2(ℓ−1)p(1−p) ). What does this mean? Explain all symbols used. -> **Find the rest**
10. In the population sizing lecture we computed the probability of making a selection decision error P[SelErr] for the counting ones problem. We have shown that:  
    A black square root of a square root of a square root

    AI-generated content may be incorrect.  
    with Φ the cumulative distribution function of the standard normal distribution. What do the symbols l and p stand for? Explain the reasoning that led to this formula, in other words, how did we obtain this result?  
    **ANSWER:**  
    This is related to schema fitness. L= string length, p=probability of having 1 at a location in schema. Consider 2 schemas, H1 good, H2 bad schema. Fitness of schemas are normally distributed with N(mean, variance) function. We make a selection error when we choose H2 over H1, meaning when P(F\_H1-H2) < 0, probability of choosing H2. P(F\_H1-H2) = CumulativeDist( (b-mean)/Stdev ), where b=0. Copy the formulas to cheat sheet, including the fitness approximations of H1 and H2 and H1-H2. Then you can derive this formula. If H1 normal dist and H2 normal dist, then H1-H2 is also normal dist,
11. A population consists of 10 individuals with fitness values: {1, 2, 2, 3, 4, 4, 5, 6, 6, 7, 8, 8, 9, 10, 10}. Using tournament selection with tournament size 2, we obtain the selected parent population. Calculate the expected number of copies of the individual with fitness 5 in the parent population. Is it in this chapter? This was in the intro chapter perhaps? Combine with the expected number of copies from Lecture 1 🡪 To cheat sheet!

# Evolutionary Multi Objective Optimization:

1. How can fitness be calculated in multi-objective evolutionary algorithms where the population needs to converge to the Pareto-optimal front?  
   **ANSWER:**Fitness is calculated based on Pareto Dominance.   
   Fitness-assignment: 1- Assign a rank to all solutions based on pareto dominance. Rank 0 = Not dominated solutions, Rank 1= dominated by 1 solution, etc. When all rank 0s are assigned and rank 0s removed, rank 1 becomes new rank 0, do this until all solutions are assigned a rank as fitness values.  
   Secondary sorting by diversity: You need to break ties when multiple solutions have same rank. If ranks are same, favor more diverse. Diversity based on calculating distance from the crowd. More distant to cluster -> more diverse is the solution.
2. Explain how selection can be performed in multi-objective optimization with evolutionary algorithms using the concept of domination ranks.  
   **ANSWER:**After ranking is complete, use one of 1- Truncation selection: select r of n domination ranks between solutions (e.g. dom rank 0 and 1 only). 2- Tournament selection: compare solutions, prefer lowest rank. If ranks are same, use secondary criterion.

# Evolutionary Strategies

1. Discuss the concept of self-adaptation in Evolution Strategies. How is mutation performed (including a mathematical description of correlated mutation)?
2. Discuss the concept of self-adaptation in Evolution Strategies. How is mutation performed (including a mathematical description of correlated mutation)?
3. How do (μ, λ) and (μ + λ) selection work? • How is an individual represented? • How is recombination performed?

# General:

1. There exist different keyboard layouts such as QWERTY, AZERTY, and DVORAK. You want to design your own layout using a genetic algorithm. How would you specify the problem representation, the mutation operator, and the crossover operator ? Explain how these operators work and why you think they would be the most suitable variation operators for the task.