COMP 5660/6660 Fall 2021 Final Exam - Canvas Quiz Key

This is a closed-book, closed-notes exam. The sum of the max points for all the questions is 144, but note that the max exam score will be capped at 136 (i.e., there are 8 bonus points but you can't score more than 100%). You have exactly two-and-a-half hours to complete this exam. Keep your answers clear and concise while complete. Good luck!

1. Alice is writing an EA to solve the binary knapsack constraint satisfaction problem. The sum of the item costs is 37 while the total cost is constrained to be below 36. Should she: [4 pts]

   (a) Ignore the constraints under the motto: all is well that ends well.

   (b) Upon generating an infeasible solution, immediately kill it and generate a new solution; repeat this step until a feasible solution is generated.

   (c) Employ a penalty function that reduces the fitness of infeasible solutions, preferably so that the fitness is reduced in proportion to the number of constraints violated, or to the distance from the feasible region.

   (d) Employ a repair function that takes infeasible solutions and "repairs" them by transforming them into a related feasible solution, typically as close as possible to the infeasible one.

   (e) Employ a closed feasible solution space which guarantees that the initial population consists of feasible solutions only and all evolutionary operations on feasible solutions are guaranteed to result in feasible solutions. Typically a combination of custom representation, initialization, recombination, and mutation is employed to achieve this.

   (f) Employ a decoder function that maps genotype space to phenotype space such that the phenotypes are guaranteed to be feasible even when the genotypes are infeasible. Typically this involves mapping multiple different genotypes to the same phenotype.

   Select one of:

   - a [0]
   - **b**
   - c [1]
   - d [3]
   - e [1]
   - f [2]
   - none of a, b, c, d, e, nor f [0]

2. In Crowding: [4 pts]

   (a) new individuals replace similar population members, resulting in the population sharing the niches equally

   (b) the fitness of individuals immediately prior to selection is adjusted according to the number of individuals falling within some prespecified distance of each other

   (c) individuals share the fitness of similar population members immediately prior to selection, resulting in the number of individuals per niche being dependent on the niche fitness

   Select one of:

   - **a**
   - b [1]
   - c [2]
   - a and b [2]
   - a and c [3]
   - b and c [1]

- all of a, b, and c [2]
- none of a, b, nor c [0]

3. In the context of multi-objective problem solving, the term scalarisation refers to combining single objective fitness scores into a weighted cumulative fitness score. This approach suffers from the following drawbacks: [4 pts]

   (a) scalarisation commonly is a computationally expensive process

   (b) the implicit assumption that all the user's preferences can be captured before the range of possible solutions is known

   (c) for repeatedly solving different instances of the same problem, either the user's preferences are assumed to be static, or the user needs to repeatedly provide new weightings

   Select one of:

   - a [0]
   - b [2]
   - c [2]
   - a and b [1]
   - a and c [1]
   - **b and c**
   - all of a, b, and c [3]
   - none of a, b, nor c [0]

4. In Evolution Strategies with uncorrelated mutation with n step sizes, the conceptual motivation for updating the mutation step sizes with the formula $\sigma'_i = \sigma_i \cdot e^{\tau' \cdot N(0,1) + \tau \cdot N_i(0,1)}$ is: [4 pts]

   (a) the sum of two normally distributed variables is also normally distributed

   (b) the common base mutation $e^{\tau' \cdot N(0,1)}$ allows for an overall change of the mutability, guaranteeing the preservation of all degrees of freedom

   (c) the coordinate-specific $e^{\tau \cdot N_i(0,1)}$ provides the flexibility to use different mutation strategies in different directions

   Select one of:

   - a [1]
   - b [1]
   - c [1]
   - a and b [3]
   - a and c [3]
   - b and c [3]
   - **all of a, b and c**
   - none of a, b, nor c [0]

5. Modern Evolutionary Programming (EP) differs from classic EP in: [4 pts]

   (a) representation

   (b) parent selection

   (c) parameter control

   Select one of:

   - a [2]

- b [0]
- c [2]
- a and b [1]
- **a and c**
- b and c [1]
- a, b, and c [3]
- none of a, b, nor c [0]

6. Koza's Automatically Defined Functions (ADFs) are: [4 pts]

   (a) the application of GP to automate the creation of functions in computer programs
   (b) the standard method of evolving reusable components in GP
   (c) the use of GP to create functions with a high AI ratio

   Select one of:

   - a [2]
   - **b**
   - c [1]
   - a and b [3]
   - a and c [1]
   - b and c [2]
   - a, b, and c [2]
   - none of a, b, nor c [0]

7. Learning Classifier Systems (LCS) have the following differences with classic expert systems in order to improve performance: [4 pts]

   (a) storing knowledge in the form of rules
   (b) improving accuracy through reinforcement learning
   (c) gaining new knowledge by generating rules employing evolutionary computation

   Select one of:

   - a [1]
   - b [2]
   - c [2]
   - a and b [2]
   - a and c [2]
   - **b and c**
   - a, b, and c [3]
   - none of a, b, nor c [0]

8. Increasing primitive granularity in hyper-heuristics is: [4 pts]

   (a) a terrible idea, because we want more sophisticated algorithms, not more primitive
   (b) a sensible idea, because it increases the search space, so the probability increases that a high quality solution is represented
   (c) a problematic idea, because it expands the search space, so the expected run time to find the global optimum increases

(d) a wonderful idea, because as the granularity approaches Turing completeness, the hyper-heuristic performance will converge on GP optimality

Select one of:

- a [0]
- b [2]
- c [2]
- d [0]
- a and b [1]
- a and c [1]
- a and d [0]
- **b and c**
- b and d [1]
- c and d [1]
- a, b, and c [3]
- a, b, and d [1]
- a, c, and d [1]
- b, c, and d [3]
- all of a, b, c, and d [2]
- none of a, b, c, nor d [0]

9. In the context of John Holland's Schema Theorem, which of the following statements are true: [4 pts]

(a) The Building Block Hypothesis is that a genetic algorithm seeks near-optimal performance through the juxtaposition of short, low-order, high-performance schemata, called the building blocks.

(b) Short, low-order, above-average schemata receive exponentially decreasing trials in subsequent generations of a genetic algorithm.

(c) The result that a population of size μ will usefully process $O(\mu^3)$ schemata is known as Implicit Parallelism.

Select one of:

- a [3]
- b [1]
- c [2]
- a and b [2]
- **a and c**
- b and c [2]
- all of a, b, and c [3]
- none of a, b, nor c [0]

10. In neuro-evolution – evolving artificial neural networks (ANNs) –: [4 pts]

(a) the high computational cost of evolution can be amortized over sufficient repeated uses of the resulting ANN

(b) the high computational cost of evolution is incurred a priori, with the resulting ANN run in production/real-time

(c) only the hyperparameters (e.g., learning rate, number of layers) can be evolved, not the weights (which the ANN has to learn) and topology/structure (which a human has to design)

(d) Lamarckian evolution is not supported because the genotype-to-phenotype mapping is per definition highly complex **that's true for generative neuro-evolution, but not in general**

Select one of:

- a [2]
- b [2]
- c [0]
- d [1]
- **a and b**
- a and c [1]
- a and d [2]
- b and c [1]
- b and d [2]
- c and d [0]
- a, b, and c [3]
- a, b, and d [3]
- a, c, and d [1]
- b, c, and d [1]
- all of a, b, c, and d [2]
- none of a, b, c, nor d [0]

11. Supportive Coevolution can be used to do which of the following: [4 pts]

(a) Automatically design EA operators

(b) Update EA parameters to more optimal settings during evolution

(c) Decrease the search space of the EA

(d) Repair invalid solutions resulting from offspring generation

Select one of:

- a [2]
- b [2]
- c [0]
- d [0]
- **a and b**
- a and c [1]
- a and d [1]
- b and c [1]
- b and d [1]
- c and d [0]
- a, b, and c [3]
- a, b, and d [3]
- a, c, and d [1]
- b, c, and d [1]
- all of a, b, c, and d [2]
- none of a, b, c, nor d [0]

12. Disengagement in a two-population competitive CoEA occurs when: [4 pts]

    (a) the individuals in both populations stop competing and start collaborating
    (b) both populations get stuck in local optimums leading to a loss of search gradient
    (c) one population gets stuck in a local optimum and the other population stops evolving because of a loss of evolutionary pressure

    Select one of:

    - a [0]
    - b [0]
    - c [2]
    - a and b [0]
    - a and c [1]
    - b and c [1]
    - all of a, b, and c [0]
    - **none of a, b, nor c**

13. Which of the following is an example of test-based competitive coevolution? [4 pts]

    (a) coevolving Pac-Man controllers and maps
    (b) coevolving Pac-Man controllers and ghost controllers **(example of interactive competitive coevolution)**
    (c) coevolving ghost controllers and maps **(example of cooperative coevolution)**

    Select one of:

    - **a**
    - b [2]
    - c [1]
    - a and b [3]
    - a and c [2]
    - b and c [1]
    - all of a, b, and c [2]
    - none of a, b, nor c [0]

14. In competitive coevolution opponent selection methods: [4 pts]

    (a) individuals under random opponent sampling only need one opponent, because it selects an unbiased and uniform random representative
    (b) comparing against only the highest-fitness opponents is effective because the highest-fitness genotypes generally also have the most diversity
    (c) shared sampling generally results in robust coevolution, because it constructs a phenotypically diverse set of opponents
    (d) comparing against similar-strength opponents needs a surrogate fitness function, because strong individuals will receive mediocre fitness against other strong opponents

    Select one of:

    - a [1]
    - b [0]
    - c [2]

- d [2]
- a and b [0]
- a and c [1]
- a and d [1]
- b and c [1]
- b and d [1]
- **c and d**
- a, b, and c [1]
- a, b, and d [1]
- a, c, and d [3]
- b, c, and d [2]
- all of a, b, c, and d [2]
- none of a, b, c, nor d [0]

15. A CIAO plot which at some point becomes uniformly dark is indicative of: [4 pts]

    (a) mediocre stability
    (b) cycling
    (c) disengagement

    Select one of:

    - a [2]
    - b [0]
    - **c**
    - a and b [1]
    - a and c [3]
    - b and c [2]
    - all of a, b, and c [2]
    - none of a, b, nor c [0]

16. Is the genotypic encoding of the Pac-Man controllers in this semester's Assignment 2 Series: [4 pts]

    (a) pleitropic
    (b) polygenetic
    (c) altruistic

    Select one of:

    - a [2]
    - b [2]
    - c [0]
    - **a and b**
    - a and c [1]
    - b and c [1]
    - a, b, and c [3]
    - none of a, b, nor c [0]

17. If we employ self-adaptation to control the value of penalty coefficients for an EA with an evaluation function which includes a penalty function, then: [4 pts]

    (a) the penalty coefficients will be self-adapted to cause fitness improvement just like, for instance, mutation step sizes

    (b) this cannot be done because it is inherently impossible to self-adapt any part of the evaluation function

    (c) the penalty coefficients will be self-adapted, but the increase in fitness achieved may not be correlated with better performance on the objective function

    Select one of:

    - a [2]
    - b [1]
    - **c**
    - none of a, b, nor c [0]

18. While in a standard EA an offspring is generated by recombination followed by mutation, in GP one usually generates an offspring either by recombination or by mutating a clone of a parent, not both. This is because: [4 pts]

    (a) the combination of recombination and mutation frequently creates too much stochastic noise, effectively resulting in random search; GP is a relatively new type of EA which allowed its creators to correct this problem by designing it from the start to do either recombination or mutation, but not both at the same time

    (b) recombination and mutation are often quite destructive in GP and doing both would effectively result in random search

    (c) performing both recombination and mutation would violate the closure property of GP

    Select one of:

    - a [2]
    - **b**
    - c [0]
    - a and b [3]
    - a and c [1]
    - b and c [2]
    - a, b, and c [3]
    - none of a, b, nor c [0]

19. To battle the "human bottleneck" in interactive evolution, one can employ: [4 pts]

    (a) surrogate fitness functions

    (b) small population sizes

    (c) multi-objective EAs for problems that exhibit a mixture of quantitative and qualitative aspects

    (d) crowdsourcing

    Select one of:

    - a [1]
    - b [1]
    - c [1]
    - d [1]

- a and b [2]
- a and c [2]
- a and d [2]
- b and c [2]
- b and d [2]
- c and d [2]
- a, b, and c [3]
- a, b, and d [3]
- a, c, and d [3]
- b, c, and d [3]
- **a, b, c, and d**
- none of a, b, c, nor d [0]

Open Questions

20. What is the binary gray code for the standard binary number 11100010100? [3 pts] *10010011110*

21. What is the standard binary number encoded by the binary gray code 110110011101? [3 pts] *100100010110*

22. Is the genotype-phenotype decoding function for this semester's Assignment 2 Series Pac-Man controllers surjective, injective, both, or neither? Explain your answer! [6 pts] **It is surjective but not injective, because all control strategies (i.e., mappings from sensory input to controller actions) are valid genotypes (surjective), but there exist control strategies than can be encoded by multiple distinct genotypes, for instance by swapping two constant terminals being fed into a summation function (not injective).**

23. Does the closure property in GP hold for the Assignment Series 2 Pac-Man controllers? Explain your answer! [4 pts] **Yes, because all the terminals input floating point numbers and all the functions consist of mathematical functions which take as input and produce as output floating point numbers.**

24. Given the following two parents with permutation representation:
$p1 = (435792168)$
$p2 = (623571489)$
Compute the first offspring with Order Crossover, using crossover points between the 2nd and 3rd loci and between the 6th and 7th loci. [6 pts]

    (a) *Child 1:* $\cdot\cdot 5792 \cdot\cdot\cdot$
    (b) *Child 1:* ***315792486***

25. Given the following two parents with permutation representation:
$p1 = (435792168)$
$p2 = (623571489)$
Compute the first offspring with Cycle Crossover. Show first the cycles you've identified and then the construction of the offspring. [6 pts]

    Cycle 1: 4-6-8-9-7-5-3-2-1
    Construction of first offspring by scanning parents from left to right, starting at parent 1 and alternating parents:

    (a) *Add cycle 1 from parent 1:* ***435792168***

26. Given the following two parents with permutation representation:
$p1 = (435792168)$
$p2 = (623571489)$
Compute the first offspring with PMX, using crossover points between the 3rd and 4thd loci and between the 7th and 8th loci. [10 pts]

(a) $\cdots$*7921*$\cdot\cdot$

(b) $\cdots$*7921*$\cdot$5

(c) $\cdot$4$\cdot$*7921*$\cdot$5

(d) ***643792185***

27. Given the following two parents with permutation representation:
$p1 = (435792168)$
$p2 = (623571489)$
Compute the first offspring with Edge Crossover, except that for each random choice you instead select the lowest element. [14 pts]

Original Edge Table:

Element	Edges	Element	Edges
1	2,6,7,4	6	1,8,9,2
2	9,1,6,3	7	5+,9,1
3	4,5+,2	8	6,4+,9
4	8+,3,1	9	7,2,8,6
5	3+,7+		

Construction Table:

Element selected	Reason	Partial result
1	Lowest	1
4	Tied shortest list, so lowest	1,4
8	Common edge	1,4,8
6	Shortest list	1,4,8,6
2	Equal list sizes, so lowest	1,4,8,6,2
3	Equal list sizes, so lowest	1,4,8,6,2,3
5	Only element	1,4,8,6,2,3,5
7	Only element	1,4,8,6,2,3,5,7
9	Last element	**148623579**

Edge Table After Step 1:

Element	Edges	Element	Edges
1	2,6,7,4	6	8,9,2
2	9,6,3	7	5+,9
3	4,5+,2	8	6,4+,9
4	8+,3	9	7,2,8,6
5	3+,7+		

Edge Table After Step 2:

Element	Edges	Element	Edges
		6	8,9,2
2	9,6,3	7	5+,9
3	5+,2	8	6,9
4	8+,3	9	7,2,8,6
5	3+,7+		

Edge Table After Step 3:

Element	Edges	Element	Edges
		6	9,2
2	9,6,3	7	5+,9
3	5+,2	8	6,9
		9	7,2,6
5	3+,7+		

Edge Table After Step 4:

Element	Edges	Element	Edges
		6	9,2
2	9,3	7	5+,9
3	5+,2		
		9	7,2
5	3+,7+		

Edge Table After Step 5:

Element	Edges	Element	Edges
2	9,3	7	5+,9
3	5+		
		9	7
5	3+,7+		

Edge Table After Step 6:

Element	Edges	Element	Edges
		7	5+,9
3	5+		
		9	7
5	7+		

Edge Table After Step 7:

Element	Edges	Element	Edges
		7	9
3			
		9	7
5	7+		

28. Given the following bit strings v_1 through v_5 and schema S
$v_1 = (01101110101001)$ $fitness(v_1) = 0.8$
$v_2 = (10110010011001)$ $fitness(v_2) = 0.1$
$v_3 = (00001010011010)$ $fitness(v_3) = 1.0$
$v_4 = (01001110111001)$ $fitness(v_4) = 1.2$
$v_5 = (01001011100011)$ $fitness(v_5) = 1.9$
$S = (01 * *11101 * 100*)$

    (a) Compute the *order* of S. [1 pt] 10
    (b) Compute the *defining length* of S. [1 pt] 13-1=12
    (c) Compute the fitness of S. [1 pt] $\frac{0.8+1.2}{2} = 1.0$
    (d) Do you expect the number of strings matching S to increase or decrease in subsequent generations? Explain your answer! [4 pts] *Average population fitness:* $\frac{0.8+0.1+1.0+1.2+1.9}{5} = 1.0$
    Decrease, because the fitness of S is equal to the average population fitness and S has a high-order and defining length so large destruction chance.

29. Say you want to purchase a new house and care most about maximizing space and affordability. You collect square footage data and pricing on ten different houses and then you normalize both the square footage data and the pricing which results in the following table, where higher space numbers indicate greater square footage and higher affordability numbers indicate lower prices:

ID	Space	Affordability
1	5	7
2	2	6
3	3	2
4	4	10
5	7	8
6	5	6
7	3	4
8	10	2
9	1	1
10	6	1

    (a) List for each element which elements it dominates; indicate elements with their IDs. [3]

ID	Dominates
1	2,3,6,7,9
2	9
3	9
4	2,3,7,9
5	1,2,3,6,7,9,10
6	2,3,7,9
7	3,9
8	3,9,10
9	None
10	9

    (b) Show the population distributed over non-dominated levels, like some multi-objective EAs employ, after each addition of an element, starting with element 1 and ending with element 10 increasing the element number one at a time; indicate elements with their IDs. So you need to show ten different population distributions, the first one consisting of a single element, and the last one consisting of ten elements. [6]

After adding element 1:

- **Level 1:** 1

After adding element 2:

- **Level 1:** 1
- **Level 2:** 2

After adding element 3:

- **Level 1:** 1
- **Level 2:** 2,3

After adding element 4:

- **Level 1:** 1,4
- **Level 2:** 2,3

After adding element 5:

- **Level 1:** 4,5
- **Level 2:** 1
- **Level 3:** 2,3

After adding element 6:

- **Level 1:** 4,5
- **Level 2:** 1
- **Level 3:** 6
- **Level 4:** 2,3

After adding element 7:

- **Level 1:** 4,5
- **Level 2:** 1
- **Level 3:** 6
- **Level 4:** 2,7
- **Level 5:** 3

After adding element 8:

- **Level 1:** 4,5,8
- **Level 2:** 1
- **Level 3:** 6
- **Level 4:** 2,7
- **Level 5:** 3

After adding element 9:

- **Level 1:** 4,5,8
- **Level 2:** 1
- **Level 3:** 6
- **Level 4:** 2,7
- **Level 5:** 3
- **Level 6:** 9

After adding element 10:

- **Level 1:** 4,5,8
- **Level 2:** 1,10
- **Level 3:** 6
- **Level 4:** 2,7
- **Level 5:** 3
- **Level 6:** 9