CS5401 FS2018 Exam 2 Key

This is a closed-book, closed-notes exam. The only items you are allowed to use are writing implements. Write your name in the designated spot on the top left of each of the exam pages. If you are caught cheating, you will receive a zero grade for this exam. The max number of points per question is indicated in square brackets after each question. The sum of the max points for all the questions is 42, but note that the max exam score will be capped at 40 (i.e., there are 2 bonus points but you can't score more than 100%). Note that this exam consists of 16 multiple-choice questions followed by a single multi-part open question. You have exactly 75 minutes to complete this exam. Good luck!

Multiple Choice Questions - circle the letter of your choice on your exam

1. One advantage of implementing survivor selection by employing a so-called reverse k-tournament selection to select who dies is that: [2]
   (a) you guarantee (k)-elitism [1]
   (b) **you guarantee $(k-1)$-elitism**
   (c) the probability of surviving is proportional to your fitness rank [0]
   (d) you guarantee 1-elitism (i.e., the fittest individual is guaranteed to survive) [$\frac{1}{2}$]
   (e) none of the above [0]

2. Genetic drift and natural selection: [2]
   (a) are different terms for the same concept [0] *(false because they are different concepts)*
   (b) are different non-related concepts [$\frac{1}{2}$] *(false because while different, they are certainly related)*
   (c) complement each other because natural selection without genetic drift would select based on phenotypes without regard for genotypes [1] *(bad answer because natural selection selects based on phenotype, whether or not there is genetic drift)*
   (d) complement each other because genetic drift without natural selection would result in random search

3. 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: [2]
   (a) the sum of two normally distributed variables is also normally distributed [$\frac{1}{2}$]
   (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 [1]
   (c) the coordinate-specific $e^{\tau \cdot N_i(0,1)}$ provides the flexibility to use different mutation strategies in different directions [1]
   (d) **all of the above**
   (e) none of the above [0]

4. Rechenberg's 1/5 success rule: [2]
   (a) refers to the minimum succesful mutation rate threshold necessary for an Evolution Strategy to reach the global optimum [1]
   (b) refers to the ratio of offspring created by mutations versus recombination in Genetic Programming [0]
   (c) **refers to a rule of thumb for the optimal ratio of successful versus total mutations in Evolution Strategies where mutation step size is increased if the ratio is greater than $\frac{1}{5}$ and decreased if the ratio is smaller than $\frac{1}{5}$**
   (d) refers to the minimum ratio of succesful offspring creation versus total offspring creation in order for a parent to survive to the next generation [$\frac{1}{2}$]
   (e) none of the above [0]

5. There is no recombination in "standard" Evolutionary Programming (EP) because: [2]
   (a) extensive research has shown that the use of recombination is counterproductive in EP [0]
   (b) EP was conceived before the invention of recombination [0]
   (c) **each individual in "standard" EP is viewed as the abstraction of a species**
   (d) all of the above [0]
   (e) none of the above [0]

6. 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: [2]
   (a) **recombination and mutation are often quite destructive in GP and doing both would effectively result in random search**
   (b) 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 [1]
   (c) performing both recombination and mutation would violate the closure property of GP [0]
   (d) all of the above [0]
   (e) none of the above [0]

7. The ramped half-and-half method is the most common technique in GP for: [2]
   (a) **initialization**
   (b) parent selection [0]
   (c) survival selection [0]
   (d) termination [0]
   (e) none of the above [0]

8. Does the closure property in GP hold for the following function & terminal set, as might be employed in Pac-Man versus the Ghosts, if each function accepts two inputs and produces one output, and the terminals consist of floating point numbers representing sensor inputs and constants: [2]

Function set	addition, subtraction, multiplication, protected division, rand(a,b)
Terminal set	$\mathbf{R}$

   (a) No, because the arity of the functions in the function set are not all equal. [0]
   (b) No, because the functions in the function set cannot accept all the terminal types present in the terminal set. [0]
   (c) Yes, because the functions in the function set have equal arity. [$\frac{1}{2}$]
   (d) No, because there are more functions in the function set than terminals in the terminal set, making it impossible to guarantee closure for each and every terminal. [0]
   (e) **None of the above.**

9. The phenomenon of bloat in GP occurs most likely because: [2]
   (a) individuals with bigger genomes have a larger chance of survival (also known as "survival of the fattest") [$\frac{1}{2}$]
   (b) **the variable length aspect of GP causes a natural tendency for the population to reflect the different possible sizes**
   (c) the ratio of alleles to genes in bloated individuals is higher than non-bloated individuals which gives them an evolutionary advantage [0]
   (d) all of the above [$\frac{1}{2}$]
   (e) none of the above [0]
10. Koza states that the aim of the fields of artificial intelligence and machine learning is to generate human-competitive results with a high artificial-to-intelligence (AI) ratio where the AI ratio of a problem-solving method means: [2]
    (a) the ratio of automation (generality) to human intelligence (speciality) needed by the problem-solving method to solve a particular problem [1]
    (b) **the ratio of that which is delivered by the automated operation of the problem-solving method to the amount of intelligence that is supplied by the human applying the method to a particular problem**
    (c) the ratio of artificial intelligence to human intelligence employed by the problem-solving method [1]
    (d) none of the above [0]
11. Koza's Automatically Defined Functions (ADFs) are: [2]
    (a) the application of GP to automate the creation of functions in computer programs [1]
    (b) **the standard method of evolving reusable components in GP**
    (c) the use of GP to create functions with a high AI ratio [$\frac{1}{2}$]
    (d) none of the above [0]
12. Over-selection is employed in GP because: [2]
    (a) GP typically uses large trees which suffer from bloat [0]
    (b) GP typically uses fitness proportionate selection which suffers from premature convergence [$\frac{1}{2}$]
    (c) GP typically uses large populations which cause excessively high selective pressure [$\frac{1}{2}$]
    (d) all of the above [0]
    (e) **none of the above**
13. 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: [2]
    (a) the penalty coefficients will be self-adapted to cause fitness improvement just like, for instance, mutation step sizes [1]
    (b) this cannot be done because it is inherently impossible to self-adapt any part of the evaluation function [$\frac{1}{2}$]
    (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**
    (d) none of the above [0]

14. The challenge when employing parameter control in order to reduce the number of EA strategy parameters which the practitioner has to configure is: [2]

    (a) the introduction of "stealth" parameters, namely new parameters to control the parameter control, which may be as hard or harder to tune than the parameter(s) eliminated by the employment of the parameter control [1]

    (b) the introduction of "stealth" parameters, namely new parameters to control the parameter control, which cause a dynamic derived variable associated with the eliminated EA strategy parameter to converge to a fixed value deterministically specified by the stealth parameters [1]

    (c) the interaction between the parameter control of different operators, such as population sizing and offspring sizing, may be complex and hard to tune [$\frac{1}{2}$]

    (d) **all of the above**

    (e) none of the above [0]

15. In Multi-Objective problems a solution x is said to be dominated by a solution y when: [2]

    (a) solution x is no better than y in all objectives [1]

    (b) solution x is strictly worse than y in no more than one objective [$\frac{1}{2}$]

    (c) only if both the above are true [1]

    (d) **none of the above**

16. In Multi-Objective EAs employing levels of non-domination, a decrease in the number of levels, generally will: [2]

    (a) not impact the amount of selective pressure [0]

    (b) increase the amount of selective pressure [0]

    (c) **decrease the amount of selective pressure**

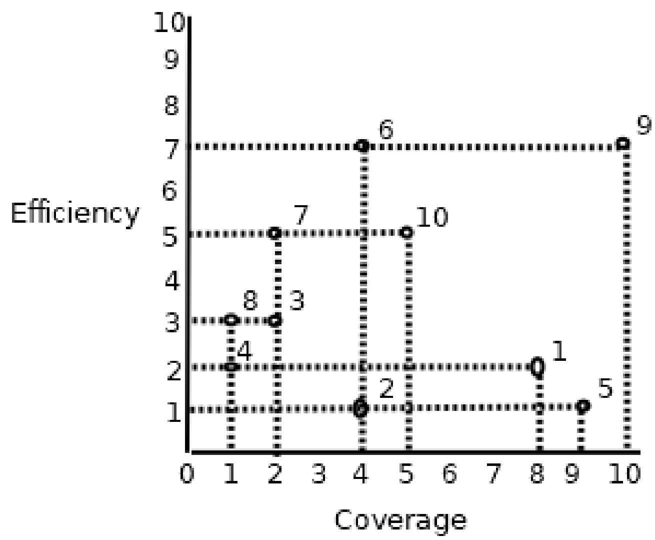
    (d) either increase or decrease the amount of selective pressure, depending on the number of conflicting objectives [0]

Regular Questions - write your answer under the question on the exam page

17. Say for the Light Up Puzzle, you want to simultaneously maximize cells lit up while minimizing number of bulbs placed. You execute a multi-objective EA and the final population contains the solutions listed in the following table, where higher coverage indicates more cells lit up, and higher efficiency indicates lower number of bulbs placed:

ID	Coverage	Efficiency
1	8	2
2	4	1
3	2	3
4	1	2
5	9	1
6	4	7
7	2	5
8	1	3
9	10	7
10	5	5

(a) Plot the above table and use dotted lines to indicate the area of domination for each element. [2]



(b) List for each element which elements it dominates; indicate elements with their IDs. [2]

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

(c) 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:** 1,3
- **Level 2:** 2

After adding element 4:
- **Level 1:** 1,3
- **Level 2:** 2,4

After adding element 5:
- **Level 1:** 1,3,5
- **Level 2:** 2,4

After adding element 6:
- **Level 1:** 1,5,6
- **Level 2:** 2,3
- **Level 3:** 4

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

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

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

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