CS5401 FS2016 Final Exam Key

This is a closed-book, closed-notes exam. The only items you are allowed to use are writing implements. Mark each sheet of paper you use with your name and the string "cs5401 fs2016 final exam". 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 77, but note that the max exam score will be capped at 75 (i.e., there are 2 bonus points but you can't score more than 100%). You have exactly two hours to complete this exam. Keep your answers clear and concise while complete. Good luck!

Multiple Choice Questions - write the letter of your choice on your answer paper

1. Is the genotypic encoding for the Assignment 2 Series of Pac-Man vs. the Ghosts: [2]
   (a) pleitropic but not polygenetic [1]
   (b) polygenetic but not pleitropic [1]
   (c) **pleitropic and polygenic**
   (d) none of the above [0]
2. Is the genotype-phenotype decoding function for the Assignment 2 Series of Pac-Man vs. the Ghosts: [2]
   (a) **surjective but not injective**
   (b) injective but not surjective [0]
   (c) bijective [1]
   (d) none of the above [0]
3. Is the phenotype to fitness mapping for the Assignment 2 Series of Pac-Man vs. the Ghosts: [2]
   (a) **surjective but not injective**
   (b) injective but not surjective [0]
   (c) bijective [1]
   (d) none of the above [0]
4. The current GP practice of strongly limiting the role of mutation in favor of recombination is because: [2]
   (a) recombination tends to increase genetic diversity in GP, unlike mutation which contrary to in standard EAs which employ a linear representation, has a tendency to destroy critical alleles [0]
   (b) **the generally shared view that in GP, crossover has a large shuffling effect, acting in some sense as a macromutation operator**
   (c) mutation tends to cause excessive bloat in GP, unlike recombination which has a natural parsimony pressure effect [0]
   (d) all of the above [$\frac{1}{2}$]
   (e) none of the above [0]
5. Countermeasures to bloat in GP include: [2]
   (a) increasing mutation rate to maintain genetic diversity [0]
   (b) **increasing parsimony pressure to penalize the fitness of large chromosomes**
   (c) reducing the number of alleles to prevent disproportional tree growth [0]
   (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) 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]

   (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 [0]

   (d) all of the above [0]

   (e) none of the above [0]

7. 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.**

8. The Pitt and Michigan approaches in Learning Classifier Systems differ in that: [2]

   (a) in the Pitt approach each individual has the option of either representing a single rule or a rule set, while in the Michigan approach each individual represents a single rule and the entire population represents the complete rule set [1]

   (b) in the Pitt approach each individual represents a single rule and the entire population represents the complete rule set, while in the Michigan approach each individual has the option of either representing a single rule or a rule set [$\frac{1}{2}$]

   (c) **in the Pitt approach each individual represents a complete rule set, while in the Michigan approach each individual represents a single rule and the entire population represents the complete rule set**

   (d) in the Pitt approach each individual represents a single rule and the entire population represents the complete rule set, while in the Michigan approach each individual represents a complete rule set [1]

   (e) in the Pitt approach each individual represents a complete rule set, while in the Michigan approach each individual has the option of either representing a single rule or a rule set [1]

   (f) none of the above [0]

9. Panmictic mate selection in EAs has the following properties: [2]

   (a) strategy parameters are fixed during an EA run [0]

   (b) **no genotypic restrictions on mating**

   (c) more fit individuals mate more often [0]

   (d) process of tuning mate selection parameters for each problem is time-consuming [0]

   (e) all of the above [$\frac{1}{2}$]

   (f) none of the above [0]

10. In Crowding: [2]

    (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 [$\frac{1}{2}$]
    (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 [1]
    (d) none of the above [0]

11. In Multi-Objective EAs employing levels of non-domination, increasing the number of conflicting objectives, generally will: [2]

    (a) not impact the number of levels of non-domination [0]
    (b) increase the number of levels of non-domination [0]
    (c) **decrease the number of levels of non-domination**
    (d) either increase or decrease the number of levels of non-domination, depending on the amount of selective pressure [0]

12. "Intelligent" initialization in a memetic algorithm can be performed by: [2]

    (a) Seeding [$\frac{1}{2}$]
    (b) Selective Initialization [$\frac{1}{2}$]
    (c) Locally Optimized Random Initialization [$\frac{1}{2}$]
    (d) Mass Mutation [$\frac{1}{2}$]
    (e) **all of the above**
    (f) none of the above [0]

13. The exacerbation of premature convergence in memetic algorithms is due to: [2]

    (a) limited seeding [$\frac{1}{2}$]
    (b) diversity preserving recombination operators [0]
    (c) non-duplicating selection operators [0]
    (d) Boltzmann selection [0]
    (e) all of the above [0]
    (f) **none of the above**

14. Is it necessary in competitive coevolution to recompute each individual's fitness every generation? [2]

    (a) **Yes, because in competitive coevolution, fitness is dependent on other individuals, so it can change every generation.**
    (b) Yes, because in competitive coevolution, the fitness of an individual in one population is dependent on the fitness of all the individuals in the competing population, so it can change every generation. [$1\frac{1}{2}$]
    (c) No, because in competitive coevolutionn, the fitness of an individual only needs to be recomputed when one or more of the opponents it was previously sampled against are eliminated. [$\frac{1}{2}$]
    (d) No, because in all evolutionary algorithms, including coevolution, fitness is an absolute measure of the quality of the solution encoded in an individual. [0]
    (e) None of the above [0]

15. One countermeasure to cycling in competitive coevolution is maintaining a hall of fame which: [2]
    (a) stores the most famous warriors of previous generations in order to intimidate current individuals in the opposing population [$\frac{1}{2}$]
    (b) stores the best solutions found in previous generations to guarantee that the global best solution is preserved and not forgotten over and over again [1]
    (c) **consists of the best individuals of previous generations against who current individuals are competed to prevent later populations from "forgetting" about the winning traits of earlier generations**
    (d) all of the above [$\frac{1}{2}$]
    (e) none of the above [0]

16. A cooperative CoEA is a CoEA where: [2]
    (a) each population tries to solve its own problem without harming the fitness of any of the other populations [1]
    (b) the populations are symbiotic species [1]
    (c) each population is a different species representing part of a larger problem [$1\frac{1}{2}$]
    (d) **all of the above**
    (e) none of the above [0]

17. A CIAO plot containing well-defined diagonal bands every five generations is indicative of: [2]
    (a) mediocre stability [0]
    (b) **cycling**
    (c) disengagement [0]
    (d) answers a and b [1]
    (e) answers a and c [0]
    (f) answers b and c [1]
    (g) answers a, b, and c [$\frac{1}{2}$]
    (h) none of the above [0]

18. On a computer system with 200 computing cores and given a population size of 100 and an offspring size of 500, employing an Asynchronous Parallel EA (APEA) for evolving GP controllers for Pac-Man: [2]
    (a) may be expected to reduce run-time versus a Synchronous Parallel EA (SPEA) because a SPEA cannot utilize more cores than the offspring size while an APEA can [1] *(while the reason given is true, it doesn't apply here because $\lambda = 500 \geq 200 = \#$ computing cores)*
    (b) may be expected to increase run-time versus a SPEA because an APEA cannot utilize more cores than the population size while a SPEA can [0]
    (c) **may be expected to reduce run-time versus a SPEA because a SPEA has to wait for the longest evaluation to complete while an APEA can exploit the heterogeneous evaluation times common to GP**
    (d) answers a and b [$\frac{1}{2}$]
    (e) answers b and c [1]
    (f) answers a and c [$1\frac{1}{2}$]
    (g) answers a, b, and c [1]
    (h) none of the above [0]

Regular Questions

19. (a) What is the binary gray code for the standard binary number 10111110? [2]

    11100001

    (b) What is the standard binary number encoded by the binary gray code 1000001? [2]

    1111110

20. Given the following two parents with permutation representation:
$p1 = (435792168)$
$p2 = (623571489)$

    (a) Compute the first offspring with Cycle Crossover. [4]

    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:

    i. *Add cycle 1 from parent 1:* ***435792168***

    (b) Compute the first offspring with PMX, using crossover points between the 3rd and 4thd loci and between the 7th and 8th loci. [5]

    i. · · ·*7921*··
    ii. · · ·*7921*·5
    iii. ·4·*7921*·5
    iv. ***643792185***

    (c) Compute the first offspring with Order Crossover, using crossover points between the 2nd and 3rd loci and between the 6th and 7th loci. [3]

    i. *Child 1:* · · 5792 · ··
    ii. *Child 1:* ***315792486***

    (d) Compute the first offspring with Edge Crossover, except that for each random choice you instead select the lowest element. [7]

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+		

21. Given the following bit strings v_1 through v_5 and schema S
$v_1 = (11101110111101)\ fitness(v_1) = 0.3$
$v_2 = (10110010001101)\ fitness(v_2) = 0.1$
$v_3 = (00001010011010)\ fitness(v_3) = 1.0$
$v_4 = (01001110111001)\ fitness(v_4) = 1.9$
$v_5 = (11001011110101)\ fitness(v_5) = 1.7$
$S = (00000001111111)$

(a) Compute the *order* of S. [1]
14

(b) Compute the *defining length* of S and show your computation. [1]
14-1=13

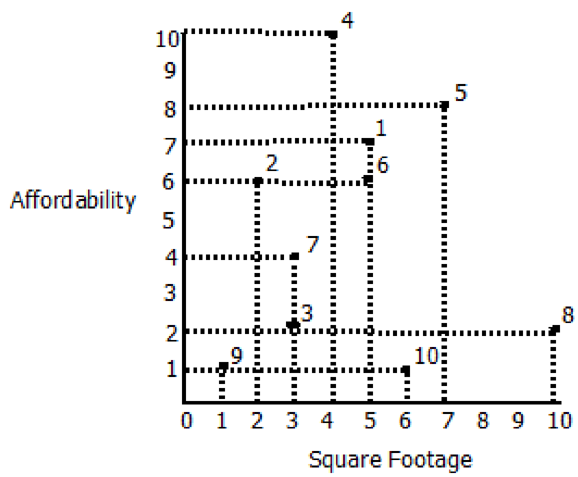
(c) Compute the fitness of S and justify your answer. [2]
Undefined because S doesn't match any of the given strings.

(d) Do you expect the number of strings matching S to increase or decrease in subsequent generations? Explain your answer! [4]
Because S currently doesn't match any strings and eventually may match strings after sufficient recombination and mutation has taken place, the number of strings matching S is expected to eventually increase.

22. 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) Plot the above table using dotted lines to indicate the area of domination for each element, with space on the horizontal axis and affordability on the vertical axis. [2]



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

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

(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
- **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