

COMP 5660/6660 Fall 2026 Exam 1 Key

This is a closed-book, closed-notes exam. The sum of the max points for all the questions is 76, but note that the max exam score will be capped at 70 (i.e., there are 6 bonus points, but you can't score more than 100%). You have exactly 50 minutes to complete this exam. Good luck!

1. Alice and Bob run their EAs on a target benchmark. Bob employs a two-tailed Welch's t-test with $\alpha = 0.05$ and sample size $n = 30$ in order to statistically analyze the difference between their EAs. Bob's EA has a higher average performance than Alice's EA. The test returns a p of 0.09. Bob repeats the experiment, this time with $n = 100$, obtaining a p of 0.04. What should Bob do now? [4 pts]
 - (a) As the difference between his p of 0.04 and the α of 0.05 is within the margin of error, he should increase n just a bit and rerun the experiment to see if he can decrease p to a statistically significant lower value than α , at which point he can claim that his EA beats Alice's.
 - (b) He should increase α to be sufficiently higher than p to be statistically meaningful in order to support the claim that his EA beats Alice's.
 - (c) He should settle for his EA not beating Alice's, because making n or α arbitrarily high in order to demonstrate a statistical difference is not scientifically meaningful.
 - (d) This analysis is flawed because Bob knows that his EA has a higher average than Alice's, so he should be using a one-tailed t-test rather than a two-tailed, which would increase the sensitivity of the test.

Select one of:

- a [1]; the margin of error argument is invalid, any p value smaller than α is sufficient to statistically determine that Bob's EA beat Alice's.
 - b [1]; this is an invalid argument because Bob's already statistically determined his EA beats Alice's; also, increasing α after reviewing the data invalidates the error control.
 - c [2]; while it's true that arbitrarily increasing n or α will lead to scientifically meaningless results, Bob's EA already beats Alice's so this option is not applicable.
 - d [0]; first of all, he should not use a one-tailed t-test when comparing two algorithms, either of which could be better; secondly, he's already beaten Alice's algorithm.
 - **None of the above, because Bob has already statistically demonstrated that his EA beats Alice's and therefore doesn't need to do anything more.**
2. A (1+1) EA is equivalent to: [4 pts]
 - (a) random search
 - (b) hill climber
 - (c) simulated annealing
 - (d) stochastic universal sampling

Select one of:

- a [0]
- **b**
- c [1]
- d [0]
- None of the above [0]

The next 5 questions are about the following EA experimental log data from Assignment 1b, showing best fitness per generation per run on the Cutting Stock Problem:

Eval	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6
1,000	30	-50	-50	31	-50	-50
2,000	37	-50	-50	49	-50	-50
3,000	98	-50	-50	104	-50	-50
4,000	123	-50	24	106	-50	30
5,000	157	-50	24	123	-50	47
6,000	183	-50	92	130	-50	65
7,000	222	84	121	141	-50	96
8,000	240	84	140	164	-50	118
9,000	248	85	184	176	-50	120
10,000	253	156	200	203	-50	131

3. What is μ ? [1 pt]

1,000

4. What is λ ? [1 pt]

1,000

5. Given that the EA is using the limit of 10,000 fitness evaluations as its termination criterion, write out the formula for computing the number of generations (i.e., generational cycles) completed by the EA each run, in terms of μ , λ , and `evaluation_limit`, and your computation of that number of generations. Note that the randomly initialized population is not counted as a generation. [3 pts]

$$\text{number_of_generations} = \frac{(\text{evaluation_limit} - \mu)}{\lambda} = \frac{(10,000 - 1,000)}{1,000} = 9$$

6. Which of the following is indicated by the log data? [3 pts]

- (a) The EA must be employing elitist survival selection.
- (b) The EA must be employing non-elitist survival selection.
- (c) There is insufficient information to determine whether the EA is employing elitist survival selection.

Select one of:

- a [1]
- b [0]
- c, because there is no evidence of maximum fitness values decreasing, but this may be due to chance.

7. Which of the following is indicated by the log data? [4 pts]

- (a) μ , λ , and the number of generations are well-configured and the EA is sufficiently tuned.
- (b) μ should be increased to raise the chance of valid solutions existing in the initial population.
- (c) λ should be increased to raise the selective pressure on the EA, as fitness is stagnant in some runs.
- (d) The number of generations should be increased, as the EA terminates before it reaches convergence.

Select one of:

- a [0]
- b [2]
- c [0]
- d [2]
- b and c [1]
- **b and d**; c is wrong, as raising λ will not increase selective pressure, and the “stagnant” fitness is mainly due to invalid solutions.
- c and d [1]
- b, c, and d [1]

The next two questions are about the following scenario:

An experimenter has noticed that their EA is failing to produce good solutions for a problem, and notes that *fitness was still steadily increasing when it reached the evaluation limit*. The EA is currently configured with k -tournament parent selection and k -tournament survival selection ($k = 10$ for both), $\mu = 10,000$, $\lambda = 2,000$, and a mutation rate per child of 10%, running for 1,000,000 evaluations. For technical reasons, they can not increase the evaluation limit.

8. Which changes to selection could improve the EA’s results? [4 pts]

- (a) Decrease k to 2
- (b) Increase k to 50
- (c) Switch to uniform random parent selection
- (d) Switch to truncation survival selection

Select one of:

- a [0]
- b [2]
- c [0]
- d [2]
- a and b [0]
- a and c [0]
- a and d [2]
- b and c [2]
- **b and d**; both increase selection pressure
- c and d [2]
- a, b, and c [0]
- a, b, and d [1]
- a, c, and d [2]
- b, c, and d [2]
- a, b, c, and d [0]

- none of a, b, c, nor d [0]

9. Which other changes could improve EA's results? [4 pts]

- (a) Decrease mutation rate to 1%
- (b) Increase mutation rate to 50%
- (c) Decrease λ to 200
- (d) Increase λ to 10,000

Select one of:

- a [0]
- b [2]
- c [2]
- d [0]
- a and b [1]
- a and c [2]
- a and d [0]
- **b and c**; an increased mutation rate could allow faster exploration, and a decreased λ would increase the number of generations for changes to accumulate over.
- b and d [2]
- c and d [0]
- a, b, and c [2]
- a, b, and d [0]
- a, c, and d [0]
- b, c, and d [2]
- a, b, c, and d [0]
- none of a, b, c, nor d [0]

The next two questions are about the following scenario:

Consider an EA designed to solve the Traveling Salesman Problem for a given list of city locations. A genotype is an ordered permutation of cities to visit, after which the path returns to the first city. The fitness function is designed to minimize the length of this looping path. Consider the phenotype to be the closed polygon composed of line segments between cities. For the genotype to phenotype encoding, consider the domain and codomain to include only valid genotypes and phenotypes.

10. Is the genetic encoding described here: [4 pts]

- (a) pleiotropic
- (b) polygenic
- (c) both
- (d) neither

Select one of:

- a [1]
- b [1]
- **c**; It is pleiotropic because one gene can affect multiple phenotypic traits (path segment lengths to the previous and next city), and polygenic because one phenotypic trait (path segment) can depend on multiple genes (the start and end city for that segment).
- d [0]

11. Is the genotype to phenotype encoding described here: [4 pts]

- (a) injective
- (b) surjective
- (c) bijective

Select one of:

- a [0]
- b; It is surjective because every valid path can be represented by the ordered list of cities it visits. It is not injective, because the same path can be represented by multiple genotypes, with the same city order but a different starting city.
- c [1]
- none of a, b, nor c [1]

12. What is the binary gray code for the standard binary number 11110111? [4 pts]

10001100

13. What is the standard binary number encoded by the binary gray code 000111001101? [4 pts]

000101110110

14. Given the following two parents with permutation representation:

$p1 = (327619548)$

$p2 = (986125347)$

compute the first offspring with Order Crossover, using crossover points between the 1st and 2nd loci and between the 4th and 5th loci. Show your offspring construction steps. [4 pts]

- (a) $\cdot 276 \dots$
- (b) 127653498

15. Given the following two parents with permutation representation:

$p1 = (426537819)$

$p2 = (981725346)$

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: 4-9-6-1

Cycle 2: 2-8-3

Cycle 3: 5-7

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: $4 \cdot 6 \dots 19$
- (b) Add cycle 2 from parent 2: $486 \cdot 2 \cdot 319$
- (c) Add cycle 3 from parent 1: 486527319

16. Given the following two parents with permutation representation:

$p1 = (912834756)$

$p2 = (145283679)$

compute the first offspring with PMX, using crossover points between the 3rd and 4th loci and between the 8th and 9th loci. Show your offspring construction steps. [6 pts]

(a) $\cdot \cdot \cdot 83475 \cdot$

(b) $\cdot 2 \cdot 83475 \cdot$

(c) $\cdot 2683475 \cdot$

(d) 126834759

17. Given the following parents with permutation representation:

$p1 = (135792468)$

$p2 = (614879532)$

compute the first offspring with Edge Crossover, except that for each random choice you instead select the lowest element. Show how you arrived at your answer by filling the following templates: [10 pts]

Edge Table:

Element	Edges
---------	-------

Construction Table:

Element Selected	Reason Selected	Partial Result
------------------	-----------------	----------------

Original Edge Table:

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

Construction Table:

Element selected	Reason	Partial result
1	Lowest	1
3	Shortest list size	13
5	Common edge	135
7	Equal list size, so lowest	1357
9	Common edge	13579
2	Only element	135792
4	Equal list size, so lowest	1357924
6	Equal list size, so lowest	13579246
8	Last element	135792468

Edge Table After Step 1:

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

Edge Table After Step 2:

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

Edge Table After Step 3:

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

Edge Table After Step 4:

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

Edge Table After Step 5:

Element	Edges	Element	Edges
		6	4,8,2
2	4,6		
		8	6,4
4	2,6,8	9	2

Edge Table After Step 6:

Element	Edges	Element	Edges
		6	4,8
2	4,6		
		8	6,4
4	6,8		

Edge Table After Step 7:

Element	Edges	Element	Edges
		6	8
		8	6
4	6,8		

Edge Table After Step 8:

Element	Edges	Element	Edges
		6	8
		8	

18. Show how one can configure an EA to be functionally equivalent to Simulated Annealing (SA) by copying and filling out the provided table on your answer paper, or, if you believe it is inherently impossible, explain why. [6]

For your convenience, here is pseudo-code for SA:

function SimulatedAnnealing(*problem*, *schedule*) **returns** a solution state

```

current  $\leftarrow$  MakeNode(InitialState[problem])
for t  $\leftarrow$  1 to  $\infty$  do
  T  $\leftarrow$  schedule[t]
  if T = 0 then
    return current
  end if
  next  $\leftarrow$  a randomly selected successor of current
   $\Delta E \leftarrow$  Value[next] – Value[current]
  if  $\Delta E > 0$  then
    current  $\leftarrow$  next
  else
    current  $\leftarrow$  next with probability  $e^{\Delta E/T}$ 
  end if
end for

```

An EA can be configured to be functionally equivalent to SA as follows:

μ	1
λ	1
Initialization	the single pop member is uniform randomly selected $t=1$
Parent selection	the lone population member is always selected to be the parent
Recombination rate	0%
Mutation rate	100%
Mutation	offspring = random successor of the parent
Survival selection type	$(\mu + \lambda)$
Survival selection	fitness $F[]$ is defined to be equal to <i>Value</i> [] $T=\text{schedule}[t]$ IF $F[\text{offspring}] - F[\text{parent}] > 0$ THEN offspring survives ELSE offspring survives with probability $e^{\frac{F[\text{offspring}] - F[\text{parent}]}{T}}$ $t++$
Termination condition	$T==0$