Tecnológico de Monterrey

Computational intelligence

Homework 5

Student:
Jacob Rivera

Professor:
Dr. José Carlos Bayliss

September 20, 2019

1 Partially mapped crossover

Apply the PMX operator to combine the following two parents. Assume that the two points selected for crossover are two and five, and that the numbering of the crossover points starts at 1.

Parent A: 2 8 3 6 1 4 5 7
Parent B: 1 2 3 6 7 5 8 4

2 Selecting the best representation

A company wants to determine the best representation for a planning problem. The problem this company tries to solve requires to find a sequence of 12 actions in order to move a robot from point A to point B (the robot is located on a grid along with some obstacles). The robot has four available actions in its repository: UP, DOWN, LEFT, and RIGHT. Any of these actions takes exactly one second to execute. Since you are a well-known expert on genetic algorithms, this company is requesting your help to make a choice that maximizes the opportunities that its genetic algorithm implementation fins a good-quality solution for the problem. Given the limited amount of resources for the project, they have only considered two representations for its genetic algorithm implementation: binary and integer-based representations. Also, they have already decided that, regardless of the chosen representation, they will use linear ranking selection and one-point crossover, and that $p_c = 1$ and $p_m = 0$.

Just hours before you submit your report with the recommended representation (and the corresponding justification), you are informed that a new action will be added to the robot's repository: JUMP (which allows the robot to jump over the obstacles in the direction of the previous move). How would this change affect your decision on the representation to recommend? Justify your answer.

R → I would prefer a integer-based representation as it is simpler to reason about and fits the intended use case rather well. It is also shorter to use than the binary representation, which can only represent two options per symbol, as opposed as 10 in a integer representation. This would also prevent problems with adding a new action to the first 4, because you would only need to use another symbol, while in a binary representation you would have to add another bit to each action and maybe modify the logic which reads and interprets this representation.

3 Schemata analysis

For this exercise, assume that the chromosomes in a genetic algorithm are coded as six-bit strings, and that the fitness of each chromosome is calculated as the number of ones in the chromosome. For example, the chromosomes 110101 and 001100 have a fitness of 4 and 2, respectively.

Imagine that, at some generation t, the population in the genetic algorithm looks as follows:

Population
101000
001101
011011
010100
110010

Calculate the expected number of chromosomes that will contain the following schemata in the new population (at generation $t + 1$): $H_1 = *01 * 10$, $H_2 = * * 01 * *$, $H_3 = *10 * **$, and $H_4 = * * *0 * *$. For your calculations, consider $p_c = 0.90$ and $p_m = 0.01$.

	Population	f
A_1	101000	2
A_2	001101	3
A_3	011011	4
A_4	010100	2
A_5	110010	3

Avg = 2.8

	Schema	Contained in	count	fitness	f(h)
H_1	*01*10		0	0	0
H_2	**01**	A_4	1	2	2
H_3	*10***	A_4, A_5	2	2,3	2.5
H_4	***0**	A_1, A_3, A_5	3	2,4,3	3

Reproduction

$$m(H_1,t)\frac{f(H_1)}{f} = 0 * \frac{0}{2.8} = 0$$
$$m(H_2,t)\frac{f(H_2)}{f} = 1 * \frac{2}{2.8} = 0.7142$$
$$m(H_3,t)\frac{f(H_3)}{f} = 2 * \frac{2.5}{2.8} = 1.7857$$
$$m(H_4,t)\frac{f(H_4)}{f} = 3 * \frac{3}{2.8} = 3.2142$$

Crossover

$H_1 = *01*10$

$$p_s = 1 - (p_c * p_d)$$
$$= 1 - (0.9 * 0.8) = 0.28$$

$H2 = **01**$

$$p_s = 1 - (p_c * p_d)$$
$$= 1 - (0.9 * 0.2) = 0.82$$

$H3 = *10***$

$$p_s = 1 - (p_c * p_d)$$
$$= 1 - (0.9 * 0.2) = 0.82$$

$H4 = ***0**$

$$p_s = 1 - (p_c * p_d)$$
$$= 1 - (0.9 * 0) = 1$$

Mutation

$p_m = 0.01$

$H_1 = *01*10$
$O(H_1) = 4$
$$(1 - 0.01)^4 = 0.9605$$

$H_2 = **01**$
$O(H_2) = 2$
$$(1 - 0.01)^2 = 0.9801$$

$H_3 = *10***$
$O(H_3) = 2$
$$(1 - 0.01)^2 = 0.9801$$

$H_4 = ***0**$
$O(H_4) = 1$
$$(1 - 0.01)^1 = 0.999$$

Expected number of chromosomes

$$m(H_1, t+1) = 0 * 0.28 * 0.9605 = 0$$
$$m(H_2, t+1) = 0.7142 * 0.82 * 0.9801 = 0.5739896844$$
$$m(H_3, t+1) = 1.7857 * 0.82 * 0.9801 = 1.435134947$$
$$m(H_4, t+1) = 3.2142 * 1 * 0.999 = 3.2109858$$

4 Probability of surviving crossover

We have already seen how to calculate the probability of surviving one-point crossover for binary strings. This time, you will explore the effect of using a two-point crossover operator and how it affects the probability of surviving crossover for the following schemata: $H_1 = *100 * 1*$, $H_2 = 1 * 0 * * * *$, and $H_3 = *00 * *11$.

Given that using a two-point crossover can only be used between indices 1 and 6, starting from 0 and we use two of them, we can use the following table to see the possibilities:

Pair	Pair	Pair
1, 2	1, 3	1, 4
1, 5	1, 6	2, 3
2, 4	2, 5	2, 6
3, 4	3, 5	3, 6
4, 5	4, 6	5, 6

After that we can then see that for the schemata to survive, it must no be cut between the fixed values. Therefore, we can then obtain the possibilities of survival by simply counting the pairs that attain this condition and divide it by the total of pairs. Which is shown here:

Schemata	Pairs in which it survives	Probability of survival
H_1	(1, 6), (4, 5)	$\frac{2}{15}$
H_2	(1, 2), (3, 4), (3, 5), (3, 6), (4, 5), (4, 6), (5, 6)	$\frac{7}{15}$
H_3	(3, 4), (3, 5), (4, 5)	$\frac{3}{15}$

5 Genetic programming

In this exercise, you will use genetic programming to generate boolean expressions (represented as functions). The following list indicates the inputs of the objective boolean expression and its corresponding outputs: f(false, false) = true, f(false, true) = false, f(true, false) = true, and f(true, true) = false. Please note that, in order to generate a function, we need to provide variables as inputs (instead of the actual values true and false). Then, the terminal set for this exercise will contain only the variables x and y, such that we can express the function as f(x, y).

The following tree-based data structure depicts one possible function generated during the evolutionary process.

In this case, the corresponding boolean expression x AND (NOT(y)) produces a correct result in three out of four cases: false AND (NOT(true)), true AND (NOT(false)), and true AND (NOT(true)); but it fails for the case false AND (NOT(false)) (produces false but the objective function requires true). Then, its fitness is 3/4 = 0.75.

Given this information, answer the following questions or do what is requested (each question or request is independent from the rest).

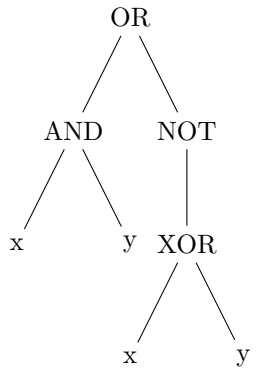
- Do we need to include ephemeral constants in the terminal set? Justify your answer.

  Its not necessary given that the expression needs to be reduced before being evaluated. And the results of each operation are the same as any of the terminals.

- What is the minimal terminal set required to fulfill the sufficiency property of the function and terminal sets? Explain your answer.

  Terminal set = $\{1, 0, x, y\}$
  Function set = $\{AND, OR, NOT\}$

- Assuming that the terminal set contains the operations AND, OR, XOR and NOT, how would you represent, by using a tree-based data structure, the expression (x AND y) OR NOT(x XOR y)?



- Assuming that the terminal set contains the operations AND, OR and NOT, what is the fitness of the expression (x AND NOT(y)) OR NOT(x OR y)?

x	y	x AND NOT(y)	NOT(x OR y)	(x AND NOT(y)) OR NOT(x OR y)
0	0	0	1	1
0	1	0	0	0
1	0	1	0	1
1	1	0	0	0

  Given the in initial conditions, this gives us a 4/4 or a 100% fitness

- What would be the fitness of the tree-based data structure depicted in the following picture:

x	y	x OR y	y AND x	(y AND x) OR Y	NOT((x OR y) OR ((y AND x) OR Y))
0	0	0	0	0	1
0	1	1	0	1	0
1	0	1	0	1	0
1	1	0	1	1	0

  Given the in initial conditions, this gives us a 3/4 or a 75gs% fitness