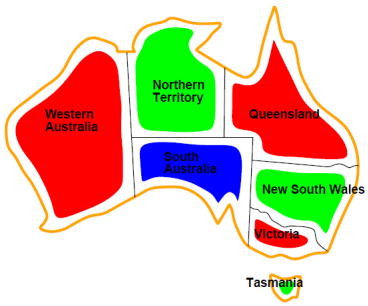
COMP 2019 Workbook Exercises Week 5 – Evolutionary Algorithms Answers

Revisit the Australia Map Colouring problem that was introduced in Week 4 Constraint Satisfaction Search, where the objective is to assign one of three colours to each region on the map so that no adjacent regions share the same colour.



In this exercise, you will solve this problem using a Genetic Algorithm.

a) Define an encoding of the problem as chromosomes.

Use a chromosome of length 7, one gene per region. Use a fixed order for the regions within the chromosome.

The values each chromosome can take on are the colours: r,g,b.

Suppose the order of the genes in the chromosome is: WA, NT, SA, Q, NSW, V, T.

The state depicted above is represented as:

[red, green, blue, red, green, red, green]

b) Define the fitness function.

The fitness can be defined as the negative of the number of conflicting pairs of colours in a chromosome. We take the negative, since the maximum is then 0 which is what the standard GA will strive to obtain. Alternatively, we could use the number of non-conflicting pairs, or change the GA to prefer minimising the fitness function (instead of maximising it).

Here is a python definition of a possible fitness function:

```
def fitness_map(chromosome):
    wa, nt, sa, q, nsw, vic, tas = chromosome
    violations = (wa == nt) + (sa == nt) + (sa == q) + (sa == nsw) + \
                 (sa == vic) + (q == nsw) + (nsw == vic)
    return -violations
```

c) Define crossover and mutation operators.

We can use single point crossover for this problem.

Mutation can be to assign a different randomly selected colour to a randomly selected gene.

d) Suppose a population sized 5 is to be used for the genetic algorithm. Define the initial population.

The initial values for each chromosome are chosen randomly.

There are 5 individuals in the population:

1. [red, red, blue, green, green, green, red]
2. [blue, green, red, red, blue, red, green]
3. [blue, blue, red, blue, red, green, red]
4. [red, blue, green, red, red, blue, blue]
5. [green, blue, red, blue, green, green, green]

e) Calculate the fitness of each individual in the initial population.

[red, red, blue, green, green, green, red] -3
[blue, green, red, red, blue, red, green] -2
[blue, blue, red, blue, red, green, red] -2
[red, blue, green, red, red, blue, blue] -1
[green, blue, red, blue, green, green, green] -1

f) Calculate the offspring of individuals 1 and 2 in the initial population. Do not apply the mutation operator.

We select a cut-point for the parents at random. Say we cut after gene number 4.

Parents:

[red, red, blue, green | green, green, red]

[blue, green, red, red | blue, red, green]

Offspring:

[red, red, blue, green, blue, red, green] (fitness -2)

[blue, green, red, red, green, green, red] (fitness -2)

In this case, the offspring have worse fitness than the parents.