


```

import random
import string

from seeds.Cell import *
from seeds.Plugin import *
from seeds.SEEDSError import *
from seeds.utils.sampling import roulette_select

__name__ = 'SentenceCell'
__description__ = 'Evolving configured target sentence using GAs'
__version__ = (1,0)
__author__ = 'Brian Connelly <bdc@msu.edu>'
__credits__ = 'Brian Connelly and Luis Zaman'
__type__ = 1

class SentenceCell(Cell, Plugin):
    alphabet = string.ascii_lowercase + string.ascii_uppercase + ' '
    types = ['Sentence']

    def __init__(self, experiment, population, node, type=None,
                 name="SentenceCell", label=None):
        super(SentenceCell, self).__init__(experiment, population,
                                           node=node, type=0, name=name, label=label)

        self.target = self.experiment.config.get(self.config_section, 'target')
        self.genome_length = len(self.target)
        self.mutation = self.experiment.config.getfloat(self.config_section, 'mutation',
                                                       default=0)

        self.genome = random.sample(self.alphabet, self.genome_length)
        self.calculate_fitness()

    def update(self):
        self.neighbors = self.get_neighbors()
        neighbor_fitnesses = [n.fitness for n in self.neighbors]

        # Use roulette wheel to find the most fit parents
        parents = roulette_select(items=self.neighbors, fitnesses=neighbor_fitnesses, k=2)

        # Choose random crossover points
        cp1 = random.randint(0, self.genome_length-1)
        cp2 = random.randint(0, self.genome_length-1)

        self.genome[:cp1] = parents[0].genome[:cp1]
        self.genome[cp1:cp2] = parents[1].genome[cp1:cp2]
        self.genome[cp2:] = parents[0].genome[cp2:]

        # Apply mutations to the offspring
        for i in range(self.genome_length):
            if random.random() < self.mutation:
                self.genome[i] = random.choice(self.alphabet)

        self.calculate_fitness()

    def calculate_fitness(self):
        """Fitness is 2^(number of matching characters)"""
        self.fitness = 1
        for i in range(self.genome_length):
            if self.genome[i] == self.target[i]:
                self.fitness *= 2

```

Listing 4: SEEDS Cell implementing a genetic algorithm to evolve to match a target string. This cell's two parameters, *target* and *mutation*, can be defined in the SentenceCell section of a configuration file.