Genetic Algorithm

Author: Caitlin-Dawn Sangcap

This was my attempt at recreating Costa's work taking a genetic algorithm approach to the League of Legends team composition based on 3 strategies:

```
1. Hard Engage
2. Team Fight
3. Poke
```

Due to missining data and not being able to find the source of that data there were certain features that I was not able to code for. Some of those features were:

- A specific Champion's winrate
- A specific Champion's recommended lane position
- A specific Champion's counter
- Ensuring that each team does not have any repeating champions
- Excluding a specific set of champions from the pool selection

Bugs remaining:

```
For some reason, there is a bug in where if I
try to find the fitness value for the Poke
strategy, it always returns 0. I don't
understand why but it is another thing I plan to
edit in the future.
```

How to change the settings for the program:

```
Step 1) In the 'Generating the size of the
population' section, change the value on for
'sol_per_pop' to the desired size of the
population.
Step 2) In the 'Passing the teams to the genetic
algorithm' section, change the number of
generations and the mutation rate to the desired
values.
Step 3) In the 'Saving the results in a .csv
file for later use' section, change the name of
the .csv file to whatever name you desire to
record the fitness value results of the current
iteration.
```

In [59]:

```
import numpy as np
from random import choice
import pandas as pd
import random
import csv
import matplotlib.pyplot as plt
```

Genetic Algorithm functions

Due to not being able to completely replicate the functions and classes from the Costa paper, I resulted to using another coder's genetic algorithm functions and modifying it as needed for my purposes.

In [46]:

```
#pulled the code from:
# https://github.com/ahmedfgad/GeneticAlgorithmPython/blob/m

#modified function
#will get passed a 2d array containing row index for the cha
def cal_pop_fitness(pop,champ_info,strategy):
    # Calculating the fitness value of each solution in the
    # The fitness function calulates the sum of products bet
```

```
    fitness = []
    for row in range(len(pop)):
        temp = 0
        for col in range(len(pop[row])):
            if(strategy == "Hard Engage"):
                temp1 = champ_info.iloc[pop[row][col]]['Atta
                temp += temp1
            elif (strategy == "Team Fight"):
                temp1 = champ_info.iloc[pop[row][col]]['Atta
                temp += temp1
            elif (strategy == "Poke"):
                temp1 = champ_info.iloc[pop[row][col]]['Atta
                temp += temp1
            else:
                print("Invalid Strategy")
        fitness.append(temp)
    return fitness

#added line 39 to line 41
#no change needed here
#fitness = 1d array
def select_mating_pool(pop, fitness, num_parents):
    #make a copy of fitness
    copied_fit = []
    for num in range(len(fitness)):
        copied_fit.append(fitness[num])

    # Selecting the best individuals in the current generati
    parents = np.empty((num_parents, pop.shape[1]))
    for parent_num in range(num_parents):
        max_fitness_idx = np.where(copied_fit == np.max(copi
        max_fitness_idx = max_fitness_idx[0][0]
        parents[parent_num, :] = pop[max_fitness_idx, :]
        copied_fit[max_fitness_idx] = -99999999999
    return parents

#original function
#no change needed here
def crossover(parents, offspring_size):
    offspring = np.empty(offspring_size)
    # The point at which crossover takes place between two p
    crossover_point = np.uint8(offspring_size[1]/2)
    #print("Crossover_point: ",crossover_point)

    for k in range(offspring_size[0]):
        # Index of the first parent to mate.
        parent1_idx = k%parents.shape[0]
```

```
        # Index of the second parent to mate.
        parent2_idx = (k+1)%parents.shape[0]
        # The new offspring will have its first half of its
        offspring[k, 0:crossover_point] = parents[parent1_id
        # The new offspring will have its second half of its
        offspring[k, crossover_point:] = parents[parent2_idx
    return offspring


def mutation(offspring_crossover,population,mutation_chance)
    random_chance = int(np.random.uniform(0,11,1))
    # Mutation changes a single gene in each offspring rando

    # to emualate the chance of mutation happening
    if (random_chance <= mutation_chance):

        for idx in range(offspring_crossover.shape[0]):

            #getting the number of genes to mutate
            mutation_counter = int(np.random.uniform(0,5,1))

            if mutation_counter > 0:

                #to mutate the number of genes
                mutate = 0
                while mutate < mutation_counter:

                    #does not account for muation in the sam
                    #the index of the gene to change
                    gene_idx = int(np.random.uniform(0,5,1))

                    check = 0
                    random_value = 0
                    while(check == 0):
                        #print("check = ", check)
                        random_value = int(np.random.uniform
                        #print("random_value = ",random_valu
                        checker = np.argwhere(population ==
                        #print("checker = ", checker)
                        if checker.size == 0:
                            if random_value < 152:
                                check = 1

                    offspring_crossover[idx, gene_idx] = ran
                    mutate +=1

    return offspring_crossover
```

```python
#new function to handle the fitness value
def fitness_value(fitness_info,strategy):
    # Calculating the fitness value of each solution in the
    fitness_val = 0
    if (strategy == "Hard Engage"):
        fitness_val = (fitness_info/2125)*100
    elif (strategy == "Team Fight"):
        fitness_val = (fitness_info/405)*100
    elif (strategy == "Poke"):
        fitness_val == (fitness_info/3630)*100
    else:
        print("Invaled strategy")
    return fitness_val
```

Load in the Champion Information

In [47]:

```python
#getting the champion dataset
champion = pd.read_csv('Champion_Info_Mod.csv', sep=",")
champion.head()
```

Out[47]:

	Unnamed: 0	ID_Num	Name	Attack Damage	Attack Damage per Level	Attack Range	Attack Speed per Level	Health Points
0	0	266	Aatrox	60.00	5.00	175	2.500	580.0
1	1	103	Ahri	53.04	3.00	550	2.000	526.0
2	2	84	Akali	62.40	3.30	125	3.200	575.0
3	3	12	Alistar	62.00	3.75	125	2.125	600.0
4	4	32	Amumu	53.38	3.80	125	2.180	615.0

```
In [48]: # Inputs of the equation.
         # can change this as needed if a champion was already select

         equation_inputs = [425,425,425,425,425]

         # Number of the weights we are looking to optimize.
         # going to be the sum of the needed stats of a given champic
         num_weights = len(equation_inputs)
```

Generating the size of the population

```
In [54]: sol_per_pop = 30
         # Defining the population size.

         pop_size = (sol_per_pop,num_weights) # The population will h

         #Creating the initial population.
         new_population = np.zeros(shape=(sol_per_pop,num_weights), d
         #print(new_population)
```

Generating the teams

```
In [55]: pop_size_real = sol_per_pop*num_weights
         trial = np.asarray(random.sample(range(0,152),pop_size_real)

         new_population=trial.reshape(sol_per_pop,num_weights)
         #print(new_population)
```

Passing the teams to the genetic algorithm

```
In [56]: records = [["Generation", "Hard Engage","Team Fight","Poke"]
         mutation_rate = 7
         num_generations = 1000

         #number of parents mating minimum = 2
         num_parents_mating = sol_per_pop -1
         for generation in range(num_generations):
             gather = []
```

```
    gather.append(generation+1)
    for num in range(3):
        #print("Generation: ", generation+1)
        # Measuring the fitness of each chromosome in the po
        strat = 'No strat'
        if (num == 0):
            strat = "Hard Engage"
            #print(strat)
        elif (num == 1):
            strat = "Team Fight"
            #print(strat)
        elif (num == 2):
            strat = "Poke"
            #print(strat)
        fitness = cal_pop_fitness(new_population,champion,st
        #print(strat,fitness)

        # Selecting the best parents in the population for m
        parents = select_mating_pool(new_population, fitness
        #print("Parents: ")
        #print(parents)

        # Generating next generation using crossover.
        if pop_size[0] == num_parents_mating:
            offspring_crossover = crossover(parents,offsprin
        else:
            offspring_crossover = crossover(parents,offsprin
        #print("Offspring_crossover:")
        #print(offspring_crossover)

        # Adding some variations to the offsrping using muta
        offspring_mutation = mutation(offspring_crossover,ne
        #print("offspring_mutation:")
        #print(offspring_mutation)
        # Creating the new population based on the parents a
        new_population[0:parents.shape[0], :] = parents
        new_population[parents.shape[0]:, :] = offspring_mut

        #print("new_population:")
        #print(new_population)

        # The best result in the current iteration.
        fitness_new = cal_pop_fitness(new_population,champio
        fit_value = fitness_value(np.max(fitness_new),strat)
        gather.append(fit_value)
        #print(strat, fit_value)
        #print("Best result after crossover and mutation (Ha
```

```python
    records.append(gather)
```

In [57]:
```python
#print(records)
```

```
[['Generation', 'Hard Engage', 'Team Fight', 'Poke'], [1, 95
.76470588235294, 89.67901234567901, 0], [2, 95.7647058823529
4, 89.9148148148148, 0], [3, 96.04705882352941, 89.914814814
8148, 0], [4, 96.04705882352941, 89.9148148148148, 0], [5, 9
6.04705882352941, 89.9148148148148, 0], [6, 96.0470588235294
1, 89.9148148148148, 0], [7, 96.04705882352941, 89.914814814
8148, 0], [8, 96.04705882352941, 89.9148148148148, 0], [9, 9
6.04705882352941, 89.9148148148148, 0], [10, 96.047058823529
41, 89.9148148148148, 0], [11, 96.04705882352941, 89.9148148
148148, 0], [12, 96.04705882352941, 89.9148148148148, 0], [1
3, 96.04705882352941, 89.9148148148148, 0], [14, 96.04705882
352941, 89.9148148148148, 0], [15, 96.04705882352941, 89.914
8148148148, 0], [16, 96.04705882352941, 89.9148148148148, 0]
, [17, 96.04705882352941, 89.9148148148148, 0], [18, 96.0470
5882352941, 89.9148148148148, 0], [19, 96.04705882352941, 89
.9148148148148, 0], [20, 96.04705882352941, 89.9148148148148
, 0], [21, 96.04705882352941, 89.9148148148148, 0], [22, 96.
04705882352941, 89.9148148148148, 0], [23, 96.04705882352941
, 89.9148148148148, 0], [24, 96.04705882352941, 89.914814814
8148, 0], [25, 96.04705882352941, 89.9148148148148, 0], [26,
96.04705882352941, 89.9148148148148, 0], [27, 96.56470588235
294, 89.9148148148148, 0], [28, 96.56470588235294, 89.914814
8148148, 0], [29, 96.04705882352941, 89.9148148148148, 0], [
30, 96.04705882352941, 89.9148148148148, 0], [31, 96.0470588
2352941, 92.83333333333333, 0], [32, 96.04705882352941, 90.8
1481481481481, 0], [33, 96.04705882352941, 91.44567901234568
, 0], [34, 96.04705882352941, 91.44567901234568, 0], [35, 96
.04705882352941, 91.44567901234568, 0], [36, 96.047058823529
41, 91.44567901234568, 0], [37, 96.04705882352941, 91.445679
01234568, 0], [38, 96.04705882352941, 91.44567901234568, 0],
[39, 96.04705882352941, 91.44567901234568, 0], [40, 96.04705
882352941, 91.44567901234568, 0], [41, 96.04705882352941, 91
.44567901234568, 0], [42, 96.04705882352941, 91.445679012345
68, 0], [43, 96.28235294117647, 91.44567901234568, 0], [44,
96.04705882352941, 91.44567901234568, 0], [45, 96.0470588235
2941, 91.44567901234568, 0], [46, 96.04705882352941, 91.4456
7901234568, 0], [47, 96.04705882352941, 91.44567901234568, 0
], [48, 96.04705882352941, 91.44567901234568, 0], [49, 96.04
705882352941, 91.44567901234568, 0], [50, 96.04705882352941,
91.44567901234568, 0], [51, 96.70588235294117, 91.4456790123
4568, 0], [52, 96.04705882352941, 91.44567901234568, 0], [53
, 95.95294117647059, 92.91358024691358, 0], [54, 96.32941176
470588, 92.91358024691358, 0], [55, 96.32941176470588, 92.91
358024691358, 0], [56, 96.32941176470588, 92.91358024691358,
0], [57, 96.32941176470588, 92.91358024691358, 0], [58, 96.3
2941176470588, 92.91358024691358, 0], [59, 96.32941176470588
```

, 92.91358024691358, 0], [60, 96.32941176470588, 92.91358024
691358, 0], [61, 96.32941176470588, 92.91358024691358, 0], [
62, 96.32941176470588, 92.91358024691358, 0], [63, 96.329411
76470588, 92.91358024691358, 0], [64, 96.32941176470588, 92.
91358024691358, 0], [65, 96.32941176470588, 92.913580246913
58, 0], [66, 96.32941176470588, 92.91358024691358, 0], [67, 9
6.32941176470588, 92.91358024691358, 0], [68, 95.67058823529
412, 91.44567901234568, 0], [69, 95.67058823529412, 91.44567
901234568, 0], [70, 95.67058823529412, 91.44567901234568, 0]
, [71, 95.67058823529412, 91.44567901234568, 0], [72, 95.670
58823529412, 91.44567901234568, 0], [73, 95.67058823529412,
91.44567901234568, 0], [74, 95.67058823529412, 91.4456790123
4568, 0], [75, 95.67058823529412, 91.44567901234568, 0], [76
, 95.67058823529412, 91.44567901234568, 0], [77, 96.51623529
411766, 91.44567901234568, 0], [78, 95.67058823529412, 91.44
567901234568, 0], [79, 95.67058823529412, 91.44567901234568,
0], [80, 95.67058823529412, 91.44567901234568, 0], [81, 95.6
7058823529412, 91.44567901234568, 0], [82, 96.61176470588235
, 91.44567901234568, 0], [83, 95.67058823529412, 91.44567901
234568, 0], [84, 96.09411764705882, 91.44567901234568, 0], [
85, 95.67058823529412, 91.44567901234568, 0], [86, 95.670588
23529412, 91.59876543209877, 0], [87, 95.67058823529412, 91.
44567901234568, 0], [88, 95.82870588235295, 91.4456790123456
8, 0], [89, 95.67058823529412, 91.44567901234568, 0], [90, 9
6.04894117647058, 91.44567901234568, 0], [91, 95.67058823529
412, 91.44567901234568, 0], [92, 95.67058823529412, 91.44567
901234568, 0], [93, 95.67058823529412, 91.44567901234568, 0]
, [94, 95.67058823529412, 91.89506172839505, 0], [95, 95.670
58823529412, 91.44567901234568, 0], [96, 95.67058823529412,
91.44567901234568, 0], [97, 95.67058823529412, 91.4456790123
4568, 0], [98, 95.67058823529412, 91.44567901234568, 0], [99
, 95.67058823529412, 91.44567901234568, 0], [100, 95.6705882
3529412, 91.44567901234568, 0], [101, 96.18823529411765, 91.
44567901234568, 0], [102, 95.67058823529412, 91.445679012345
68, 0], [103, 95.67058823529412, 91.44567901234568, 0], [104
, 95.67058823529412, 91.44567901234568, 0], [105, 95.6705882
3529412, 91.44567901234568, 0], [106, 95.67058823529412, 91.
44567901234568, 0], [107, 96.04141176470588, 91.445679012345
68, 0], [108, 95.67058823529412, 91.44567901234568, 0], [109
, 96.56470588235294, 91.44567901234568, 0], [110, 95.6705882
3529412, 91.44567901234568, 0], [111, 95.67058823529412, 91.
44567901234568, 0], [112, 95.67058823529412, 91.445679012345
68, 0], [113, 95.67058823529412, 91.44567901234568, 0], [114
, 95.67058823529412, 90.51975308641977, 0], [115, 95.6705882
3529412, 90.51975308641977, 0], [116, 95.7176470588235 3, 90.
51975308641977, 0], [117, 95.67058823529412, 90.519753086419
77, 0], [118, 95.67058823529412, 90.51975308641977, 0], [119
, 95.67058823529412, 90.51975308641977, 0], [120, 95.6705882
3529412, 90.51975308641977, 0], [121, 95.67058823529412, 90.
51975308641977, 0], [122, 96.0, 90.51975308641977, 0], [123,

95.67058823529412, 90.51975308641977, 0], [124, 96.094117647
05882, 90.51975308641977, 0], [125, 95.67058823529412, 90.51
975308641977, 0], [126, 95.67058823529412, 90.51975308641977
, 0], [127, 96.09411764705882, 91.62345679012347, 0], [128,
95.67058823529412, 90.51975308641977, 0], [129, 95.670588235
29412, 90.51975308641977, 0], [130, 95.67058823529412, 90.51
975308641977, 0], [131, 95.67058823529412, 90.51975308641977
, 0], [132, 95.67058823529412, 90.51975308641977, 0], [133,
95.67058823529412, 90.51975308641977, 0], [134, 95.670588235
29412, 90.51975308641977, 0], [135, 96.32941176470588, 90.51
975308641977, 0], [136, 95.67058823529412, 90.51975308641977
, 0], [137, 96.0, 90.51975308641977, 0], [138, 96.8, 90.5197
5308641977, 0], [139, 95.67058823529412, 90.51975308641977,
0], [140, 95.67058823529412, 90.51975308641977, 0], [141, 95
.67058823529412, 90.51975308641977, 0], [142, 96.23529411764
706, 90.51975308641977, 0], [143, 95.67058823529412, 90.5197
5308641977, 0], [144, 95.95294117647059, 91.10493827160494,
0], [145, 95.67058823529412, 90.51975308641977, 0], [146, 95
.67058823529412, 90.51975308641977, 0], [147, 95.67058823529
412, 90.51975308641977, 0], [148, 95.67058823529412, 90.5197
5308641977, 0], [149, 95.67058823529412, 90.51975308641977,
0], [150, 95.67058823529412, 90.51975308641977, 0], [151, 95
.67058823529412, 90.51975308641977, 0], [152, 95.67058823529
412, 90.51975308641977, 0], [153, 95.67058823529412, 90.5197
5308641977, 0], [154, 95.67058823529412, 90.51975308641977,
0], [155, 96.23529411764706, 90.82098765432097, 0], [156, 96
.23529411764706, 91.48271604938272, 0], [157, 96.23529411764
706, 90.82098765432097, 0], [158, 96.1195294117647, 90.76666
666666668, 0], [159, 96.32941176470588, 90.76666666666668, 0
], [160, 96.32941176470588, 90.76666666666668, 0], [161, 95.
85882352941177, 90.76666666666668, 0], [162, 95.858823529411
77, 90.76666666666668, 0], [163, 96.32941176470588, 91.48271
604938272, 0], [164, 95.71764705882353, 90.76666666666668, 0
], [165, 96.32941176470588, 90.76666666666668, 0], [166, 95.
71764705882353, 90.76666666666668, 0], [167, 95.717647058823
53, 90.51975308641977, 0], [168, 95.71764705882353, 90.51975
308641977, 0], [169, 95.81176470588235, 90.51975308641977, 0
], [170, 95.67058823529412, 90.51975308641977, 0], [171, 95.
67058823529412, 90.51975308641977, 0], [172, 95.670588235294
12, 90.51975308641977, 0], [173, 95.67058823529412, 90.51975
308641977, 0], [174, 95.67058823529412, 90.51975308641977, 0
], [175, 95.67058823529412, 90.51975308641977, 0], [176, 95.
67058823529412, 90.51975308641977, 0], [177, 95.670588235294
12, 91.27283950617283, 0], [178, 95.67058823529412, 90.51975
308641977, 0], [179, 95.67058823529412, 90.51975308641977, 0
], [180, 95.67058823529412, 90.51975308641977, 0], [181, 95.
67058823529412, 91.38888888888889, 0], [182, 96.564705882352
94, 90.51975308641977, 0], [183, 96.5176470588235, 91.10493
827160494, 0], [184, 95.67058823529412, 90.51975308641977, 0
], [185, 95.67247058823529, 90.51975308641977, 0], [186, 95.

67058823529412, 90.51975308641977, 0], [187, 95.764705882352
94, 90.51975308641977, 0], [188, 95.67058823529412, 90.51975
308641977, 0], [189, 95.67058823529412, 90.51975308641977, 0
], [190, 96.0, 90.51975308641977, 0], [191, 95.670588235294l
2, 90.51975308641977, 0], [192, 95.67058823529412, 90.519753
08641977, 0], [193, 95.67058823529412, 90.51975308641977, 0]
, [194, 95.67058823529412, 90.51975308641977, 0], [195, 95.6
7058823529412, 90.51975308641977, 0], [196, 95.7176470588235
3, 90.51975308641977, 0], [197, 95.67058823529412, 90.519753
08641977, 0], [198, 95.67058823529412, 90.51975308641977, 0]
, [199, 96.47058823529412, 90.51975308641977, 0], [200, 95.6
7058823529412, 90.51975308641977, 0], [201, 95.670588235294l
2, 90.51975308641977, 0], [202, 95.67058823529412, 90.877777
77777777, 0], [203, 95.67058823529412, 90.87777777777777, 0]
, [204, 95.67058823529412, 90.87777777777777, 0], [205, 95.6
7058823529412, 90.87777777777777, 0], [206, 96.5428705882353
, 90.51975308641977, 0], [207, 95.67058823529412, 90.5197530
8641977, 0], [208, 96.18823529411765, 90.51975308641977, 0],
[209, 95.67058823529412, 90.51975308641977, 0], [210, 95.670
58823529412, 90.51975308641977, 0], [211, 96.32941176470588,
90.51975308641977, 0], [212, 95.67058823529412, 90.519753086
41977, 0], [213, 95.67058823529412, 90.51975308641977, 0], [
214, 95.67058823529412, 90.51975308641977, 0], [215, 95.6705
8823529412, 90.51975308641977, 0], [216, 96.56470588235294,
90.51975308641977, 0], [217, 95.67058823529412, 90.519753086
41977, 0], [218, 96.56470588235294, 90.51975308641977, 0], [
219, 95.67058823529412, 90.51975308641977, 0], [220, 95.6705
8823529412, 90.51975308641977, 0], [221, 95.67058823529412,
90.51975308641977, 0], [222, 95.67058823529412, 90.519753086
41977, 0], [223, 95.67058823529412, 90.51975308641977, 0], [
224, 95.67058823529412, 90.51975308641977, 0], [225, 95.6705
8823529412, 90.51975308641977, 0], [226, 95.67058823529412,
90.51975308641977, 0], [227, 95.67058823529412, 90.519753086
41977, 0], [228, 95.67058823529412, 90.51975308641977, 0], [
229, 95.67058823529412, 90.51975308641977, 0], [230, 95.6705
8823529412, 90.51975308641977, 0], [231, 95.67058823529412,
90.51975308641977, 0], [232, 95.67058823529412, 90.519753086
41977, 0], [233, 95.67058823529412, 90.51975308641977, 0], [
234, 95.84282352941176, 90.51975308641977, 0], [235, 95.8294
5882352941, 90.51975308641977, 0], [236, 96.7529411764705 9,
90.51975308641977, 0], [237, 95.67058823529412, 90.519753086
41977, 0], [238, 95.8136470588235 2, 90.51975308641977, 0], [
239, 95.67058823529412, 90.51975308641977, 0], [240, 95.6705
8823529412, 90.51975308641977, 0], [241, 95.67058823529412,
90.51975308641977, 0], [242, 95.67058823529412, 90.519753086
41977, 0], [243, 95.67058823529412, 90.51975308641977, 0], [
244, 95.67058823529412, 90.51975308641977, 0], [245, 95.6705
8823529412, 90.51975308641977, 0], [246, 96.2823529411764 7,
90.51975308641977, 0], [247, 95.67058823529412, 90.519753086
41977, 0], [248, 95.67058823529412, 90.51975308641977, 0], [

249, 95.67058823529412, 90.51975308641977, 0], [250, 95.6705
8823529412, 90.51975308641977, 0], [251, 95.67058823529412,
90.51975308641977, 0], [252, 95.67058823529412, 90.519753086
41977, 0], [253, 95.67058823529412, 90.51975308641977, 0], [
254, 96.04705882352941, 90.51975308641977, 0], [255, 96.7058
8235294117, 90.51975308641977, 0], [256, 95.67058823529412,
90.51975308641977, 0], [257, 95.85882352941177, 90.519753086
41977, 0], [258, 95.85882352941177, 90.51975308641977, 0], [
259, 95.67058823529412, 90.51975308641977, 0], [260, 96.8941
1764705882, 92.15432098765433, 0], [261, 95.67058823529412,
90.51975308641977, 0], [262, 95.67058823529412, 90.519753086
41977, 0], [263, 95.67058823529412, 90.57530864197531, 0], [
264, 96.28235294117647, 90.57530864197531, 0], [265, 95.9058
8235294118, 90.57530864197531, 0], [266, 95.67058823529412,
90.57530864197531, 0], [267, 95.67058823529412, 90.519753086
41977, 0], [268, 95.67058823529412, 90.51975308641977, 0], [
269, 95.81176470588235, 90.81111111111111, 0], [270, 96.04705
882352941, 90.81111111111111, 0], [271, 95.67058823529412, 90
.81111111111111, 0], [272, 95.67058823529412, 90.811111111111
1, 0], [273, 95.86070588235293, 90.81111111111111, 0], [274,
95.67058823529412, 90.81111111111111, 0], [275, 95.6705882352
9412, 90.81111111111111, 0], [276, 96.14117647058823, 91.1370
3703703703, 0], [277, 95.67058823529412, 90.51975308641977,
0], [278, 95.67058823529412, 90.51975308641977, 0], [279, 95
.67058823529412, 90.51975308641977, 0], [280, 95.67058823529
412, 90.51975308641977, 0], [281, 95.67058823529412, 90.5197
5308641977, 0], [282, 95.67058823529412, 90.51975308641977,
0], [283, 95.67058823529412, 90.51975308641977, 0], [284, 95
.67058823529412, 90.51975308641977, 0], [285, 95.67058823529
412, 90.9753086419753, 0], [286, 95.67058823529412, 90.51975
308641977, 0], [287, 95.67058823529412, 90.51975308641977, 0
], [288, 95.67058823529412, 90.51975308641977, 0], [289, 95.
67058823529412, 90.51975308641977, 0], [290, 95.670588235294
12, 90.51975308641977, 0], [291, 96.89411764705882, 90.51975
308641977, 0], [292, 95.67058823529412, 90.51975308641977, 0
], [293, 95.67058823529412, 90.51975308641977, 0], [294, 95.
67058823529412, 90.51975308641977, 0], [295, 95.670588235294
12, 90.51975308641977, 0], [296, 95.67058823529412, 90.51975
308641977, 0], [297, 95.67058823529412, 90.51975308641977, 0
], [298, 95.67058823529412, 90.51975308641977, 0], [299, 95.
67058823529412, 90.51975308641977, 0], [300, 95.670588235294
12, 90.51975308641977, 0], [301, 95.67058823529412, 90.51975
308641977, 0], [302, 95.67058823529412, 90.51975308641977, 0
], [303, 95.67058823529412, 90.51975308641977, 0], [304, 95.
67058823529412, 90.51975308641977, 0], [305, 95.670588235294
12, 90.51975308641977, 0], [306, 95.67058823529412, 90.51975
308641977, 0], [307, 95.67058823529412, 90.51975308641977, 0
], [308, 95.67058823529412, 90.51975308641977, 0], [309, 95.
67058823529412, 90.51975308641977, 0], [310, 95.670588235294
12, 90.51975308641977, 0], [311, 95.67058823529412, 90.51975

308641977, 0], [312, 95.67058823529412, 90.51975308641977, 0
], [313, 95.67058823529412, 90.51975308641977, 0], [314, 95.
67058823529412, 91.84074074074076, 0], [315, 95.670588235294
12, 90.51975308641977, 0], [316, 95.90588235294118, 90.51975
308641977, 0], [317, 95.67058823529412, 90.51975308641977, 0
], [318, 95.67058823529412, 90.51975308641977, 0], [319, 95.
67058823529412, 90.51975308641977, 0], [320, 95.670588235294
12, 90.51975308641977, 0], [321, 95.67058823529412, 90.51975
308641977, 0], [322, 95.67058823529412, 90.51975308641977, 0
], [323, 95.67058823529412, 90.51975308641977, 0], [324, 95.
67058823529412, 90.51975308641977, 0], [325, 95.670588235294
12, 90.51975308641977, 0], [326, 95.67058823529412, 90.51975
308641977, 0], [327, 95.67058823529412, 90.51975308641977, 0
], [328, 96.0, 90.51975308641977, 0], [329, 95.670588235294l
2, 90.51975308641977, 0], [330, 95.67058823529412, 90.519753
08641977, 0], [331, 95.67058823529412, 90.51975308641977, 0]
, [332, 97.03529411764707, 91.8259259259259 2, 0], [333, 95.6
7058823529412, 90.51975308641977, 0], [334, 95.670588235294l
2, 90.85308641975307, 0], [335, 95.67058823529412, 90.519753
08641977, 0], [336, 95.67058823529412, 90.51975308641977, 0]
, [337, 95.67058823529412, 90.51975308641977, 0], [338, 95.6
7058823529412, 90.51975308641977, 0], [339, 95.670588235294l
2, 90.51975308641977, 0], [340, 95.67058823529412, 90.519753
08641977, 0], [341, 95.67058823529412, 90.51975308641977, 0]
, [342, 95.67058823529412, 90.51975308641977, 0], [343, 95.6
7058823529412, 90.51975308641977, 0], [344, 95.670588235294l
2, 90.51975308641977, 0], [345, 95.67058823529412, 90.519753
08641977, 0], [346, 95.67058823529412, 90.51975308641977, 0]
, [347, 95.81176470588235, 90.51975308641977, 0], [348, 95.8
1364705882352, 90.51975308641977, 0], [349, 95.813647058823 5
2, 90.51975308641977, 0], [350, 95.81364705882352, 90.519753
08641977, 0], [351, 95.81364705882352, 90.51975308641977, 0]
, [352, 95.81364705882352, 90.51975308641977, 0], [353, 95.8
1364705882352, 90.51975308641977, 0], [354, 95.8136470588235
2, 90.51975308641977, 0], [355, 96.37647058823529, 90.519753
08641977, 0], [356, 95.76470588235294, 90.51975308641977, 0]
, [357, 95.67058823529412, 90.51975308641977, 0], [358, 95.6
7058823529412, 90.51975308641977, 0], [359, 95.6705882352941
2, 90.51975308641977, 0], [360, 95.67058823529412, 90.519753
08641977, 0], [361, 95.67058823529412, 90.51975308641977, 0]
, [362, 95.67058823529412, 90.51975308641977, 0], [363, 95.7
12, 90.51975308641977, 0], [364, 95.712, 90.51975308641977,
0], [365, 95.67058823529412, 90.51975308641977, 0], [366, 95
.67058823529412, 90.51975308641977, 0], [367, 95.6705882352
9
412, 90.51975308641977, 0], [368, 95.67058823529412, 90.5197
5308641977, 0], [369, 95.67058823529412, 90.51975308641977,
0], [370, 95.67058823529412, 90.51975308641977, 0], [371, 95
.67058823529412, 90.51975308641977, 0], [372, 95.6705882352
9
412, 90.51975308641977, 0], [373, 95.67058823529412, 90.5197
5308641977, 0], [374, 96.28235294117647, 90.9938271604938 2,

0], [375, 95.67058823529412, 90.51975308641977, 0], [376, 95
.67058823529412, 90.51975308641977, 0], [377, 95.7647058823529
294, 90.51975308641977, 0], [378, 95.67058823529412, 90.5197
5308641977, 0], [379, 95.67058823529412, 90.51975308641977,
0], [380, 95.67058823529412, 90.51975308641977, 0], [381, 95
.67058823529412, 90.51975308641977, 0], [382, 95.67058823529
412, 90.51975308641977, 0], [383, 95.67058823529412, 90.5197
5308641977, 0], [384, 95.67058823529412, 90.51975308641977,
0], [385, 95.67058823529412, 90.51975308641977, 0], [386, 96
.2009411764706, 90.51975308641977, 0], [387, 95.905882352941
18, 90.51975308641977, 0], [388, 95.67058823529412, 90.51975
308641977, 0], [389, 95.67058823529412, 90.51975308641977, 0
], [390, 95.67058823529412, 90.51975308641977, 0], [391, 95.
67058823529412, 90.51975308641977, 0], [392, 95.670588235294
12, 90.51975308641977, 0], [393, 95.67058823529412, 90.51975
308641977, 0], [394, 95.67058823529412, 90.51975308641977, 0
], [395, 95.67058823529412, 90.51975308641977, 0], [396, 95.
67058823529412, 90.51975308641977, 0], [397, 95.670588235294
12, 90.51975308641977, 0], [398, 95.67058823529412, 90.51975
308641977, 0], [399, 95.67058823529412, 90.51975308641977, 0
], [400, 95.67058823529412, 90.51975308641977, 0], [401, 95.
67058823529412, 90.51975308641977, 0], [402, 95.670588235294
12, 90.51975308641977, 0], [403, 95.67058823529412, 90.51975
308641977, 0], [404, 96.37647058823529, 90.67283950617283, 0
], [405, 95.67058823529412, 90.51975308641977, 0], [406, 95.
67058823529412, 90.51975308641977, 0], [407, 95.670588235294
12, 90.51975308641977, 0], [408, 95.67058823529412, 90.51975
308641977, 0], [409, 95.67058823529412, 90.51975308641977, 0
], [410, 95.90588235294118, 90.51975308641977, 0], [411, 95.
67058823529412, 90.51975308641977, 0], [412, 95.670588235294
12, 90.51975308641977, 0], [413, 96.04413647058823, 90.51975
308641977, 0], [414, 95.67058823529412, 90.51975308641977, 0
], [415, 96.32941176470588, 90.51975308641977, 0], [416, 95.
67058823529412, 90.51975308641977, 0], [417, 96.329411764705
88, 90.51975308641977, 0], [418, 95.67058823529412, 90.51975
308641977, 0], [419, 95.67058823529412, 90.51975308641977, 0
], [420, 95.67058823529412, 90.51975308641977, 0], [421, 96.
4235294117647, 90.51975308641977, 0], [422, 95.6705882352941
2, 90.51975308641977, 0], [423, 97.45882352941176, 90.519753
08641977, 0], [424, 95.67058823529412, 90.51975308641977, 0]
, [425, 95.67058823529412, 90.51975308641977, 0], [426, 95.6
7058823529412, 90.51975308641977, 0], [427, 95.6705882352941
2, 90.51975308641977, 0], [428, 95.67058823529412, 90.519753
08641977, 0], [429, 95.67058823529412, 90.51975308641977, 0]
, [430, 95.67058823529412, 90.51975308641977, 0], [431, 95.6
7058823529412, 90.51975308641977, 0], [432, 95.9311058823529
4, 90.51975308641977, 0], [433, 95.67058823529412, 91.420987
65432098, 0], [434, 95.67058823529412, 90.51975308641977, 0]
, [435, 95.67058823529412, 90.51975308641977, 0], [436, 95.6
7058823529412, 90.51975308641977, 0], [437, 95.6705882352941

2, 90.51975308641977, 0], [438, 95.67058823529412, 90.519753
08641977, 0], [439, 95.67058823529412, 90.51975308641977, 0]
, [440, 95.67058823529412, 90.51975308641977, 0], [441, 95.6
7058823529412, 90.51975308641977, 0], [442, 95.6705882352941
2, 90.51975308641977, 0], [443, 95.67058823529412, 90.519753
08641977, 0], [444, 95.97063529411764, 90.51975308641977, 0]
, [445, 95.67058823529412, 90.51975308641977, 0], [446, 95.9
5294117647059, 90.51975308641977, 0], [447, 95.7647058823529
4, 90.51975308641977, 0], [448, 95.67058823529412, 90.519753
08641977, 0], [449, 95.67058823529412, 90.51975308641977, 0]
, [450, 96.32941176470588, 90.51975308641977, 0], [451, 96.9
4117647058823, 92.06543209876543, 0], [452, 95.6705882352941
2, 90.51975308641977, 0], [453, 97.1764705882353, 90.5864197
5308642, 0], [454, 96.8, 90.51975308641977, 0], [455, 96.188
23529411765, 90.51975308641977, 0], [456, 95.67058823529412,
90.51975308641977, 0], [457, 96.14117647058823, 90.519753086
41977, 0], [458, 95.67058823529412, 90.51975308641977, 0], [
459, 95.67058823529412, 90.51975308641977, 0], [460, 95.6705
8823529412, 90.51975308641977, 0], [461, 95.67058823529412,
90.51975308641977, 0], [462, 96.0, 90.51975308641977, 0], [4
63, 95.71764705882353, 90.51975308641977, 0], [464, 95.67058
823529412, 90.51975308641977, 0], [465, 95.67058823529412, 9
0.51975308641977, 0], [466, 95.67058823529412, 90.5197530864
1977, 0], [467, 95.67058823529412, 90.51975308641977, 0], [4
68, 95.67058823529412, 90.51975308641977, 0], [469, 97.45882
352941176, 90.82098765432097, 0], [470, 96.47058823529412, 9
0.70493827160495, 0], [471, 96.18823529411765, 90.7049382716
0495, 0], [472, 96.47058823529412, 90.70493827160495, 0], [4
73, 96.47058823529412, 90.70493827160495, 0], [474, 96.47058
823529412, 90.51975308641977, 0], [475, 95.67058823529412, 9
0.51975308641977, 0], [476, 95.67058823529412, 90.5197530864
1977, 0], [477, 96.18823529411765, 90.51975308641977, 0], [4
78, 95.67058823529412, 90.51975308641977, 0], [479, 95.67058
823529412, 90.51975308641977, 0], [480, 95.67058823529412, 9
0.51975308641977, 0], [481, 95.76470588235294, 90.5197530864
1977, 0], [482, 95.67058823529412, 90.51975308641977, 0], [4
83, 95.67058823529412, 90.51975308641977, 0], [484, 95.67058
823529412, 90.51975308641977, 0], [485, 95.67058823529412, 9
0.51975308641977, 0], [486, 96.65882352941176, 91.9074074074
0742, 0], [487, 95.67058823529412, 90.51975308641977, 0], [4
88, 95.67058823529412, 90.51975308641977, 0], [489, 95.67058
823529412, 90.51975308641977, 0], [490, 95.67058823529412, 9
0.51975308641977, 0], [491, 95.67058823529412, 90.5197530864
1977, 0], [492, 95.67058823529412, 90.51975308641977, 0], [4
93, 96.56470588235294, 90.51975308641977, 0], [494, 96.8, 90
.51975308641977, 0], [495, 95.67058823529412, 90.51975308641
977, 0], [496, 96.18823529411765, 91.41358024691358, 0], [49
7, 95.67058823529412, 90.51975308641977, 0], [498, 95.670588
23529412, 90.51975308641977, 0], [499, 95.67058823529412, 90
.85308641975307, 0], [500, 96.37647058823529, 90.51975308641

977, 0], [501, 95.67058823529412, 90.51975308641977, 0], [50
2, 95.67058823529412, 91.16666666666667, 0], [503, 95.670588
23529412, 90.51975308641977, 0], [504, 95.67058823529412, 90
.51975308641977, 0], [505, 95.67058823529412, 90.51975308641
977, 0], [506, 95.67058823529412, 90.51975308641977, 0], [50
7, 96.4178823529411‌9, 90.51975308641977, 0], [508, 95.670588
23529412, 90.51975308641977, 0], [509, 95.75905882352941, 90
.51975308641977, 0], [510, 97.17082352941176, 90.51975308641
977, 0], [511, 95.67058823529412, 90.51975308641977, 0], [51
2, 95.73534117647058, 90.51975308641977, 0], [513, 96.188235
29411765, 90.51975308641977, 0], [514, 95.67058823529412, 90
.51975308641977, 0], [515, 95.67058823529412, 91.47530864197
532, 0], [516, 95.71764705882353, 90.51975308641977, 0], [51
7, 95.67058823529412, 90.51975308641977, 0], [518, 96.329411
76470588, 90.51975308641977, 0], [519, 95.67058823529412, 90
.51975308641977, 0], [520, 95.67058823529412, 90.51975308641
977, 0], [521, 95.67058823529412, 90.51975308641977, 0], [52
2, 96.03294117647059, 90.51975308641977, 0], [523, 95.670588
23529412, 90.51975308641977, 0], [524, 95.67058823529412, 90
.51975308641977, 0], [525, 95.67058823529412, 90.51975308641
977, 0], [526, 96.09411764705882, 90.51975308641977, 0], [52
7, 95.67058823529412, 90.51975308641977, 0], [528, 95.670588
23529412, 90.92716049382716, 0], [529, 95.95294117647059, 90
.51975308641977, 0], [530, 95.67058823529412, 90.61111111111
111, 0], [531, 95.67058823529412, 90.51975308641977, 0], [53
2, 95.67058823529412, 90.51975308641977, 0], [533, 95.717647
05882353, 90.51975308641977, 0], [534, 95.67058823529412, 90
.51975308641977, 0], [535, 96.23529411764706, 90.51975308641
977, 0], [536, 95.67058823529412, 90.51975308641977, 0], [53
7, 95.67058823529412, 90.51975308641977, 0], [538, 95.670588
23529412, 90.51975308641977, 0], [539, 95.67058823529412, 90
.51975308641977, 0], [540, 95.67058823529412, 90.84567901234
567, 0], [541, 95.67058823529412, 90.51975308641977, 0], [54
2, 95.81176470588235, 90.51975308641977, 0], [543, 95.670588
23529412, 90.51975308641977, 0], [544, 96.14117647058823, 90
.51975308641977, 0], [545, 95.67058823529412, 90.51975308641
977, 0], [546, 95.67058823529412, 90.51975308641977, 0], [54
7, 96.56470588235294, 90.51975308641977, 0], [548, 95.670588
23529412, 90.51975308641977, 0], [549, 95.67058823529412, 90
.51975308641977, 0], [550, 96.61176470588235, 90.51975308641
977, 0], [551, 95.67058823529412, 90.51975308641977, 0], [55
2, 95.67058823529412, 90.51975308641977, 0], [553, 95.670588
23529412, 90.51975308641977, 0], [554, 95.67058823529412, 90
.51975308641977, 0], [555, 95.67058823529412, 90.51975308641
977, 0], [556, 96.89599999999999, 90.51975308641977, 0], [55
7, 95.67058823529412, 90.51975308641977, 0], [558, 95.670588
23529412, 90.51975308641977, 0], [559, 95.74287058823529, 91
.9024691358024‌8, 0], [560, 96.47058823529412, 90.8950617283‌9
506, 0], [561, 95.67058823529412, 90.51975308641977, 0], [56
2, 95.67058823529412, 90.51975308641977, 0], [563, 95.670588

23529412, 90.51975308641977, 0], [564, 95.67058823529412, 90
.51975308641977, 0], [565, 95.67058823529412, 90.51975308641
977, 0], [566, 95.67058823529412, 90.51975308641977, 0], [56
7, 95.67058823529412, 90.51975308641977, 0], [568, 95.670588
23529412, 90.51975308641977, 0], [569, 95.67058823529412, 90
.51975308641977, 0], [570, 95.67058823529412, 90.51975308641
977, 0], [571, 95.67058823529412, 90.51975308641977, 0], [57
2, 95.67058823529412, 90.51975308641977, 0], [573, 95.670588
23529412, 90.51975308641977, 0], [574, 95.67058823529412, 90
.51975308641977, 0], [575, 95.67058823529412, 90.51975308641
977, 0], [576, 95.67058823529412, 90.51975308641977, 0], [57
7, 95.85882352941177, 90.51975308641977, 0], [578, 95.670588
23529412, 90.51975308641977, 0], [579, 95.67058823529412, 90
.51975308641977, 0], [580, 95.67058823529412, 90.51975308641
977, 0], [581, 95.67058823529412, 90.51975308641977, 0], [58
2, 95.90588235294118, 90.51975308641977, 0], [583, 95.670588
23529412, 90.51975308641977, 0], [584, 95.81176470588235, 90
.51975308641977, 0], [585, 95.67058823529412, 90.51975308641
977, 0], [586, 96.47058823529412, 90.89506172839506, 0], [58
7, 95.67058823529412, 90.51975308641977, 0], [588, 95.670588
23529412, 90.51975308641977, 0], [589, 95.67058823529412, 90
.51975308641977, 0], [590, 95.67058823529412, 90.51975308641
977, 0], [591, 95.67058823529412, 90.51975308641977, 0], [59
2, 97.03529411764707, 91.82592592592592, 0], [593, 95.811764
70588235, 90.51975308641977, 0], [594, 95.81176470588235, 90
.51975308641977, 0], [595, 95.81176470588235, 90.51975308641
977, 0], [596, 96.28235294117647, 90.51975308641977, 0], [59
7, 96.28235294117647, 91.49506172839504, 0], [598, 96.282352
94117647, 90.51975308641977, 0], [599, 96.28235294117647, 90
.51975308641977, 0], [600, 95.81176470588235, 90.51975308641
977, 0], [601, 95.81176470588235, 90.51975308641977, 0], [60
2, 95.81176470588235, 90.51975308641977, 0], [603, 95.811764
70588235, 90.51975308641977, 0], [604, 95.81176470588235, 90
.51975308641977, 0], [605, 95.81176470588235, 90.51975308641
977, 0], [606, 95.81176470588235, 90.51975308641977, 0], [60
7, 95.81176470588235, 90.51975308641977, 0], [608, 95.811764
70588235, 90.51975308641977, 0], [609, 95.81176470588235, 90
.51975308641977, 0], [610, 95.81176470588235, 90.51975308641
977, 0], [611, 95.81176470588235, 90.51975308641977, 0], [61
2, 95.81176470588235, 90.51975308641977, 0], [613, 95.811764
70588235, 90.51975308641977, 0], [614, 95.81176470588235, 90
.51975308641977, 0], [615, 95.81176470588235, 90.51975308641
977, 0], [616, 95.81176470588235, 90.51975308641977, 0], [61
7, 95.81176470588235, 90.51975308641977, 0], [618, 96.423529
4117647, 90.51975308641977, 0], [619, 95.81176470588235, 90.
67283950617283, 0], [620, 95.81176470588235, 90.519753086419
77, 0], [621, 95.81176470588235, 90.51975308641977, 0], [622
, 95.81176470588235, 90.51975308641977, 0], [623, 95.8117647
0588235, 90.51975308641977, 0], [624, 95.81176470588235, 90.
5197530864197, 0], [625, 95.81176470588235, 90.519753086419

77, 0], [626, 95.81176470588235, 90.51975308641977, 0], [627
, 95.81176470588235, 90.51975308641977, 0], [628, 95.8117647
0588235, 90.51975308641977, 0], [629, 95.81176470588235, 90.
51975308641977, 0], [630, 95.81176470588235, 90.519753086419
77, 0], [631, 95.81176470588235, 90.51975308641977, 0], [632
, 95.81176470588235, 90.51975308641977, 0], [633, 95.8117647
0588235, 90.51975308641977, 0], [634, 95.81176470588235, 90.
51975308641977, 0], [635, 95.81176470588235, 90.519753086419
77, 0], [636, 96.17411764705882, 90.51975308641977, 0], [637
, 95.81176470588235, 90.51975308641977, 0], [638, 95.8117647
0588235, 90.51975308641977, 0], [639, 95.81176470588235, 90.
51975308641977, 0], [640, 95.81176470588235, 90.519753086419
77, 0], [641, 95.81176470588235, 90.51975308641977, 0], [642
, 95.81176470588235, 90.51975308641977, 0], [643, 95.8117647
0588235, 90.51975308641977, 0], [644, 95.81176470588235, 90.
51975308641977, 0], [645, 95.81176470588235, 90.519753086419
77, 0], [646, 95.81176470588235, 90.51975308641977, 0], [647
, 95.81176470588235, 90.51975308641977, 0], [648, 96.1383529
4117648, 90.51975308641977, 0], [649, 95.81176470588235, 90.
51975308641977, 0], [650, 95.81176470588235, 90.519753086419
77, 0], [651, 95.81176470588235, 90.51975308641977, 0], [652
, 96.47058823529412, 90.79135802469138, 0], [653, 95.8117647
0588235, 90.79135802469138, 0], [654, 95.81176470588235, 89.
82839506172839, 0], [655, 95.81176470588235, 90.260493827160
49, 0], [656, 95.81176470588235, 89.82839506172839, 0], [657
, 96.32941176470588, 90.29012345679011, 0], [658, 96.3294117
6470588, 90.29012345679011, 0], [659, 96.8, 90.2901234567901
1, 0], [660, 96.32941176470588, 90.29012345679011, 0], [661,
96.32941176470588, 90.29012345679011, 0], [662, 96.329411764
70588, 90.29012345679011, 0], [663, 95.81176470588235, 89.82
839506172839, 0], [664, 95.81176470588235, 89.82839506172839
, 0], [665, 95.81176470588235, 89.82839506172839, 0], [666,
96.23529411764706, 89.82839506172839, 0], [667, 95.811764705
88235, 89.82839506172839, 0], [668, 96.4235294117647, 89.828
39506172839, 0], [669, 95.81176470588235, 89.82839506172839,
0], [670, 95.81176470588235, 89.88395061728394, 0], [671, 95
.81176470588235, 89.82839506172839, 0], [672, 95.81176470588
235, 89.82839506172839, 0], [673, 95.81176470588235, 89.8283
9506172839, 0], [674, 95.81176470588235, 89.82839506172839,
0], [675, 96.8, 89.82839506172839, 0], [676, 95.811764705882
35, 89.82839506172839, 0], [677, 96.61176470588235, 90.80370
37037037, 0], [678, 95.81176470588235, 90.8037037037037, 0],
[679, 95.81176470588235, 89.82839506172839, 0], [680, 95.811
76470588235, 89.82839506172839, 0], [681, 95.81176470588235,
89.82839506172839, 0], [682, 95.81176470588235, 89.828395061
72839, 0], [683, 95.81176470588235, 89.82839506172839, 0], [
684, 95.85317647058824, 89.82839506172839, 0], [685, 95.8117
6470588235, 89.82839506172839, 0], [686, 95.81176470588235,
89.82839506172839, 0], [687, 95.81176470588235, 89.828395061
72839, 0], [688, 95.81176470588235, 89.82839506172839, 0], [

689, 95.8117647058235, 89.82839506172839, 0], [690, 95.8117
6470588235, 89.82839506172839, 0], [691, 95.81176470588235,
89.82839506172839, 0], [692, 96.18823529411765, 89.828395061
72839, 0], [693, 95.81176470588235, 89.82839506172839, 0], [
694, 95.81176470588235, 89.82839506172839, 0], [695, 95.8117
6470588235, 89.82839506172839, 0], [696, 95.81176470588235,
89.82839506172839, 0], [697, 95.81176470588235, 89.828395061
72839, 0], [698, 95.81176470588235, 89.82839506172839, 0], [
699, 95.81176470588235, 89.82839506172839, 0], [700, 95.8117
6470588235, 89.82839506172839, 0], [701, 95.81176470588235,
89.82839506172839, 0], [702, 95.81176470588235, 89.828395061
72839, 0], [703, 96.0, 89.82839506172839, 0], [704, 96.0, 89
.82839506172839, 0], [705, 95.81176470588235, 89.82839506172
839, 0], [706, 95.81176470588235, 89.82839506172839, 0], [70
7, 95.81176470588235, 89.82839506172839, 0], [708, 95.811764
70588235, 89.82839506172839, 0], [709, 95.81176470588235, 89
.82839506172839, 0], [710, 96.04705882352941, 89.82839506172
839, 0], [711, 96.18823529411765, 90.0962962962963, 0], [712
, 95.81176470588235, 89.82839506172839, 0], [713, 97.0823529
4117648, 91.37407407407409, 0], [714, 95.81176470588235, 89.
82839506172839, 0], [715, 95.88404705882353, 90.129629629629
62, 0], [716, 96.51764705882353, 89.82839506172839, 0], [717
, 96.51764705882353, 89.82839506172839, 0], [718, 95.8117647
0588235, 90.6024691358025, 0], [719, 95.81176470588235, 89.8
2839506172839, 0], [720, 95.95294117647059, 89.8283950617283
9, 0], [721, 95.95294117647059, 89.82839506172839, 0], [722,
95.81176470588235, 89.82839506172839, 0], [723, 95.811764705
88235, 89.82839506172839, 0], [724, 95.81176470588235, 89.82
839506172839, 0], [725, 95.81176470588235, 89.82839506172839
, 0], [726, 95.81176470588235, 90.00617283950618, 0], [727,
95.81176470588235, 89.82839506172839, 0], [728, 95.811764705
88235, 89.82839506172839, 0], [729, 96.14117647058823, 89.82
839506172839, 0], [730, 95.81176470588235, 89.82839506172839
, 0], [731, 95.81176470588235, 89.82839506172839, 0], [732,
95.81176470588235, 89.82839506172839, 0], [733, 95.811764705
88235, 89.82839506172839, 0], [734, 95.81176470588235, 89.82
839506172839, 0], [735, 96.18823529411765, 89.82839506172839
, 0], [736, 95.81176470588235, 89.82839506172839, 0], [737,
95.81176470588235, 89.82839506172839, 0], [738, 95.811764705
88235, 89.82839506172839, 0], [739, 95.81176470588235, 89.82
839506172839, 0], [740, 95.81176470588235, 89.82839506172839
, 0], [741, 95.81176470588235, 89.82839506172839, 0], [742,
95.81176470588235, 89.82839506172839, 0], [743, 95.811764705
88235, 89.82839506172839, 0], [744, 95.81176470588235, 89.82
839506172839, 0], [745, 96.37647058823529, 90.30246913580248
, 0], [746, 96.32941176470588, 89.82839506172839, 0], [747,
95.81176470588235, 89.82839506172839, 0], [748, 95.811764705
88235, 89.82839506172839, 0], [749, 96.8, 89.82839506172839,
0], [750, 95.81176470588235, 89.82839506172839, 0], [751, 95
.81176470588235, 89.82839506172839, 0], [752, 95.81176470588

235, 89.82839506172839, 0], [753, 95.81176470588235, 89.8283
9506172839, 0], [754, 95.81176470588235, 89.82839506172839,
0], [755, 96.32941176470588, 89.82839506172839, 0], [756, 96
.18823529411765, 89.82839506172839, 0], [757, 95.81176470588
235, 89.82839506172839, 0], [758, 95.81176470588235, 89.8283
9506172839, 0], [759, 95.81176470588235, 90.75432098765431,
0], [760, 95.81176470588235, 89.82839506172839, 0], [761, 95
.81176470588235, 89.82839506172839, 0], [762, 95.81176470588
235, 89.82839506172839, 0], [763, 95.81176470588235, 89.8283
9506172839, 0], [764, 95.89035294117647, 89.82839506172839,
0], [765, 95.81176470588235, 89.82839506172839, 0], [766, 95
.81176470588235, 89.82839506172839, 0], [767, 95.81176470588
235, 89.82839506172839, 0], [768, 95.81176470588235, 89.8283
9506172839, 0], [769, 95.81176470588235, 89.82839506172839,
0], [770, 95.81176470588235, 89.82839506172839, 0], [771, 95
.81176470588235, 89.82839506172839, 0], [772, 96.84705882352
941, 90.78395061728394, 0], [773, 95.81176470588235, 89.8283
9506172839, 0], [774, 95.81176470588235, 89.82839506172839,
0], [775, 95.81176470588235, 89.82839506172839, 0], [776, 95
.81176470588235, 89.82839506172839, 0], [777, 95.81176470588
235, 89.82839506172839, 0], [778, 95.81176470588235, 89.8283
9506172839, 0], [779, 95.81176470588235, 89.82839506172839,
0], [780, 95.81176470588235, 89.82839506172839, 0], [781, 95
.81176470588235, 89.82839506172839, 0], [782, 95.81176470588
235, 89.82839506172839, 0], [783, 96.47058823529412, 90.4938
2716049382, 0], [784, 96.47058823529412, 90.49382716049382,
0], [785, 95.81176470588235, 89.82839506172839, 0], [786, 95
.81176470588235, 89.82839506172839, 0], [787, 95.81176470588
235, 89.82839506172839, 0], [788, 95.90588235294118, 89.8283
9506172839, 0], [789, 95.81176470588235, 89.82839506172839,
0], [790, 96.23529411764706, 89.82839506172839, 0], [791, 95
.81176470588235, 89.82839506172839, 0], [792, 95.81176470588
235, 89.82839506172839, 0], [793, 95.81176470588235, 89.8283
9506172839, 0], [794, 95.81176470588235, 89.82839506172839,
0], [795, 95.81176470588235, 89.82839506172839, 0], [796, 95
.81176470588235, 89.82839506172839, 0], [797, 95.81176470588
235, 89.82839506172839, 0], [798, 95.81176470588235, 89.8283
9506172839, 0], [799, 97.41176470588235, 89.82839506172839,
0], [800, 95.81176470588235, 89.82839506172839, 0], [801, 96
.8, 89.87777777777778, 0], [802, 95.81176470588235, 89.82839
506172839, 0], [803, 95.81176470588235, 89.82839506172839, 0
], [804, 96.94117647058823, 90.84567901234567, 0], [805, 95.
81176470588235, 89.82839506172839, 0], [806, 95.811764705882
35, 89.82839506172839, 0], [807, 96.14117647058823, 89.82839
506172839, 0], [808, 95.81176470588235, 89.82839506172839, 0
], [809, 95.81176470588235, 89.82839506172839, 0], [810, 95.
81176470588235, 89.82839506172839, 0], [811, 95.811764705882
35, 89.82839506172839, 0], [812, 95.81176470588235, 89.82839
506172839, 0], [813, 95.81176470588235, 89.82839506172839, 0
], [814, 96.94117647058823, 90.64814814814814, 0], [815, 95.

86070588235293, 89.82839506172839, 0], [816, 95.811764705882
35, 90.63086419753084, 0], [817, 95.81176470588235, 89.82839
506172839, 0], [818, 95.81176470588235, 89.82839506172839, 0
], [819, 95.81176470588235, 89.82839506172839, 0], [820, 95.
81176470588235, 89.82839506172839, 0], [821, 96.563294117647
08, 89.82839506172839, 0], [822, 95.81176470588235, 89.82839
506172839, 0], [823, 95.81176470588235, 89.82839506172839, 0
], [824, 95.81176470588235, 90.58148148148149, 0], [825, 95.
81176470588235, 89.82839506172839, 0], [826, 95.811764705882
35, 89.82839506172839, 0], [827, 95.81176470588235, 89.82839
506172839, 0], [828, 95.81176470588235, 89.82839506172839, 0
], [829, 95.81176470588235, 89.82839506172839, 0], [830, 96.
47058823529412, 89.82839506172839, 0], [831, 95.811764705882
35, 89.82839506172839, 0], [832, 95.81176470588235, 89.82839
506172839, 0], [833, 95.81176470588235, 89.82839506172839, 0
], [834, 95.81176470588235, 89.82839506172839, 0], [835, 95.
81176470588235, 89.82839506172839, 0], [836, 95.811764705882
35, 89.82839506172839, 0], [837, 95.81176470588235, 89.82839
506172839, 0], [838, 95.81176470588235, 89.82839506172839, 0
], [839, 95.81176470588235, 89.82839506172839, 0], [840, 96.
70588235294117, 89.82839506172839, 0], [841, 95.811764705882
35, 89.82839506172839, 0], [842, 95.81176470588235, 89.82839
506172839, 0], [843, 95.81176470588235, 89.82839506172839, 0
], [844, 95.81176470588235, 89.82839506172839, 0], [845, 95.
81176470588235, 89.82839506172839, 0], [846, 95.811764705882
35, 89.82839506172839, 0], [847, 95.81176470588235, 89.82839
506172839, 0], [848, 95.81176470588235, 89.82839506172839, 0
], [849, 95.81176470588235, 89.82839506172839, 0], [850, 95.
81176470588235, 89.82839506172839, 0], [851, 95.811764705882
35, 89.82839506172839, 0], [852, 95.81176470588235, 89.82839
506172839, 0], [853, 95.81176470588235, 89.82839506172839, 0
], [854, 96.65882352941176, 89.82839506172839, 0], [855, 95.
81176470588235, 89.82839506172839, 0], [856, 95.811764705882
35, 89.82839506172839, 0], [857, 96.04705882352941, 89.82839
506172839, 0], [858, 95.81176470588235, 89.82839506172839, 0
], [859, 95.81176470588235, 89.82839506172839, 0], [860, 95.
94729411764706, 89.82839506172839, 0], [861, 95.811764705882
35, 89.82839506172839, 0], [862, 96.4235294117647, 89.828395
06172839, 0], [863, 95.81176470588235, 89.82839506172839, 0]
, [864, 95.81176470588235, 89.82839506172839, 0], [865, 95.8
1176470588235, 89.82839506172839, 0], [866, 95.8117647058823
5, 89.82839506172839, 0], [867, 95.81176470588235, 89.828395
06172839, 0], [868, 95.81176470588235, 89.82839506172839, 0]
, [869, 95.81176470588235, 89.82839506172839, 0], [870, 95.8
1176470588235, 89.82839506172839, 0], [871, 95.8117647058823
5, 89.82839506172839, 0], [872, 95.81176470588235, 89.828395
06172839, 0], [873, 96.37647058823529, 89.82839506172839, 0]
, [874, 95.81176470588235, 89.82839506172839, 0], [875, 95.8
1176470588235, 89.82839506172839, 0], [876, 95.8117647058823
5, 89.82839506172839, 0], [877, 95.81176470588235, 89.972839

50617285, 0], [878, 95.81176470588235, 89.82839506172839, 0]
, [879, 95.81176470588235, 89.82839506172839, 0], [880, 95.8
1176470588235, 89.82839506172839, 0], [881, 95.8117647058823
5, 89.82839506172839, 0], [882, 95.81176470588235, 89.828395
06172839, 0], [883, 95.81176470588235, 89.82839506172839, 0]
, [884, 95.81176470588235, 89.82839506172839, 0], [885, 95.8
1176470588235, 89.82839506172839, 0], [886, 95.8117647058823
5, 89.82839506172839, 0], [887, 95.81176470588235, 89.828395
06172839, 0], [888, 95.81176470588235, 90.22345679012345, 0]
, [889, 95.81176470588235, 90.22345679012345, 0], [890, 96.8
9411764705882, 89.82839506172839, 0], [891, 95.8117647058823
5, 89.82839506172839, 0], [892, 95.81176470588235, 89.828395
06172839, 0], [893, 96.94117647058823, 90.64814814814814, 0]
, [894, 95.81176470588235, 89.82839506172839, 0], [895, 95.8
1176470588235, 89.82839506172839, 0], [896, 95.8117647058823
5, 89.89012345679012, 0], [897, 95.81176470588235, 89.828395
06172839, 0], [898, 95.81176470588235, 89.82839506172839, 0]
, [899, 95.81176470588235, 89.82839506172839, 0], [900, 96.1
8823529411765, 89.82839506172839, 0], [901, 95.8117647058823
5, 89.82839506172839, 0], [902, 96.77176470588236, 89.956790
1234568, 0], [903, 95.81176470588235, 89.82839506172839, 0],
[904, 95.81176470588235, 89.82839506172839, 0], [905, 96.342
11764705883, 89.82839506172839, 0], [906, 95.81176470588235,
89.82839506172839, 0], [907, 95.81176470588235, 89.828395061
72839, 0], [908, 95.81176470588235, 89.82839506172839, 0], [
909, 95.81176470588235, 89.82839506172839, 0], [910, 95.8117
6470588235, 89.82839506172839, 0], [911, 95.81176470588235,
89.82839506172839, 0], [912, 95.81176470588235, 89.828395061
72839, 0], [913, 95.81176470588235, 89.82839506172839, 0], [
914, 95.81176470588235, 89.82839506172839, 0], [915, 95.8117
6470588235, 89.82839506172839, 0], [916, 95.82870588235295,
89.82839506172839, 0], [917, 95.81176470588235, 89.828395061
72839, 0], [918, 95.81176470588235, 89.82839506172839, 0], [
919, 95.81176470588235, 89.82839506172839, 0], [920, 95.8117
6470588235, 89.82839506172839, 0], [921, 95.81176470588235,
89.82839506172839, 0], [922, 95.81176470588235, 89.828395061
72839, 0], [923, 95.81176470588235, 89.82839506172839, 0], [
924, 95.81176470588235, 89.82839506172839, 0], [925, 96.3764
7058823529, 89.82839506172839, 0], [926, 95.81176470588235,
89.82839506172839, 0], [927, 96.04705882352941, 89.828395061
72839, 0], [928, 95.81176470588235, 89.82839506172839, 0], [
929, 95.81176470588235, 89.82839506172839, 0], [930, 95.8117
6470588235, 89.82839506172839, 0], [931, 95.81176470588235,
89.82839506172839, 0], [932, 95.81176470588235, 89.828395061
72839, 0], [933, 95.81176470588235, 89.82839506172839, 0], [
934, 95.81176470588235, 89.82839506172839, 0], [935, 95.8117
6470588235, 89.82839506172839, 0], [936, 95.81176470588235,
89.82839506172839, 0], [937, 95.81176470588235, 89.828395061
72839, 0], [938, 95.81176470588235, 89.82839506172839, 0], [
939, 95.81176470588235, 89.82839506172839, 0], [940, 96.9411

7647058823, 89.82839506172839, 0], [941, 95.81176470588235,
89.82839506172839, 0], [942, 95.95294117647059, 89.828395061
72839, 0], [943, 95.95294117647059, 89.82839506172839, 0], [
944, 95.95294117647059, 89.82839506172839, 0], [945, 95.8117
6470588235, 89.82839506172839, 0], [946, 95.81176470588235,
89.82839506172839, 0], [947, 95.81176470588235, 89.828395061
72839, 0], [948, 96.04705882352941, 89.82839506172839, 0], [
949, 95.81176470588235, 89.82839506172839, 0], [950, 95.8117
6470588235, 89.82839506172839, 0], [951, 95.81176470588235,
89.82839506172839, 0], [952, 95.81176470588235, 89.828395061
72839, 0], [953, 96.14117647058823, 89.82839506172839, 0], [
954, 95.81176470588235, 89.82839506172839, 0], [955, 95.8117
6470588235, 89.82839506172839, 0], [956, 95.81176470588235,
89.82839506172839, 0], [957, 96.14117647058823, 89.828395061
72839, 0], [958, 95.81176470588235, 89.82839506172839, 0], [
959, 95.81176470588235, 89.82839506172839, 0], [960, 95.8117
6470588235, 89.82839506172839, 0], [961, 95.81176470588235,
89.82839506172839, 0], [962, 95.81176470588235, 89.828395061
72839, 0], [963, 95.81176470588235, 89.82839506172839, 0], [
964, 95.81176470588235, 89.82839506172839, 0], [965, 95.8117
6470588235, 89.82839506172839, 0], [966, 95.81176470588235,
89.82839506172839, 0], [967, 95.85882352941177, 89.828395061
72839, 0], [968, 95.81176470588235, 89.82839506172839, 0], [
969, 95.81176470588235, 89.82839506172839, 0], [970, 96.0941
1764705882, 89.82839506172839, 0], [971, 95.81176470588235,
89.82839506172839, 0], [972, 95.81176470588235, 89.828395061
72839, 0], [973, 95.81176470588235, 89.82839506172839, 0], [
974, 95.81176470588235, 89.82839506172839, 0], [975, 95.8117
6470588235, 89.82839506172839, 0], [976, 95.81176470588235,
89.82839506172839, 0], [977, 95.81176470588235, 89.828395061
72839, 0], [978, 95.81176470588235, 89.82839506172839, 0], [
979, 95.81176470588235, 89.82839506172839, 0], [980, 95.8117
6470588235, 89.82839506172839, 0], [981, 95.81176470588235,
89.82839506172839, 0], [982, 95.85882352941177, 89.828395061
72839, 0], [983, 95.81176470588235, 89.82839506172839, 0], [
984, 95.81176470588235, 89.82839506172839, 0], [985, 95.8117
6470588235, 89.82839506172839, 0], [986, 95.85882352941177,
89.82839506172839, 0], [987, 95.85882352941177, 89.828395061
72839, 0], [988, 95.81176470588235, 89.82839506172839, 0], [
989, 95.81176470588235, 89.82839506172839, 0], [990, 95.8117
6470588235, 89.82839506172839, 0], [991, 95.81176470588235,
89.82839506172839, 0], [992, 95.81176470588235, 89.828395061
72839, 0], [993, 95.81176470588235, 89.82839506172839, 0], [
994, 95.81176470588235, 89.82839506172839, 0], [995, 95.8117
6470588235, 89.82839506172839, 0], [996, 96.18823529411765,
89.82839506172839, 0], [997, 96.0, 89.82839506172839, 0], [9
98, 95.85882352941177, 89.82839506172839, 0], [999, 96.42352
94117647, 89.82839506172839, 0], [1000, 95.81176470588235, 8
9.82839506172839, 0]]

Saving the results in a .csv file for later use

In [58]:

```
with open('P_30_MR_7_G_1000.csv', 'w', newline='') as file:
    writer = csv.writer(file, quoting=csv.QUOTE_NONNUMERIC,
    writer.writerows(records)
```

Displaying the results

Due to my unfamiliarity in collecting all the data in one go, I had to manually go through the code above, change the needed values to mimic the settings that were used to test the Genetic Algorithm from Costa's work and collect that data bit by bit.

Settings:

```
Setting 1: Population = 10, Mutation Rate = 0.3,
Generations = 10
Setting 2: Population = 20, Mutation Rate = 0.5,
Generations = 100
Setting 3: Population = 30, Mutation Rate = 0.7,
Generations = 1000
```

The corresponding results are in the labeled .csv files.

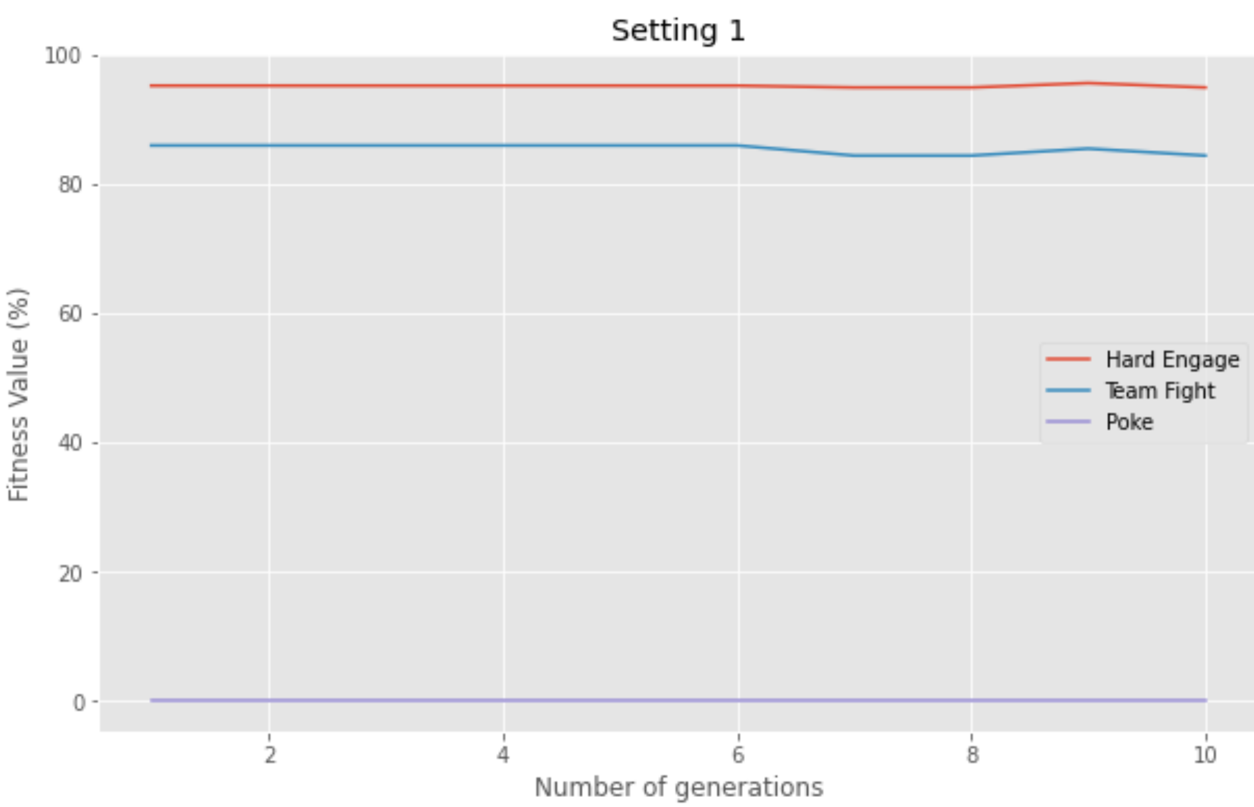
Below are the results.

Setting 1 Results

In [60]:
```
results_s1 = pd.read_csv('P_10_MR_3_G_10.csv', sep=",")
#results_s1.head()

x_axis = results_s1["Generation"]
hard_engage = results_s1["Hard Engage"]
#print(hard_engage)
team_fight = results_s1["Team Fight"]
poke = results_s1["Poke"]

plt.style.use('ggplot')
fig = plt.figure(figsize=(10, 6))
plt.title('Setting 1')
plt.plot(x_axis, hard_engage, label='Hard Engage')
plt.plot(x_axis, team_fight, label='Team Fight')
plt.plot(x_axis, poke, label='Poke')
plt.legend()
plt.xlabel('Number of generations')
plt.ylabel('Fitness Value (%)')
plt.show()
```

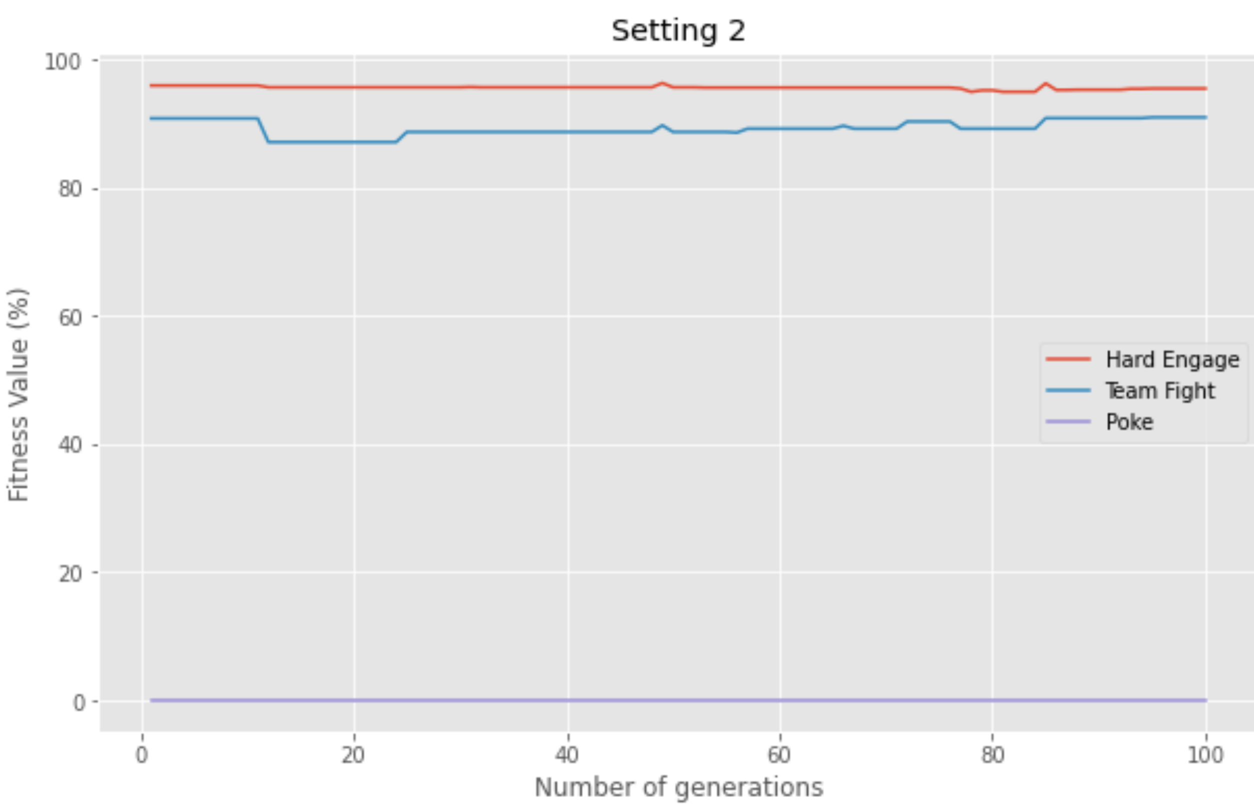


Setting 2 Results

In [61]:
```
results_s2 = pd.read_csv('P_20_MR_5_G_100.csv', sep=",")
#results_s1.head()

x_axis2 = results_s2["Generation"]
hard_engage2 = results_s2["Hard Engage"]
#print(hard_engage)
team_fight2 = results_s2["Team Fight"]
poke2 = results_s2["Poke"]

plt.style.use('ggplot')
fig = plt.figure(figsize=(10, 6))
plt.title('Setting 2')
plt.plot(x_axis2, hard_engage2, label='Hard Engage')
plt.plot(x_axis2, team_fight2, label='Team Fight')
plt.plot(x_axis2, poke2, label='Poke')
plt.legend()
plt.xlabel('Number of generations')
plt.ylabel('Fitness Value (%)')
plt.show()
```



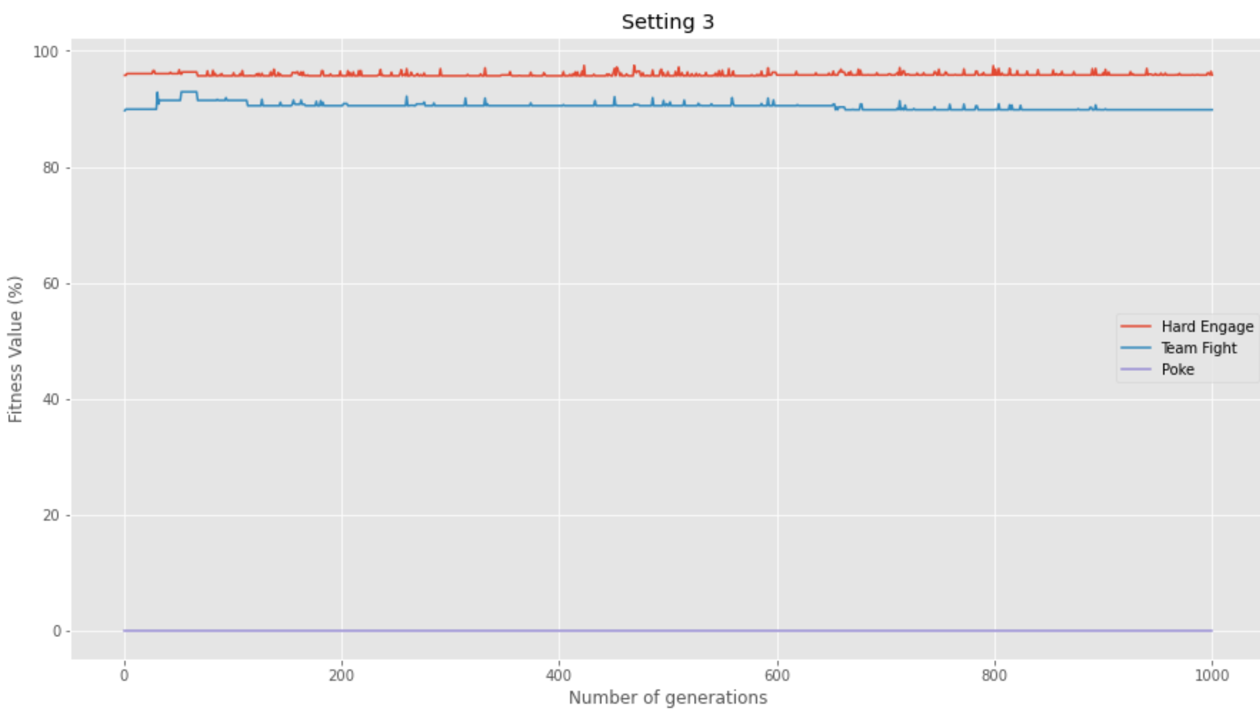
Setting 3 Results

In [62]:

```
results_s3 = pd.read_csv('P_30_MR_7_G_1000.csv', sep=",")
#results_s1.head()

x_axis3 = results_s3["Generation"]
hard_engage3 = results_s3["Hard Engage"]
#print(hard_engage)
team_fight3 = results_s3["Team Fight"]
poke3 = results_s3["Poke"]

plt.style.use('ggplot')
fig = plt.figure(figsize=(15, 8))
plt.title('Setting 3')
plt.plot(x_axis3, hard_engage3, label='Hard Engage')
plt.plot(x_axis3, team_fight3, label='Team Fight')
plt.plot(x_axis3, poke3, label='Poke')
plt.legend()
plt.xlabel('Number of generations')
plt.ylabel('Fitness Value (%)')
plt.show()
```



Future Work

This current form of the genetic algorithm that I had envisioned is far from perfect and still has it's bugs. This is still just another thing to improve on.

A small list of things I pla to improve on in the future I plan to improve a list of things:

```
- use websites to gather the most up to date win
rate of specific champions
- use websites to gather the most up to date
champion counters
- coding to ensure that each team does not have
any repeat champions
- making it useful for any player from novice to
professional
```

```
In [ ]:
```