

# GeneticAlgorithm

December 11, 2020

## 1 Genetic Algorithm

```
[14]: import francium.algorithms.genetic_algorithm as ga
import francium.core.eval_functions as eval_functions
```

### 1.1 using an environment with $z = x^2 + y^2$

```
[15]: agent = ga.Agent()
env = ga.Environment(x_bounds=(-5.0, 5.0), y_bounds=(-5.0, 5.0),
    ↪eval_func=eval_functions.convex_x_square)
solver = ga.Solver(agent=agent, environment=env, pop_size=100)
```

```
[16]: solver.init_solver()
```

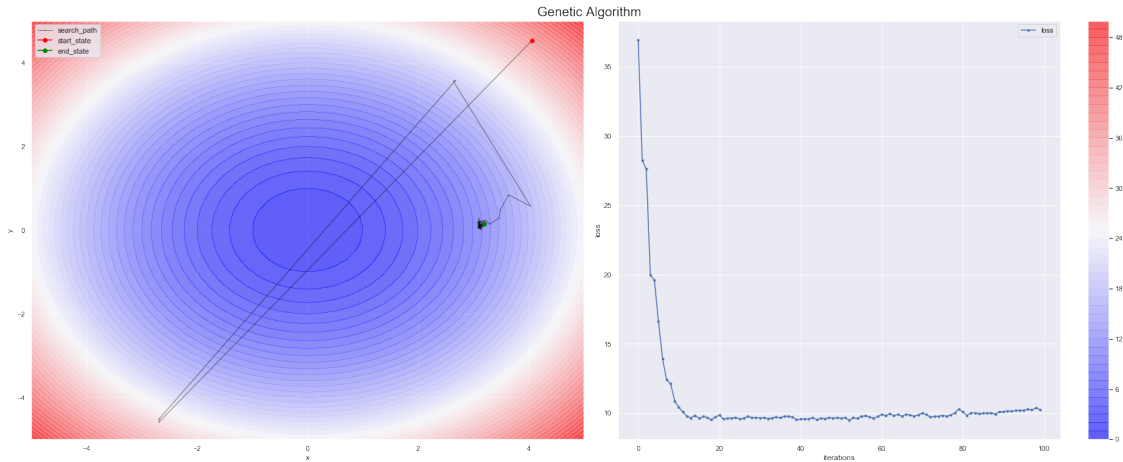
```
[ 2020-12-06 21:18:46,805 - francium.algorithms.genetic_algorithm.solver ] INFO:
=> Initialized Agent !
```

```
[17]: for episode in range(100):
    trainable = solver.train_step()
    if not trainable:
        break
```

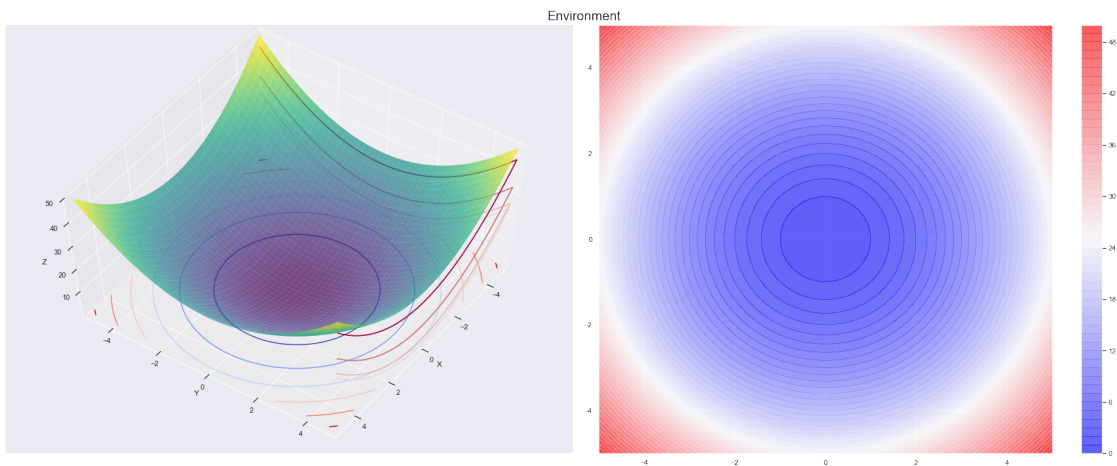
```
[18]: solver.memory.best_episode
```

```
[18]: {'x': 3.0785559458172402, 'y': 0.07634261349107278, 'z': 9.48333490616133}
```

```
[19]: solver.plot_history()
```



```
[20]: env.plot_environment()
```



## 1.2 using an environment with $z = 5 * \sin(x^2 + y^2) + x^2 + y^2$

```
[21]: agent = ga.Agent()
env = ga.Environment(x_bounds=(-5.0, 5.0), y_bounds=(-5.0, 5.0),
    eval_func=eval_functions.sinx_plus_x)
solver = ga.Solver(agent=agent, environment=env, pop_size=100)
```

```
[22]: solver.init_solver()
```

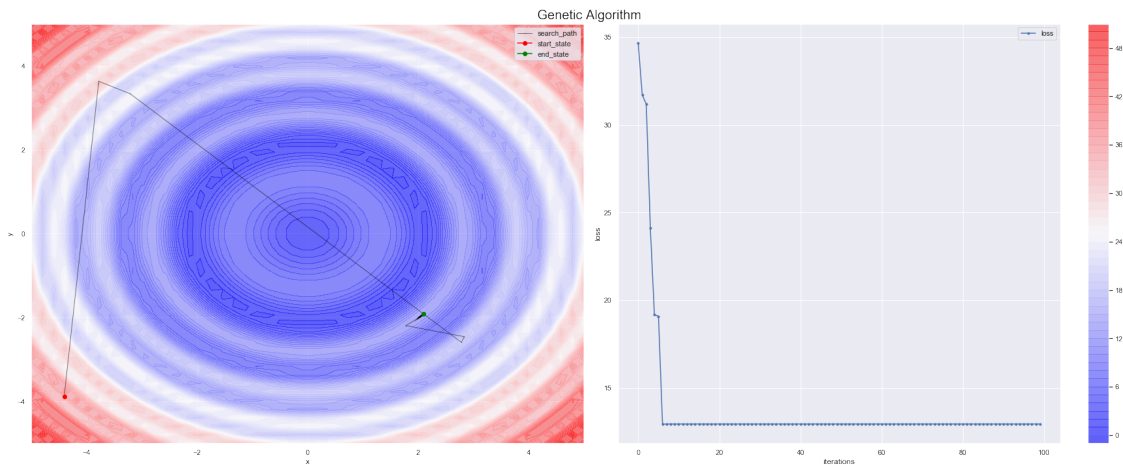
```
[ 2020-12-06 21:18:49,826 - francium.algorithms.genetic_algorithm.solver ] INFO:
=> Initialized Agent !
```

```
[23]: for episode in range(100):
        trainable = solver.train_step()
        if not trainable:
            break
```

```
[24]: solver.memory.best_episode
```

```
[24]: {'x': 1.7793197640828229, 'y': -2.1944559850071386, 'z': 12.940944890399273}
```

```
[25]: solver.plot_history()
```



```
[26]: env.plot_environment()
```

C:\Users\shadowleaf\anaconda3\envs\thetensorclan-aws\lib\site-packages\numpy\core\\_asarray.py:136: VisibleDeprecationWarning: Creating an ndarray from ragged nested sequences (which is a list-or-tuple of lists-or-tuples-or ndarrays with different lengths or shapes) is deprecated. If you meant to do this, you must specify 'dtype=object' when creating the ndarray

```
return array(a, dtype, copy=False, order=order, subok=True)
```

