Genetic Algorithm-introduction

Genetic Algorithms(GA) are useful search and application techniques that mimic the evolution process.Solves problems in the same way that nature solves the problem of adapting living organisms to the harsh realities of life in a hostile world: evolution. To design artificial systems software that retains the robustness of natural systems. John Holland American scientist. The original form of the GA, as illustrated by John Holland in 1975 had distinct features: a bit string representation, proportional selection and cross-over to produce new individuals. Several variations to the original GA have been developed using different representation schemes, selection and reproduction operators.i.e.“Select The Best, Discard The Rest”.

Basic Idea Behind GA:

◇ A class of probabilistic optimization algorithms
◇ Inspired by the biological evolution process
◇ Uses concepts of “Natural Selection” and “Genetic Inheritance” (Darwin 1859).
◇ Take a population of candidate solutions to a given problem.
◇ Use operators inspired by the mechanisms of natural genetic variation.
◇ Apply selective pressure toward certain properties.
◇ Evolve a more fit solution.

Characteristics of GA:

(1)The genetic algorithm works by encoding a set of parameters, not the limits themselves.

(2)The genetic algorithm starts its search from a number of points, not a single point.

(3)The genetic algorithm uses paid data, not derivatives.

(4)The genetic algorithm uses possible mutation rules, not determining.

GA Parameters and Operators

(1)Selection
(2)Crossover
(3)Mutation

Simple Genetic Algorithm(sga)

```
function sga ()

{
        Initialize population;
        Calculate fitness function;
        While(fitness value != termination criteria)

                {
                        Selection;
                        Crossover;
                        Mutation;
                        Calculate fitness function;
                }

}
```

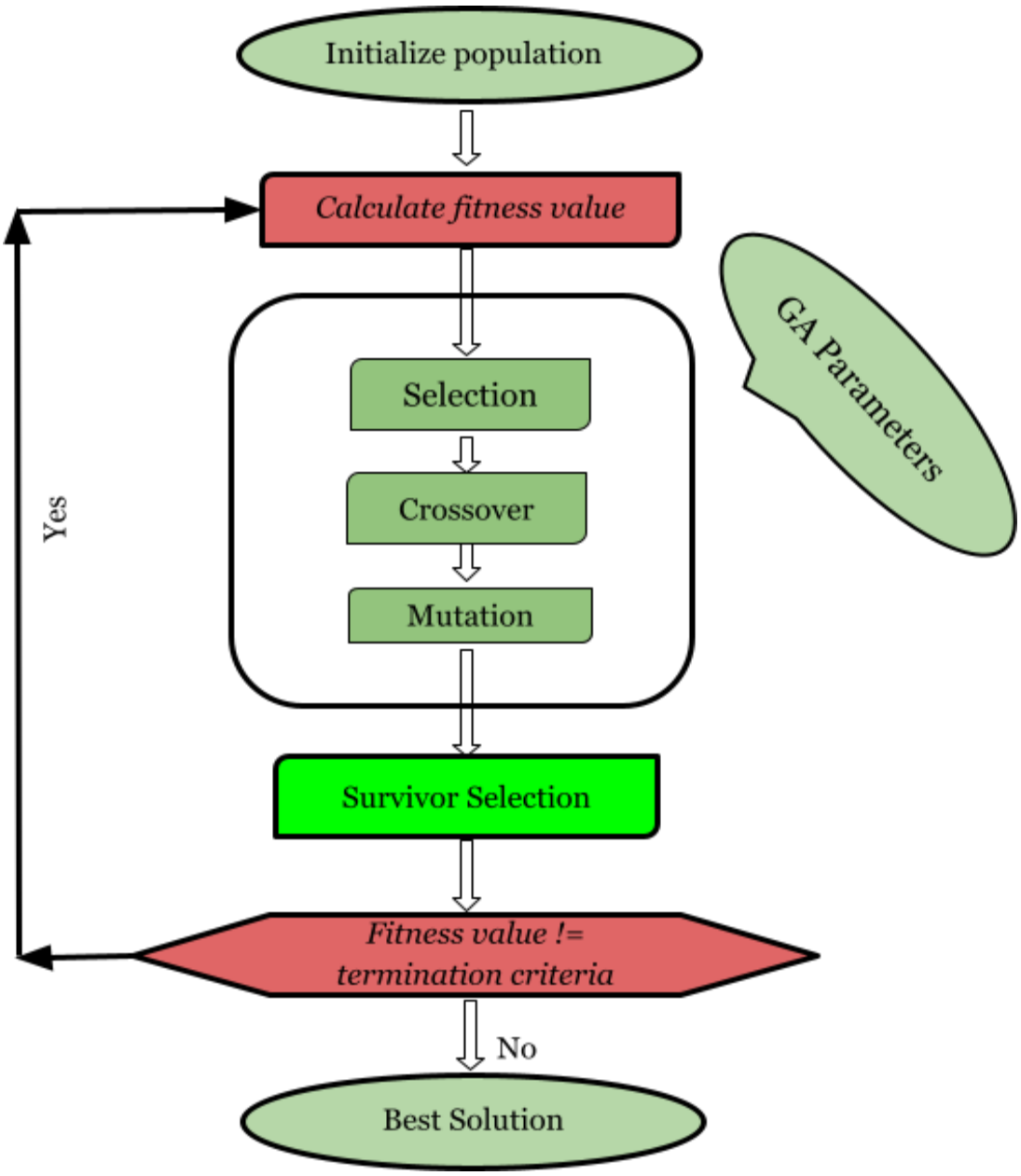

Fig. Genetic Algorithm(GA) Process

Applications of GA

1. Recurrent Neural Network
2. Mutation testing
3. Code breaking
4. Filtering and signal processing
5. Learning fuzzy rule base, etc