# NNA GA Prototype Review

Erich Seamon

September 2019

## Genetic Algorithm review

The following set of steps provide a general review of a genetic algorithm approach, using simulated data from a optimization test function. Genetic algorithms (GA) fall within the the grouping metaheuristic procedures, modeled after the process of natural selection. A large set of research (Holland, 1960; Goldberg, 1989; Mitchell, 1996; Banzhaf et al., 1998; Bies et al., 2006) have explored the use and value of genetic algorithmic modeling, primarly for optimization and search efforts that apply evolutionary features such as selection, crossover, and mutation.

GA theory takes a population of outcomes/solutions to a problem set, and attempts to model ("evolve") said solutions in a manner that allows for zeroing in on the most optimized result. Each solution may be multi-dimensional (i.e. an N-numbered array) which is altered ("mutated") in an iterative fashion. After each iteration ("generation"), the fitness of each outcome is evaluated - against a known/predetermined objective function. As such, these two components: 1) the outcome/solution domain, as well as 2) the function for evaluating generational fitness - are essential in an appropriate genetic algorithmic model. For our proposed GA applied effort, the solution domain would be a multi-dimensional representation of input data that varied across space, and potentially time. Our fitness function would be represented as an known overall spatiotemporal pattern (ie. ocean acidification).

In order to prototype a GA approach, we have constructed a modeling process which uses an known objective test function, to evaluate a typical GA iterative process. For our purposes, we are using the Rastrigin optimization function. The Rastrigin function is non-convex, ,non-linear, and multimodal, and is frequently used to test optimization algorithms, given its large search space and numerous local minima. The Rastrigin function can be defined as:

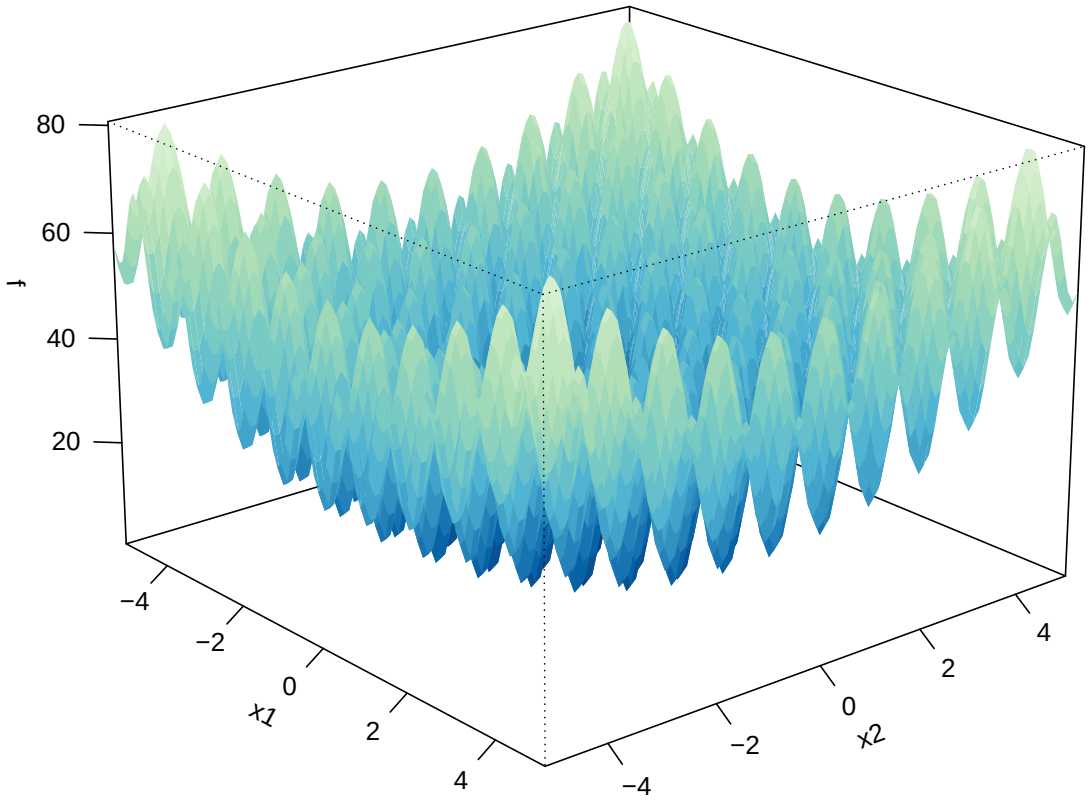
$$f(x) = \sum_{i=1}^{n} [x_i^2 - Acos(2\pi x_i)]$$
$$where\ A = 10\ and\ x_i\ \epsilon\ [-5.12, 5.12]$$
$$with\ a\ global\ minimum\ at\ x = 0$$
$$where\ f(x) = 0$$

## Rastrigin Function Plotted in 3d

Here we construct the function and plot in 3 dimensions.

```
usePackage("rgl")
usePackage("GA")
#usePackage("animation")
#usePackage("ffmpeg")
```

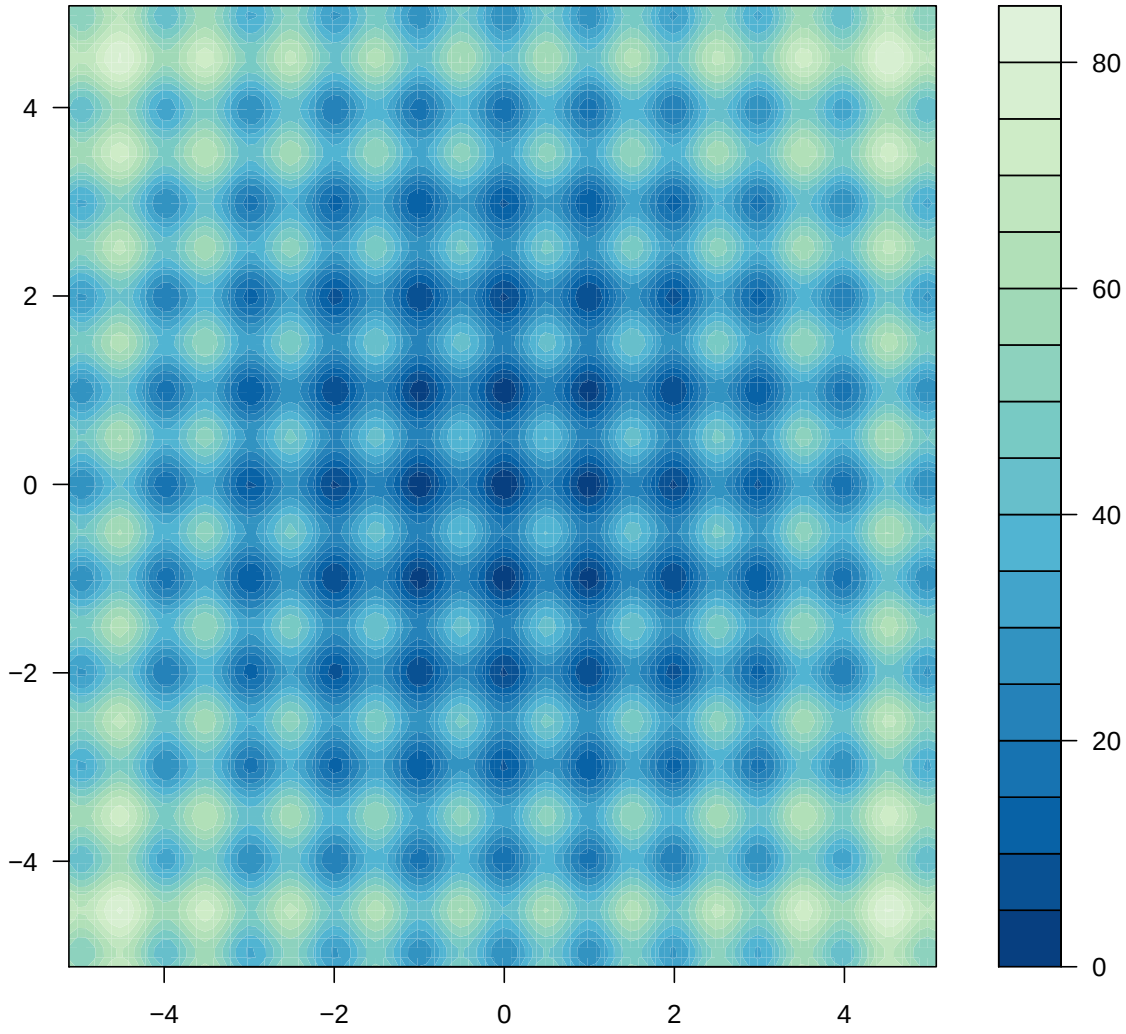
```r
#--use of Rastrigin objective function to test optimization

Rastrigin <- function(x1, x2)
{
  20 + x1^2 + x2^2 - 10*(cos(2*pi*x1) + cos(2*pi*x2))
}


x1 <- x2 <- seq(-5.12, 5.12, by = 0.1)
f <- outer(x1, x2, Rastrigin)
persp3D(x1, x2, f, theta = 50, phi = 20, col.palette = bl2gr.colors)
```



### Rastrigin Function Plotted in 2d

Plotted in 2 dimensions provides a simpler depiction of the global and local minima.

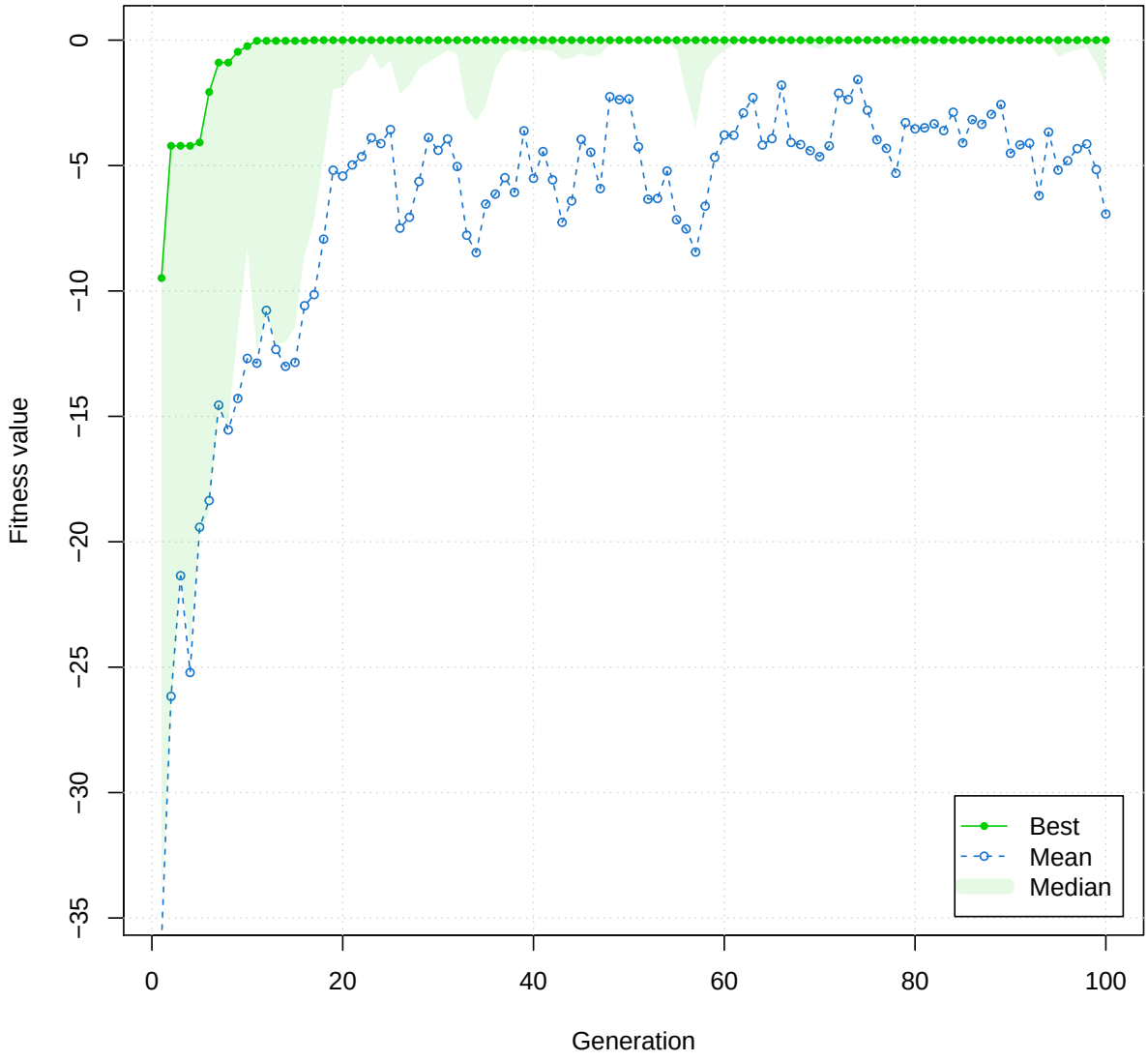
```
filled.contour(x1, x2, f, color.palette = bl2gr.colors)
```




### Construction of a monitoring function and running of GA over 100 iterations

Here we set up a monitoring function which generates an animation of our GA run from each of the iterations. In addition, we plot the iterations based on their fitness function prediction vs. the optimum function plot.

```
monitor <- function(obj)
{
  #setwd("/tmp/seamon")
  pdf(paste(obj@iter, "Rastrigin.pdf", sep=""),width=7,height=5)
  contour(x1, x2, f, drawlabels = FALSE, col = grey(0.5))
  title(paste("iteration =", obj@iter), font.main = 1)
  points(obj@population, pch = 20, col = 2)
  Sys.sleep(0.2)
  dev.off()
}

suggestedSol <- matrix(c(0.2,1.5,-1.5,0.5), nrow = 2, ncol = 2, byrow = TRUE)

#dir.create("/tmp/seamon")

GA <- ga(type = "real-valued",
         fitness =  function(x) -Rastrigin(x[1], x[2]),
         lower = c(-5.12, -5.12), upper = c(5.12, 5.12),
         suggestions = suggestedSol,
         popSize = 50, maxiter = 100, run = 100, monitor = monitor)

plot(GA)
```




## GA summary notation

The summary output of our GA run provides the number of observation, iterations, and set elitism, crossover, and mutation parameters. These parameters can be altered and evaluted as part of our iterative model approach.

```
summary(GA)
```

```
## -- Genetic Algorithm -------------------
##
## GA settings:
## Type                  =  real-valued
```

```
## Population size       =  50
## Number of generations =  100
## Elitism               =  2
## Crossover probability =  0.8
## Mutation probability  =  0.1
## Search domain =
##          x1    x2
## lower -5.12 -5.12
## upper  5.12  5.12
## Suggestions =
##     x1  x2
## 1  0.2 1.5
## 2 -1.5 0.5
##
## GA results:
## Iterations             = 100
## Fitness function value = -4.180628e-06
## Solution =
##                x1           x2
## [1,] 0.0001056118 -9.95927e-05
```