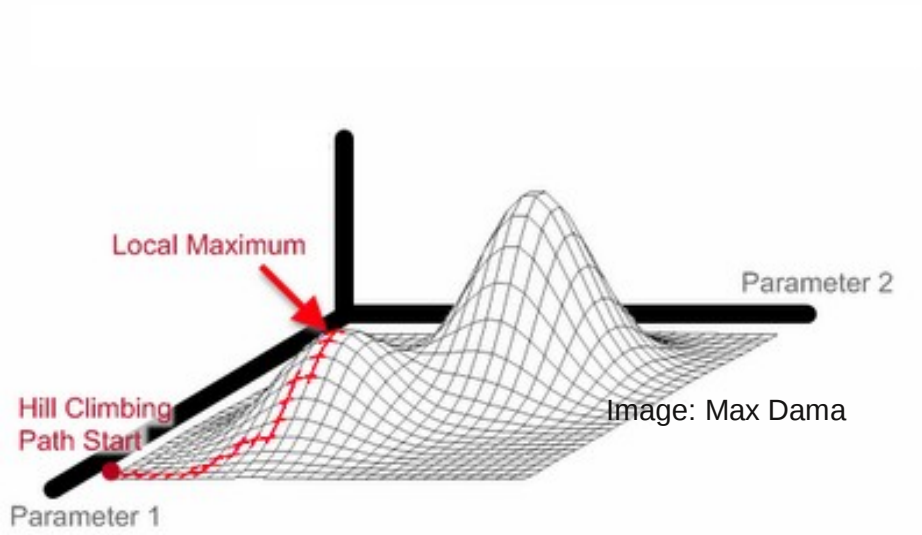
Generalized Global Optimization with DEoptim

Brian Peterson

Global Optimization Problem



- Two common approaches to optimization are differentiation and hill climbing
- Both extremely subject to local minima/maxima
- The risk of finding local solutions increases with the number of parameters

Improving Global Optimization

- Most methods for improving optimization results add some concept of randomization or jumps
- Jumps may be in the wrong direction,
  - but enough randomization should provide at least one path towards the global optima
- This idea was first employed in 'simulated annealing'

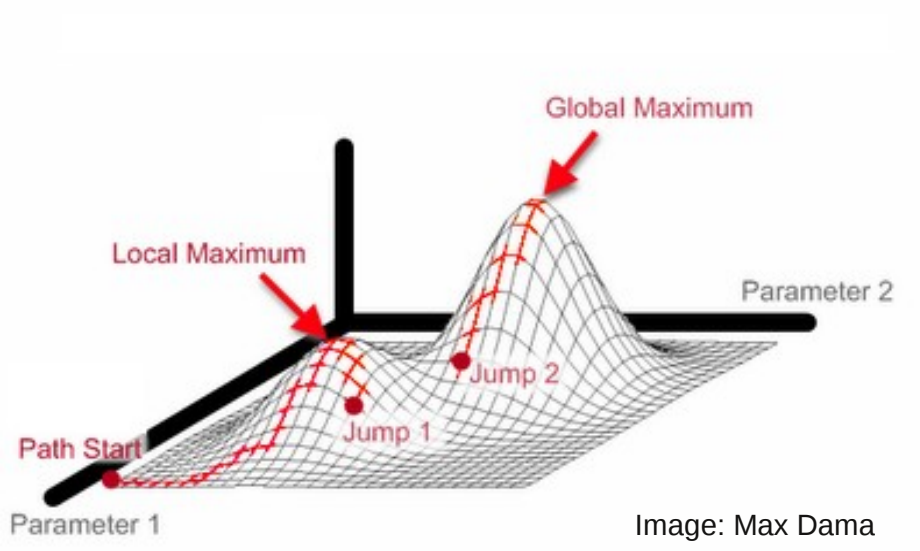


Image: Max Dama

About Differential Evolution

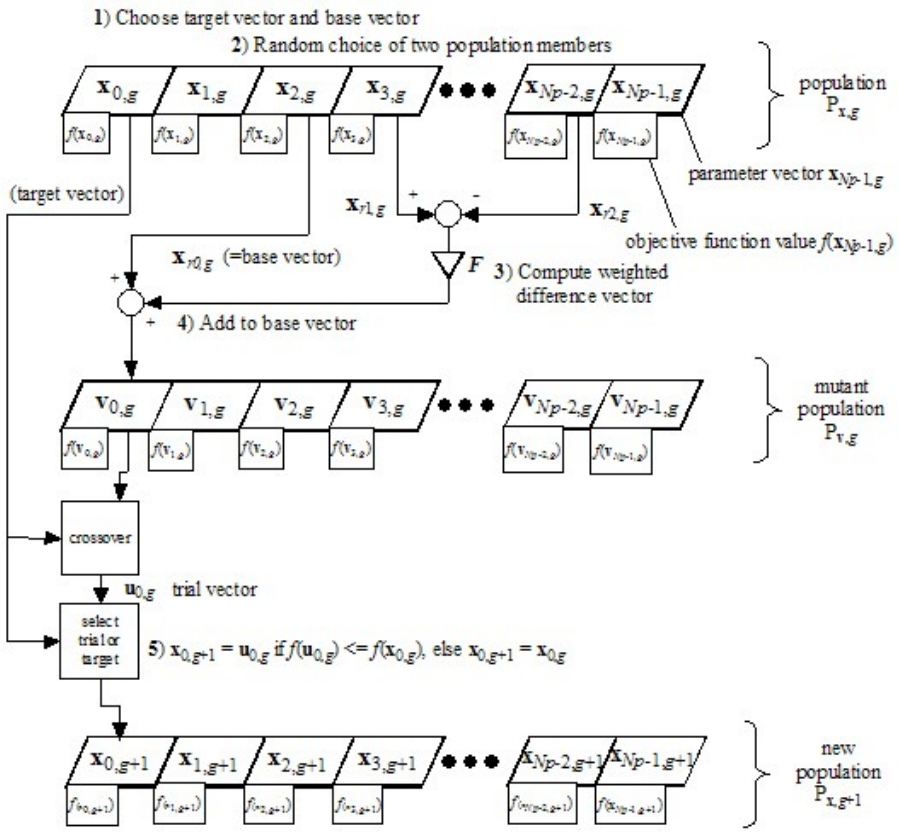


Image: Storn et. al.

- Differential Evolution is an elegant population based stochastic function minimizer.
  - Continuous, evolutionary optimization.
  - Uses real-number parameters.
  - Key innovation lies in the parameter vector generation
- Package *DEoptim* provides the algorithm in R.
- Implementation of the algorithm distributed with the book:
  - *Differential Evolution - A Practical Approach to Global Optimization* by Price, K.V., Storn, R.M., Lampinen J.A, Springer-Verlag, 2005.

Using DEoptim

- Why DEoptim?
  - All numerical optimizations involve tradeoffs between speed and accuracy
  - The problem space may well be non-convex in real problems
  - Differential Evolution will get more directed with each generation, rather than a random grid search or a stepwise function differentiation
  - It continues to use random jumps throughout the process, but is not controlled by these if they do not improve the solution
  - Allows more logical 'space' to be searched with the same number of trial populations for more complex objectives

Example: Rastrigen

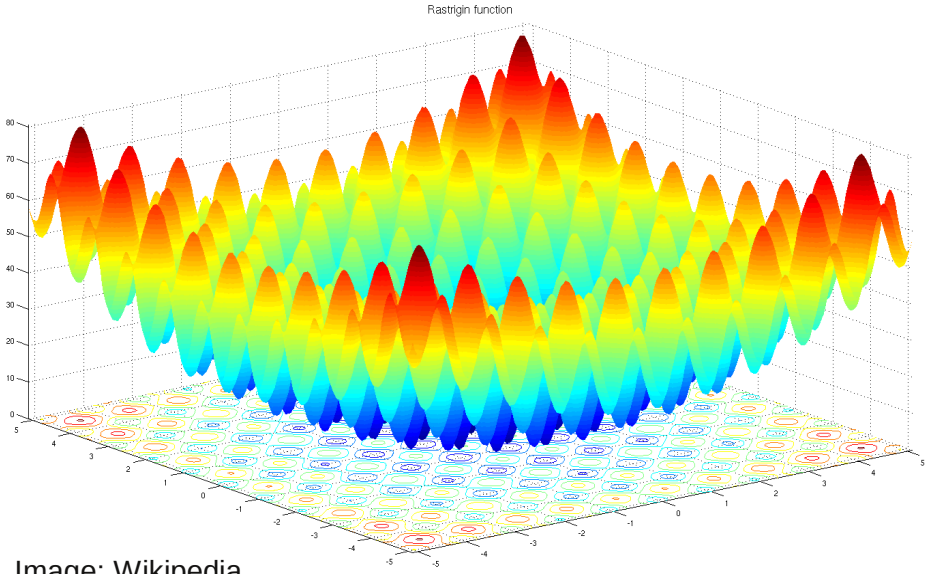
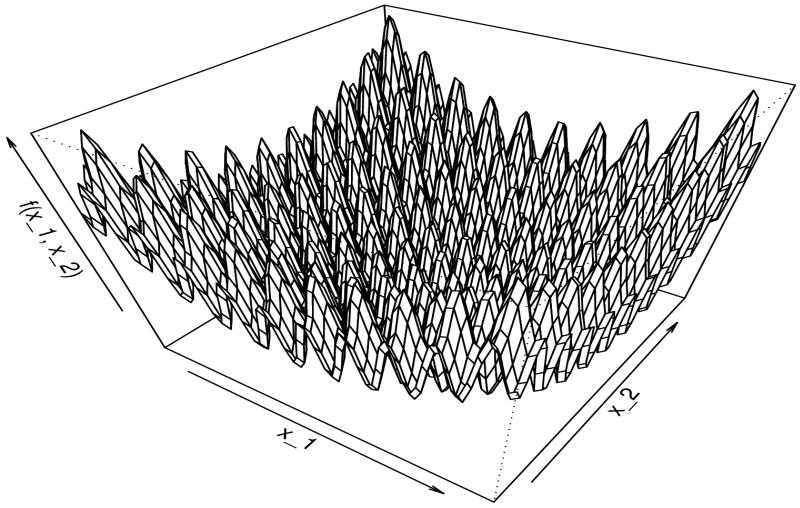


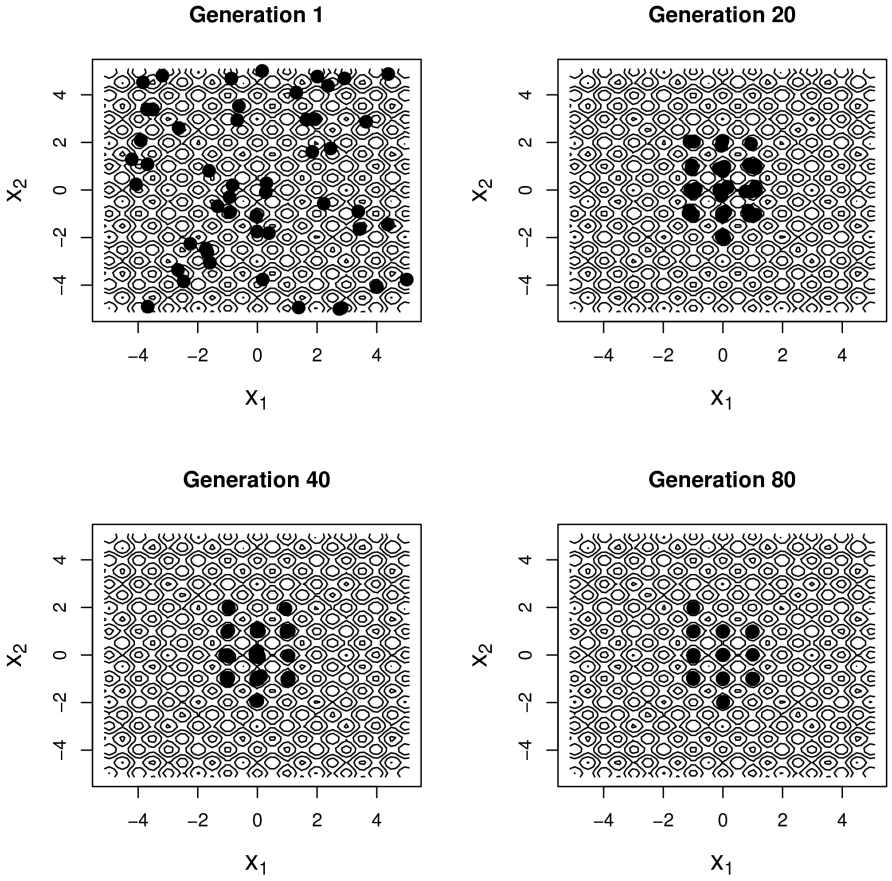
Image: Wikipedia



- Rastrigin is a non-convex function with multiple local minima designed for testing optimization algorithms
- Obviously, continuous differentiation would find a local minima
- Random grid search might find the true optima, or might get close, but results aren't guaranteed

Example: Rastrigen

```
> Rastrigin <- function(x) {
  sum(x^2 - 10 * cos(2 * pi
  * x)) + 20 }
> opt_result <- DEoptim
  (fn=Rastrigin, lower =
  c(-5,-5), upper = c(5,5),
  control = list
  (storepopfrom = 1) )
```



- Examining the results shows how DEoptim narrows the search,
  - starting from a wide range of possibilities,
  - getting to a much tighter grouping
- DEoptim will find the global optima within 100 generations

What do I use DEoptim for?

- Portfolio optimization:
  - see R package PortfolioAnalytics
  - see R package LSPM
- Strategy Parameter optimization
  - to be added to R package quantstrat
- Volatility (or other) Regime Detection
  - Not yet packaged, but in an upcoming JSS paper

Thank You for Your Attention

References

- CRAN Optimization Task View:
  - http://cran.r-project.org/web/views/Optimization.html
- Differential Evolution:
  - http://www.icsi.berkeley.edu/~storn/code.html
- DEoptim:
  - http://cran.r-project.org/web/packages/DEoptim/index.html
  - http://r-forge.r-project.org/projects/deoptim
- Rastrigin:
  - http://en.wikipedia.org/wiki/Rastrigin_function