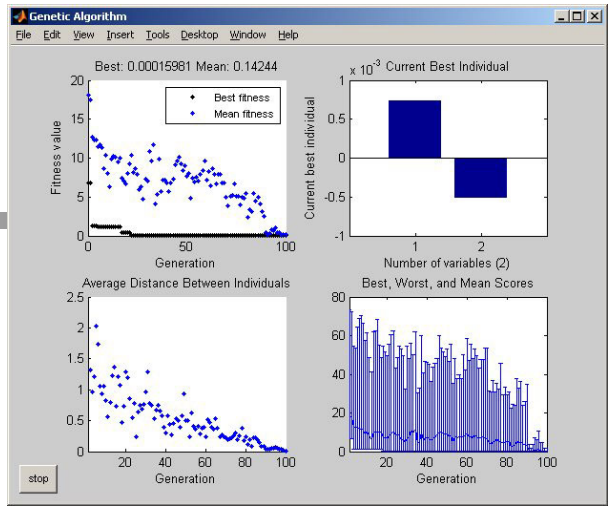
The MathWorks
Accelerating the pace of engineering and science



Genetic Algorithm and Direct Search Toolbox

V2.3 (R2008a)

Seungjae Min
School of Mechanical Engineering
Department of Automotive Engineering
Hanyang University

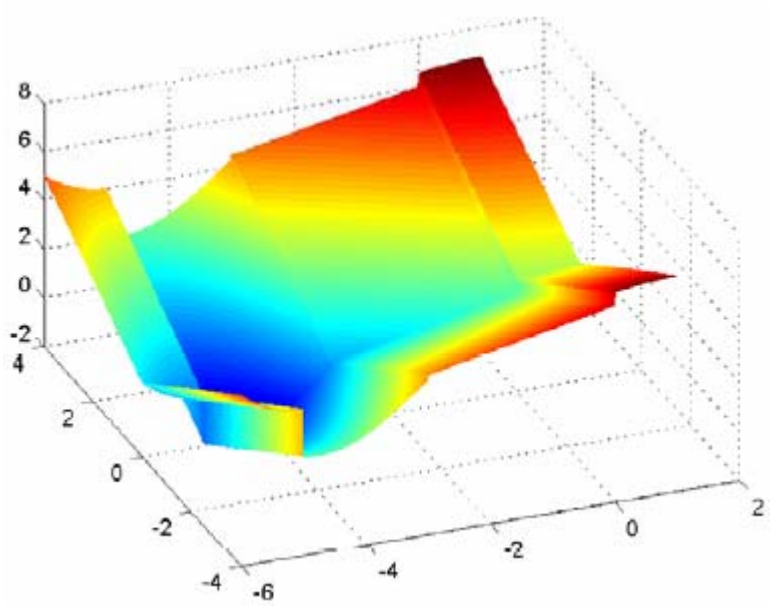
Direct Search

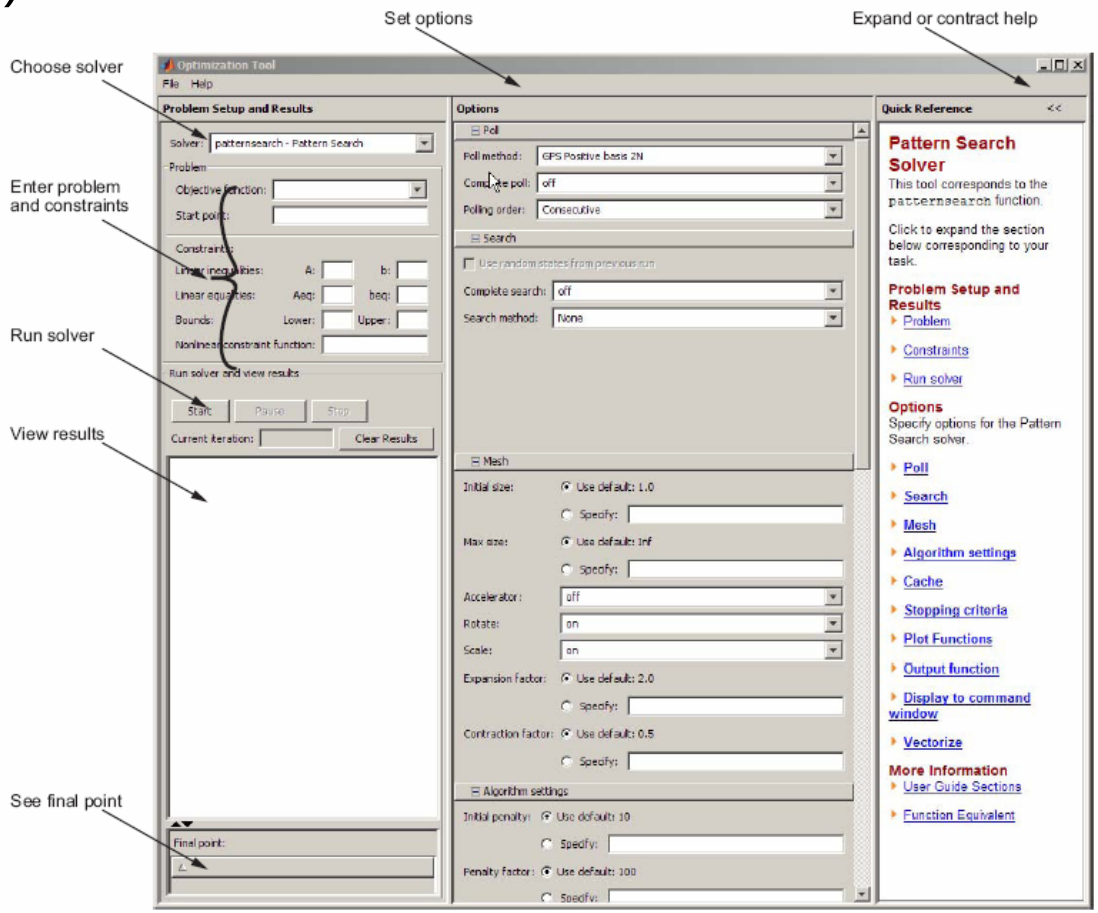
- method for solving optimization problems that does not require any information about the gradient of the objective function
- searches a set of points around the current point, looking for one where the value of the objective function is lower than the value at the current point
- solves problems for which the objective function is not differentiable, stochastic, or even continuous

- generalized pattern search (GPS) algorithm
- mesh adaptive search (MADS) algorithm

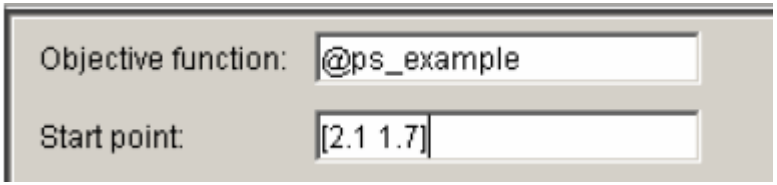
DS: Function Call

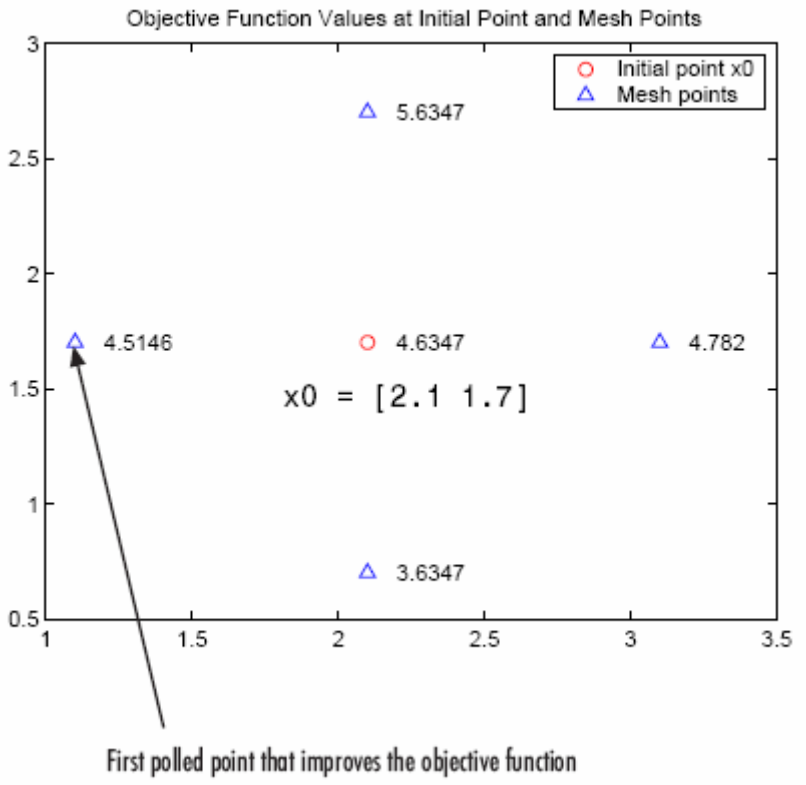
- [x fval exitflag output] =patternsearch(@objfun, x0, A, b, Aeq, beq, lb, ub, nonlcon, options)
- optimtool('patternsearch')





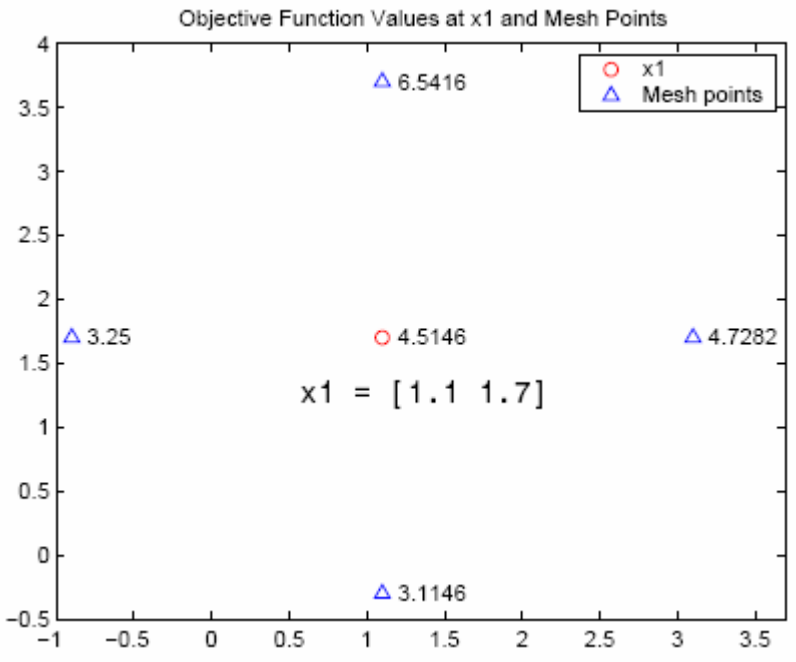
How Pattern Search Works






```
[1 0] + x0 = [3.1 1.7]
[0 1] + x0 = [2.1 2.7]
[-1 0] + x0 = [1.1 1.7]
[0 -1] + x0 = [2.1 0.7]
```

(GPS Positive basis 2N)

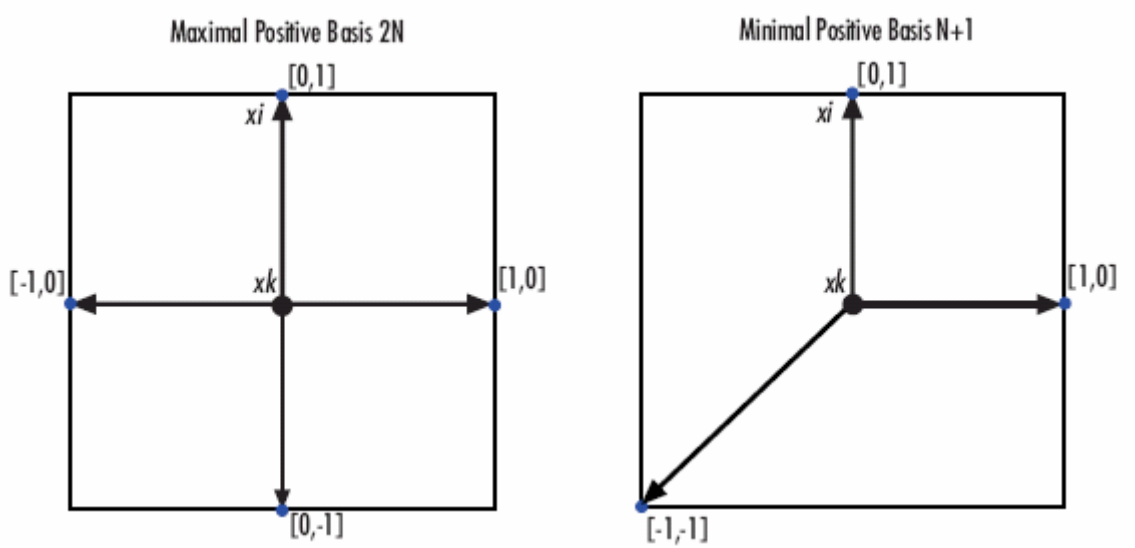



```
2*[1 0] + x1 = [3.1 1.7]
2*[0 1] + x1 = [1.1 3.7]
2*[-1 0] + x1 = [-0.9 1.7]
2*[0 -1] + x1 = [1.1 -0.3]
```

x2 = [-0.9 1.7]

DS: Terminology

- Pattern: direction



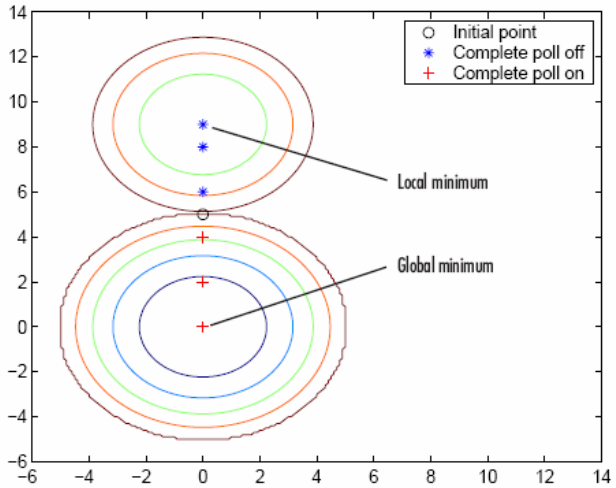
- Mesh: step size (expansion/contraction factor)
- Polling: finds a point whose objective function value is less than that of the current point

DS: Options (1)

– options = psoptimset(@patternsearch)

```
options =
               TolMesh: 1.0000e-006
                TolCon: 1.0000e-006
                  TolX: 1.0000e-006
                TolFun: 1.0000e-006
               TolBind: 1.0000e-003
               MaxIter: '100*numberofvariables'
           MaxFunEvals: '2000*numberofvariables'
             TimeLimit: Inf
       MeshContraction: 0.5000
         MeshExpansion: 2
       MeshAccelerator: 'off'
            MeshRotate: 'on'
       InitialMeshSize: 1
             ScaleMesh: 'on'
           MaxMeshSize: Inf
        InitialPenalty: 10
         PenaltyFactor: 100
            PollMethod: 'gpspositivebasis2n'
          CompletePoll: 'off'
          PollingOrder: 'consecutive'
          SearchMethod: []
        CompleteSearch: 'off'
               Display: 'final'
           OutputFcns: []
             PlotFcns: []
          PlotInterval: 1
                 Cache: 'off'
             CacheSize: 10000
              CacheTol: 2.2204e-016
            Vectorized: 'off'
           UseParallel: 'never'
```

DS: Options (2)



- Poll Method
  - GPS/MADS + Positive basis 2N/Np1
- Complete Poll: Off/On
- Using a Search Method
- Mesh Expansion and Contraction
- Mesh Accelerator
  - When the mesh size is below a certain value
- Using Cache
  - Store a history, eliminate redundant computations
- Setting Tolerances for the Solver
  - Mesh, X, function, nonlinear constraint, bind
- Constrained Minimization Using pattern search

Genetic Algorithm

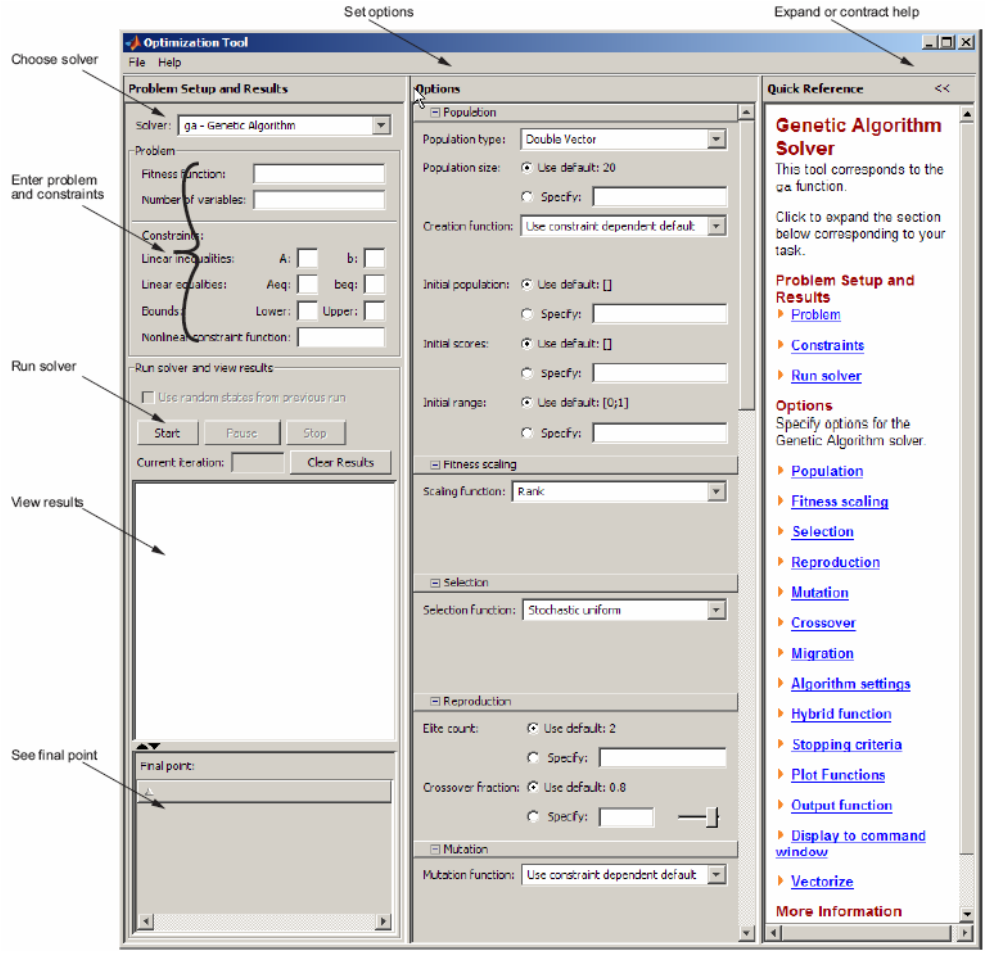
- method for solving both constrained and unconstrained optimization problems that is based on natural selection, the process that drives biological evolution
- solves a variety of optimization problems that are not well suited for standard optimization algorithms, including problems in which the objective function is discontinuous, nondifferentiable, stochastic, or highly nonlinear

Classical Algorithm	**Genetic Algorithm**
Generates a single point at each iteration. The sequence of points approaches an optimal solution.	Generates a population of points at each iteration. The best point in the population approaches an optimal solution.
Selects the next point in the sequence by a deterministic computation.	Selects the next population by computation which uses random number generators.

GA: Function Call

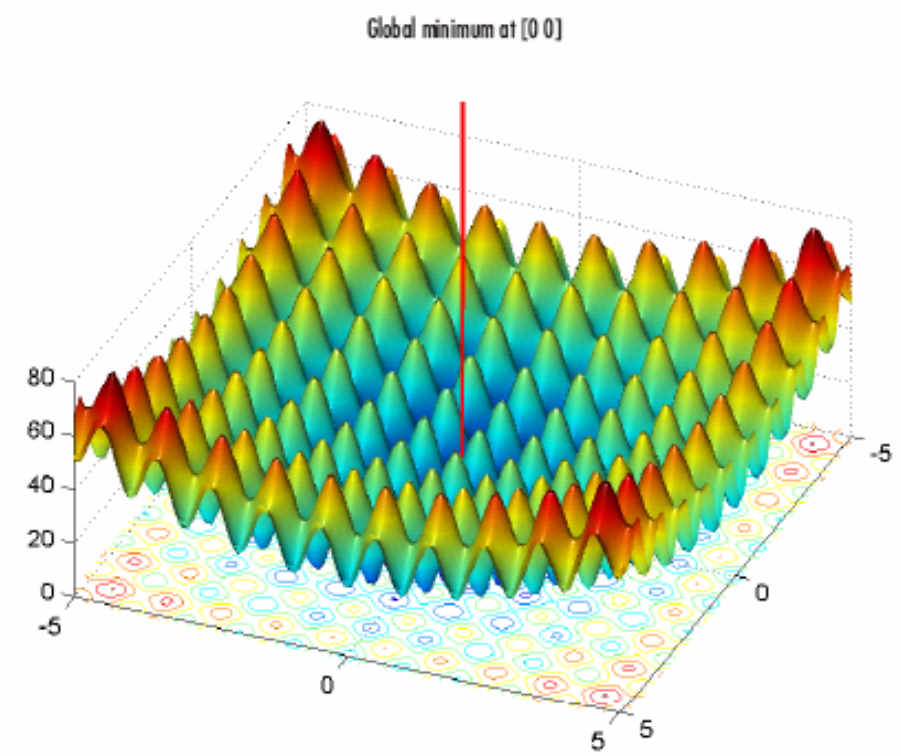
– [x fval exitflag output population scores] = ga(@fitnessfun, nvars, A, b, Aeq, beq, lb, ub, nonlcon, options)

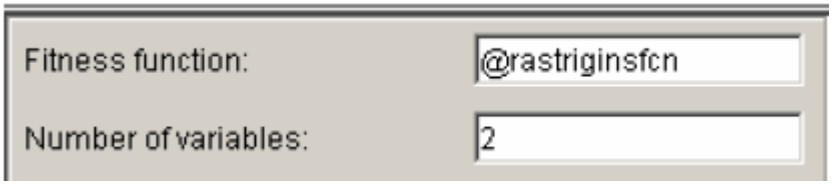
– optimtool('ga')

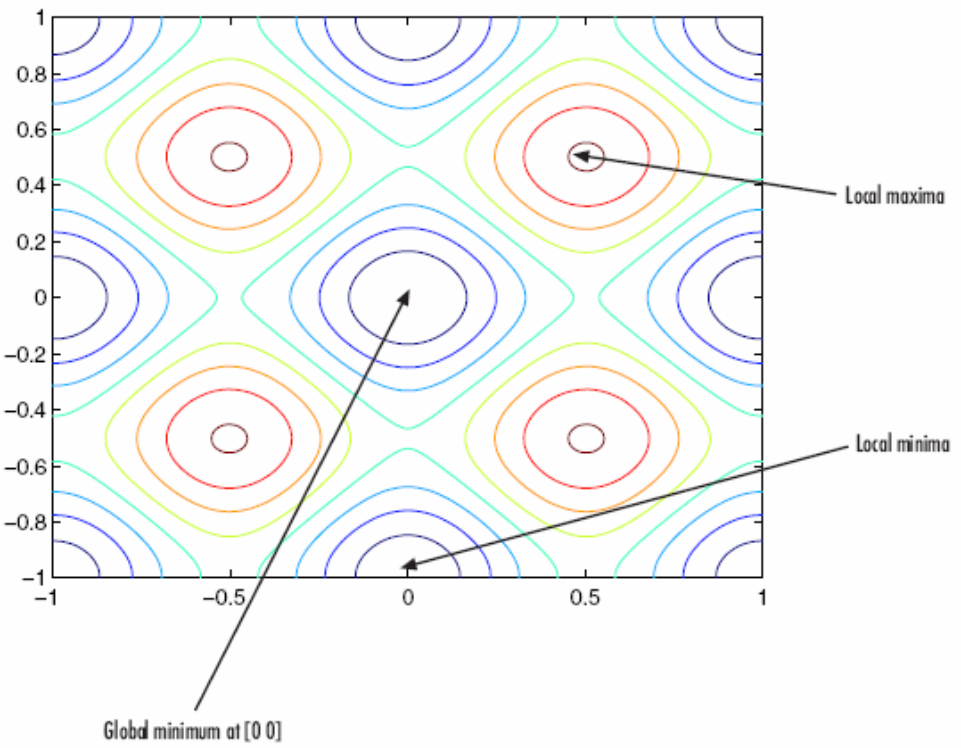


Rastrigin's Function

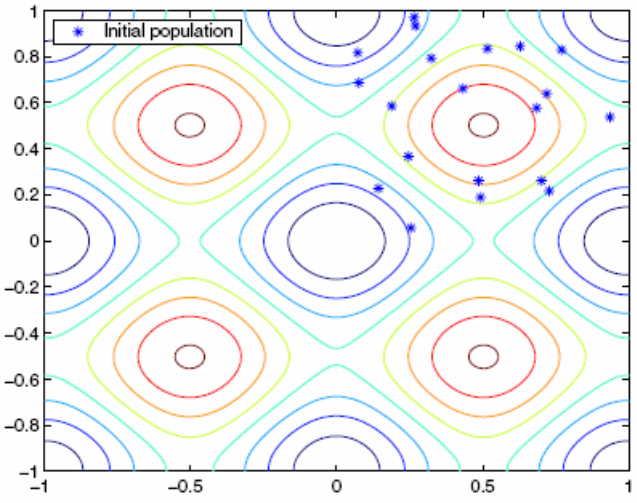
$$Ras(\boldsymbol{x}) = 20 + x_1^{\,2} + x_2^{\,2} - 10\left(\cos 2\pi x_1 + \cos 2\pi x_2\right)$$

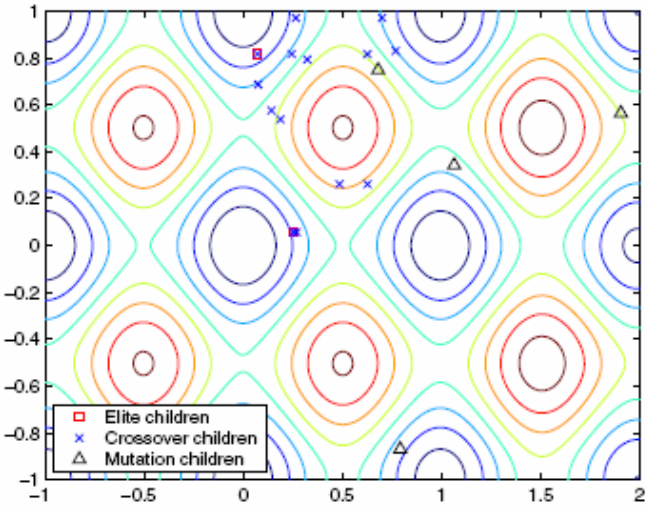


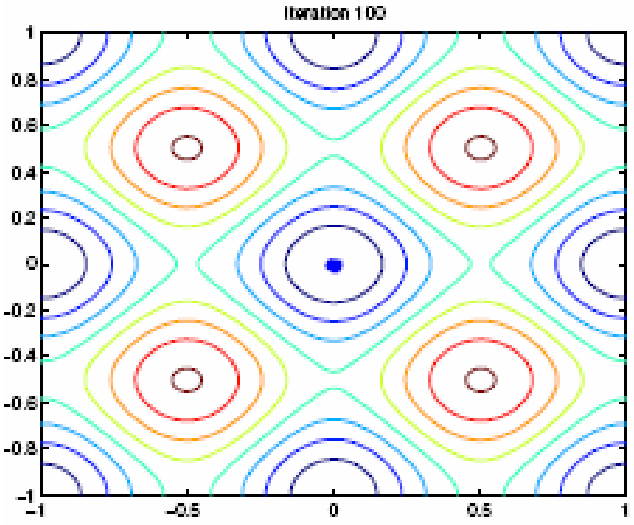


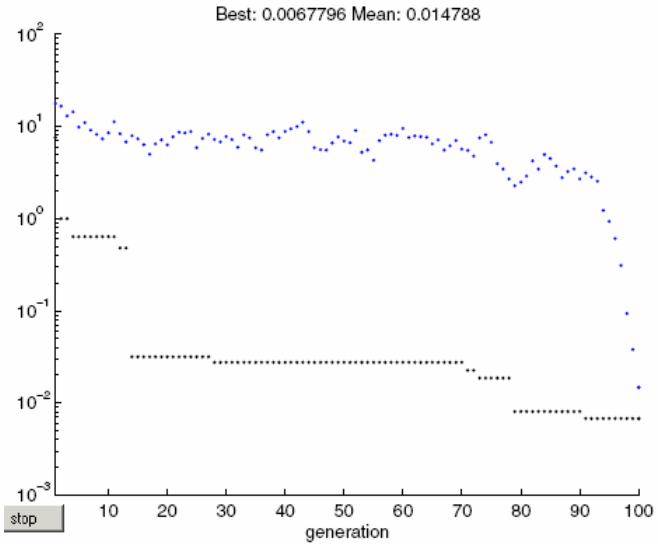


Iteration History



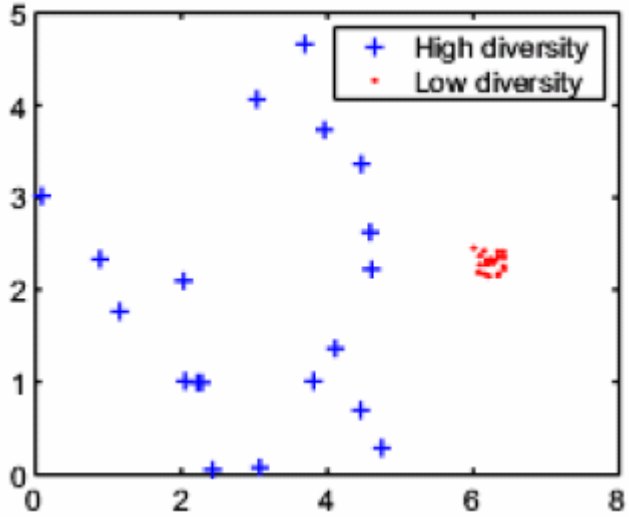






GA: Terminology

- Fitness function: objective function
- Individual: genome (genes)
  - any point to which you can apply the fitness function
- Population: an array of individuals
- Generation: Each successive population
- Diversity
  - average distance between individuals in a population
- Fitness value and best fitness
- Parent and children



GA: Options (1)

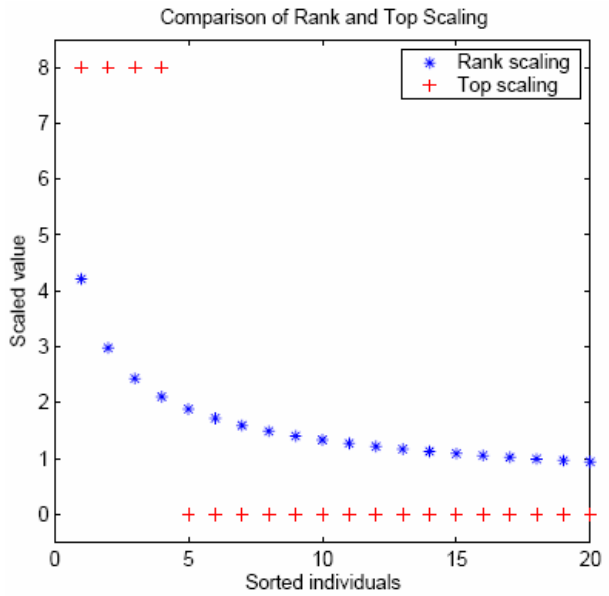
– options = gaoptimset(@ga)

```
options =
        PopulationType: 'doubleVector'
          PopInitRange: [2x1 double]
        PopulationSize: 20
           EliteCount: 2
      CrossoverFraction: 0.8000
         ParetoFraction: []
     MigrationDirection: 'forward'
      MigrationInterval: 20
      MigrationFraction: 0.2000
            Generations: 100
              TimeLimit: Inf
           FitnessLimit: -Inf
          StallGenLimit: 50
          StallTimeLimit: Inf
                 TolFun: 1.0000e-006
                 TolCon: 1.0000e-006
```

```
  InitialPopulation: []
      InitialScores: []
     InitialPenalty: 10
      PenaltyFactor: 100
       PlotInterval: 1
        CreationFcn: @gacreationuniform
  FitnessScalingFcn: @fitscalingrank
       SelectionFcn: @selectionstochunif
       CrossoverFcn: @crossoverscattered
        MutationFcn: {[1x1 function_handle]  [1]  [1]}
 DistanceMeasureFcn: []
          HybridFcn: []
            Display: 'final'
           PlotFcns: []
         OutputFcns: []
         Vectorized: 'off'
        UseParallel: 'never'
```

GA: Options (2)

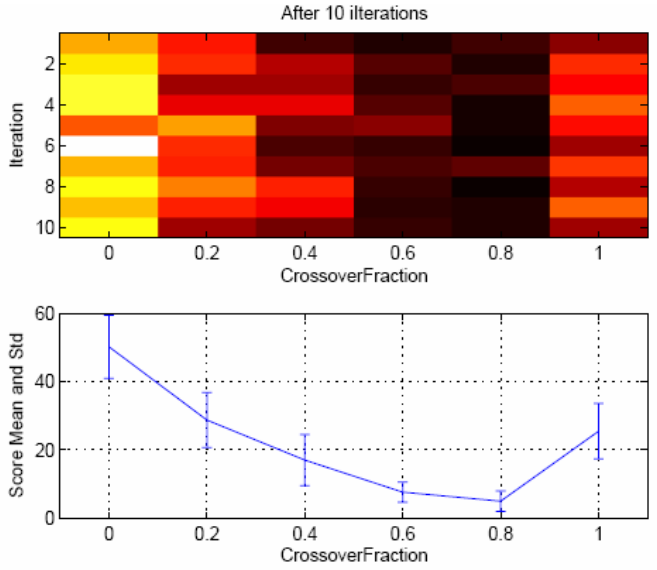
- Population Diversity
  - Setting the initial range / population size
- Fitness Scaling
  - Rank, Proportional, Top, Shift linear
- Selection
  - Stochastic uniform, Remainder, Uniform, Roulette, Tournament



GA: Options (3)

- Reproduction Options
  - Elite count, Crossover fraction
  - >>deterministicstudy
- Crossover
  - Scattered, Single point, Two point, Intermediate, Heuristic, Arithmetic

Population size	20	Children
Elite count	2	2
Crossover fraction	0.8	0.8*18→14
Mutation		4



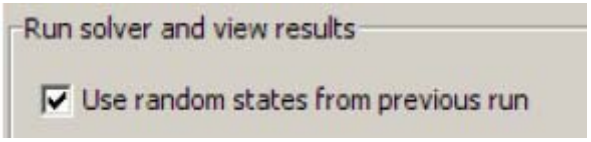
GA: Options (4)

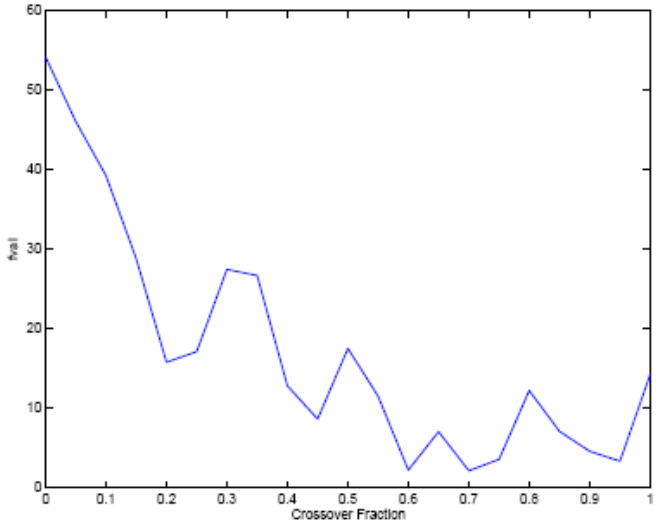
- Mutation
  - Mutation function: Gaussian, Uniform, Adaptive feasible
  - Scale: controls the standard deviation of the mutation
  - Shrink: controls the rate at which the average amount of mutation decreases
- Global vs. Local minima
  - Increase the initial range
- Using a Hybrid Function
  - optimization function that runs after the genetic algorithm terminates in order to improve the value of the fitness function
  - uses the final point from the genetic algorithm as its initial point
- Setting the Maximum Number of Generations
- Vectorizing the Fitness Function
  - 'Vectorized' On (compute the fitness for all individuals at once)
  - tic; ga(@fitnessfun, nvars); toc

Reproducing Results

– run the genetic algorithm with different settings for **Crossover fraction** to see which one gives the best results

```
options = gaoptimset('Generations',300);
rand('twister', 71); % These two commands are only included to
randn('state', 59); % make the results reproducible.
record=[];
for n=0:.05:1
 options = gaoptimset(options,'CrossoverFraction', n);
 [x fval]=ga(@rastriginsfcn, 10,[],[],[],[],[],[],[],options);
 record = [record; fval];
end
plot(0:.05:1, record);
xlabel('Crossover Fraction');
ylabel('fval')
```





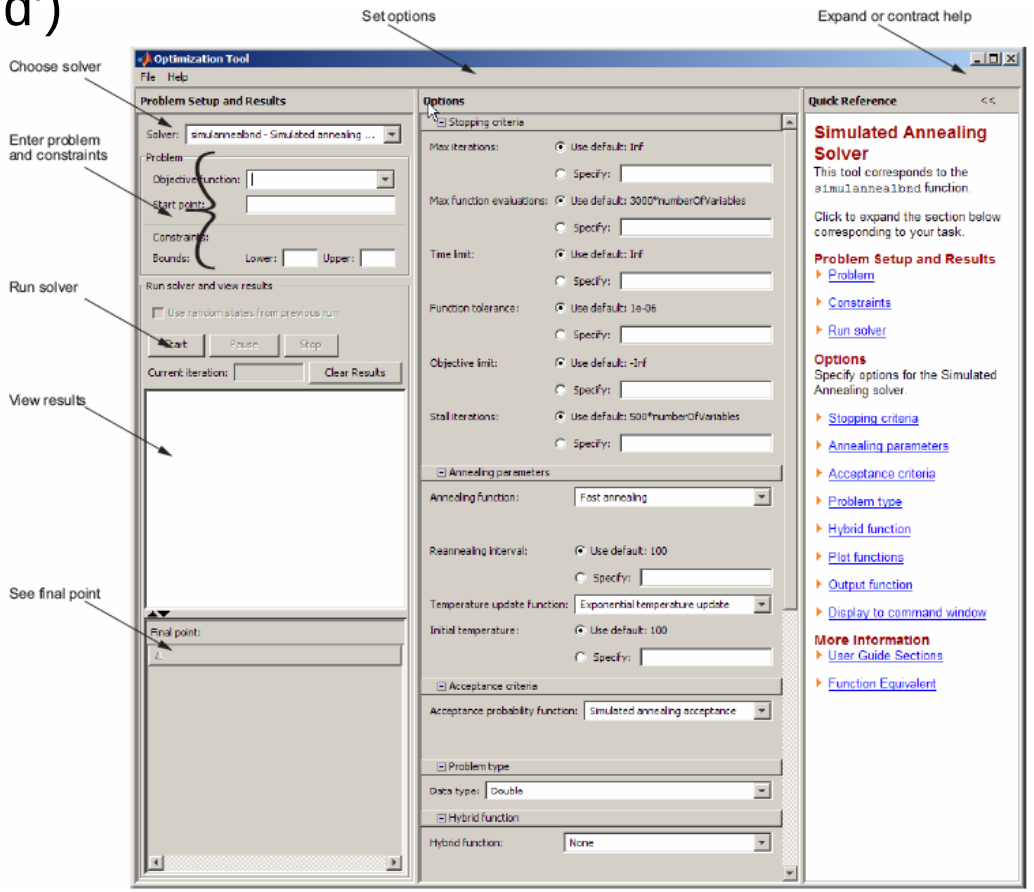
Simulated Annealing and Threshold Acceptance

- method for solving unconstrained and bound-constrained optimization problems
- The method models the physical process of heating a material and then slowly lowering the temperature to decrease defects, thus minimizing the system energy

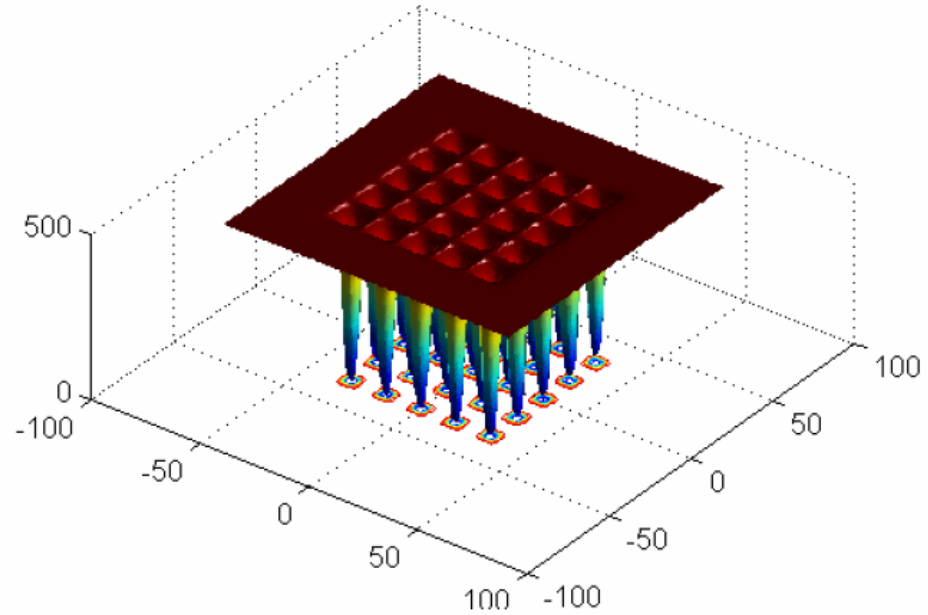
- Threshold acceptance
  - Instead of accepting new points that raise the objective with a certain probability, it accepts all new points below a fixed threshold (faster than SA)
  - The threshold is then systematically lowered, just as the temperature is lowered in an annealing schedule

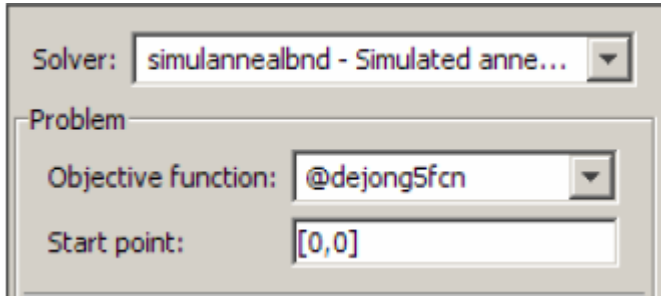
Function Call

- [x fval exitflag output] = simulannealbnd(@objfun, x0, lb, ub, options)
- [x, fval exitflag output] = threshacceptbnd(@objfun, x0, lb, ub, options)
- optimtool('simulannealbnd')



De Jong's Fifth Function






```
fun = @dejong5fcn;
[x fval] = simulannealbnd(fun, [0 0])
```

```
x =
   -31.9779  -31.9595
fval =
    0.9980
```

Terminology

- Objective function
- Temperature: `InitialTemperature,TemperatureFcn`
  - It determines the probability of accepting a worse solution at any step and is used to limit the extent of the search in a given dimension
- Annealing schedule: `TemperatureFcn`
  - The rate by which the temperature is decreased
  - The slower the rate of decrease, the better the chances are of finding an optimal solution, but the longer the run time
- Reannealing: `ReannealInterval`
  - Raises the temperature after a certain number of new points have been accepted, and starts the search again at the higher temperature
- Threshold acceptance
  - accepts a worse point if the objective function is raised by less than a fixed threshold

SA: Options

- options=saoptimset('simulannealbnd')
- options=saoptimset('threshacceptbnd')

```
options =
          AnnealingFcn: @annealingfast
       TemperatureFcn: @temperatureexp
        AcceptanceFcn: @acceptancesa
              TolFun: 1.0000e-006
       StallIterLimit: '500*numberofvariables'
          MaxFunEvals: '3000*numberofvariables'
            TimeLimit: Inf
              MaxIter: Inf
       ObjectiveLimit: -Inf
              Display: 'final'
      DisplayInterval: 10
            HybridFcn: []
       HybridInterval: 'end'
             PlotFcns: []
         PlotInterval: 1
           OutputFcns: []
    InitialTemperature: 100
      ReannealInterval: 100
              DataType: 'double'
```

Nonlinear Constraint Solver

- Augmented Lagrangian Pattern Search (ALPS) algorithm
- Augmented Lagrangian Genetic Algorithm (ALGA)

$$\begin{aligned} &\underset{x}{\text{Minimize}} \quad f(x) \\ &\text{such that} \\ &\quad C_i(x) \le 0, i = 1 \ldots m \\ &\quad C_i(x) = 0,\ i = m+1 \ldots mt \\ &\quad Ax \le b \\ &\quad A_{eq}x = beq \\ &\quad LB \le x \le UB \end{aligned}$$

$$\Theta(x,\lambda,s,\rho) = f(x) - \sum_{i=1}^{m} \lambda_i s_i \log(s_i - c_i(x)) + \sum_{i=m+1}^{mt} \lambda_i c_i(x) + \frac{\rho}{2} \sum_{i=m+1}^{mt} c_i(x)^2$$