Genetic algorithms

- Based on “survival of the fittest.”
- Start with “population of points.”
- Retain better points
- Based on “natural selection.”
- (as in genetic processes)
Genetic algorithms

• Maximise $f(x)$, $x_i^L \leq x_i \leq x_i^u$
• Code every variable using binary string
  Eg.(00000) – (11111)
  gives $2^5$ values
• The range of each variable is mapped to this range
• $x_i=x_i^L+(x_i^U-x_i^L)/(2^{li}-1) \times \text{decoded value}(s_i)$
• A value in this range represents actual value of variable. (eg. If $0 \leq x < 8$ needs values if desired accuracy desired is 0.5)
• Every variable is represented by set of strings like this.
Objective/Fitness function
- Convert minimization problem to maximization problems
- \( F(x) = \frac{1}{1+f(x)} \)
- Not \( F(x) = -f(x) \)
- Evaluates “Fitness” of a string.

Generate a population of strings
- operate using operators (reproduction, crossover, and mutation)
- get a new population.
GA operators

• Reproduction/Selection
  - select a set of above average strings
  - probabilistically add more instances of the strings to the mating pool
    e.g. probability of selecting $i^{th}$ string $= \frac{F_i}{\sum F_i}$
• Population size remains constant at each stage
Initial Population and the mating pool
Crossover operator

- Choose two strings randomly from the mating pool.
- Choose a point in the strings randomly
- Exchange all bits to one side of the point in both strings.
- Two new points in the population

```
0 0 0 0 0
1 1 1 1 1
```

Parent string

```
0 0 1 1 1
1 1 0 0 0
```

Child string
Cross over

- Not all strings are chosen for cross over.
- Randomly chosen with a cross over probability $P_c$
- $N \times P_c$ strings are used in crossover
- $N \times (1 - P_c)$ are left unchanged
Population after crossover
Mutation operator

- Consider a population

0 0 1 1 0 0 1
0 1 1 0 1 1 0
0 1 0 1 0 1 0
0 1 1 0 1 1 0
0 0 1 0 1 0 1

- No amount of reproduction /crossover can change 1<sup>st</sup> bit to 1: optimum may be missed.

- Change bits from 0 to 1 (or 1 to 0) with a probability of $P_m$.

- $P_m$ should be very small.
Population after mutation
Characteristics of the operators

- Reproduction selects good strings.
- Crossover combines two good strings to come with better strings (hopefully).
- Mutation alters a string.
- Bad strings are removed in next generation (by the reproduction operator).
Result from iteration to iteration
GAs vs Traditional

- Uses string coding of variables.
- Search space becomes discrete.
- Discrete functions can also be handled easily.
- Uses a population of points.
- Larger likelihood of getting a global solution
- Multiple optimals can be found easily
- No information is needed for problem domain (like slope etc.)
Constrained Optimization problems using Gas

• Add penalty functions to minimization problem
  \[ P(x) = f(x) + \sum u_j \langle g_j(x) \rangle^2 + \sum v_k [h_k(x)]^2 \]

• Convert to maximization problem
  \[ F(x) = 1/(1+P(x)) \]

• Penalty parameter need not be updated from iteration to iteration

• A large value of R can be taken

• Multi modal functions can be handled easily
Modifications in GAs

• If one point in the population is very good
  – Many copies of the same after reproduction
  – Other optimal points may be difficult to get
  – Use scaling after each generation
  – $S(x) = aF(x) + b$
Modifications in Gas

• In crossover the right side bits have a greater probability of selection
• Hence use a two point crossover
• Choose two points.
• Swap bits between the two locations.
Simulated Annealing

• Resembles the cooling process of molten metals through annealing
• At high temperature the atoms can move freely
• At low temperatures, movement gets restricted
• To obtain, the absolute minimum energy state, the system is cooled slowly
SA (contd)

• Introduce a Temperature like parameter
• Boltzmann Probability Distribution

\[ P(E) = e^{-\frac{E}{kT}} \]

• K – Boltzman’s constant
• P – Probability of being in energy state E
• At high T, finite probability of being at any state
SA (contd)

- $P(E(t+1)) = \min [1, \exp(-\Delta E/kT)]$
  - Where $\Delta E = E(t+1) - E(t)$
- If $\Delta E$ is negative, $P = 1$, so accept new point
- Otherwise, accept only with a small probability
- Reduce $T$ slowly over the iterations
1. Choose $x^0$, $\varepsilon$, $T$, $n$ (number of iterations at each $T$); $t = 0$

2. Calculate $x^{t+1} =$ random point in neighborhood of $x^t$.

3. If $\Delta E = E(x^{t+1} - E(x^t)) < 0$, set $t = t + 1$
   1. Else generate a random number $r$ in $(0, 1)$
   2. If $r < \exp(- \Delta E / T)$ $t = t + 1$ else GOTO 2

4. If $x^{t+1} - x^t < \varepsilon$ and $T$ is small STOP
   1. Else if $(t \mod n) = 0$ lower $T$
   2. GOTO 2