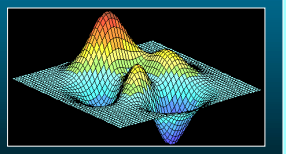
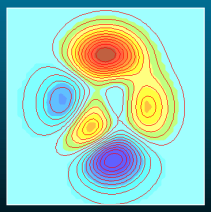
Derivative-based Optimization

(chapter 6)

Bill Cheetham cheetham@cs.rpi.edu
Kai Goebel goebel@cs.rpi.edu

Determine search direction according to an objective function's derivative information





Does not work with local maxima

Derivative-Free Optimization

(chapter 7)

Bill Cheetham cheetham@cs.rpi.edu
Kai Goebel goebel@cs.rpi.edu

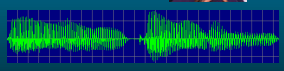
Derivative-Free Optimization

Genetic algorithms (GAs)
Simulated annealing (SA)

Genetic Algorithms

Motivation

- Look at what evolution brings us?
  - Vision
  - Hearing
  - Smell
  - Taste
  - Touch
  - Learning and reasoning
- Can we emulate the evolutionary process with today's fast computers?



Genetic Algorithms

Terminology:

- Population - a set of possible solutions to a problem
- Fitness function - how to evaluate the quality of a member of the population
- Encoding schemes - a binary representation for a member of the population
- Selection - how to determine which members of the population will survive to the next generation
- Crossover - combining two members of the population
- Mutation - changing a single member of the population
- Elitism - allowing the best members to pass to the next generation

Soft Computing: Derivative-Free Optimization

Genetic Algorithms

Binary encoding

(11, 6, 9) → 1011 0110 1001

Chromosome

Gene

Crossover

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

→

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

Crossover point

Mutation

1 0 0 1 1 1 1 0 → 1 0 0 1 1 0 1 0

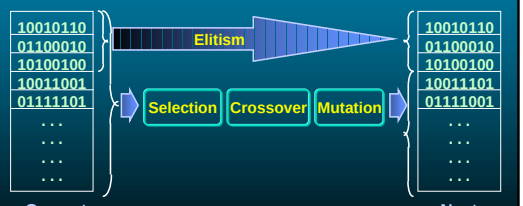
Mutation bit

7

Soft Computing: Derivative-Free Optimization

Genetic Algorithms

Flowchart



Current generation

Next generation

8

Soft Computing: Derivative-Free Optimization

Genetic Algorithms

Example: Find the max. of the "peaks" function

z = f(x, y) = 3*(1-x)^2*exp(-(x^2) - (y+1)^2) - 10*(x/5 - x^3 - y^5)*exp(-x^2-y^2) -1/3*exp(-(x+1)^2 - y^2).

9

Soft Computing: Derivative-Free Optimization

Genetic Algorithms

Derivatives of the "peaks" function

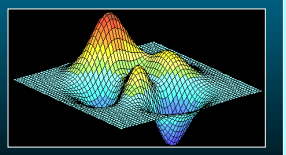
- dz/dx = -6*(1-x)*exp(-x^2-(y+1)^2) - 6*(1-x)^2*x*exp(-x^2-(y+1)^2) - 10*(1/5-3*x^2)*exp(-x^2-y^2) + 20*(1/5*x-x^3-y^5)*x*exp(-x^2-y^2) - 1/3*(-2*x-2)*exp(-(x+1)^2-y^2)
- dz/dy = 3*(1-x)^2*(-2*y-2)*exp(-x^2-(y+1)^2) + 50*y^4*exp(-x^2-y^2) + 20*(1/5*x-x^3-y^5)*y*exp(-x^2-y^2) + 2/3*y*exp(-(x+1)^2-y^2)
- d(dz/dx)/dx = 36*x*exp(-x^2-(y+1)^2) - 18*x^2*exp(-x^2-(y+1)^2) - 24*x^3*exp(-x^2-(y+1)^2) + 12*x^4*exp(-x^2-(y+1)^2) + 72*x*exp(-x^2-y^2) - 148*x^3*exp(-x^2-y^2) - 20*y^5*exp(-x^2-y^2) + 40*x^5*exp(-x^2-y^2) + 40*x^2*exp(-x^2-y^2)*y^5 - 2/3*exp(-(x+1)^2-y^2) - 4/3*exp(-(x+1)^2-y^2)*x^2 -8/3*exp(-(x+1)^2-y^2)*x
- d(dz/dy)/dy = -6*(1-x)^2*exp(-x^2-(y+1)^2) + 3*(1-x)^2*(-2*y-2)^2*exp(-x^2-(y+1)^2) + 200*y^3*exp(-x^2-y^2)-200*y^5*exp(-x^2-y^2) + 20*(1/5*x-x^3-y^5)*exp(-x^2-y^2) - 40*(1/5*x-x^3-y^5)*y^2*exp(-x^2-y^2) + 2/3*exp(-(x+1)^2-y^2)-4/3*y^2*exp(-(x+1)^2-y^2)

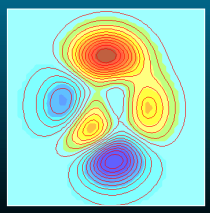
10

Soft Computing: Derivative-Free Optimization

Genetic Algorithms

Plotting the function



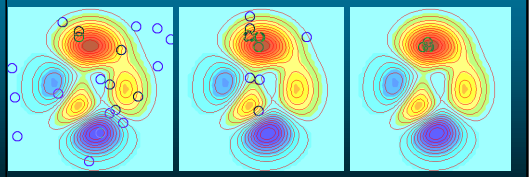


11

Soft Computing: Derivative-Free Optimization

Genetic Algorithms

GA process:

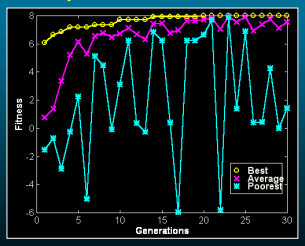


Initial population | 5th generation | 10th generation

12

Genetic Algorithms

Performance profile



Example

Let us consider the equation:

a+2b+3c+4d=30,

where a,b,c,d are positive integers

given the constraints 1 =< a,b,c,d =< 30

GA systems allow for a solution to be reached quicker since "better" solutions have a better chance of surviving and procreating, as opposed to random search

Example

First we will choose 5 random initial solution sets

Chrom	(a,b,c,d)	bit representation
1	(1,28,15,3)	(00001 11100 01111 00011)
2	(14,9,2,4)	(01110 01001 00010 00100)
3	(13,5,7,3)	(01101 00101 00111 00011)
4	(23,8,16,19)	(10111 01000 10000 10011)
5	(9,13,5,2)	(01001 01101 00101 00010)

Example

fitness function

a+2b+3c+4d - 30

Chromosome	Fitness Value
1	|114-30|=84
2	|54-30|=24
3	|58-30|=28
4	|163-30|=133
5	|58-30|=28

Example

more desirable fitness values are more likely to be chosen as parents

Chromosome	Likelihood
1	(1/84)/0.135266 = 8.80%
2	(1/24)/0.135266 = 30.8%
3	(1/28)/0.135266 = 26.4%
4	(1/133)/0.135266 = 5.56%
5	(1/28)/0.135266 = 26.4%

Example

Crossover

Father	Mother	Offspring
(13 | 5,7,3)	(1 | 28,15,3)	(13,28,15,3)
(9,13 | 5,2)	(14,9 | 2,4)	(9,13,2,4)
(13,5,7 | 3)	(9,13,5 | 2)	(13,5,7,2)
(14 | 9,2,4)	(9 | 13,5,2)	(14,13,5,2)
(13,5 | 7, 3)	(9,13 | 5, 2)	(13,5,5,2)

Example

Mutation

Before-Mutation	After Mutation
(1,28,15,3)	(1,29,15,3)
(14,9,2,4)	(14,25,2,4)
(00001 11100 01111 00011)	(00001 11101 01111 00011)
(01110 01001 00010 00100)	(01110 11001 00010 00100)

Example

Now we can calculate the fitness values for the new generation of offspring.

Offspring	Fitness Value
(13,28,15,3)	|126-30|=96
(9,13,2,4)	|57-30|=27
(13,5,7,2)	|57-30|=22
(14,13,5,2)	|63-30|=33
(13,5,5,2)	|46-30|=16

Last GA Slide

Break up into groups of 3 or 4
Design a GA to determine the maximum of the peaks function in the range $-2 < x < 3$, $-2 < y < 3$

1) Design a binary chromosome that can represent the possible solutions accurate to two digits after the decimal point?
2) How do you decode your chromosome to evaluate the peaks function?
3) How do you determine the probability of selection?
4) What parameters do you use for population size, generations, elitism, crossover rate, mutation rate?

2) How do you decode your chromosome to evaluate the peaks function?

```
function num = bit2num(bit, range)
%
%  bit2num([1 1 0 1], [0, 15])
%  bit2num([0 1 1 0 0 0 1], [0, 127])

%  Roger Jang, 12-24-94

integer = polyval(bit, 2);
num = integer*((range(2)-range(1))/(2^length(bit)-1))
    + range(1);
```

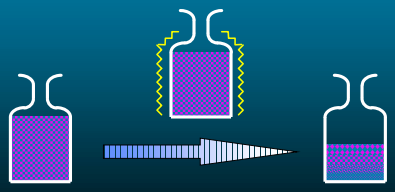

3) How do you determine the probability of selection?

```
% rescaling the fitness
fitness = fitness - min(fitness);  % keep it positive
total = sum(fitness);
if total == 0,
    fprintf(' Warning: converge to a single point\n');
    fitness = ones(popu_s, 1)/popu_s; % sum is 1
else
    fitness = fitness/sum(fitness);       % sum is 1
end
cum_prob = cumsum(fitness);
```

Soft Computing: Derivative-Free Optimization

Simulated Annealing

Analogy



25

Soft Computing: Derivative-Free Optimization

Simulated Annealing

Terminology:

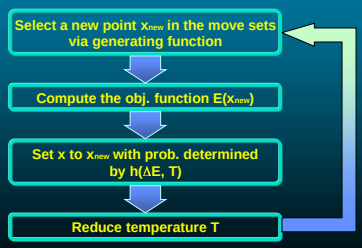
- Objective function E(x): function to be optimized
- Move set: set of next points to explore
- Generating function: to select next point
- Acceptance function h(ΔE, T): to determine if the selected point should be accept or not. Usually h(ΔE, T) = 1/(1+exp(ΔE/(cT)).
- Annealing (cooling) schedule: schedule for reducing the temperature T

26

Soft Computing: Derivative-Free Optimization

Simulated Annealing

Flowchart



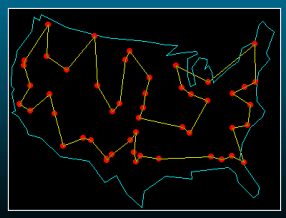
27

Soft Computing: Derivative-Free Optimization

Simulated Annealing

Example: Travel Salesperson Problem (TSP)

How to transverse n cities once and only once with a minimal total distance?

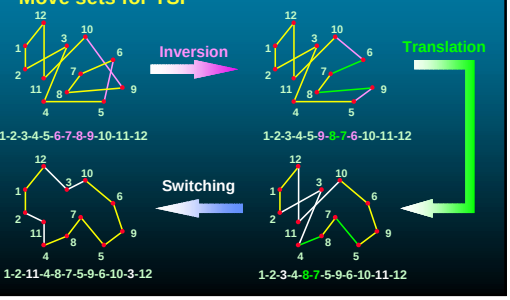


28

Soft Computing: Derivative-Free Optimization

Simulated Annealing

Move sets for TSP

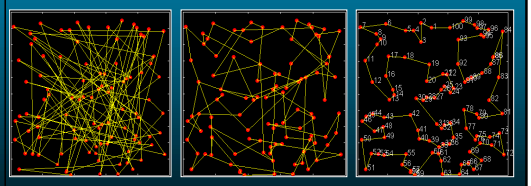


29

Soft Computing: Derivative-Free Optimization

Simulated Annealing

A 100-city TSP using SA

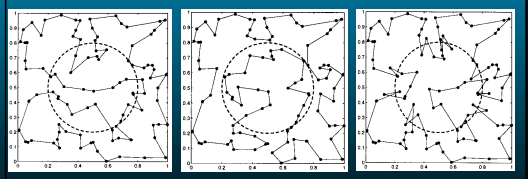


Initial random path | During SA process | Final path

30

Simulated Annealing

100-city TSP with penalties when crossing the circle



Penalty = 0 | Penalty = 0.5 | Penalty = -0.3