MAE 552 Heuristic Optimization

Instructor: John Eddy

Lecture #14

2/25/02

Evolutionary Algorithms

Practical Implementation Issues

The next set of slides will deal with some issues encountered in the practical implementation of a genetic algorithm.

Specifically constraint handling and convergence criteria.

Constraint Handling

What if we have a constrained problem?

Typically would use a penalty function to worsen the fitness of any designs which violate constraints.

Constraint Handling

Example of a penalty term.

$$P(\bar{x}) = \sum_{j=1}^{m} \max[g_j(\bar{x}), 0]^2 + \sum_{k=1}^{l} h_k(\bar{x})^2$$

This will provide a numerical violation value. It is up to you to decide how this will effect your fitness.

Convergence

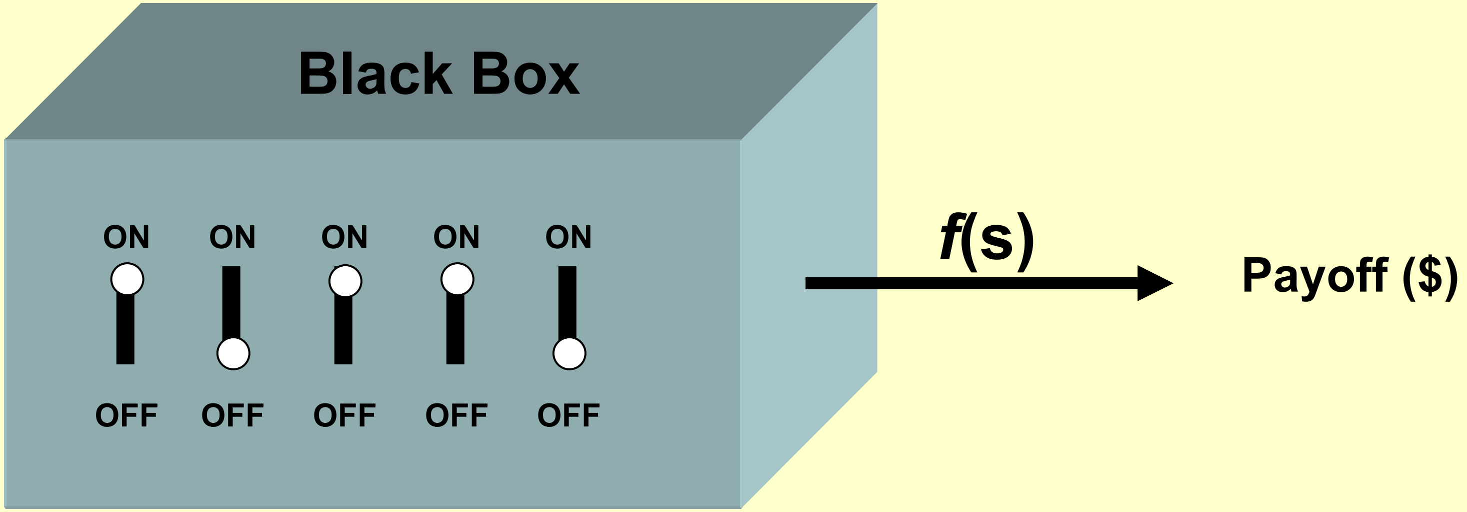
Common ways of determining convergence:

1. No change in best quality design over chosen number of gens.
2. No change in average quality of population of chosen number of gens.
3. Set a maximum allowable number of generations.
4. Set a maximum allowable number of objective function evaluations.

Putting It All Together

```
begin
    t = 0
    initialize P(t)
    evaluate P(t)
    while (not converged) do
        t = t + 1
        select P(t) from P(t-1)
        alter P(t) (variation operators)
        evaluate P(t)
    end do while
end
```

Example



We will toggle the switches to alter the payoff.

Example

According to our 5 basic components, what should our first step be?

Considering the nature of our problem, is there an encoding scheme that we think may work well?

How can we encode solutions to our problem in this way?

Example

We have decided on a vector of bit encoding because each of our switches has 2 states (on and off). So each of our encoded strings will look like this:

[{0,1}, {0,1}, {0,1}, {0,1}, {0,1}]

An example of which would be:

[0, 1, 0, 1, 0]

Example

What’s next?

We must decide how to generate an initial population.

We will use a random approach. This would be like flipping a coin a sufficient number of times to fill each of our initial designs with 1’s and 0’s.

Example

Now what?

We must decide how we are going to evaluate our design fitness. This is not (necessarily) the objective function value.

In our case, we will consider the output (payoff) of our black box to be the design fitness (We will treat the switch settings as a binary number and our fitness will be the square of the decimal equivalent).

Example

At this point, we have to decide how to implement our selection and variation operators.

Selection

As mentioned when presenting selection strategies, we will use the f/f_{avg} approach.

Example

Variation - Crossover

Recall the approaches presented for crossover on vectors of bits. (random parameter selection and n-pt crossover)

We will choose to use single point uniform crossover. (uniform refers to the means by which we generate our random crossover pt.)

Example

Variation – Mutation

We will use random bit mutation which in this case is exactly the same as random design variable reassignment since each of our design variables is in itself a bit.

Example

Parameters

We need to decide on values for the parameters of our algorithm. ie: mutation and crossover rate, population size, convergence criteria (note that we are not going to run this optimization to completion so we will not explicitly worry about convergence).

Example

Execution

We will maintain a population of 4 members and perform 2 crossovers per generation. With 4 members, there are 6 possible parings (excluding order) so this is a 33% crossover rate. We will use a 5% mutation rate (bitwise).

Example

Initial Population – Gen. #0 (recall $f = x^2$):

string	X	f	f/f_{avg}	# cop.
0 1 1 0 1	13	169	0.6	1
1 1 0 0 0	24	576	1.97	2
0 1 0 0 0	8	64	0.22	0
1 0 0 1 1	19	361	1.23	1
	avg	292.5		

Example

Generation #1 - variation

string	after cross.	after mutation
0 1 1 0 1	0 1 1 0 0	0 1 1 0 0
1 1 0 0 0	1 1 0 0 1	1 1 0 0 1
1 1 0 0 0	1 1 0 1 1	1 1 0 1 1
1 0 0 1 1	1 0 0 0 0	1 0 1 0 0

Example

Generation #1 - evaluation + selection for 2

string	X	f	f/f_{avg}	# cop.
0 1 1 0 0	12	144	0.30	0
1 1 0 0 1	25	625	1.32	1
1 1 0 1 1	27	729	1.54	2
1 0 1 0 0	20	400	0.84	1

avg 474.5

Example

Generation #2 - variation

string	after cross.	after mutation
1 1 0 0 1	1 1 0 1 1	1 1 0 0 1
1 1 0 1 1	1 1 0 0 1	1 1 0 0 1
1 1 0 1 1	1 1 0 0 0	1 1 0 0 0
1 0 1 0 0	1 0 1 1 1	1 0 1 1 1

Example

Generation #2 - evaluation + selection for 3:

string	X	f	f/f_{avg}	# cop.
1 1 0 0 1	25	625	1.06	1
1 1 0 0 1	25	625	1.06	1
1 1 0 0 0	24	576	0.98	1
1 0 1 1 1	24	529	0.90	1

avg 588.8