

# **CSE/ECE 848**

## **Introduction to Evolutionary Computation**

**Module 2, Lecture 8, Part 4c**  
**More Theory—Practical Training**

**Erik D. Goodman, Executive Director  
BEACON Center for the Study of Evolution in  
Action  
Professor, ECE, ME, and CSE**

# Now for some GA “Practical Training”

## A Traditional Way to Do GA Search...

- Population large ( $\sim 64$  to  $1024\dots$ )
- Mutation rate (per locus)  $\sim 1/L$
- Crossover rate moderate ( $< 0.3$ )
- Selection scaled (or rank/tournament, etc.) such that Schema Theorem allows new BB's to grow in number, but not lead quickly to premature convergence

# Schema Theorem and Representation/ Crossover Types

If we use a different type of representation or different crossover operator:

Will be governed by a different schema theorem, using same ideas about disruption of schemata.

# Uniform Crossover & Linkage

- 2-pt crossover is superior to 1-point
- *Uniform* crossover chooses allele for each locus at random from either parent
- If “biased”, chooses most loci from same parent
- Uniform crossover is more disruptive than 1-pt or 2-pt crossover
- BUT uniform is unbiased relative to linkage
- If all you need is small populations and a “rapid scramble” to find good solutions, uniform crossover sometimes works better – but is this what you need a GA for? Hmmmm...
- Otherwise, try to lay out chromosome for good linkage, and use 2-pt crossover (or Booker’s 1987 *reduced surrogate crossover*, (described soon!))

## Inversion – An Idea to Try to Improve Linkage

- Tries to re-order loci on chromosome – BUT NOT changing the meanings of loci in the process
- Means must treat each locus as (index, value) pair. Can then reshuffle pairs at random, let crossover work with them in order APPEAR on chromosome, but fitness function unscrambles, keeps association of values with indices of fields, unchanging.

# Classical Inversion Operator

- Example: reverses field pairs  $i$  through  $k$  on chromosome  $(a, v_a), (b, v_b), (c, v_c), (d, v_d), (e, v_e), (f, v_f), (g, v_g)$
- After inversion of positions 2-4, yields:  
 $(a, v_a), (d, v_d), (c, v_c), (b, v_b), (e, v_e), (f, v_f), (g, v_g)$
- Now fields  $a, d$  are more closely linked, 1-pt or 2-pt crossover less likely to separate them
- Holland invented it, but in practice, seldom used – must run problem for an enormous time to have such a second-level effect be useful. Need to do on population level or tag each inversion pattern (and force mates to have matching tags) or do repairs to crossovers to keep chromosomes “legal” – i.e., possess one pair of each type.

# Inversion is NOT for use as an Operator to Solve Permutation Problems

In contrast, if trying to solve for the best permutation of  $[0, N]$ , use reordering crossovers or other representations (like Random Keys) – we'll discuss later. That's NOT inversion!