Real-Coded GA and Interval-Schemata

Interval-Schemata and Crossover

Real-coded genes for functions of continuous variables

Forms of crossover in real-coded GAs

- Parameter-bounded crossover (Davis, 1991)
- Linear crossover (Wright, 1991)
- Flat crossover (Radcliff, 1990): BLX-0.0

Blend crossover, BLX-α

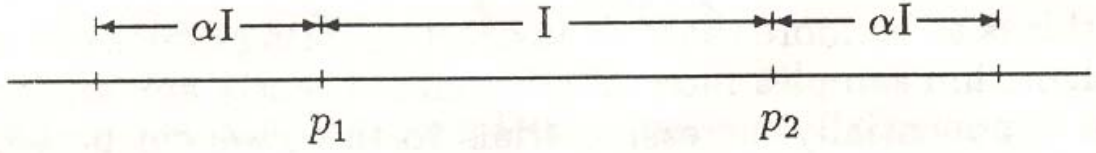

Figure 1: BLX-α

Interval-Schemata and Crossover

Interval schema vs. symbol schema

36 interval schemata for a parameter ranged over [0, 7]

[0, 7], [0, 6], [1,7],…, [0, 0], [1, 1], …,[7, 7]

Number of interval schemata over the range of integers $n = 2^L$ is $\Sigma\, i = n(n+1)/2$

Interval-Schemata and Crossover

How crossover preserves and explores interval schemata?

Parameter-bounded crossover:

- Offspring are members of the same interval schemata of which the parents are common members
- Strongly biased toward certain interval schemata over others

Flat crossover (BLX-0.0): much less biased, i.e., many new interval schemata are potentially reachable in a single crossover

Interval-Schemata and Crossover

IPGA vs. SPGA

- long interval schemata ≅ low order symbol schemata
  - [4, 7] = 1**, [6,7] = 11*
- IPGA (SPGA) progressively focus on shorter interval (higher order) schemata
- IPGA (SPGA) narrows the search to certain contiguous regions (partition)
- IPGA is limited by the max and min values of the parameters represented in the population
- IPGA (SPGA) exploits the local continuity of a function (discrete similarity)

IPGA fails to propagate good schemata

Consider a parameter with optimum at one of the extrema (i.e., 0 in IPGA and 000 in SPGA) that ranges over 2^L values

2X and UX are more successful than BLX-0.0

In BLX-0.0, the expected value of a randomly generated gene differs from the optimum by one half the range 2^{L-1}

The probability of propagating the optimum when mated with randomly chosen individual is $2/2^L = 1/2^{L-1}$

In UX, one half the bits and the probability is $1/2^{L/2}$

x-----x-----x-----x-----x-----x-----x-----x

IPGA	0	1	2	3	4	5	6	7
SPGA	000	001	010	011	100	101	110	111

Premature convergence

BLX-0.0 is less likely to prematurely converge to the values that correspond to the lower order bits (closer to the parents)

2X is much more likely to prematurely converge on the lower order bits because 2X is good at preserving contiguous chunks of the string intact

UX has no positional bias: Better at searching the lower order bits than 2X, but not as good as BLX-0.0

Unless the extrema in the initial population envelop the optimal point, it cannot be reached via BLX-0.0

This optimal-extrema can be overcome by the BLX-α

$\alpha = 0.5$ balances the convergent and divergent tendencies in the absence of selection pressure