EVOLUTIONARY COMPUTATION THEORY

Schema Theorem

- Holland's schema theorem.
- Add symbol ‘*’ to binary alphabet. It matches both 0 and 1
- Assume fixed symbols are part of the solution, * is don’t care
- Order of a schema $o(S)$: number of fixed symbols
- Defining length of a schema $\delta(S)$: the distance between the first and the last fixed symbols
- Fitness proportionate selection. Crossover (with p_c) and mutation (with p_m) assumed to be destructive
- Number of matches for S at time t in the pool: $\xi(S; t)$

- $$\xi(S,t+1) \geq \frac{\xi(S,t)\,f(S)}{f_{avg}}\left(1-\frac{p_c\,\delta(S)}{(l-1)}-p_m\,o(S)\right)$$

- **Schema Theorem**:
  Short, low-order, above-average schemata receive exponentially increasing trials in subsequent generations of a genetic algorithm.

- **Building Block Hypothesis:**
  A genetic algorithm seeks near-optimal performance through the juxtaposition of short, low-order, high performance schemata called the building blocks.

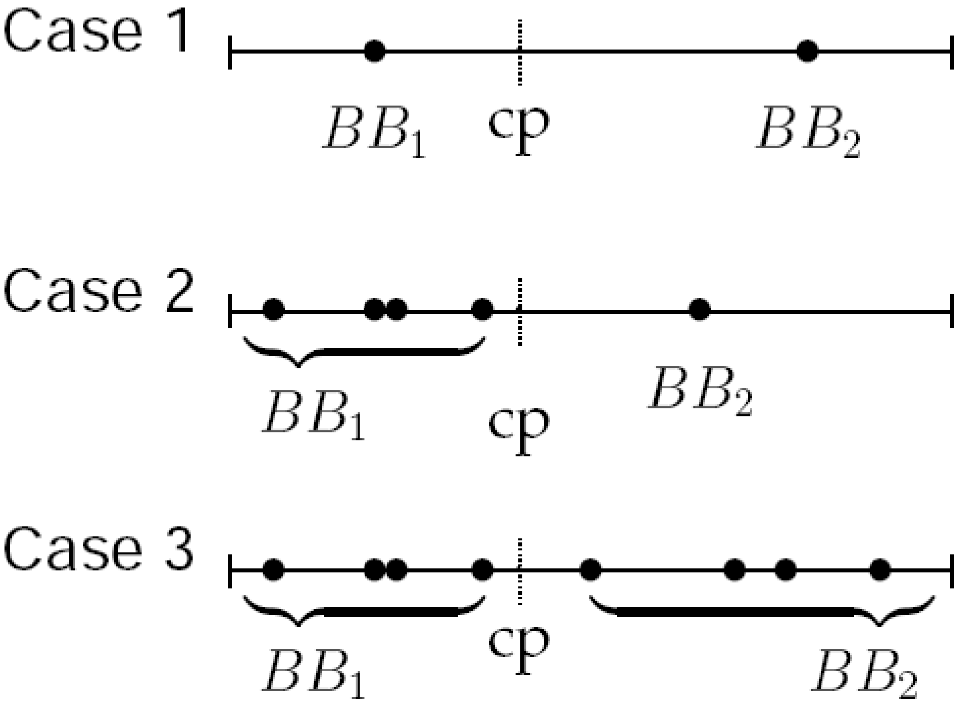
Crossover is destructive?

- Schema theorem assumes mutation and crossover are destructive in a schema. If both parents do not have the same schemata this assumption is correct.
- Find a better schema as a result of genetic operator is possible.
- **Mixing building blocks:**
  Combining two building blocks together in an individual as a result of crossover.
- Crossover of (`1***1*****`) and (`******0*0*`) at position 5 results in combination of two building blocks in one of the offsprings, (`1***1*0*0*`)

Mixing Building Blocks

- Single point crossover analysis
- Assume real value length 1 chromosomes containing infinite number of genes. (each gene is a point in the continuous line of chromosomes)
- Cases are:
  - Case 1
    Elements of the both building block are tightly placed.
  - Case 2
    Elements of the one building block are tightly placed the other is randomly placed.
  - Case 3
    Both elements are randomly placed

- Assume tight building blocks are single points. Mixing probabilities for crossover point α



$$P_{mixing}$$

$$2\alpha(1-\alpha)$$

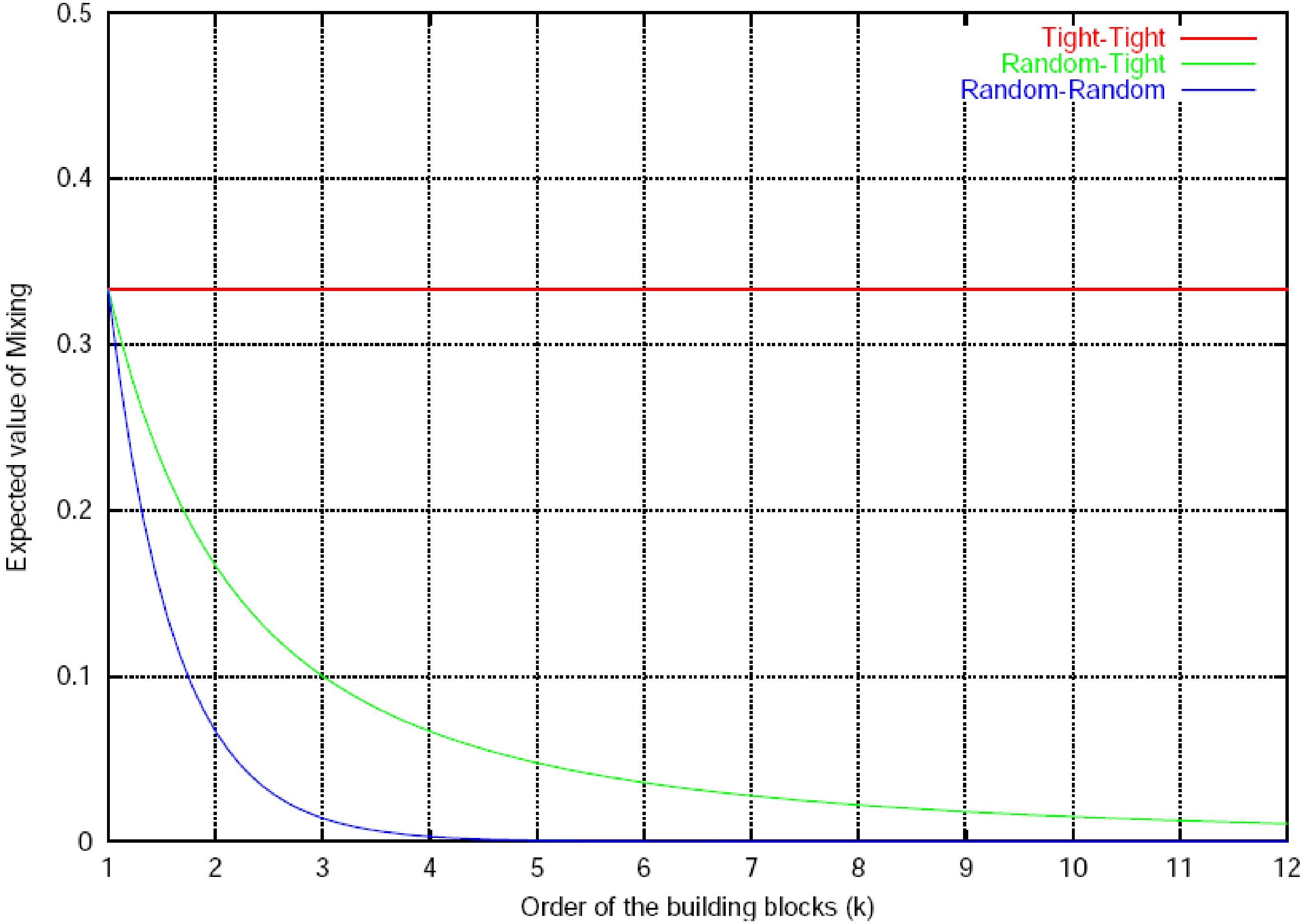
$$\alpha^k(1-\alpha)+\alpha(1-\alpha)^k$$

$$2\left[\alpha^k(1-\alpha)^k\right]$$

$$EV_{tight-tight}(k) = \int_0^1 2\alpha(1-\alpha)d\alpha = \frac{1}{3}$$

$$EV_{tight-random}(k) = \int_0^1 \alpha^k(1-\alpha) + \alpha(1-\alpha)^k d\alpha = \frac{2}{(k+1)(k+2)}$$

$$EV_{random-random}(k) = \int_0^1 2\left[\alpha^k(1-\alpha)^k\right] d\alpha = \frac{2(k!)^2}{(2k+1)!}$$



Expected value of mixing depending on k

Schema Context

- Fixed positions in each schema defines a partitioning of the search space
  `{*0**0*, *0**1*, *1**0*, *1**1*}`
- When a building block dominates the population, the search space is reduced to one of these partitions.
- What about non-fixed positions? Their frequency, how BB's interact with the others? Context of a schema
- Context of schema H is defined as a set of conditional probability values p_i for each individual in schema H.

$$C(H)=\left\{ p'_i=\frac{p_i}{\sum_{j\in H} p_j}, \qquad i\in H \right\}$$

- **Flat context**: all individuals have equal probability.

$$C(H)=\left\{p'_i=2^{k-l},\quad i\in H\right\}$$

- **Context in a population:**

m_i: number of individual representing schema i in population P

$$C_P(H)=\left\{p'_i=\frac{m_i}{\sum_{j\in H} m_j},\quad i\in H\right\}$$

- **Converged population:** schemata come from the same individual I.

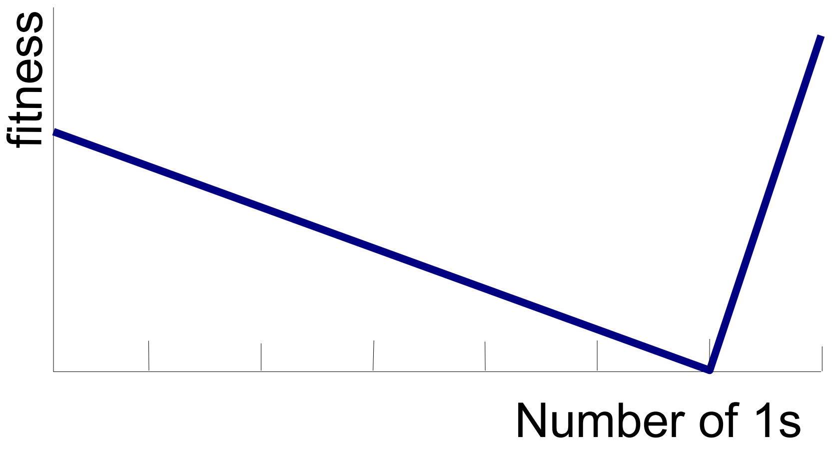
$$C(H)=\left\{p'_i=1,\ i\in I\,;\ p'_i=0\ \text{otherwise}\right\}$$

BB Superiority

- Survival of the BB depends on if it is superior in the population context.
- Assume:
  $f(0{*}{*}{*}{*}{*}) > f(1{*}{*}{*}{*}{*}), f(00{*}{*}{*}{*}) > f(01{*}{*}{*}{*}) > f(10{*}{*}{*}{*}) > f(11{*}{*}{*}{*})$
  but
  $f(111{*}{*}{*})$ is greater than all others.
- $f(111{*}{*}{*})$ is locally superior BB of solution 111111
- Deceptive problems: problems having superiority of low order schema favors a sub-optimal solution.

Deceptive Problems

- Example, a six bit trap function



- All lower order schema cause 000000 to converge.
- Some problems are deceptive in nature. Like problems with high ephistasis (semantic interaction among genes).

Linkage Learning

- Genes semantically linked however positioned not close in genotype have a smaller probability of survival and mixing.
- Linkage learning methods try to discover the linkage among the genes so that they treated specially during recombination.
- Messy GA (mGA) by Goldberg (1989) introduces a variable length position free encoding for adapting/preserving linkage. Other methods include:
  - LLGA (Harik)
  - Symbiotic Evolution
  - BOA (Pelikan), estimation of distribution.

Messy GA

- Messy encoding. Keep the position of gene in the chromosome together with its value:
  (3,0) (2,0) (5,1) (4,0) (1,1) = 10001
- Problems: under-specification and over-specification
- Over-specification:
  - Majority voting, majority wins. Deception favored?
  - First come first served. Positional priority.
- Under-specification. Harder problem:
  - Random pick
  - k-bit perturbations. Start from k=1 and as long as k+1 achieves better fitness, use it for k+1.

- Crossover. Cut and splice operations on chromosomes.
- Phases of mGA
  - Initialization: generation of all sub-strings of a certain length k
  - Primordial: selection of these enumerated building blocks against a particular template.
  - Juxtapositional: selection, cut, splice, and usual GA operations.
- A building block order k is assumed at each epoch. Start with k=1 to find a competitive template. Increase k at each epoch.
- Fast mGA, a modification of mGA to make operations more efficient.

Forces in Mixing

- Phenomena occurring in EC that affects the proper convergence:
  - Takeover: whole population consists of a single superior individual
  - Drift: random takeover any individual.
  - Cross-competition of BB's: BB's from different individuals not superior to each other end of with a poor representation of the solution.
- Mixing occurs if it finds a chance among these forces.

Forces in Mixing: Takeover

- Takeover time analysis. Use *s* tournament selection. Try to find # of generations for a superior individual to have copies through the whole the population.
- *n* tournaments of size *s* . Each superior individual can be selected with probability *s/n* . We expect to see it *s* times larger frequency at each generation.

$$s^{t_s} = n$$
$$t_s = \frac{\ln(n)}{\ln(s)}$$

- Mixing should occur before this. $t_s > c\, t_{\text{mix}}$

Forces of Mixing: Mixing time

- Mixing probability of two BB's of size *k* with uniform crossover (ignore the linkage).

$$p_{\mathrm{mix}} = 2\left(\frac{1}{2}\right)^{2\mathrm{k}}, \quad \text{if some bits are common } , \mu \le 2 \qquad p_{mix} = \frac{2}{2^{\mu k}}$$

- If $(n/2)p_c$ recombinations in one generation. Mixing time, the number of generations required to get a mixing is:

$$t_{\mathrm{mix}} = \frac{1}{(n/2)\, p_c\, p_{\mathrm{mix}}} = \frac{2^{\mu k}}{n\, p_c}$$

- Mixing time should be less than take over time:

$$t_s > c\, t_{\mathrm{mix}} \qquad \frac{\ln n}{\ln s} > c\, \frac{2^{\mu k}}{n\, p_c} \qquad n \ln(n) > c\, \frac{2^{\mu k} \ln(s)}{p_c}$$

Forces of Mixing: Drift

- Even with no selection pressure. Traits of an individual can have a random increase in population that is called drift.
- Drift time is proportional to the population size.

$$t_d = c' n$$

- Drift time should be less than takeover time:

$$t_d < t_s \quad \ln(s) < \frac{\ln(n)}{c' n}$$

- Not a practical consideration for *n* is sufficiently large.

Forces of Mixing: Cross-competition

- Selection gives *s* copies of each best individual. After first generation, *n/s* independent individuals which can construct better individuals.

p_0: ratio of bit positions already fixed correctly

$(1-p_0)^{n/s}$: correct alleleles that does not exist in the population

$\left[1-(1-p_0)^{n/s}\right]l$ alleles covered by the population

$(1-\alpha)l<\left[1-(1-p_0)^{n/s}\right]l \quad ,\alpha$:failure ratio

$$s>\frac{n\ln(1-p_0)}{\ln(\alpha)}$$

- If selection pressure is a significant fraction of the population size, we may expect to have cross-competition failure.

Forces of Mixing Revisited

- Takeover vs. mixing:

$$n \ln(n) > c \frac{2^{\mu k} \ln(s)}{p_c}$$

- Drift:

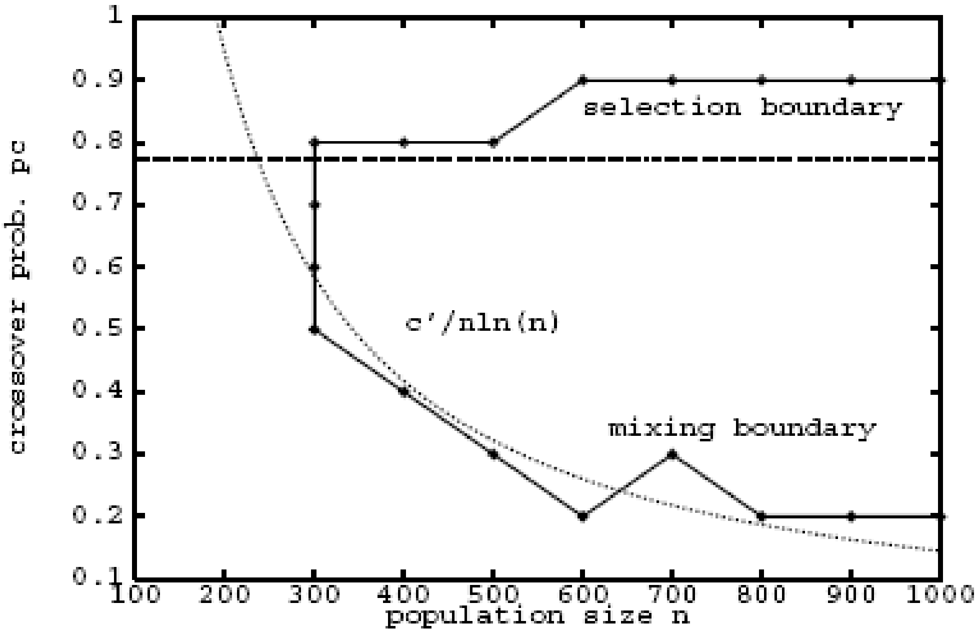
$$\ln(s) < \frac{\ln(n)}{c' n}$$

- Cross-competition:

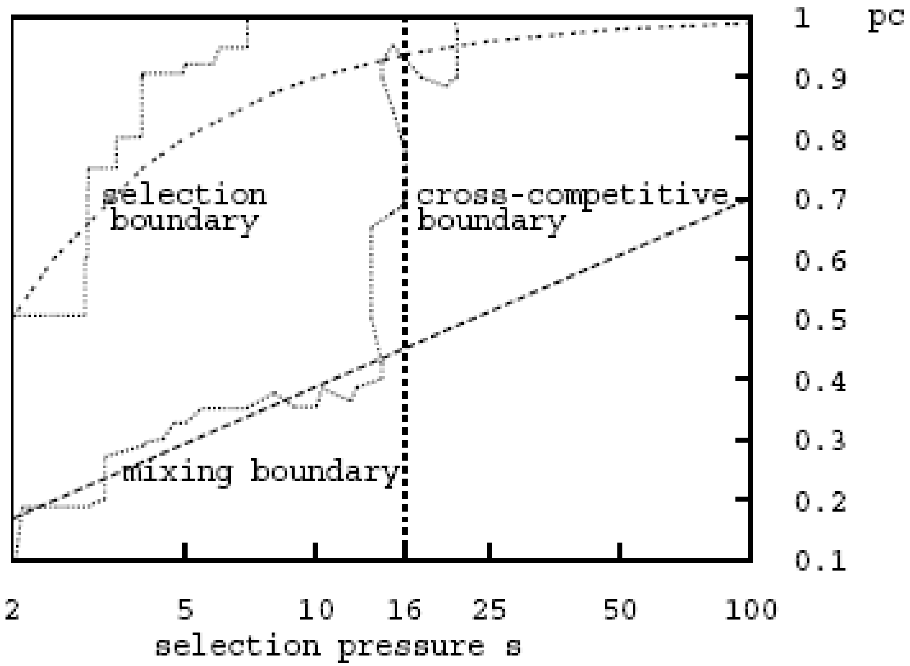
$$s < \frac{n \ln(1 - p_0)}{\ln(\alpha)}$$

Mixing Results

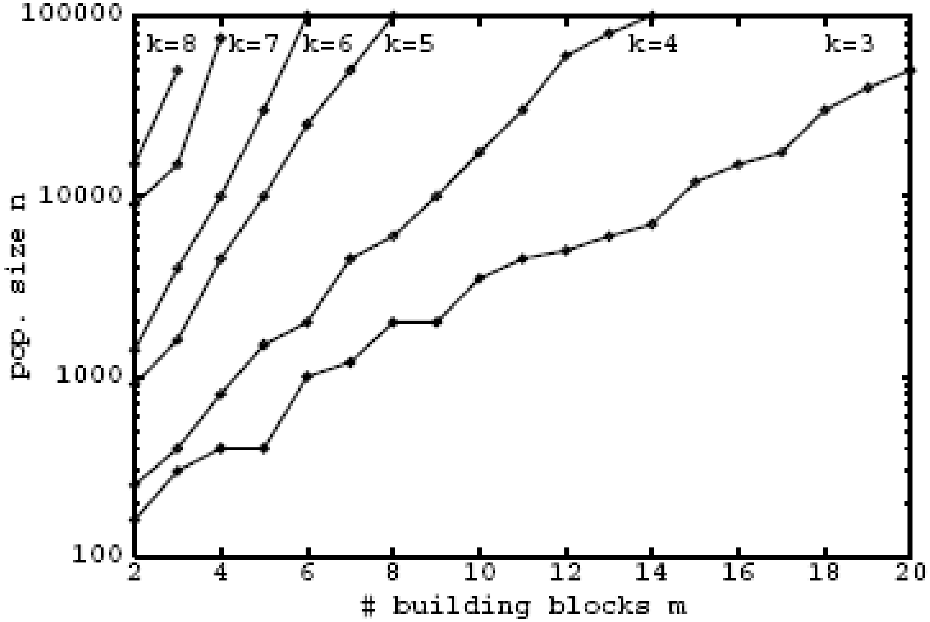
- Thierens, 1999, EC 7(4).



- Selection pressure vs. crossover probability.



- Building block size vs. population size



- Goldberg, Deb, Thierens, 1992.