# Population sizing models in GAs

Some of the images in this presentation were taken from the papers of Goldberg, Deb, and Clark (1991), and from the paper of Harik, Cantú-Paz, Goldberg, and Miller (1996).

**Fernando Lobo**

**University of Algarve**

# Solution quality and speed

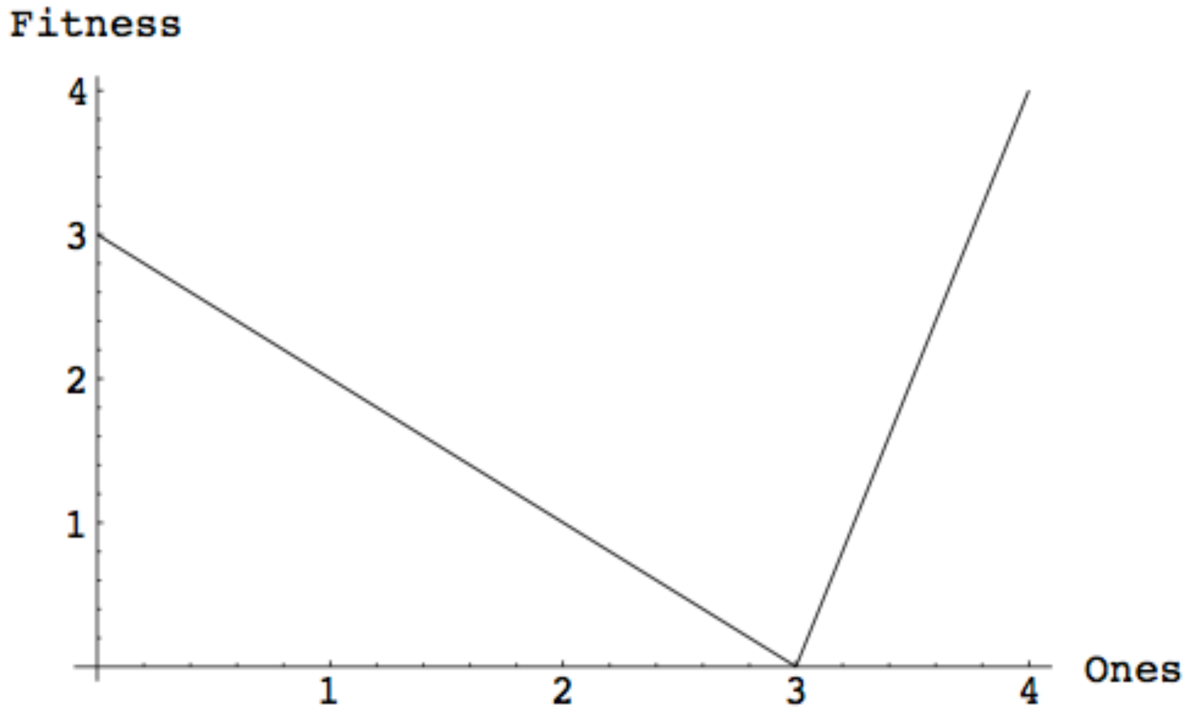
- We would like to get as good a solution as possible.
- And we would like to get that solution as quick as possible.
- In general, the larger the population size, the better the chance of obtaining a good solution.
  - But a very large population implies more computational time.
  - The two objectives are contradictory.

# Population sizing models

- Today we will look at a theoretical model that can predict the solution quality of a GA on a special class of problems.

- The class of problems consists of additive decomposable problems of uniform scale.
  - Onemax and *m-k* trap functions are examples of such problems.

# *k*-bit trap function



# Example with *m*=6, *k*=4

- x = 111101110000010110111000
  - 1st trap: bits 1-4
  - 2nd trap: bits 5-8
  - …
  - 6$^{th}$ trap: bits 21-24

$$\underbrace{1111}_{4}\underbrace{0111}_{0}\underbrace{0000}_{3}\underbrace{0101}_{1}\underbrace{1011}_{0}\underbrace{1000}_{2}$$

- f(x) = 4 + 0 + 3 + 1 + 0 + 2 = 10

# Based on the notion of building block

- In onemax, BBs are of size 1.
  - BBs are not destroyed by crossover in this case.

- In *m*-*k* trap functions, BBs are of size *k*.
  - Crossover can destroy them.
  - But the model that we will be looking at assumes they are not destroyed (this assumption is well approximated if the crossover operator is not too disruptive: 1pt or 2pt crossover for tightly coded BBs).

# Based on the notion of BB (cont.)

- Model also assumes that BBs are not constructed
  - Deception implies it's unlikely that a BB is constructed by combining sub-parts of it.
  - BBs are like atomic structures that need to be present in the initial population, and be propagated for future generations.
  - We assume variation operators do not create nor destroy them.

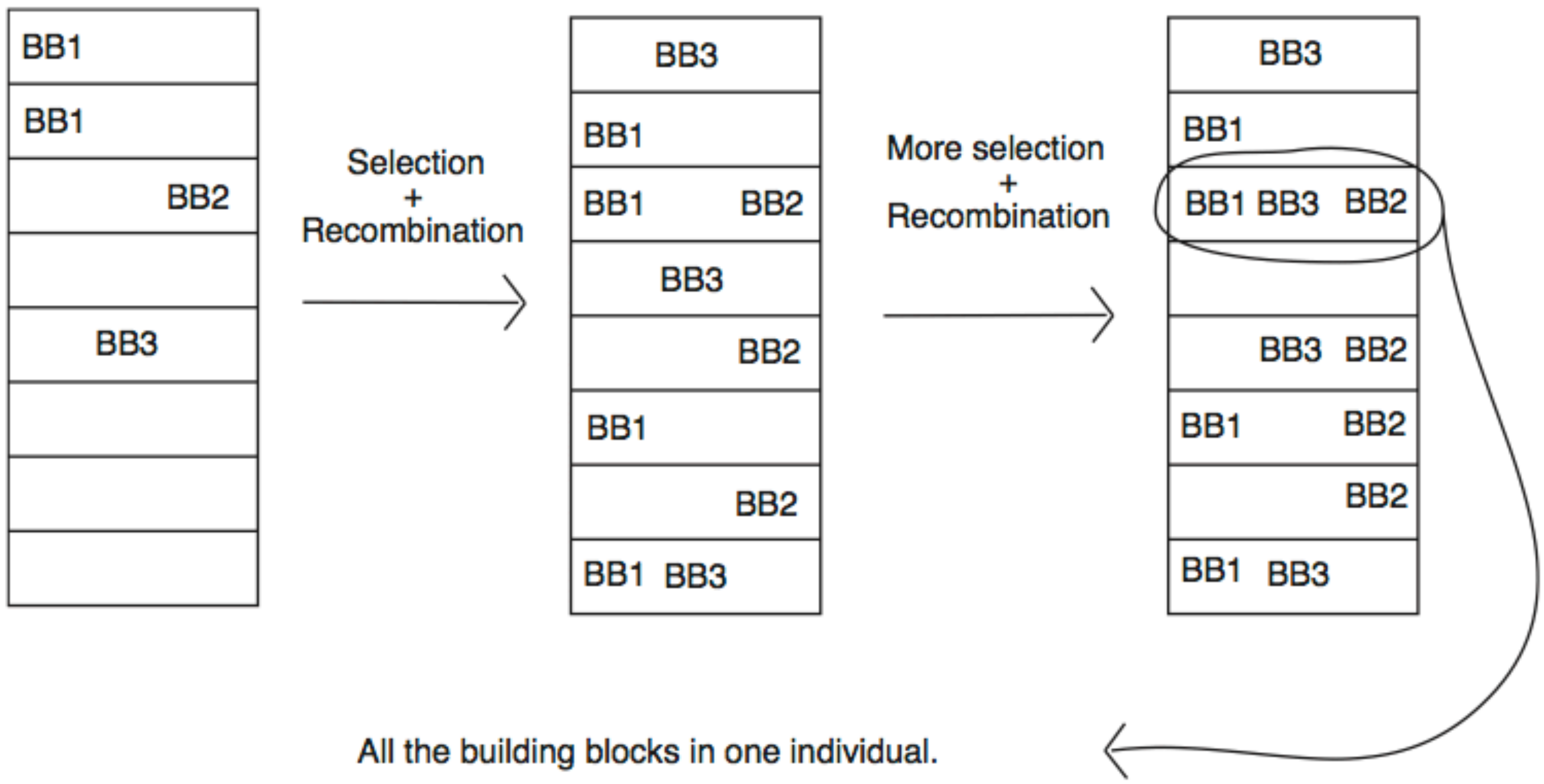



Image taken from Georges Harik's Phd thesis.

# Uniformly scaled problems

- For uniformly scaled problems (i.e., problems where each BB is worth the same amount), BB convergence occurs more or less at the same speed.

- Theoretical models that we are about to see focuses on one BB partition alone, and assume similar behaviour occurs in other partitions.

# Consider 20-bit onemax problem

| string | fit |
|---|---|
| 0 0 0 0 1 0 1 0 1 1 0 0 1 1 1 1 0 0 1 0 | 9 |
| 1 0 1 1 1 0 1 1 0 0 1 1 1 1 1 0 0 0 1 0 | 12 |
| 0 0 1 1 1 0 1 0 0 0 1 0 0 0 1 0 1 1 0 0 | 8 |
| 0 1 1 1 1 0 0 0 1 0 0 0 0 0 1 0 1 0 1 0 | 8 |
| 1 0 1 1 1 0 0 1 0 0 0 0 1 0 0 0 1 0 0 0 | 7 |
| 1 1 0 0 0 1 1 1 1 1 1 0 0 1 0 0 0 0 0 0 | 9 |
| 1 0 0 1 1 0 1 1 0 1 0 1 0 1 1 0 0 1 0 0 | 10 |
| 1 1 0 0 1 1 1 0 1 0 0 0 1 1 0 1 0 1 0 0 | 10 |
| 0 0 1 0 0 0 0 0 1 0 0 0 1 0 0 1 0 1 0 0 | 5 |
| 0 0 0 0 0 0 1 0 0 1 0 1 1 0 1 0 1 0 1 1 | 8 |

- For each bit position, 1s compete with 0s:

```
* * 1 * * * * * * * * * * * * * * * * *
* * 0 * * * * * * * * * * * * * * * * *
```

# Consider concatenation of 4-bit trap functions

- Now the competition takes place at the BB level, not single bits.

```
* * * * 1 1 1 1 * * * * * * * * * * * *
* * * * 1 1 1 0 * * * * * * * * * * * *
* * * * 1 1 0 1 * * * * * * * * * * * *
* * * * 1 1 0 0 * * * * * * * * * * * *
* * * * 1 0 1 1 * * * * * * * * * * * *
* * * * 1 0 1 0 * * * * * * * * * * * *
* * * * 1 0 0 1 * * * * * * * * * * * *
* * * * 1 0 0 0 * * * * * * * * * * * *
* * * * 0 1 1 1 * * * * * * * * * * * *
* * * * 0 1 1 0 * * * * * * * * * * * *
* * * * 0 1 0 1 * * * * * * * * * * * *
* * * * 0 1 0 0 * * * * * * * * * * * *
* * * * 0 0 1 1 * * * * * * * * * * * *
* * * * 0 0 1 0 * * * * * * * * * * * *
* * * * 0 0 0 1 * * * * * * * * * * * *
* * * * 0 0 0 0 * * * * * * * * * * * *
```

# Supply models

- Consider onemax. On a randomly initialized population, how many BBs do we have at a given partition (bit position)?
  - Answer: $n / 2$

- Consider m-k trap function. How many do we have at a given partition.
  - Answer: $n / 2^k$

- Ex: N=160
  - Onemax → 80 BBs per partition on average
  - 4-bit trap → 10 BBs per partition on average

# Decision models

- During selection, the GA chooses the better string. But it does not always chooses the correct BB.

- Again, consider onemax:

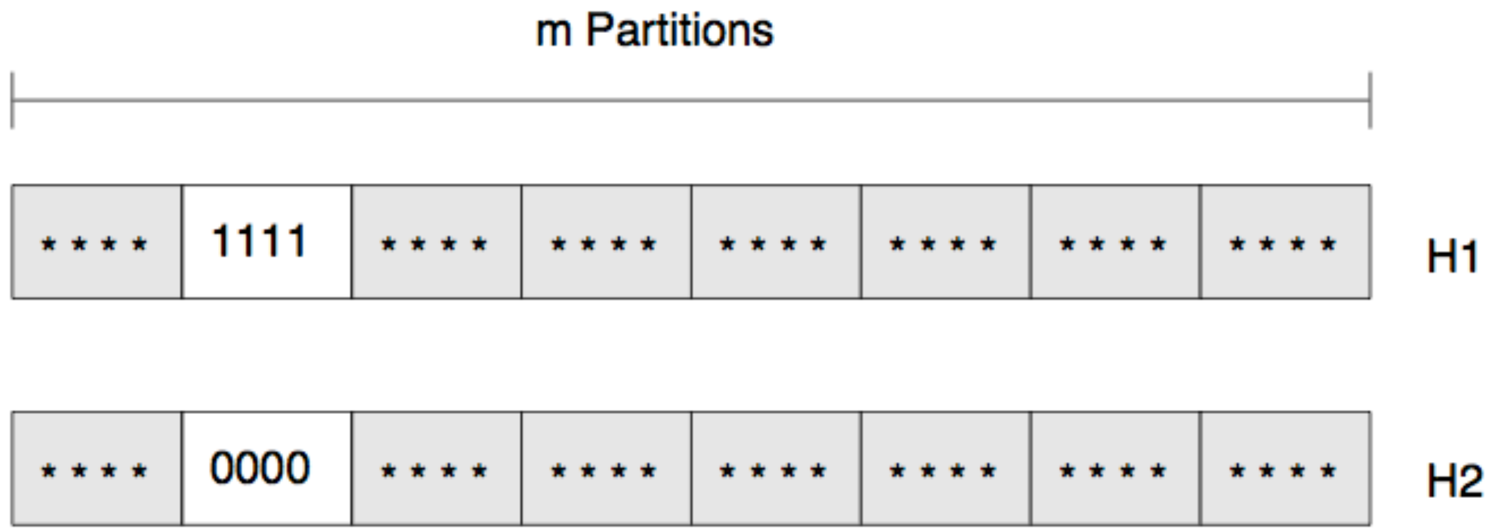
```
              string                      fit
0 0 0 0 1 0 1 0 1 1 0 0 1 1 1 1 0 0 1 0    9
1 0 1 1 1 0 1 1 0 0 1 1 1 1 1 0 0 0 1 0   12
```

- On a single competition, mistakes can be made at the BB level:

```
* * * * * * * * * * * * * * * 1 * * * *
* * * * * * * * * * * * * * * 0 * * * *
```

# Decision models (cont.)

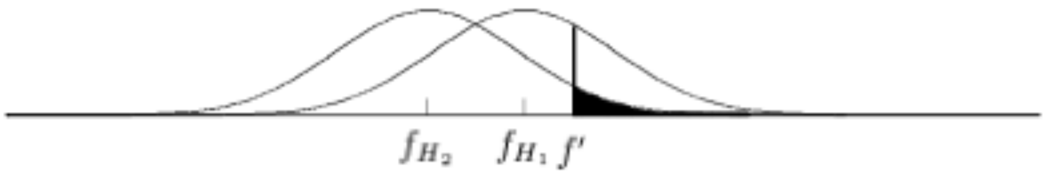
- Same thing happen for larger-sized BBs.




- The other partitions are a source of noise for the correct decision between a BB and its competitors.

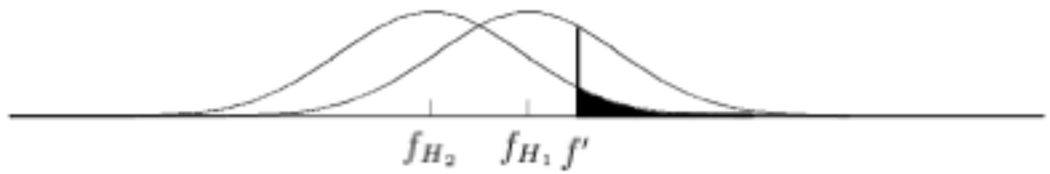
# Decision models (cont.)

- Consider the competition between the best and the 2nd best schema.

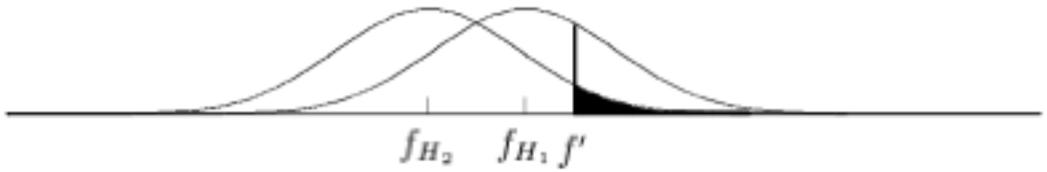


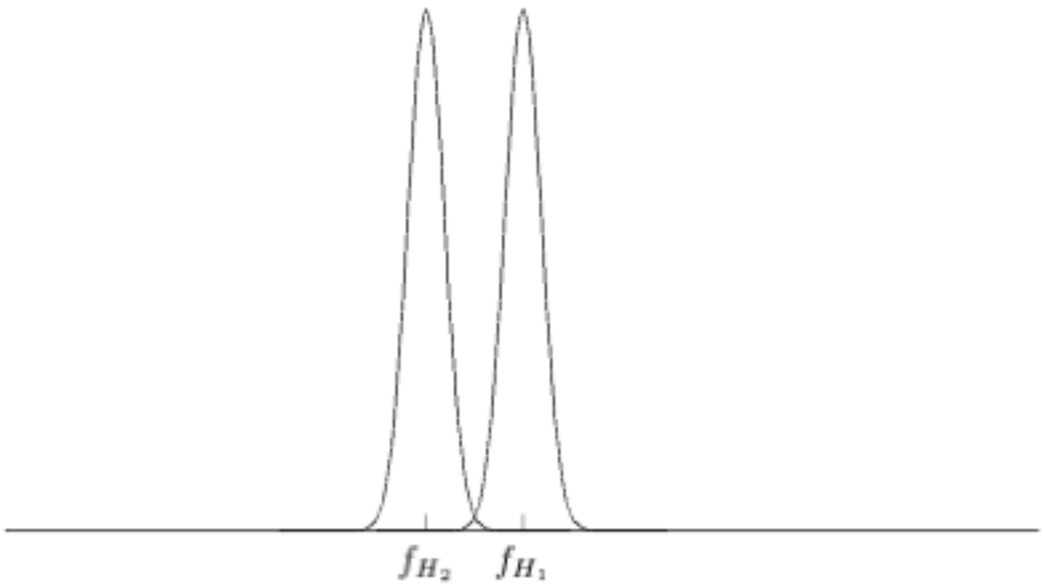

- On average, members of the best schema ($H_1$) are better than members of the 2nd best schema ($H_2$).
- Schema fitness approximately normally distributed.

# Decision models (cont.)

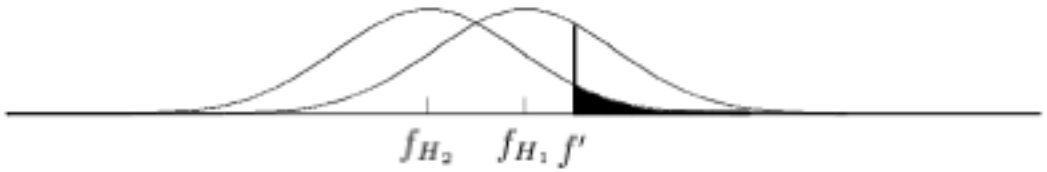



- With multiple competitions, the overlapping between the two distributions decreases, allowing more accurate decisions.


$f_{H_2}$
$f_{H_1}$
$f'$


$f_{H_2}$
$f_{H_1}$

# Decision models (cont.)




- On a single competition, what is the probability that a string from $H_1$ is better than a string from $H_2$?
- It turns out to be: $$p = \Phi\left(\frac{d}{\sqrt{\sigma_{H_1}^2 + \sigma_{H_2}^2}}\right)$$
- with $d = \overline{f_{H_1}} - \overline{f_{H_2}}$
- and $\Phi$ is the cumulative distribution function of a standard Normal distribution.

# Decision models (cont.)

- How to calculate $\sigma_{H_1}^2$ and $\sigma_{H_2}^2$ ?

- If the fitness function $F$ is the sum of $m$ independent subfunctions $F_i$, then the overall variance is:

$$\sigma_F^2 = \sum_{i=1}^{m} \sigma_{F_i}^2$$

- The total noise coming from the $m' = m - 1$ remaining partitions is $\sigma^2 = m'\sigma_{bb}^2$

# Decision models (cont.)

- The probability of deciding well in a single competition becomes:

$$p = \Phi\left(\frac{d}{\sqrt{2m'}\sigma_{bb}}\right)$$

- Example: for onemax with 100 bits we get,
  - $d = 1$
  - $m' = 99$
  - $\sigma_{bb}^2 = 0.25 \implies \sigma_{bb} = 0.5$
  - plugging these values, we get $p = \Phi(0.142) \approx 0.5565$

# The Gambler's Ruin problem

- We will now look at a mathematical model that combines the supply and decision models.
- The model looks at the operation of GAs as random walks.
- First let us make a detour and look at the Gambler's Ruin problem, a classical problem in probability theory.

# The Gambler’s Ruin problem

- Imagine 2 players, each has 500 Euros. They decide to play the following game.
  - Each one puts 1 Euro on the table, and then toss a coin.
  - The winner takes the 2 Euros (stays with 501 Euros, the loser stays with 499 Euros).
  - They keep playing the game until one of them gets bankrupt (0 Euros).

# The Gambler’ s Ruin problem

- We can make the game more general, by letting the players start with different amounts of money (one with $x_0$ Euros, the other with $n$-$x_0$ Euros).

- Also, the coin toss can be biased, with probability of heads being p (tails = $q$ = 1-$p$).

- If you start with $x_0$ Euros, what’s the probability that you win the game?

- Answer: $$\frac{1-\left(\frac{q}{p}\right)^{x_0}}{1-\left(\frac{q}{p}\right)^{n}}$$

# The Gambler's Ruin problem

- For you to have a feeling, here are some examples:
  1. p = 0.51, $x_0$ = 500, $n$-$x_0$ = 500
  2. p = 0.51, $x_0$ = 50, $n$-$x_0$ = 50

- Answer 1: $\dfrac{1-\left(\frac{q}{p}\right)^{x_0}}{1-\left(\frac{q}{p}\right)^{n}} = 0.9999$

- Answer 2: $\dfrac{1-\left(\frac{q}{p}\right)^{x_0}}{1-\left(\frac{q}{p}\right)^{n}} = 0.8808$

# What does this has to do with GAs?

- We can model the GA as doing a sequence of competitions until either the BB is lost from the population (ruin) or the BB fully takes over its partition (wins all the money).

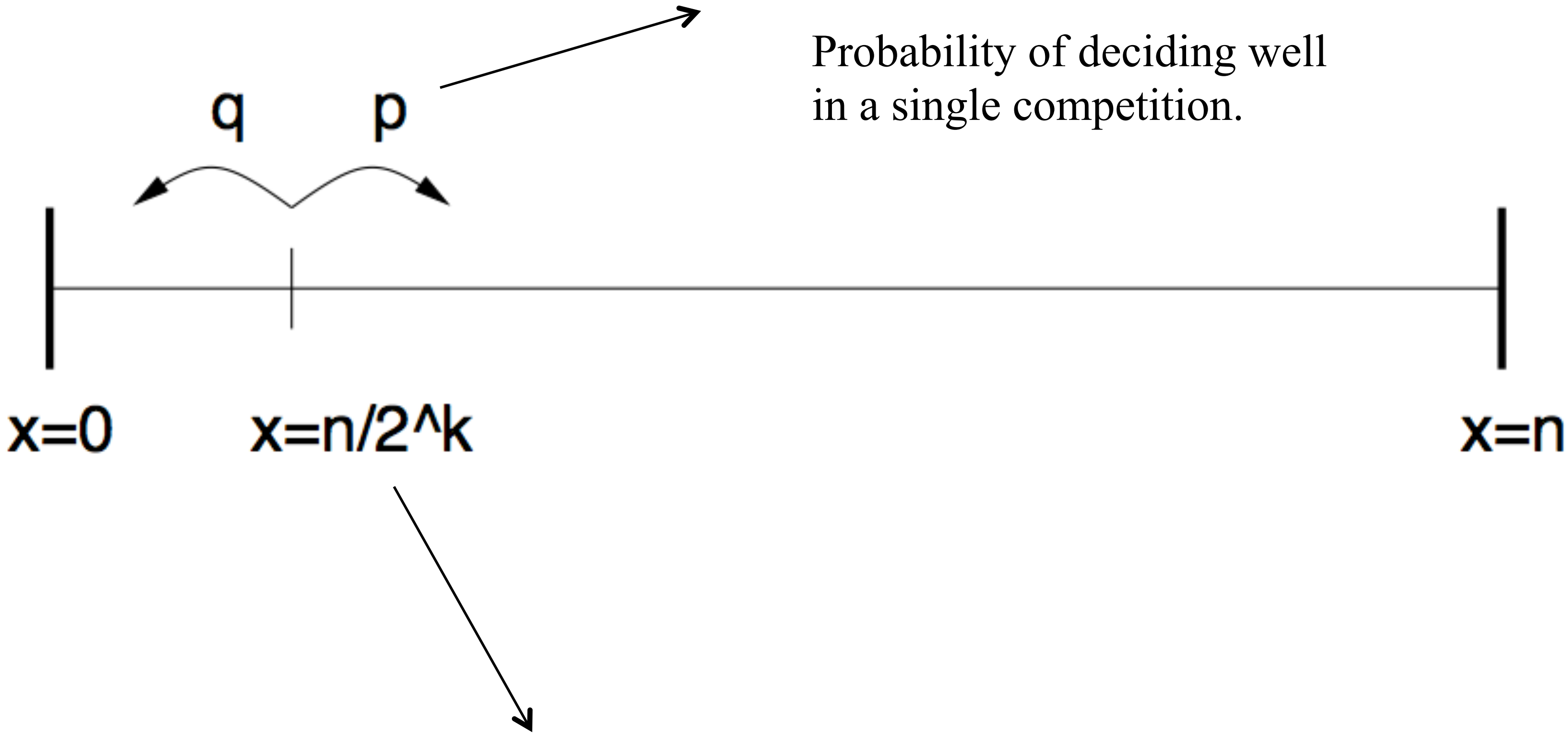
- The starting position is given by the BB supply in the initial population: $x_0 = \frac{n}{2^k}$

- The probability of winning a single competition is the probability of deciding well between the BB and its most tough competitor.

- Proposed by Harik, Cantú-Paz, Goldberg & Miller (1996)

# GAs and the Gambler's Ruin problem

Probability of deciding well
in a single competition.




Initial building block supply.

# Gambler' s ruin model

- Given the initial supply of BBs ($x_0$), the probability ($p$) of deciding well between a BB and its toughest competitor, and a population size ($n$), we can use the Gambler's Ruin model to predict the probability that the BB takes over the entire population (i.e., reaches the $n$ barrier).

$$P_{bb} = \frac{1 - \left(\frac{q}{p}\right)^{x_0}}{1 - \left(\frac{q}{p}\right)^{n}}$$

- Examples with100-bit onemax ($p = 0.5565$)
  - $n = 20 \implies x_0 = 10 \implies P_{bb} = 0.906$
  - $n = 100 \implies x_0 = 50 \implies P_{bb} = 0.999$

# Obtaining a population sizing equation

- We can obtain a population sizing equation by solving the gambler's ruin formula in terms of *n*.

- By doing several approximations, we can obtain the desired equation. Let's do it.

- First note that $p > q$,
  - For large *n*, $\left(\frac{q}{p}\right)^n \to 0$ (den. of GR formula approaches 1)

$$P_{bb} \approx 1 - \left(\frac{1-p}{p}\right)^{n/2^k}$$

# Obtaining a population sizing equation

- Setting $P_{bb} = 1 - \alpha$ , (alpha is the probability of error, i.e, probability of losing the BB from the population), and solving for *n*, gives:

$$n = \frac{2^k \ln(\alpha)}{\ln\left(\frac{1-p}{p}\right)},$$

- *p* can be approximated using the first two terms of the power series expansion for the Normal distribution,

$$p = \Phi(z) \approx \frac{1}{2} + \frac{1}{\sqrt{2\pi}} z$$

with $z = d/(\sigma_{bb}\sqrt{2m'})$

# Obtaining a population sizing equation

- Substituting gives $$n = 2^k \ln(\alpha) / \ln\left(\frac{1-\frac{z\sqrt{2}}{\sqrt{\pi}}}{1+\frac{z\sqrt{2}}{\sqrt{\pi}}}\right)$$

- When $z$ is small $\Longrightarrow ln\left(1 \pm \frac{z\sqrt{2}}{\sqrt{\pi}}\right) \approx \pm\frac{z\sqrt{2}}{\sqrt{\pi}}$

- Using the approximation and substituting the value of $z$ gives,

$$n = -2^{k-1} \ln(\alpha) \frac{\sigma_{bb}\sqrt{\pi m'}}{d}$$

# Observations

$$n = -2^{k-1} \ln(\alpha) \frac{\sigma_{bb}\sqrt{\pi m'}}{d}$$

- Population size $n = O(\sqrt{m})$
- Grows with $2^k$ (not so bad because $k$ is much smaller than the string length).
- Is inversely proportional to the signal-to-noise ratio
  - high variability imply larger population size.
- These results give an indication that GAs scale well for this class of problems.

# Using the population sizing equation

- Let's say we have a 100-bit onemax problem.
- We would like to obtain a solution with 95 BBs ($\alpha = 0.05$)
- What's the necessary population size needed to reach that quality?

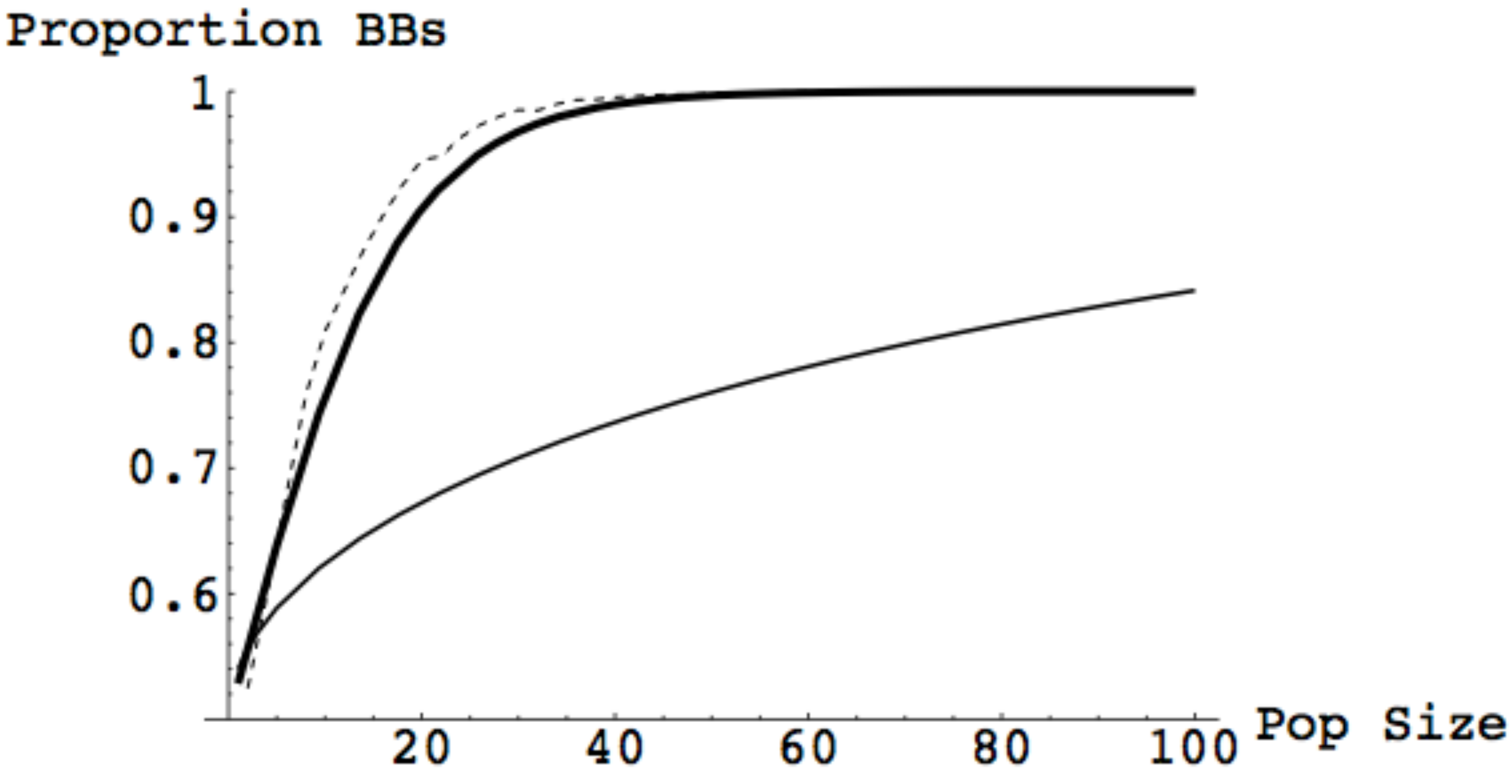
$$n = -2^{k-1}\ln(\alpha)\frac{\sigma_{bb}\sqrt{\pi m'}}{d}$$

$$\left\{\begin{array}{l} k = 1 \\ m' = 99 \\ d = 1 \\ \sigma_{bb} = 0.5 \\ \ln(\alpha = 0.05) \approx 0.05 \end{array}\right. \qquad \Longrightarrow \qquad n = 26$$

# Experiments

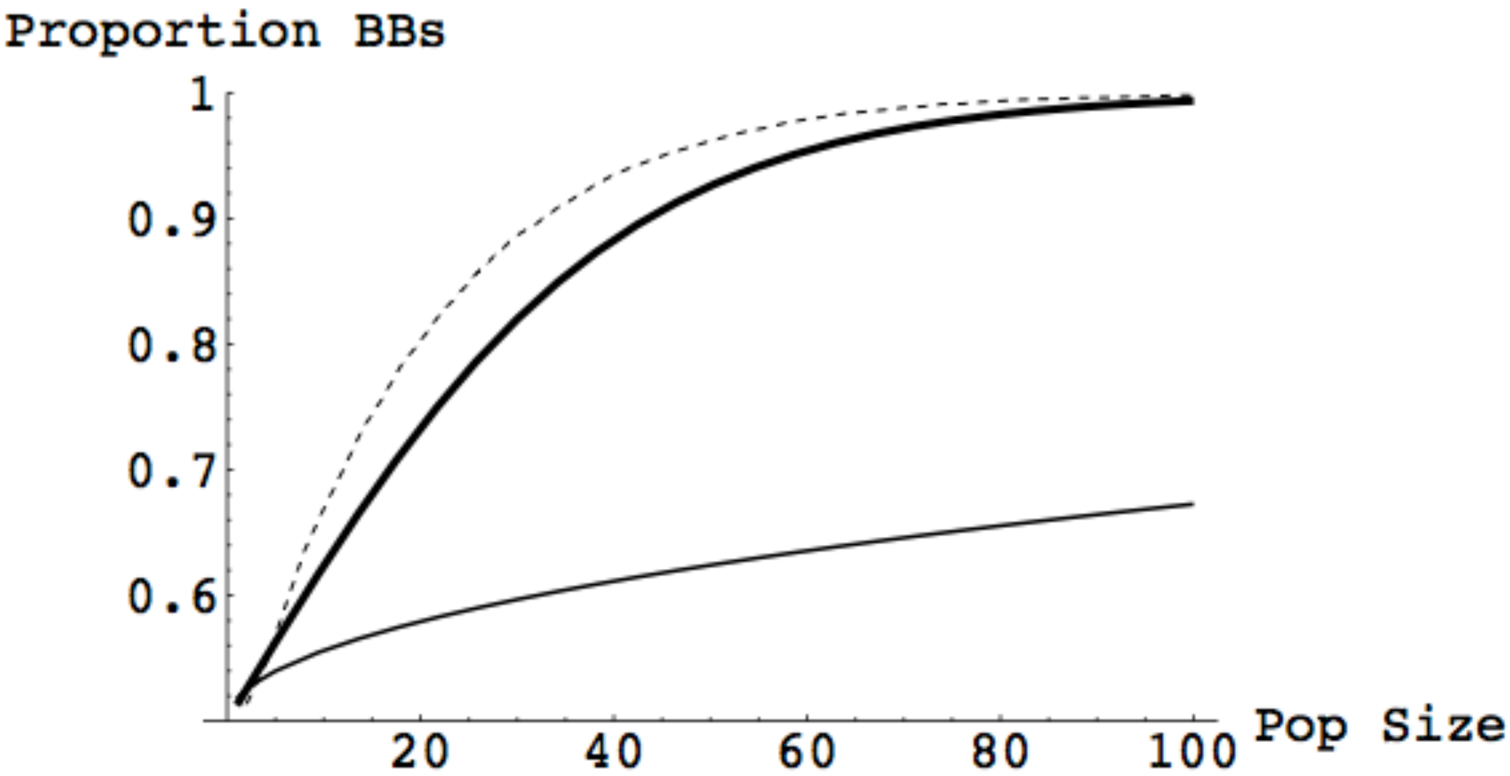
- Model assumes binary tournament selection.
- Model also assumes decorrelation among BB partitions
  - For onemax, this behaviour is well approximated using uniform crossover.

# Experiments with 100-bit onemax (uniform crossover is used)



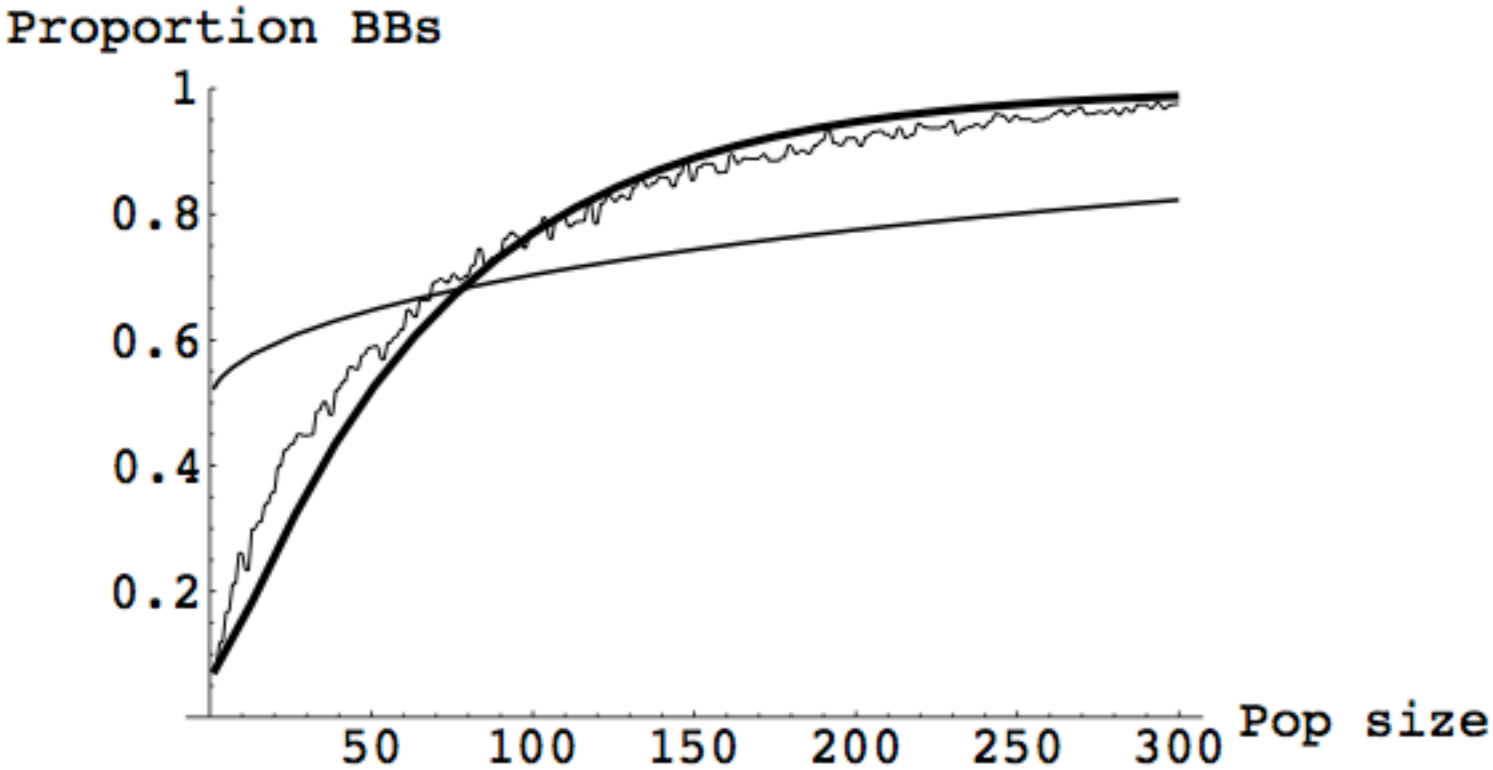

Dotted line is experiments, bold solid line is gambler's ruin model,
The other solid line is a previous (less accurate) model due to Goldberg, Deb & Clark.

# Experiments with 500-bit onemax (uniform crossover is used)



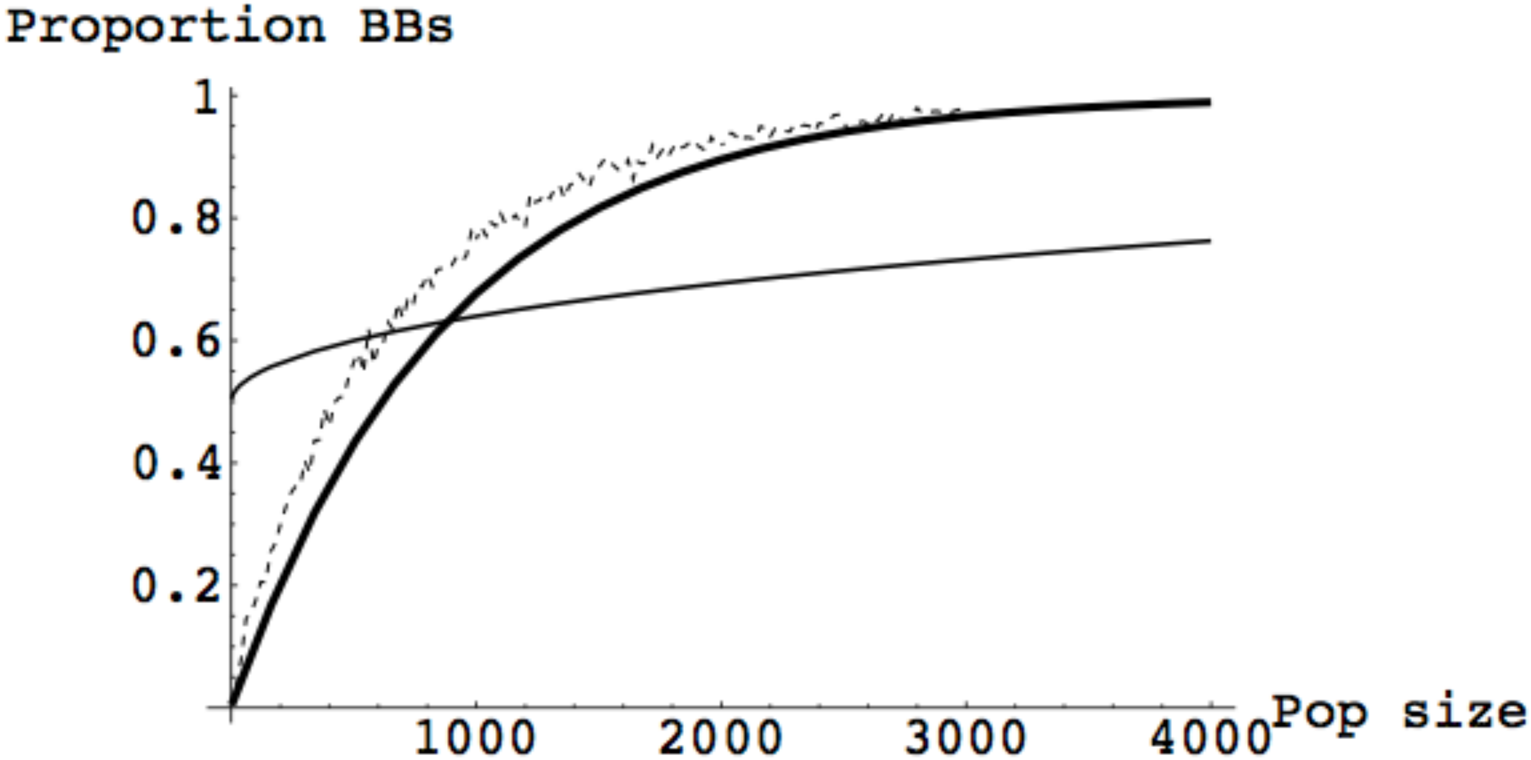

Dotted line is experiments, bold solid line is gambler's ruin model,
The other solid line is a previous (less accurate) model due to Goldberg, Deb & Clark.

# Experiments with $m$=20 $k$=4 trap function (two-point crossover is used)




Dotted line is experiments, bold solid line is gambler's ruin model,
The other solid line is a previous (less accurate) model due to Goldberg, Deb & Clark.

# Experiments with $m$=10 $k$=8 trap function (one-point crossover is used)




Dotted line is experiments, bold solid line is gambler's ruin model,
The other solid line is a previous (less accurate) model due to Goldberg, Deb & Clark.

# So what?

- Difficult to apply in an arbitrary problem. Why?
- But this is the first accurate model of GA population sizing requirements.
  - Tells us that GAs scale very well if the variation operators are not too disruptive.
  - Tells us that population size in range 50-100 for all problems is a mistake. Population sizing is problem dependent.
- In a forthcoming lecture, we will look at a way to address population sizing in practice.

# Want to know more?

- Read the original paper:
  - *The Gambler's Ruin Problem, Genetic Algorithms, and the Sizing of Populations* (1999), by Harik, Cantú-Paz, Goldberg, and Miller.

- Also read the paper that contain a previous (less accurate) model, a very important precursor to the gambler's ruin paper.
  - *Genetic Algorithms, Noise, and the Sizing of Populations* (1991), by Goldberg, Deb, and Clark.