Building Block Mixing and the Scalability of Genetic Algorithms

Some of the images in this presentation were taken from the papers of Thierens and Goldberg (1993), and Thierens (1999).

Fernando Lobo

University of Algarve

Limitations of simple GAs

- GAs work well when the variation operators are not too disruptive.
- If we choose operators that are not too disruptive, then we are likely to observe good performance.
- Being disruptive or not, depends on the encoding.
- In order to have a low disruptive operator, we need to have problem specific information.
  - But many times we do not have such information (or we have but it is unreliable).

Limitations of simple GAs

- We saw that we can solve bounded difficult problems, as long as the BBs are tightly coded.

- What if they are not tightly coded?
  - one- or two-point crossover fail.
  - what about uniform crossover?

- Uniform crossover is independent on the encoding.
  - Its performance is the same whether interacting genes are near each other or not.

Limitations of simple GAs

- As we shall see, uniform crossover doesn’t scale well for bounded difficult problems (such as concatenated trap functions).

Remember the schema theorem

$$m(H,t+1) \geq m(H,t) \underbrace{\phi(H,t)}_{} [1-\underbrace{\epsilon(H,t)}_{}]$$

Reproduction ratio
(due to selection)

Disruption factor
(due to variation ops.)

Remember the schema theorem

- Growth ratio: $\phi(H,t)\ [1-\epsilon(H,t)]$
- Or simply: $\phi\ (1-\epsilon)$
- A schema grows if its growth ratio is greater than 1.

$$\phi\ (1-\epsilon) > 1$$

- We can always enforce that by raising the selection pressure, or lowering the disruption factor.

Remember the schema theorem

- We want BBs to grow.
- It seems that we can do that even with a very disruptive crossover.
  - We simply raise the selection pressure in order to counterbalance for the disruptive effect of crossover.
  - Or we can control the amount of BB disruption by using a low probability of crossover P_c

$$s\ (1 - P_c) > 1$$

What is the problem?

$$s\ (1 - P_c) > 1$$

- A very high s leads to quick loss of diversity, and no BB exchange.
- If P_c is very low, there's also very little BB exchange!
- Lesson:
  - Growing BBs is one thing.
  - Exchanging (or mixing) them is another thing.

What is the problem?

- Good BBs of one string have to be combined with good BBs of another string.

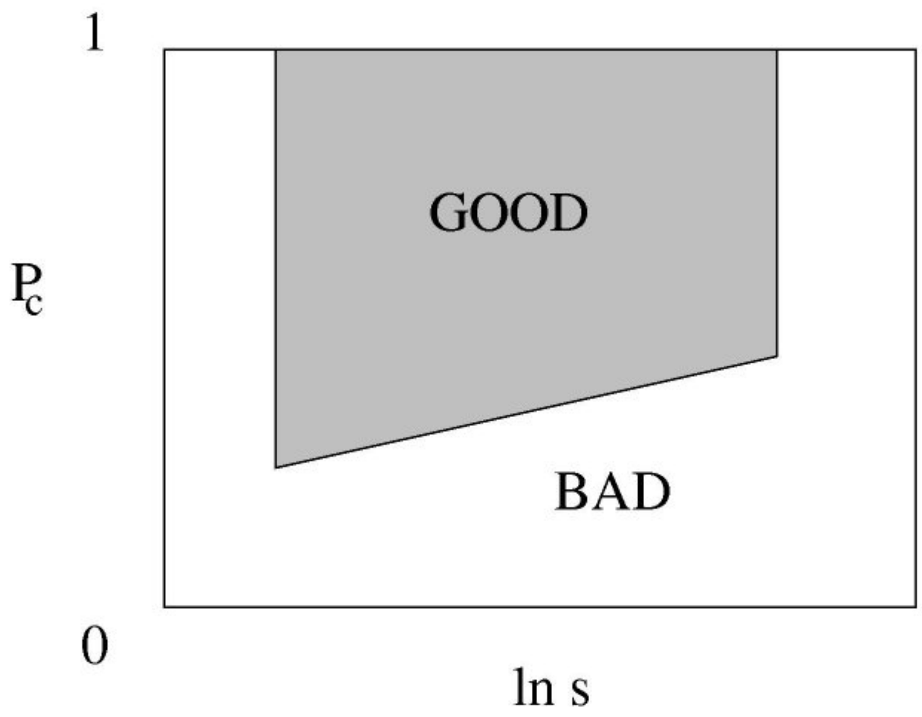
- Two contradictory requirements of a crossover operator:
  - Minimize BB disruption
  - Maximize BB exchange

- Making copies of BBs and exchanging them are opposing processes.

Mixing model

- Thierens and Goldberg (1993) made a BB mixing model for simple GAs using uniform crossover.
- We are skipping the details of the model (those interested can read the original papers).
- Results:
  - Good news: for simple problems (onemax and the like), the GA works well for a wide range of parameter values.
  - Bad news: For more difficult problems, the population size requirements grow exponentially!

Good news



Goldberg, Deb, & Thierens (1993)

Bad news

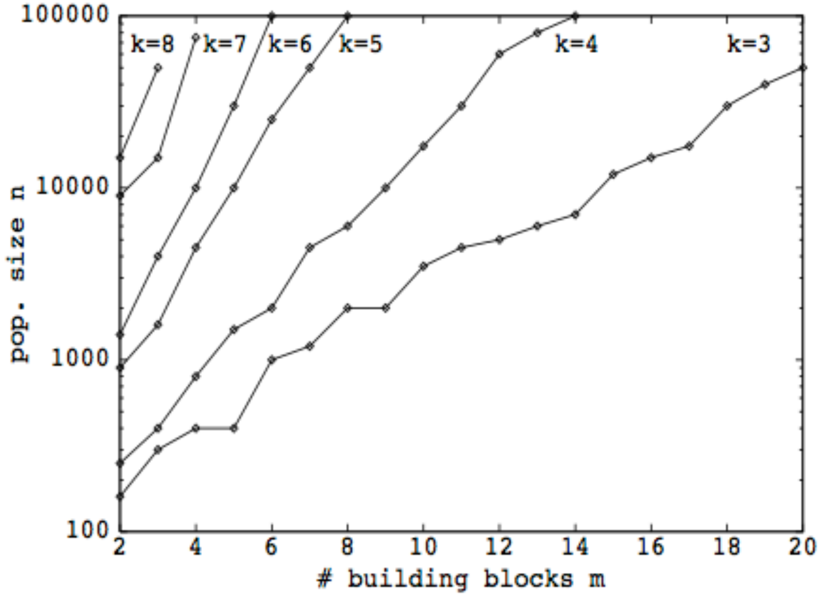


Figure 5: Minimal population size n needed as a function of the number of building blocks m in order to converge to the global optimum in at least 49 out of 50 runs. Results are shown for different building-block lengths k, with selection pressure $s = 4$ and crossover probability $p_c = 0.5$.

from Thierens and Goldberg (1993)

Want to know more?

- Read the papers:
  - *Toward a better understanding of mixing in genetic algorithms* (1993), by Goldberg, Deb and Thierens.
  - *Mixing in Genetic Algorithms* (1993), by Thierens and Goldberg.
  - *Scalability Problems of Simple Genetic Algorithms* (1999), by Thierens.