Digital Evolution

Genetic Algorithms and Artificial Life (ALife)

Overview

- Genetic Algorithm basics
- GA example
- Alife basics
- Tierra
- References

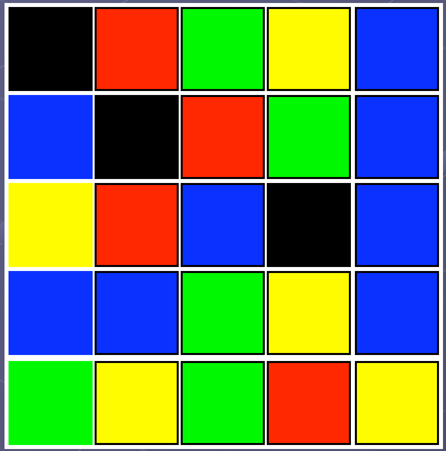
Genetic Algorithms

- Revolutionary method of computing.
    - Bottom up, instead of traditional top down approaches.
- **Outline of the Basic Genetic Algorithm**
- **[Start]** Generate random population of *n* chromosomes (suitable solutions for the problem)
- **[Fitness]** Evaluate the fitness *f(x)* of each chromosome *x* in the population
- **[New population]** Create a new population by repeating following steps until the new population is complete
    - **[Selection]** Select two parent chromosomes from a population according to their fitness (the better fitness, the bigger chance to be selected)
    - **[Crossover]** With a crossover probability cross over the parents to form new offspring (children). If no crossover was performed, offspring is the exact copy of parents.
    - **[Mutation]** With a mutation probability mutate new offspring at each locus (position in chromosome).
    - **[Accepting]** Place new offspring in the new population
- **[Replace]** Use new generated population for a further run of the algorithm
- **[Test]** If the end condition is satisfied, **stop**, and return the best solution in current population
- **[Loop]** Go to step 2
- Quoted from http://cs.felk.cvut.cz/~xobitko/ga/

Example

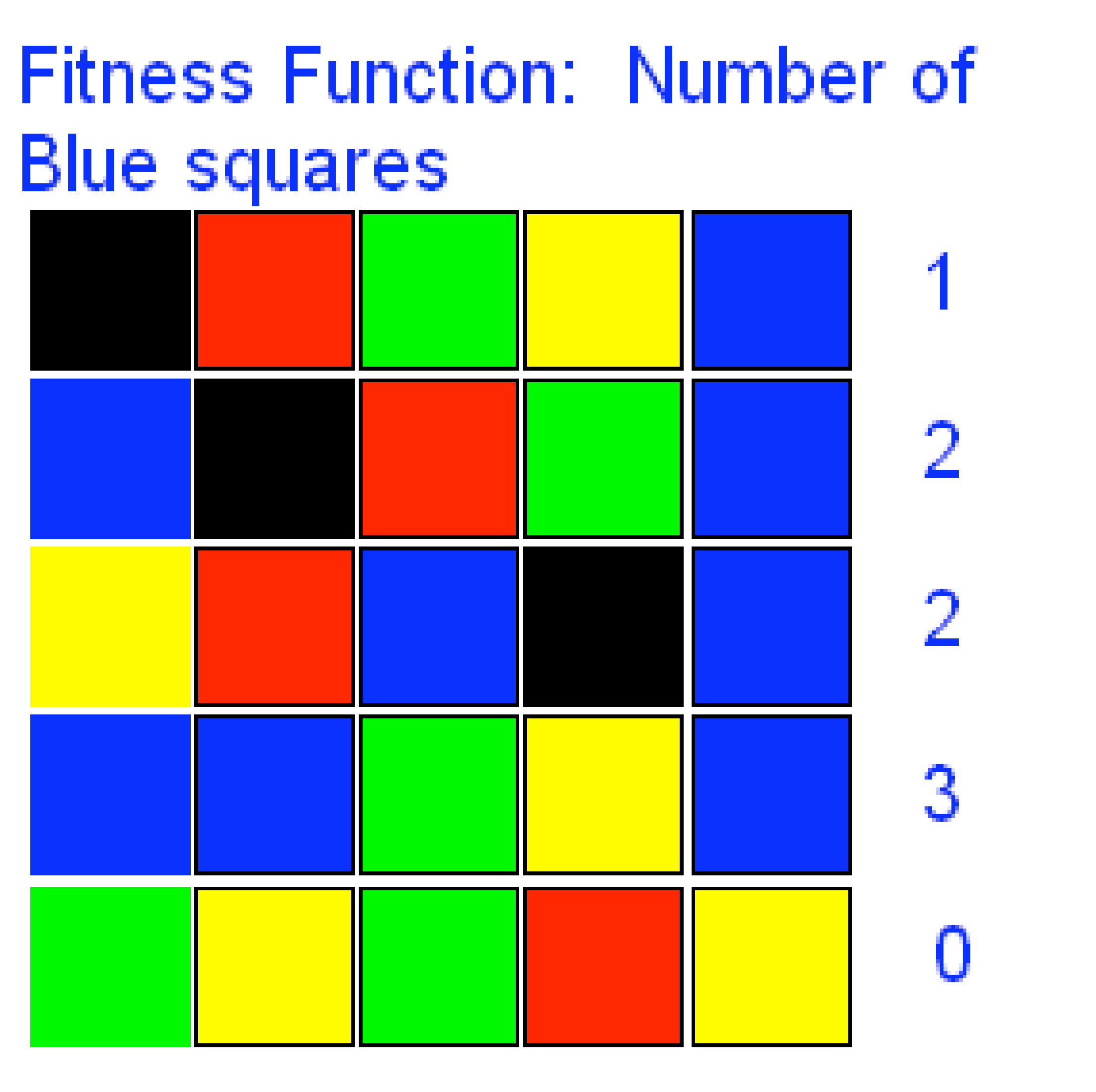
- Problem: Generate a block of squares that is all blue.

- Step 1: Generate Random Possible Solution set



⬛	🟥	🟩	🟨	🟦
🟦	⬛	🟥	🟩	🟦
🟨	🟥	🟦	⬛	🟦
🟦	🟦	🟩	🟨	🟦
🟩	🟨	🟩	🟥	🟨

Step 2: Evaluate Fitness

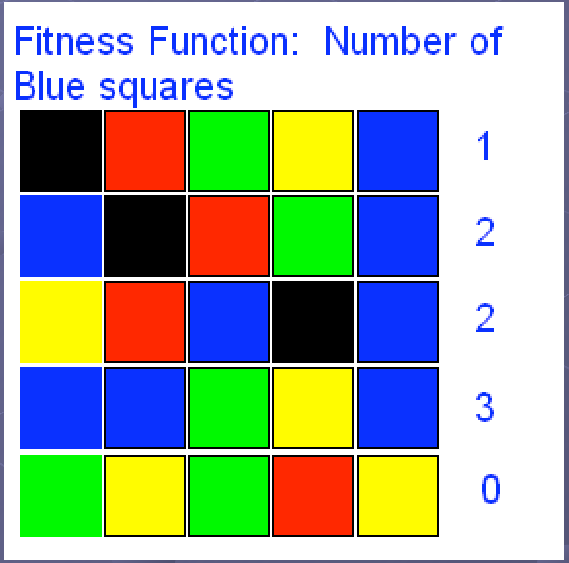


Fitness Function: Number of Blue squares

⬛	🟥	🟩	🟨	🟦	1
🟦	⬛	🟥	🟩	🟦	2
🟨	🟥	🟦	⬛	🟦	2
🟦	🟦	🟩	🟨	🟦	3
🟩	🟨	🟩	🟥	🟨	0

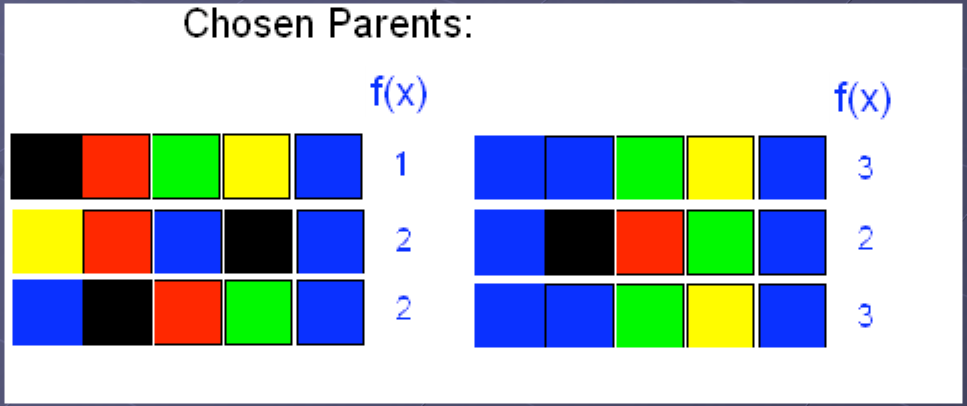
Step 3: Create a new population

- Selection: Randomly pick parents
  - population is 6, thus 3 pairs of parents are needed
  - Pick parents randomly but based on fitness



Fitness Function: Number of Blue squares

⬛	🟥	🟩	🟨	🟦	1
🟦	⬛	🟥	🟩	🟦	2
🟨	🟥	🟦	⬛	🟦	2
🟦	🟦	🟩	🟨	🟦	3
🟩	🟨	🟩	🟥	🟨	0

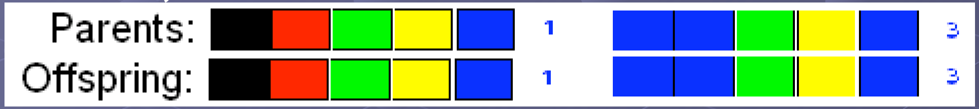


Chosen Parents:

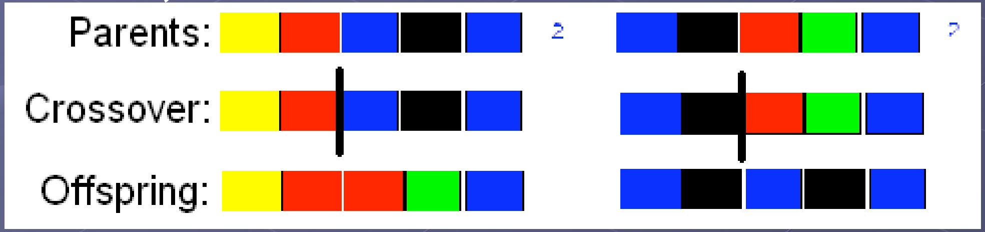
					f(x)						f(x)
⬛	🟥	🟩	🟨	🟦	1	🟦	🟦	🟩	🟨	🟦	3
🟨	🟥	🟦	⬛	🟦	2	🟦	⬛	🟥	🟩	🟦	2
🟦	⬛	🟥	🟩	🟦	2	🟦	🟦	🟩	🟨	🟦	3

Step 3: Create a new population

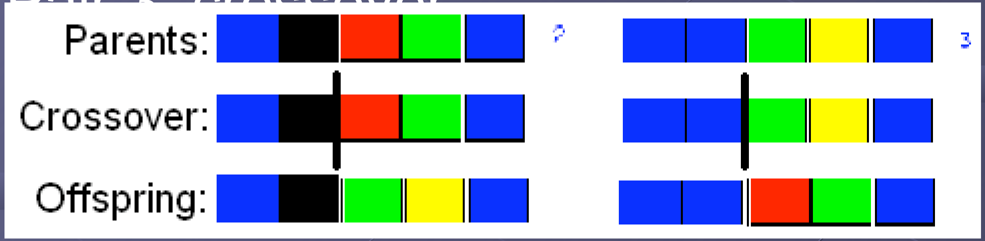
- Crossover: a constant that reflects how much genetic similarity the children will share with the parents
  - Pair 1, no crossover



  - Pair 2, crossover

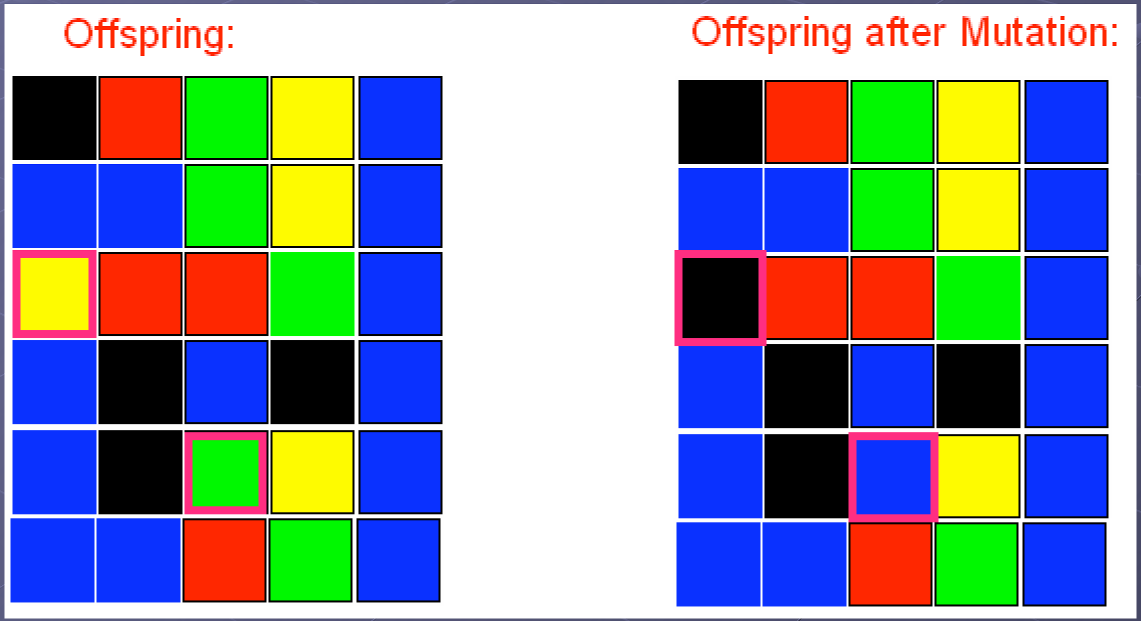


  - Pair 3, crossover



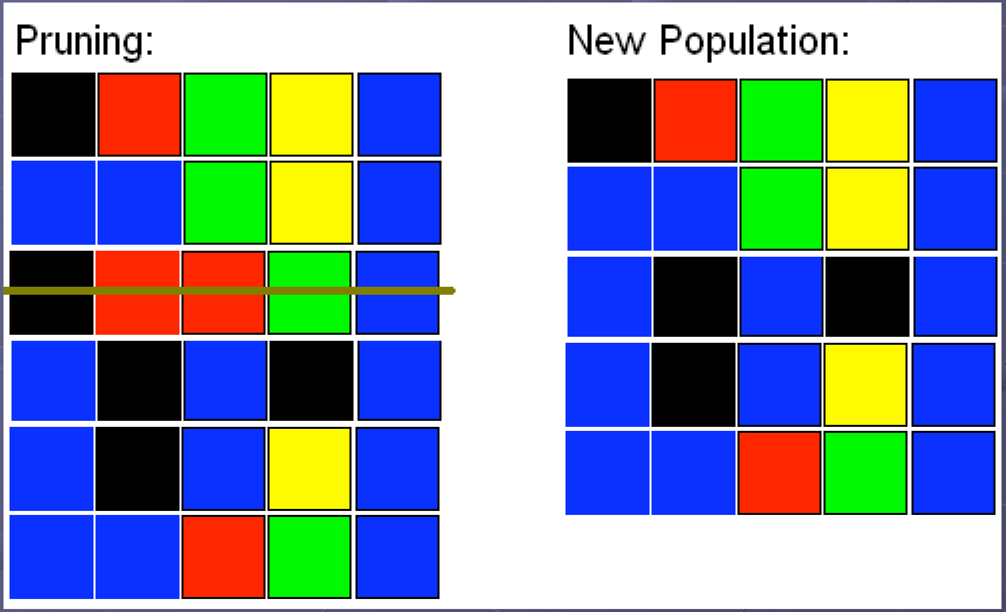
Step 3: Create a new population

- Mutation: A contant that reflects the random chance that one of the genes will mutate



Step 4: Replace

- Prune population to keep the same number (can be done randomly, by order, or by fitness)



Test

- Is the problem solved (or a certain number of iterations ran)
- Not yet… So run again with the new population!

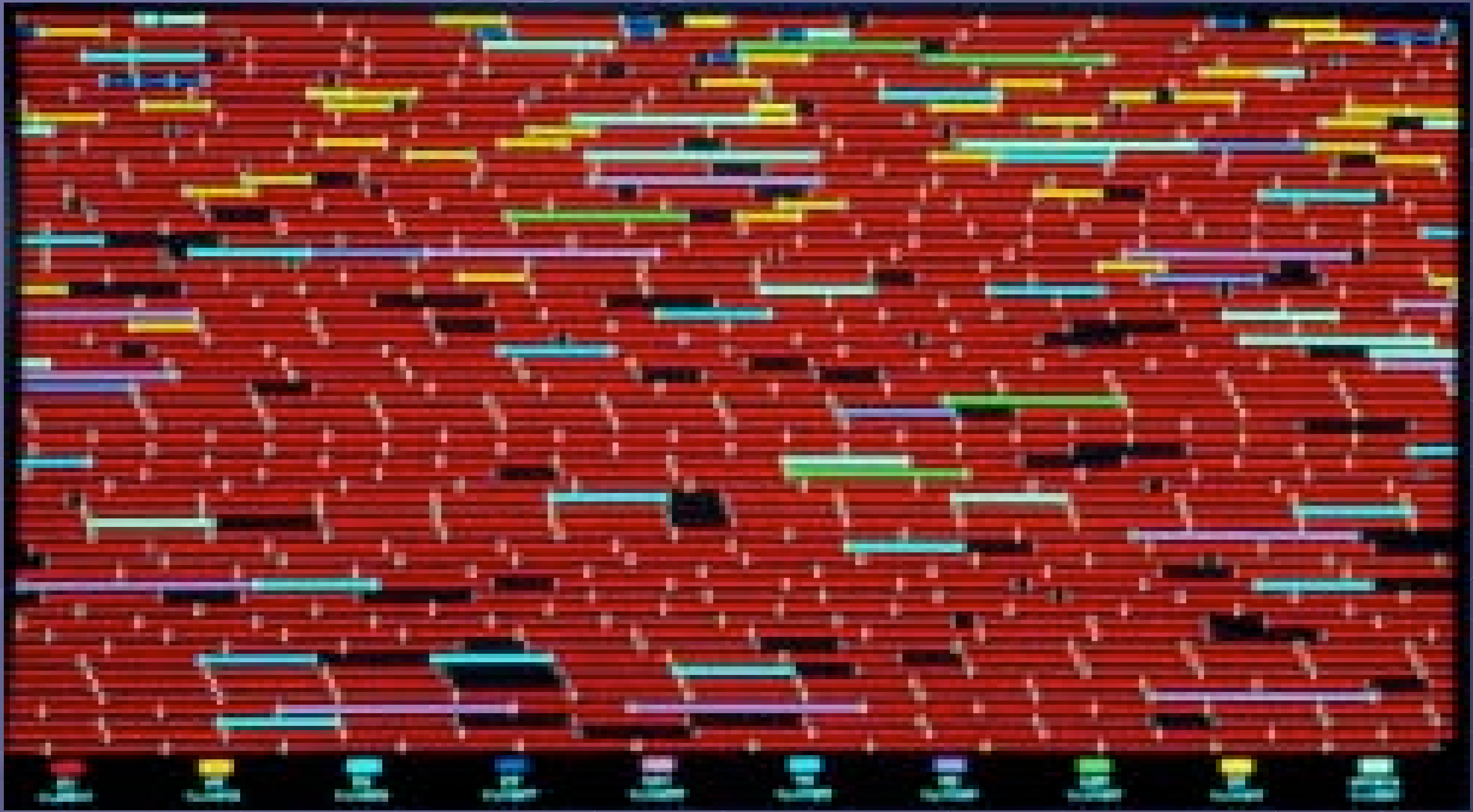
ALife basics

- Use genetic algorithms to simulate evolution.
- Types:
  - Hard – using hardware, like robots, for agents
  - Soft – simulate the agents digitally
  - Wet – grow agents using biochemical processes
- Instead of having a problem to solve, agents in an Alife system are forced to adapt to their environment. The fitness function becomes whether or not the agent is surviving (a kind of natural selection)

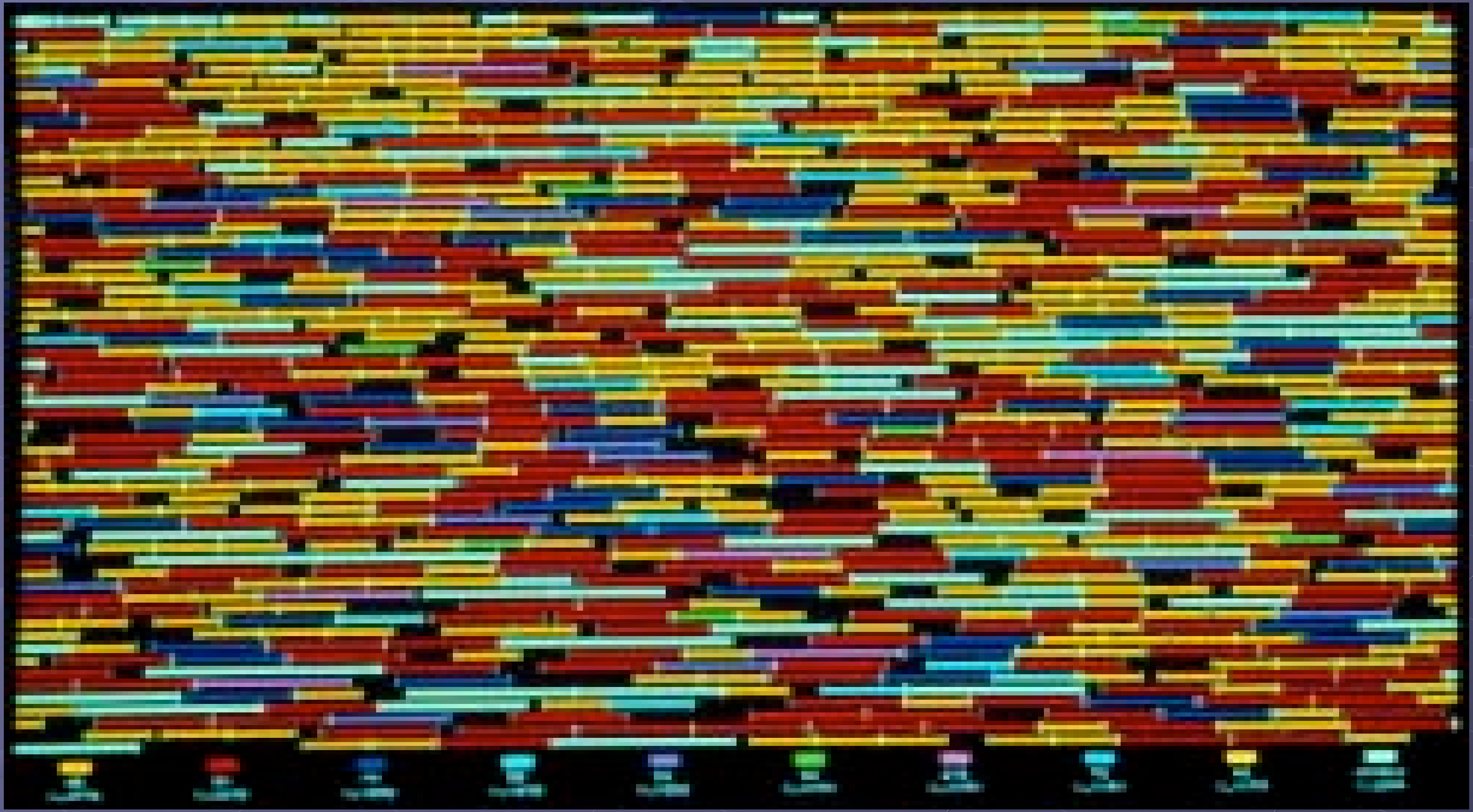
Tierra

- A soft Alife paradigm designed by Tom Ray
- Starts with a single agent (ancestor), a self-replicating bit of code.
- As it continues, older agents are removed from system…
- The vehicle of evolution here is solely mutation, as agents reproduce by division and not mating.
- If a mutation allows the offspring to replicate faster, this new type of agent will tend to dominate the space…
- Thus many different types of agents soon develop from a single ancestor. A remarkable genetic diversity will result.

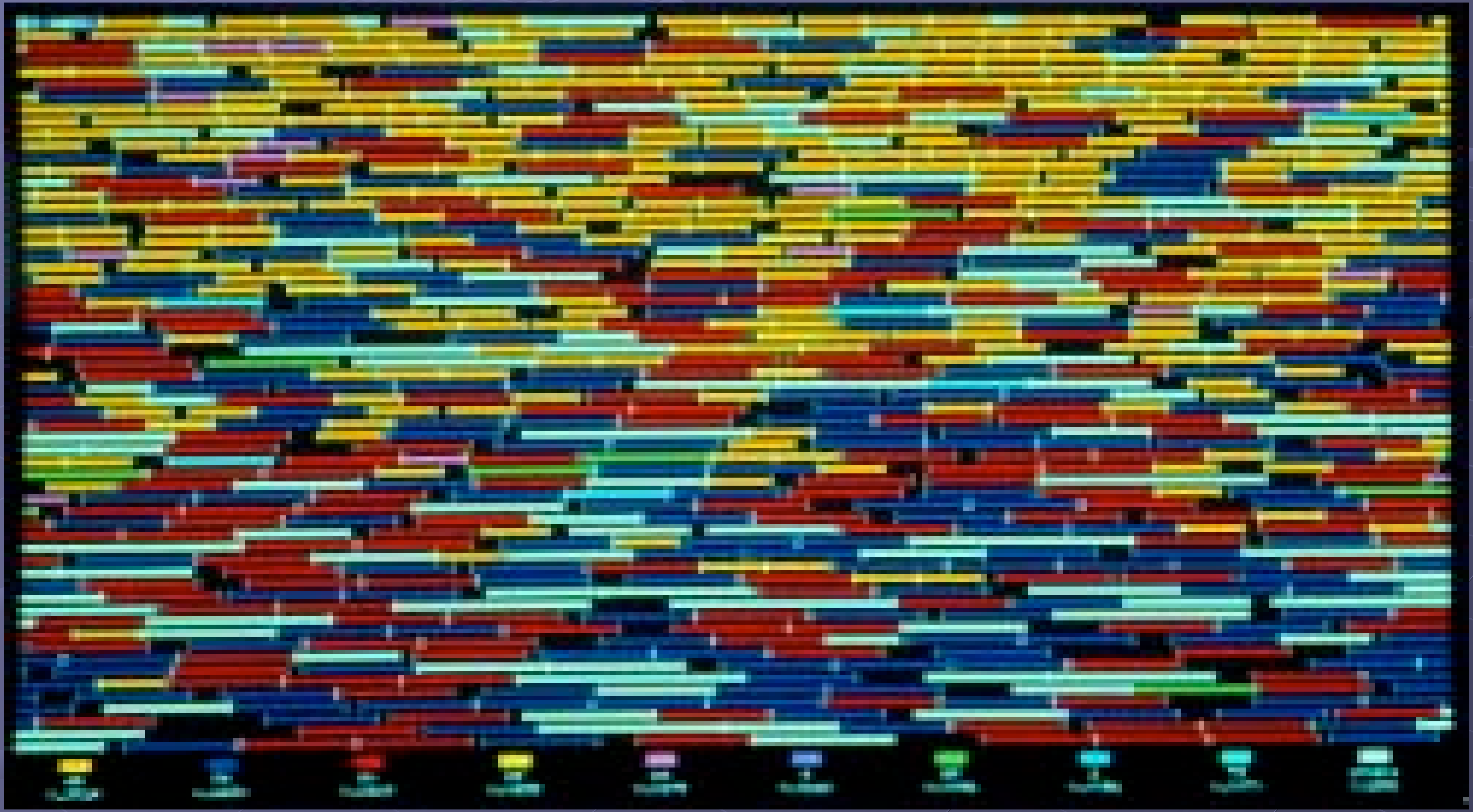
Tierra Pictures



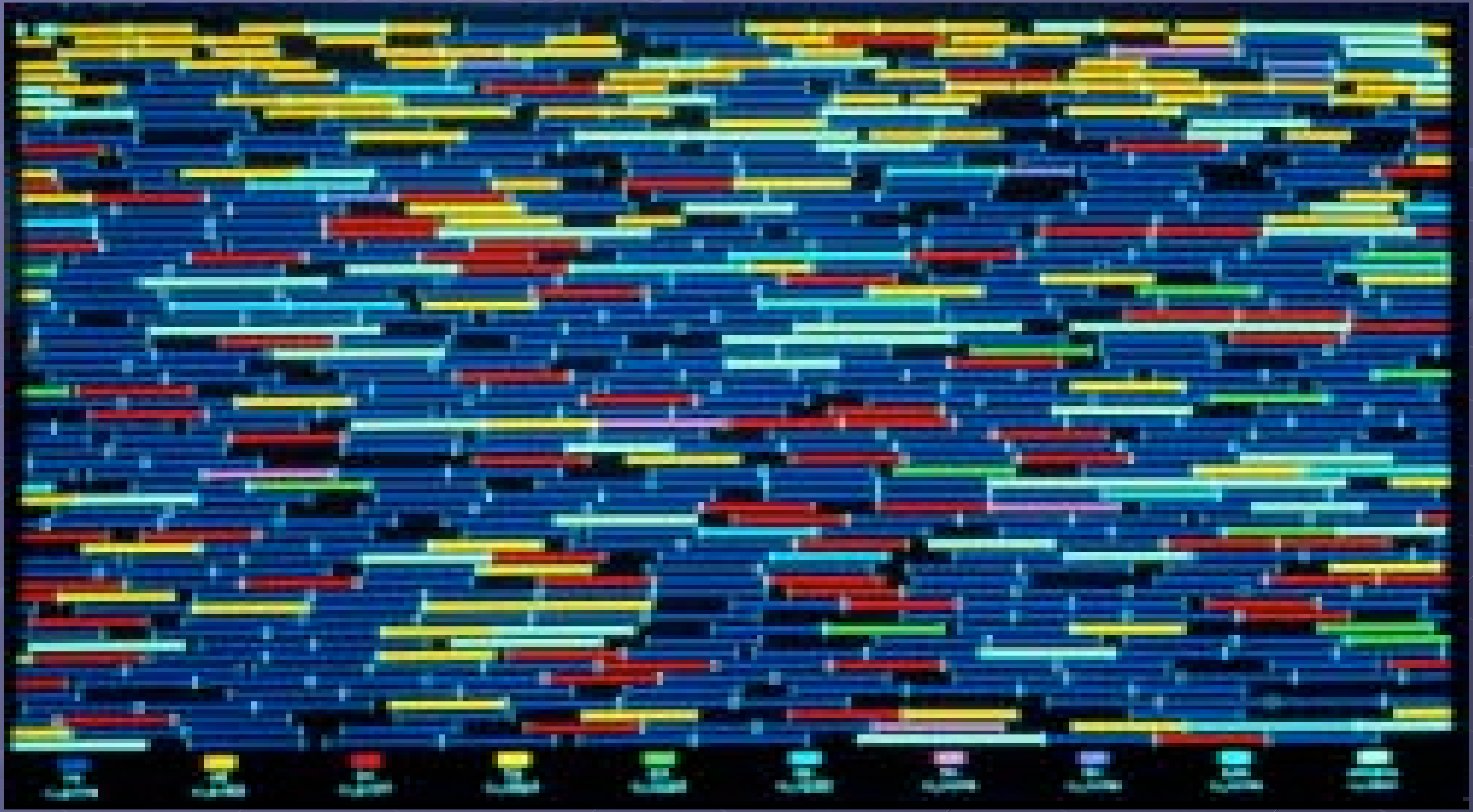
Tierra Pictures



Tierra Pictures



Tierra Pictures



Network Tierra

- A proposal to increase the power of Tierra
- Regular Tierra quickly reaches a point of genetic convergence.
  - As shown in the last picture, eventually, genetic variability collapses as the most successful form is developed.
- Network Tierra is a way to create a massive Tierra system with smaller niches that can communicate with other niches.
- Thus genetic mixing will continue because genetic specialization will only effect a single niche at a time.
- A nice idea, but eventually, genetic stagnation will occur just the same, as the environment is relatively simple without enough variability…

References

- Bedau, Mark. Artificial life: Organization, adaptation and complexity from the bottom up. Trends in Cognitive Sciences. 2003. Vol 7)11 505-12
- Holland, J.H. (1975/1992) Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence (2nd edn, expanded, 1992), University of Michigan Press
- Ray, T.S. (1992) An approach to the synthesis of life. In Artificial Life II (Langton, C.G. et al., eds), pp. 371–408, Addison-Wesley
- Very good simple genetic algorithm resource: http://cs.felk.cvut.cz/~xobitko/ga/
- Tierra's homepage:
  http://www.his.atr.jp/~ray/tierra/index.html