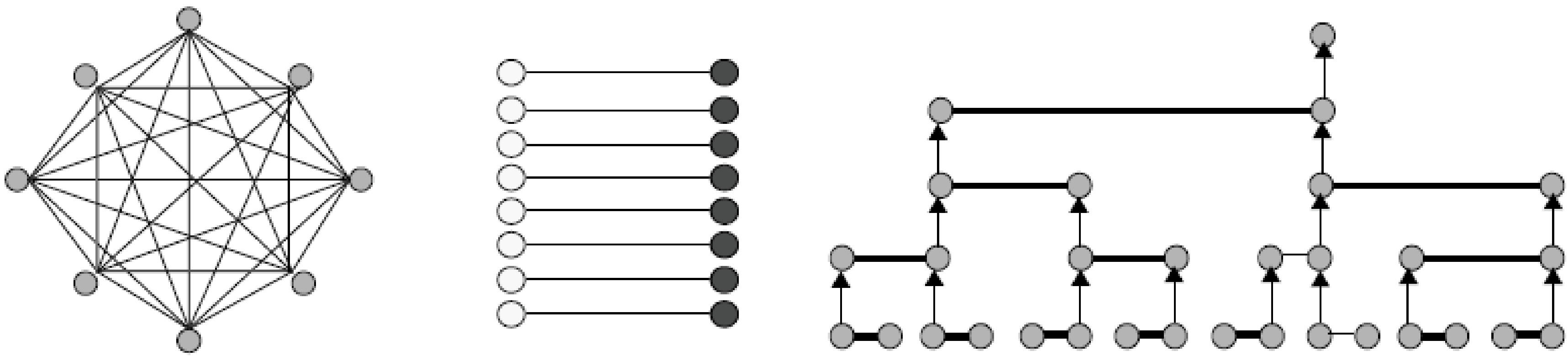
Competitive Fitness



Competitive Environments Evolve Better Solutions for Complex Tasks

Peter J. Angeline and Jordan B. Pollack

May 5, 1993

Ali
Artificial Life
Fall 2005

Reminder: Genetic Algorithm

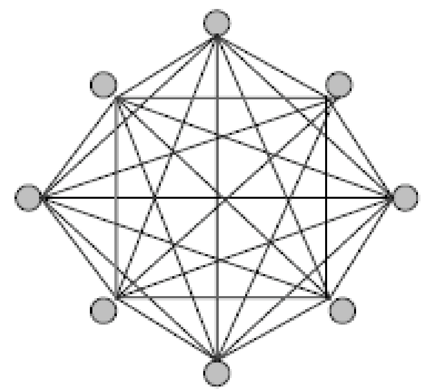
- Generate an initial population of random compositions and Iteratively perform the following
  - Using the fitness measure assign a fitness value to each individual.
  - Create a new population by applying the following operations. The operations are applied to individuals chosen from the population with a probability based on fitness.
    - (i) *Darwinian Reproduction*:
    - (ii) *Crossover*:
    - (iii) *Mutation*:
- *Genetic Algorithms* transform a population of individuals, each with an associated fitness value, into a new generation of the population using reproduction, crossover, and mutation.

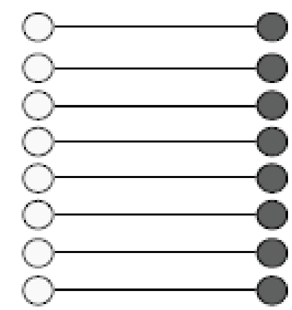
Fitness Function

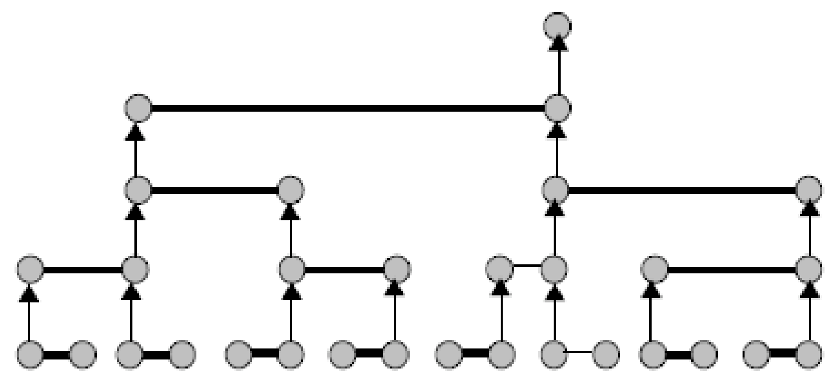
- Fitness Function is any way a GA rates its individuals for the purposes of creating the next generation.
- Static Fitness Function
    - independent of the contents of the population, and rate individuals based on closeness to the “goal”.
  - Potential Problem
    - Knowing something is close to the “goal” requires significant knowledge about the search space, in general this is as difficult as knowing the solution.
  - Suggested Solution
    - Use Competitive Fitness Function
- Applet (Minimum of 1D function)

Competitive Fitness Function

- Measure the individual's ability relative to the current population rather than the global optima.
- Three types of competitive fitness functions
  - Full Competition (a)
  - Bipartite Competition (b)
  - Tournament Fitness (c)


(a)


(b)


(c)

Competitive Fitness Function

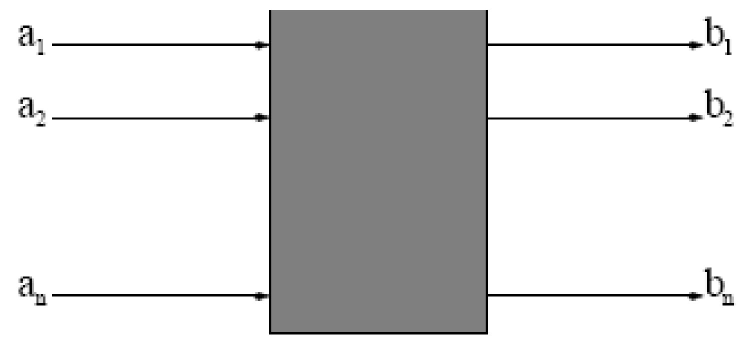
- Full Competition (Axelrod 1989)
  - Test every population member against every other population member
    - Number of competitions in each generation is n^2.
- Bipartite Competition (Hillis 1992)
  - Two co-evolving populations, members of one population are tested against members of the other population.
    - Reduce the number of competitions per generation, in Hillis's case n/2.
- Tournament Fitness (Angeline and Pollack 1993)
  - A single-elimination tournament used to establish relative fitness ranking. The fitness of an individual is its height in the playoff tree.
    - Number of competitions in each generation is n-1.

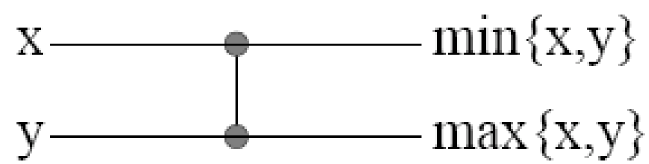
Hillis

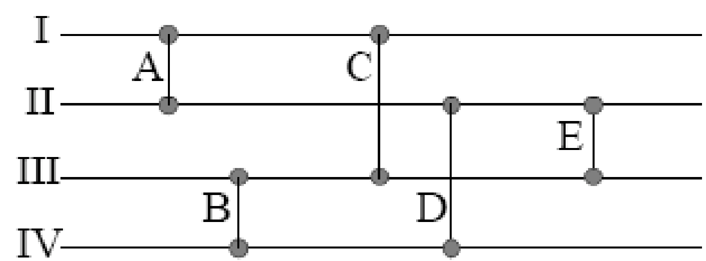
- Evolve a sorting network for sorting any arrangement of 16 integers using as few comparators as possible.
- In 1980's using an Independent Fitness Function
  - Best evolved sorting network used 65 comparators.
- In 1992 using an Competitive Fitness Function
  - The Fitness of the sorting network depended on how well it solved sorting problems.
  - The Fitness of the sorting problem depended on how well it found flaws in the sorting networks.
  - Best evolved sorting network used 61 comparators.
- Note: In 1969 best possible sorting network n = 16 was discovered that uses only 60 comparators.

Sorting Network

- n unsorted inputs $a_1, a_2, \ldots, a_n$.
- n sorted outputs $b_1 < b_2 < \ldots < b_n$.
- The set $\{b_i\}_{i=1,..n}$ is the same as the set $\{a_i\}_{i=1,,n}$.
- The network can use only comparators.







Angeline and Pollack

- Four experiments to evolve a Tic-Tac-Toe player.
- In the first three experiments, players evolved against a static "expert" strategy
  - RAND – choose a legal position at random.
  - NEAR – performs near optimally (it can be forked)
  - BEST – choose the optimal position to play (unbeatable)
    - A individual's average score over four games is its fitness.
- For the last experiment, evolving programs played against each other in a tournament structure.
  - A program wins against its opponent if it had the greater score after two games, with each player taking the first move in one game.
- In all experiments
  - A programs score is the number of moves it makes, 5 bonus points for a draw, 20 bonus points for a win.
  - Population size of 256, and ran for 150 generations.

Results and Discussion

Table 1: Performance of best evolved program from each experiment against the various "experts."

Fitness Function Used	*Evolved Program vs. RAND*			*Evolved Program vs. NEAR*			*Program vs. BEST*	
	Wins	Draws	Losses	Wins	Draws	Losses	Draws	Losses
RAND	1125	0	875	0	0	2000	0	2000
NEAR	802	104	1094	144	123	1733	0	2000
BEST	310	535	1155	0	360	1640	0	2000
POP	781	471	748	61	588	1351	481	1519

- The Independent fitness functions were unable to evolve an effective player
  - Player evolved against RAND can only beat a random player _ of the time.
  - Player evolved against BEST preferred to draw its opponent rather than win, and lost more than _ of its games against a random player.
  - Player evolved against NEAR lost 87% of games against a NEAR player
- The Tournament fitness function evolved a more robust player
  - It was able to draw against a perfect player _ of the time.
  - It lost less often against the experts than the players evolved against the experts.

Conclusions

- Advantages
  - Requires no particular knowledge of the search space.
  - Presents an adaptive development environment for the population. (The fitness landscape evolves along with the individuals)
  - Prevent large portions of the population from getting stuck at local optima.
  - Allows the GA to evolve a more general solution that approximates the global optima.
- Disadvantage
  - Since the population is only being compare with itself it is possible that the population become specialized at beating itself (red queen problem).