Genetic Algorithms: Principles of Natural Selection Applied to Computation

Stephanie Forrest

February 17, 2013

Presented By

April Suknot
Steven Garcia
Jeff Bowles

Overview

"Genetic algorithms allow engineers to use a computer to evolve solutions over time, instead of designing them by hand."

- Genetic Algorithms for Solving Problems
- Genetic Algorithms for Making Models
- Mathematical Analysis of Genetic Algorithms

Genetic Algorithms for Solving Problems

In the design of a genetic algorithm to solve a specific problem, there are two major decisions:

- Defining a concrete measure of fitness
- Representation Problem

Genetic Algorithms for Solving Problems

Function Optimization

Most obvious application: DeJong's Multiparameter Optimization - many problems can be formulated as a search for some optimal value.

- Assign regions of bit string to represent each parameter and determine the order
- Specify the domains for parameters

The precision of the solution is determined by how many bits are used to represent each parameter.

Genetic Algorithms for Solving Problems

Function Optimization

Gray codes address the increment/decrement issue because they have the property that incrementing or decrementing any number by one requires only a one bit change.

Table 1. Gray codes for three-bit numbers.

Decimal	Binary code	Gray code
0	000	000
1	001	001
2	010	011
3	011	010
4	100	110
5	101	111
6	110	101
7	111	100

Genetic Algorithms for Solving Problems

Ordering Problems

Ordering problems involve finding an optimal ordering for a sequence of N items. Some examples are:

- Travelling Salesman problem
- Bin-Packing
- Graph Coloring
- DNA Fragment Assembly

Genetic Algorithms for Solving Problems

Automatic Programming

Genetic algorithms have evolved to use special computer programs:

- Programs written in a subset of the programming language Lisp.
- Can be naturally represented as trees.
- Sometimes "junk" components of genetic programs can be useful in other areas.
- Appear less elegant and more redundant than human-designed programs.

For Example:

(-1 (* 2 (* sin (sin x))))

versus

(sin (- (- 2 (* x 2)) (sin (sin (sin (sin (sin (sin (* (sin (sin 1)) (sin (sin 1))))))))))

Genetic Algorithms for Solving Problems

Automatic Programming

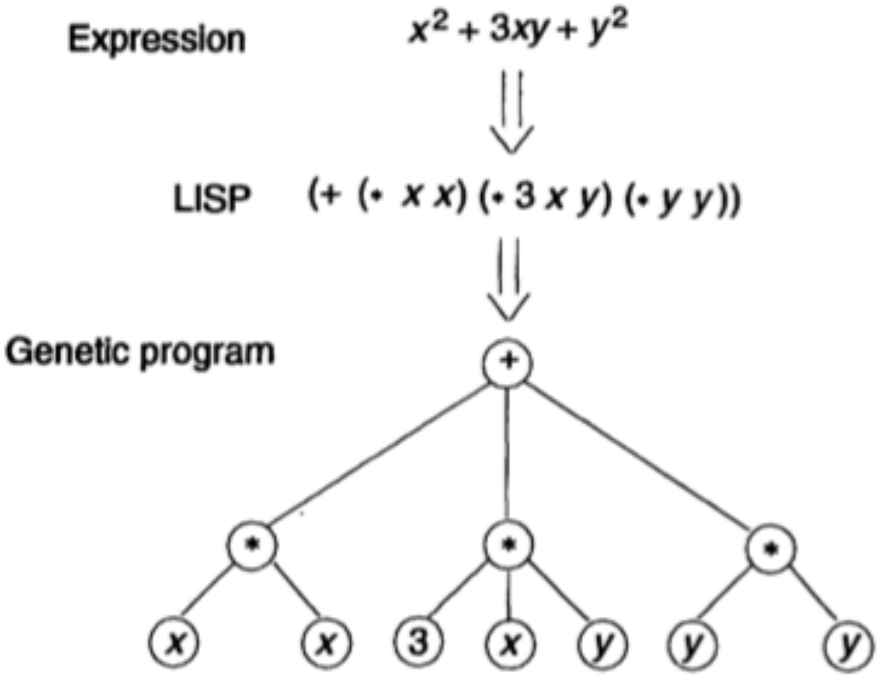


Fig. 4. Tree representation of computer programs. The displayed tree corresponds to the expression $x^2 + 3xy + y^2$. Operators for each expression are displayed as a root, and the operands for each expression are displayed as children.

Genetic Algorithms for Making Models

Modeling Ecological Systems

Characteristics

- highly abstract
- unlikely to make exact numerical predictions
- reveal conditions under which certain qualitative behaviors are likely to rise
  - diversity of phenotypes in a resource rich/poor environment
- discover critical parameters that have drastic effects on the outcomes

Genetic Algorithms for Making Models

Modeling Ecological Systems

the Echo system

1. there is no explicit fitness function
2. individuals (agents) have local storage
3. genetic representation is based on a higher cardinality alphabet e.g. $\{A, B, C, D\}$.

Genetic Algorithms for Making Models

Modeling Ecological Systems

Implicit Fitness Evaluation

1. resources modeled by elements of alphabet
2. agent genomes consist of elements of the alphabet
3. agents can make copies of themselves if resources allow
4. resources exist free in the environment or stored locally by agent

Genetic Algorithms for Making Models

Modeling Ecological Systems

Interactions

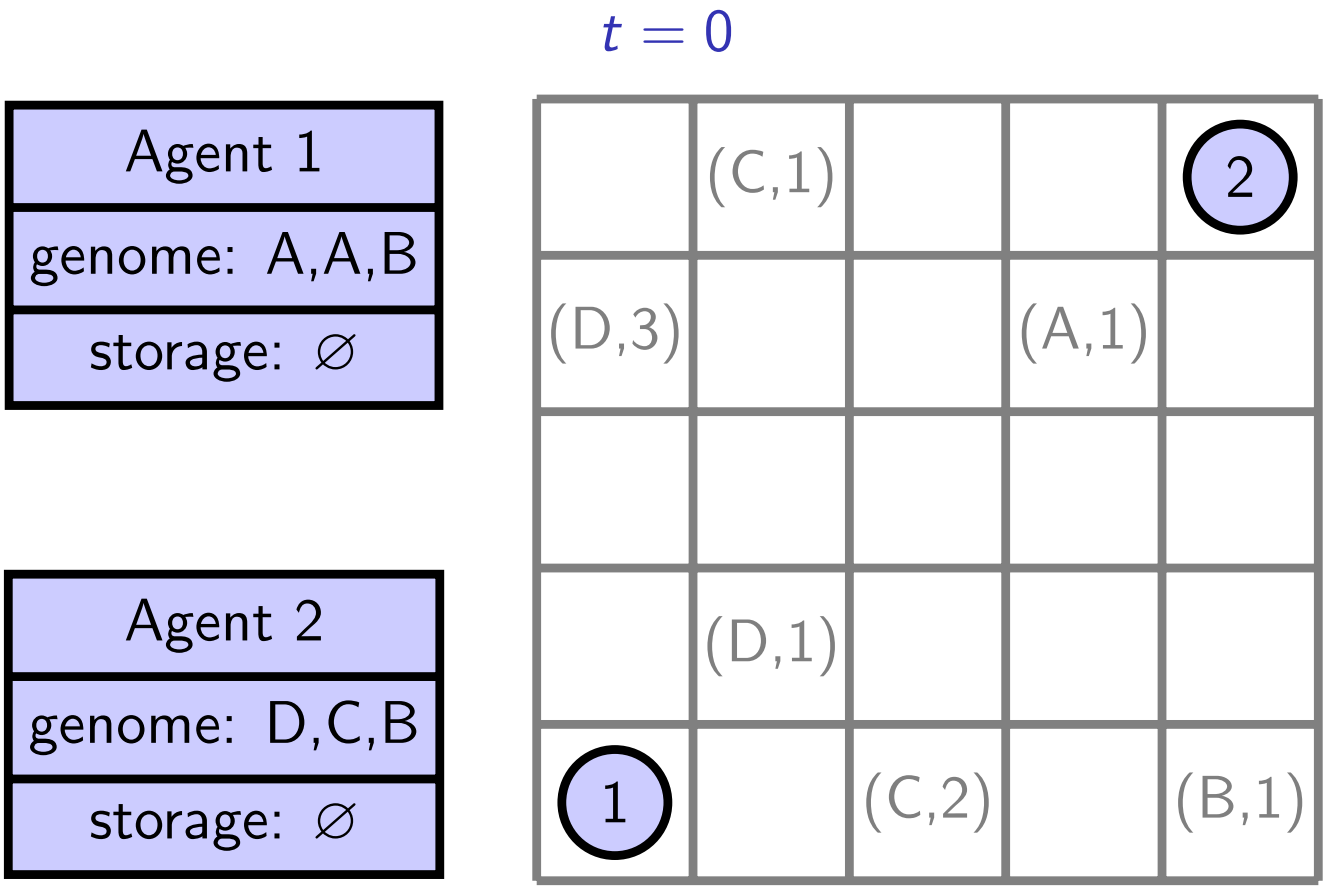
- trade between agents
- combat
- mating
  - exchange genetic material through crossover
  - mutation
  - evolution of new agents

Environment

- 2-D grid
- agents can cohabit sites and migrate
- at each timestep a fixed amount of resources become available at each site

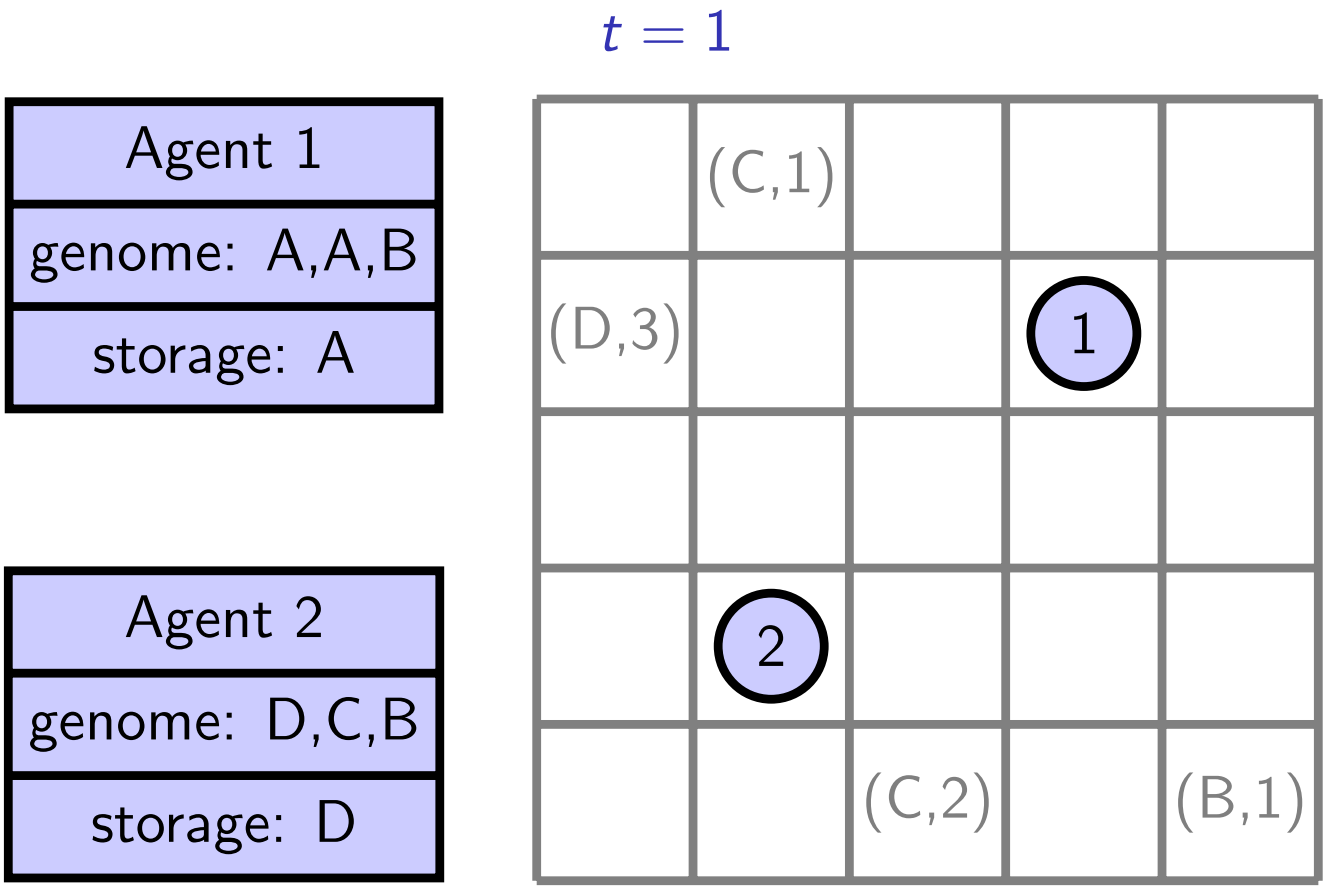
Genetic Algorithms for Making Models

Modeling Ecological Systems



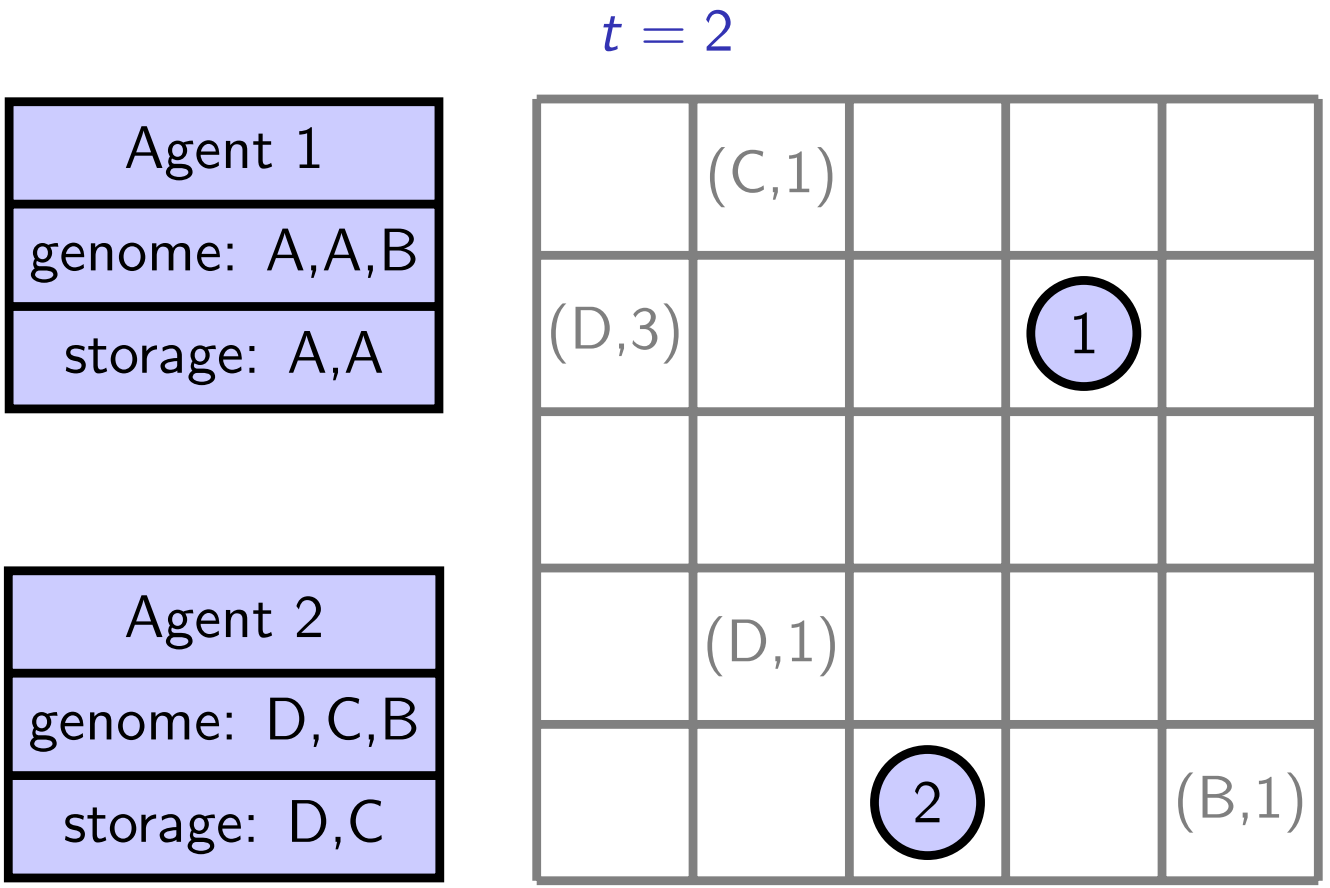
Genetic Algorithms for Making Models

Modeling Ecological Systems



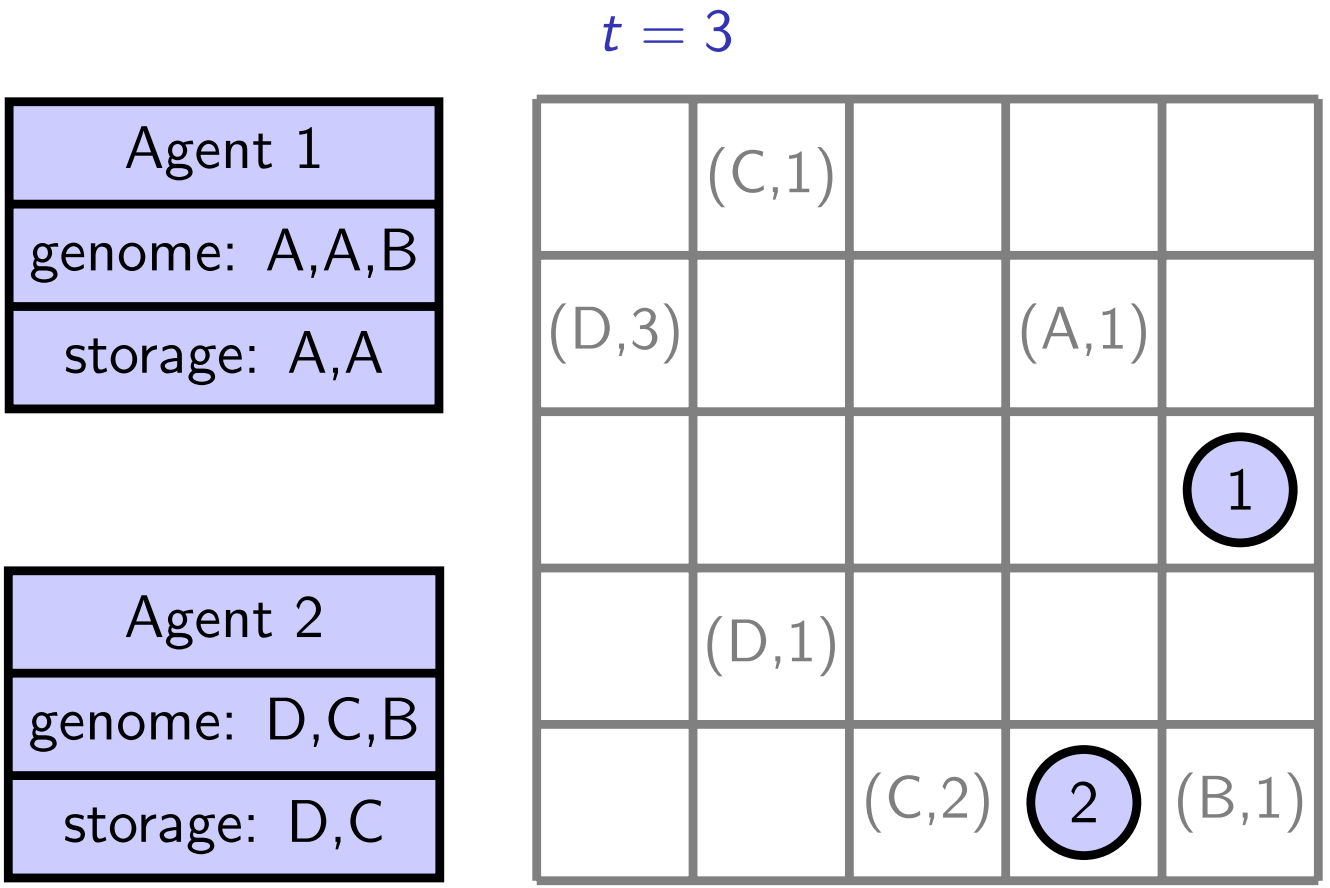
Genetic Algorithms for Making Models

Modeling Ecological Systems



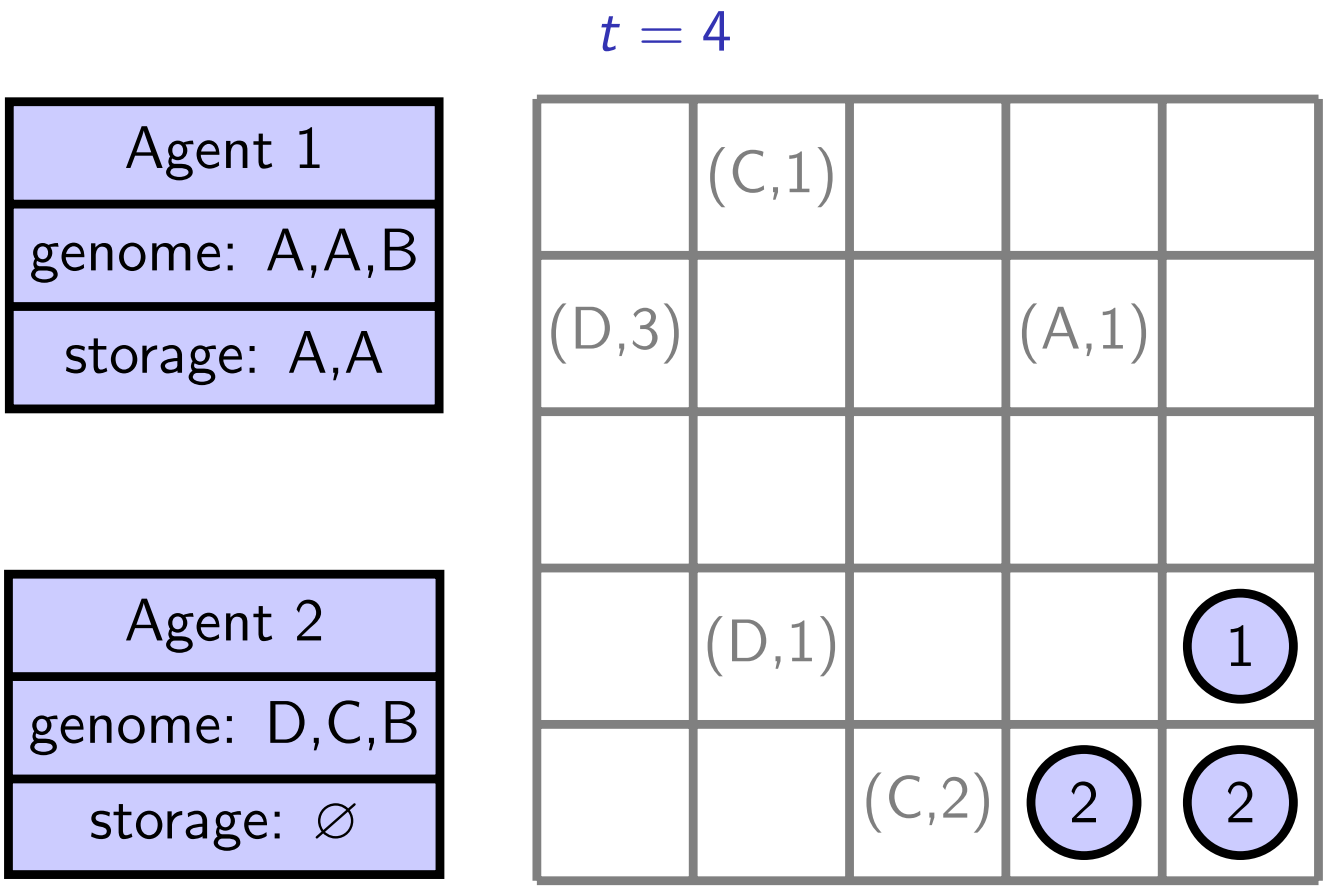
Genetic Algorithms for Making Models

Modeling Ecological Systems



Genetic Algorithms for Making Models

Modeling Ecological Systems



Genetic Algorithms for Making Models

Modeling Ecological Systems

Preliminary Simulations

- biological arms race
- ecological dependence among different species
- sensitivity to differing levels of renewable resources

Mathematical Analysis

- Let's predict how well a genetic algorithm (GA) will perform!
- Good luck with that
- Consider problems with the following characteristics:
  - Little analytical knowledge
  - Complex
  - Noisy
  - Dynamic
- Virtually impossible to predict how well **any** algorithm will perform
  - Deterministic
  - Non-deterministic
- This may be one reason why research in this area has not produced definitive answers.

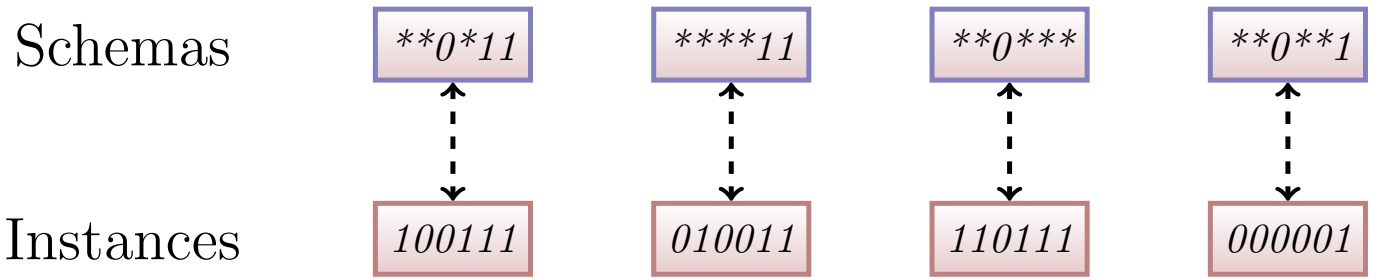
Mathematical Analysis

- If it's hard to predict how well a GA will work, can it work well?
- Yes, GAs can work well, if a global optimum is not needed.
- GAs search for bit strings in $\{0,1\}^n$.
- If n is large, it is not feasible to search entire space
- GA is searching for bit strings with high fitness

Mathematical Analysis

Schemas

Bit string abstraction: a schema is a template of a bit string.



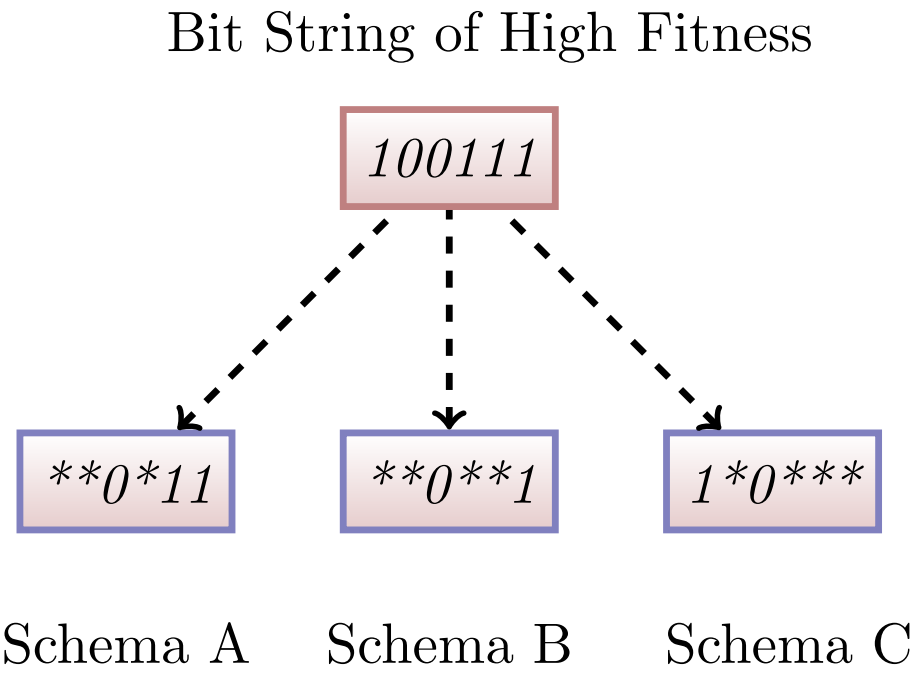
Bit strings that match the templates are *instances* of a schema.

Mathematical Analysis

- A bit string belongs to 2^n schemas
- Fit bitstring == fit schema
- Implicit parallelism
  - All bitstrings that match a schema can be judged at the same time
  - This is a very nice optimization

Mathematical Analysis

Matching Schemas

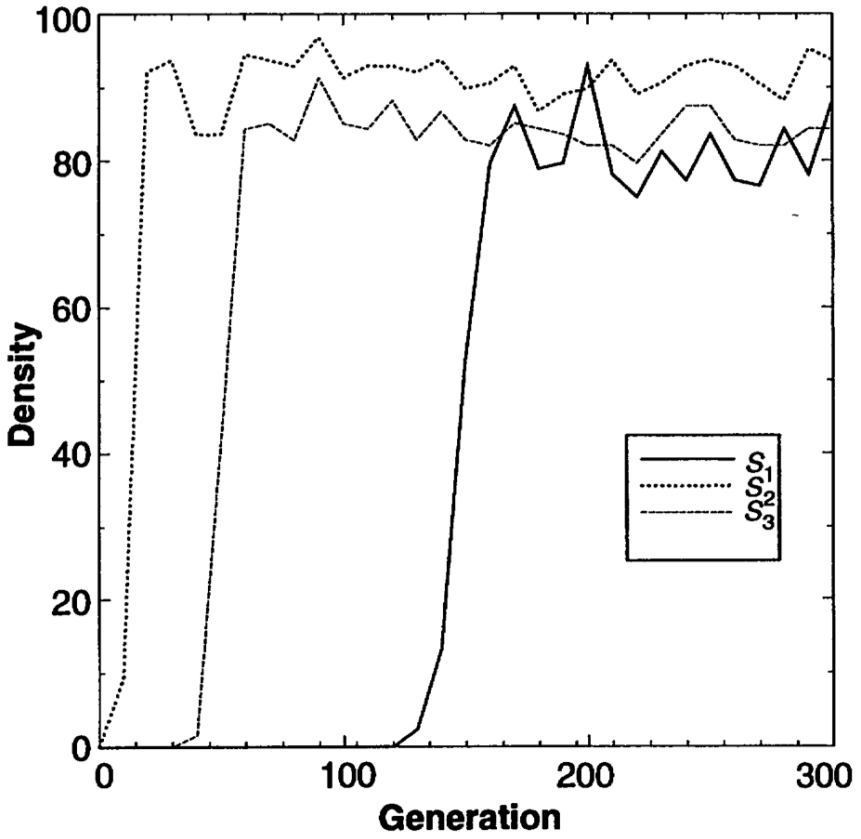


Example of a high fitness bitstring matching three schemas and the resulting implicit parallelism

Mathematical Analysis

Schema Theorem

The observed best schemas are expected to receive an exponentially increasing number of samples in successive generations.



Mathematical Analysis

Schema Theorem

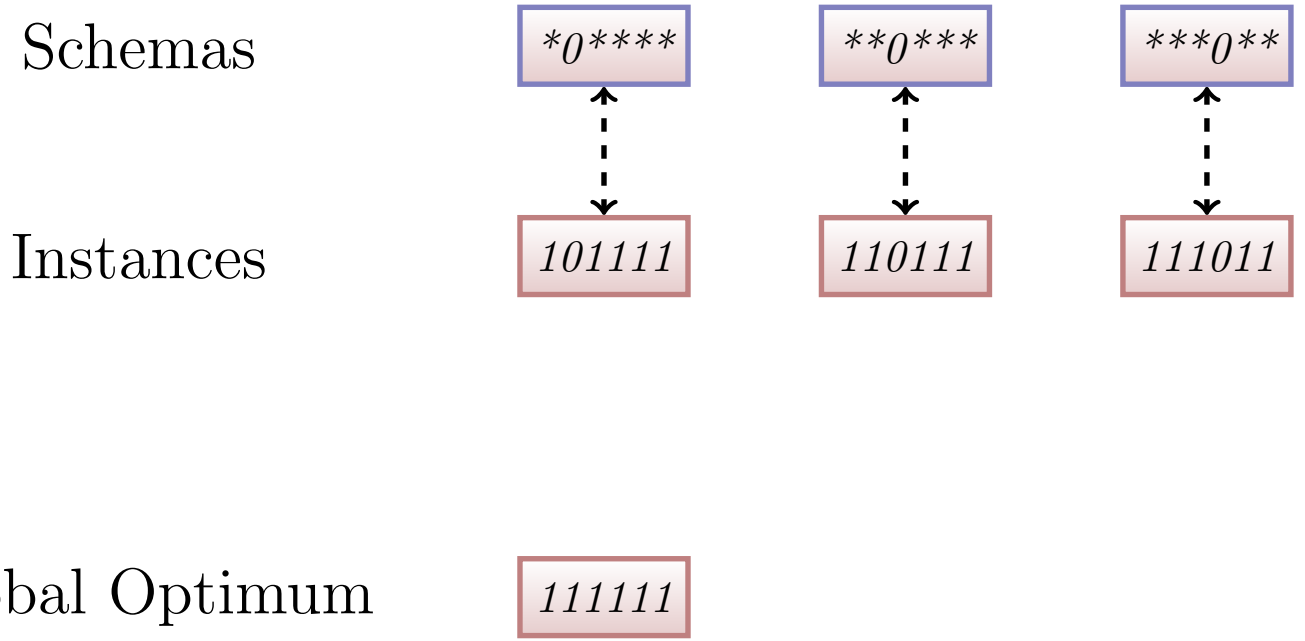
This property of GAs is a double edged sword:

- The strong convergence property can find good solutions fast
- The strong convergence property can find bad solutions fast

Mathematical Analysis

Deception

There may be some schemas that are templates for high fitness bit strings:



But they lead away from the global optimum. In this case, the GA would be deceived and led away from the global optimum

Summary

We discussed:

- Genetic Algorithms for Solving Problems
- Genetic Algorithms for Making Models
- Mathematical Analysis of Genetic Algorithms

Discussion

- Can genetic algorithms be used to evolve robots?
  - What are the characteristics of robots?

Discussion

- Can the stock market be predicted with GAs?

Discussion

- Is the signal in the stock market noisy? Dynamic? Chaotic?
- When predicting the stock market, would a GA be looking for local minima and maxima?