- # Welcome

- Welcome
- Committee
  - Dr. Arvin Agah, Professor in Charge
  - Dr. Victor Frost
  - Dr. Costas Tsatsoulis

Overview of Presentation

- # Problem and Solution Overview

Overview of Presentation

- Problem and Solution Overview
- Playing Go

Overview of Presentation

- Problem and Solution Overview
- Playing Go
- Relevant Computational Methods

Overview of Presentation

- **Problem and Solution Overview**
- **Playing Go**
- **Relevant Computational Methods**
- **State of the Art**
  - **Genetic algorithms**
  - **Traditional programs**

Overview of Presentation

- Problem and Solution Overview
- Playing Go
- Relevant Computational Methods
- State of the Art
  - Genetic algorithms
  - Traditional programs
- Implementation

Overview of Presentation

- Problem and Solution Overview
- Playing Go
- Relevant Computational Methods
- State of the Art
  - Genetic algorithms
  - Traditional programs
- Implementation
- Experiments and Results

Overview of Presentation

- Problem and Solution Overview
- Playing Go
- Relevant Computational Methods
- State of the Art
  - Genetic algorithms
  - Traditional programs
- Implementation
- Experiments and Results
- Conclusion
  - Contributions
  - Limitations
  - Future

- Perfect Information

- Perfect Information
- Board is 19 by 19

- Perfect Information
- Board is 19 by 19
- Two players

- Perfect Information
- Board is 19 by 19
- Two players
- Territory

- **Perfect Information**
- **Board is 19 by 19**
- **Two players**
- **Territory**
- **Capturing**

- Go has simple rules, but tactics and strategies are complex
  - Go has emergent complexity
  - Multiagent systems have emergent complexity

- Go has simple rules, but tactics and strategies are complex
  - Go has emergent complexity
  - Multiagent systems have emergent complexity
- Current go programs play on a beginner level, why?

- Go has simple rules, but tactics and strategies are complex
  - Go has emergent complexity
  - Multiagent systems have emergent complexity
- Current go programs play on a beginner level, why?
- Search space

Search Ply	Go	Chess	Checkers
1	361	20	7
2	129,960	400	49
3	445,145,640	approx. 10,000	approx. 343

Goals of This Project

- Use multiple agents to suggest solutions based on a narrow world perspective

Goals of This Project

- **Use multiple agents to suggest solutions based on a narrow world perspective**
- **Bring these solutions together to obtain a better overall solution**

Goals of This Project

- Use multiple agents to suggest solutions based on a narrow world perspective
- Bring these solutions together to obtain a better overall solution
- Agents are not fully decentralized

Goals of This Project

- Use multiple agents to suggest solutions based on a narrow world perspective
- Bring these solutions together to obtain a better overall solution
- Agents are not fully decentralized
- Algorithmic composition of individual agent solutions

Goals of This Project

- Use multiple agents to suggest solutions based on a narrow world perspective
- Bring these solutions together to obtain a better overall solution
- Agents are not fully decentralized
- Algorithmic composition of individual agent solutions
- Illustrate this method in a non-trivial environment: go

- Multiagent Architecture
  - Specialized agents
  - Each has its own perspective of the game
  - Outputs an array representing move qualities

Research Approach

- **Multiagent Architecture**
  - Specialized agents
  - Each has its own perspective of the game
  - Outputs an array representing move qualities
- **Agents connected via a summation network to generate output**
  - No communication
  - Allows a passive combination of agent output into a solution

- **Multiagent Architecture**
  - **Specialized agents**
  - **Each has its own perspective of the game**
  - **Outputs an array representing move qualities**
- **Agents connected via a summation network to generate output**
  - **No communication**
  - **Allows a passive combination of agent output into a solution**
- **Weights for this network were evolved using genetic algorithms**

Chromosome Description

- **Network weights are four-bit integers**

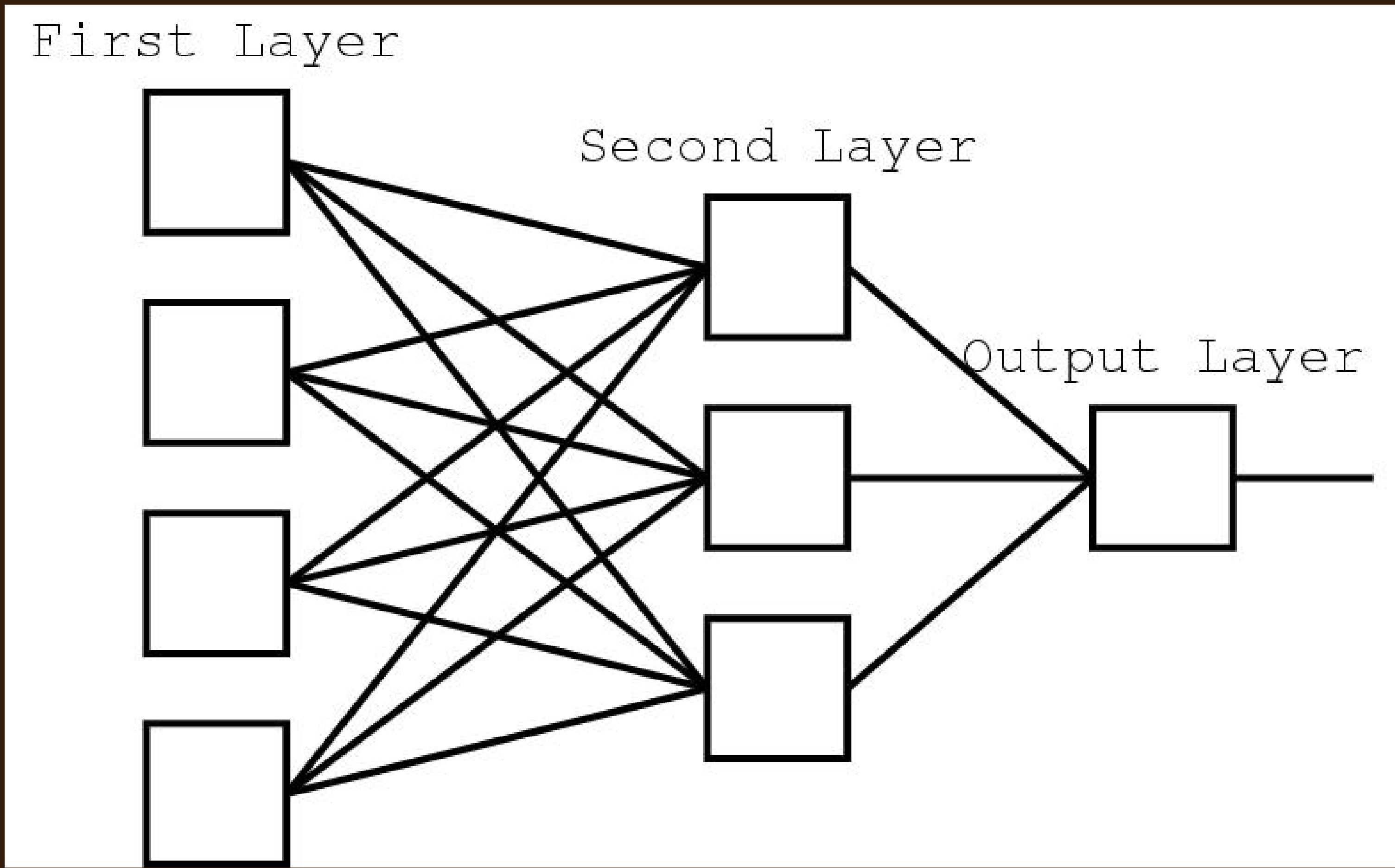
Chromosome Description

- Network weights are four-bit integers
- These four-bit integers make up chromosome

Chromosome Description

- Network weights are four-bit integers
- These four-bit integers make up chromosome
- Extra bits at the end of chromosome are available
  - Extra bits for internal use by agents
  - Extender agent uses these extra bits

Diagram of Summation Network



Why Use Go?

- Traditional search provides little help

Why Use Go?

- Traditional search provides little help
- Complex

Why Use Go?

- Traditional search provides little help
- Complex
- Heavily pattern-oriented

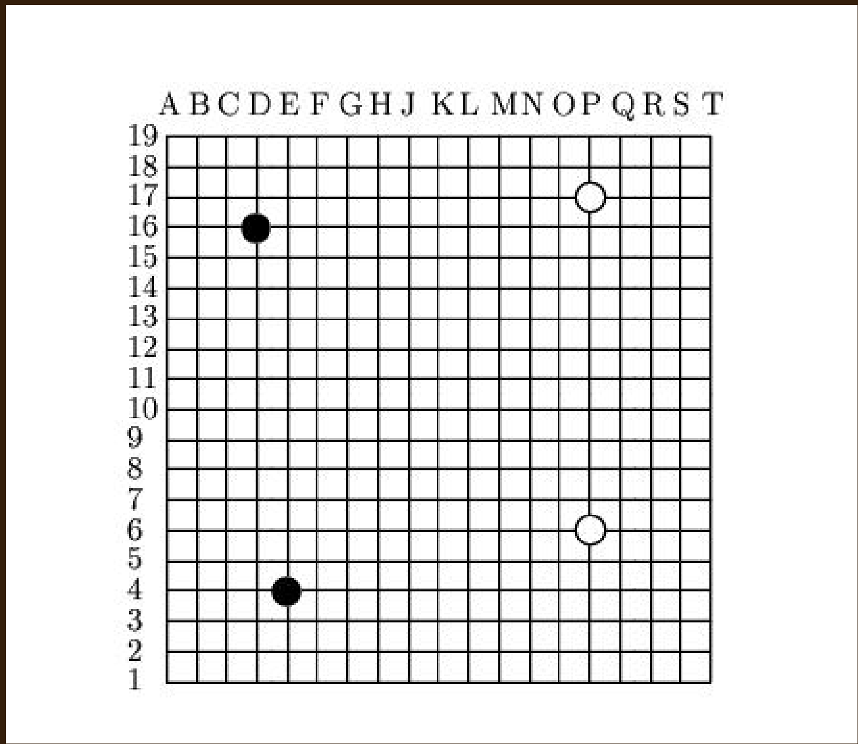
Why Use Go?

- Traditional search provides little help
- Complex
- Heavily pattern-oriented
- Unsolved now and in the near future

Why Use Go?

- Traditional search provides little help
- Complex
- Heavily pattern-oriented
- Unsolved now and in the near future
- Analogues to more complex environments
  - Local versus global concerns
  - Many choices at any point
  - Adversarial

The Go Board



- Groups

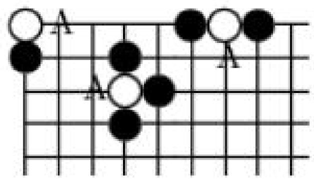
Playing Go (continued)

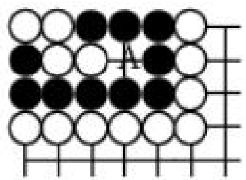
- **Groups**
- **Eyes**

Playing Go (continued)

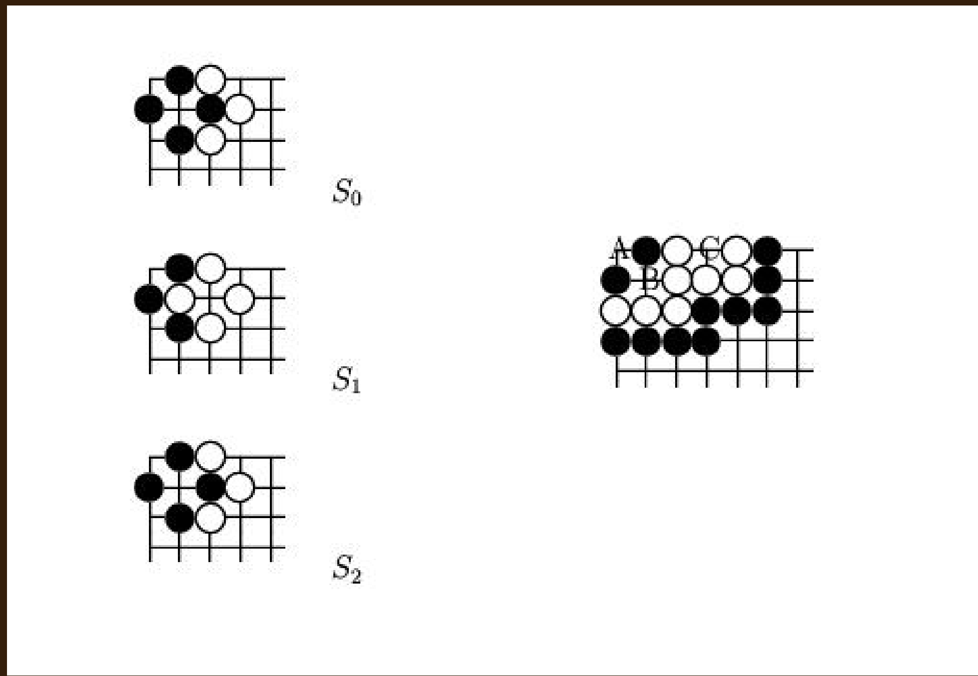
- **Groups**
- **Eyes**
- **Live and Dead Stones**

Capturing

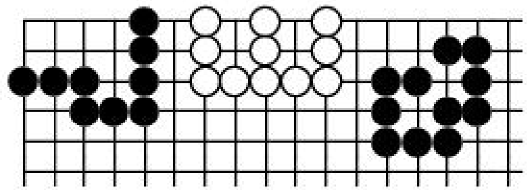




Ko and Seki Rules



Territory



Playing Go (continued)

- **Scoring**

Playing Go (continued)

- **Scoring**
- **Other board sizes**

Playing Go (continued)

- Scoring
- Other board sizes
- Handicaps

- Random search

- Random search
- Populations

- Random search
- Populations
- Chromosomes representing parameters or solutions

- Random search
- Populations
- Chromosomes representing parameters or solutions
- Fitness functions

- Random search
- Populations
- Chromosomes representing parameters or solutions
- Fitness functions
- Crossover, mating, and mutations

- **Autonomous agents**

- Autonomous agents
- Sense environment

- Autonomous agents
- Sense environment
- Interacts with environment

- Autonomous agents
- Sense environment
- Interacts with environment
- Cooperative or adversarial

- No Soft Methods

Good Traditional Programs

- No Soft Methods
- Müller
  - Patricia trees variant
  - 3000 pattern database

- No Soft Methods
- Müller
  - Patricia trees variant
  - 3000 pattern database
- Many Faces of Go
  - Opening database of 45,000 moves
  - Pattern database of 1000 patterns
  - 200 rules hardcoded

Good Traditional Programs

- No Soft Methods
- Müller
  - Patricia trees variant
  - 3000 pattern database
- Many Faces of Go
  - Opening database of 45,000 moves
  - Pattern database of 1000 patterns
  - 200 rules hardcoded
- Others
  - Patterns
  - Try to create small set of possible moves to look into

- Da Silva
  - Evaluation function evolved

Other Programs Using Genetic Algorithms

- Da Silva
  - Evaluation function evolved
- Jeffrey Greenberg
  - Evolved Prolog-like statements

- Da Silva
  - Evaluation function evolved
- Jeffrey Greenberg
  - Evolved Prolog-like statements
- Ecological Models
  - Sets of rules were evolved

Other Programs Using Genetic Algorithms

- Da Silva
  - Evaluation function evolved
- Jeffrey Greenberg
  - Evolved Prolog-like statements
- Ecological Models
  - Sets of rules were evolved
- Neural Network Hybrids: SANE
  - Neural network configuration and weights evolved
  - Entire board fed into neural network

Other Programs Using Genetic Algorithms

- Da Silva
  - Evaluation function evolved
- Jeffrey Greenberg
  - Evolved Prolog-like statements
- Ecological Models
  - Sets of rules were evolved
- Neural Network Hybrids: SANE
  - Neural network configuration and weights evolved
  - Entire board fed into neural network
- Common Themes
  - Small Boards
  - No Meta-processing
  - Not multiagent

Support Classes

- Bit-level operations for Stone class for speed

Support Classes

- Bit-level operations for Stone class for speed
- Board class is a 1D array of stone classes

Support Classes

- Bit-level operations for Stone class for speed
- Board class is a 1D array of stone classes
- Game class is a linked list of Boards

Support Classes

- Bit-level operations for Stone class for speed
- Board class is a 1D array of stone classes
- Game class is a linked list of Boards
- Probability Board class
  - Parallel to board array
  - Each offset is a move quality
  - Summation, normalization, and scaling provided
  - Spin

- Moderator class, a template

Interfaces

- Moderator class, a template
- Multiagent genetic algorithm player

Interfaces

- Moderator class, a template
- Multiagent genetic algorithm player
- Genetic algorithm trainer player
  - Fitness function

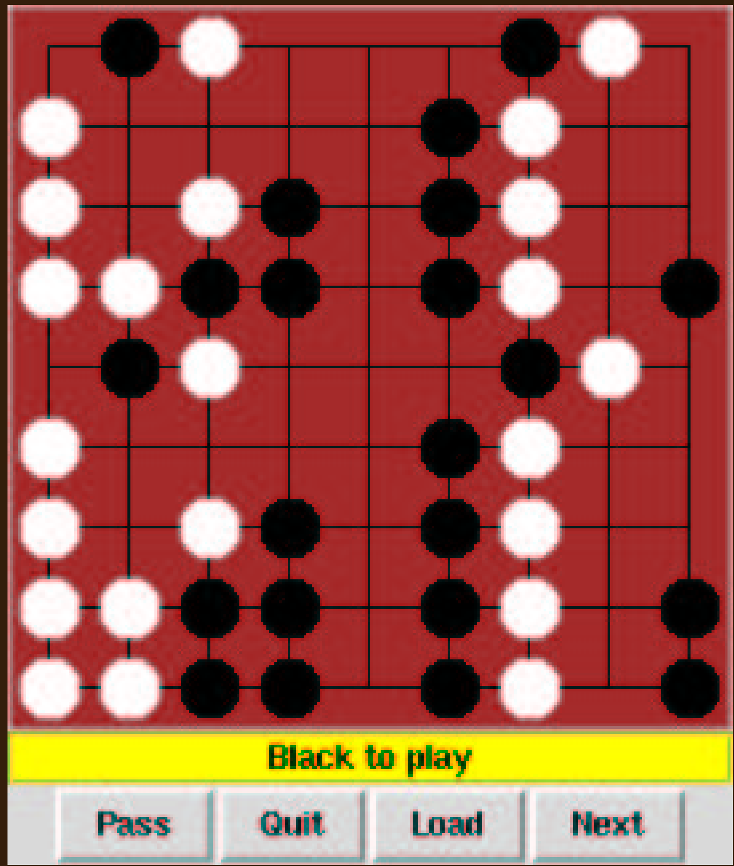
Text User Interface

```
 8| . . # . . . . . . | 8
 7| . . o . . . o . . | 7
 6| . . . . . . . . . | 6
 5| . . . . . . . . . | 5
 4| . . . . . . . . . | 4
 3| . . . . . . # . . | 3
 2| . . . # . . . . . | 2
 1| . . . . . . . . . | 1
   _A_B_C_D_E_F_G_H_J_

   _A_B_C_D_E_F_G_H_J_
 9| . . . . . . . . . | 9
 8| . . # . . . . . . | 8
 7| . . o . . . o . . | 7
 6| . . . . . . . . . | 6
 5| . . . . . . . . . | 5
 4| . . . . . . . . . | 4
 3| . . o . . . # . . | 3
 2| . . . # . . . . . | 2
 1| . . . . . . . . . | 1
   _A_B_C_D_E_F_G_H_J_

P=pass, A3=play at (A,3), R=redisplay, Q=quit
Move>
```

Graphical User Interface



Agents

- **Random**

Agents

- Random
- Follower

Agents

- Random
- Follower
- Opener

Agents

- Random
- Follower
- Opener
- Capture

Agents

- Random
- Follower
- Opener
- Capture
- Tiger’s Mouth

Agents

- Random
- Follower
- Opener
- Capture
- Tiger’s Mouth
- Extender
  - Uses GA values internally

Experiments Overview

- **Each Agent Individually**

Experiments Overview

- Each Agent Individually
- Random Agent

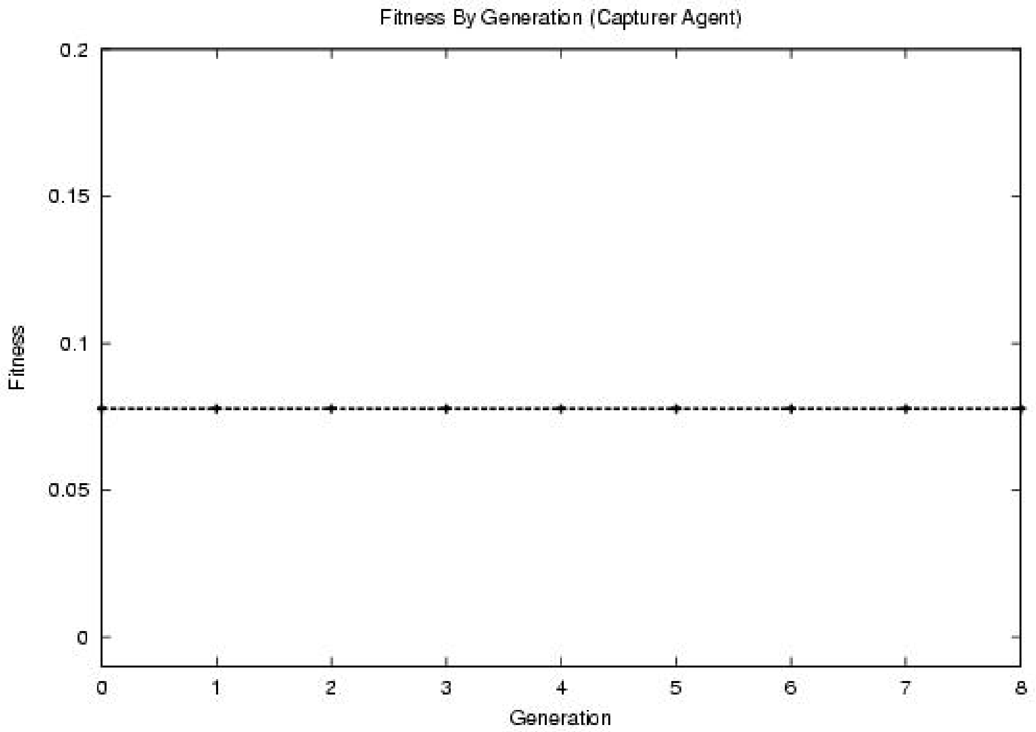
Experiments Overview

- Each Agent Individually
- Random Agent
- Multiagent

- Each Agent Individually
- Random Agent
- Multiagent
- GA parameters
  - Crossover 0.4
  - Mutation 0.0333
  - Population size: 10 and 100

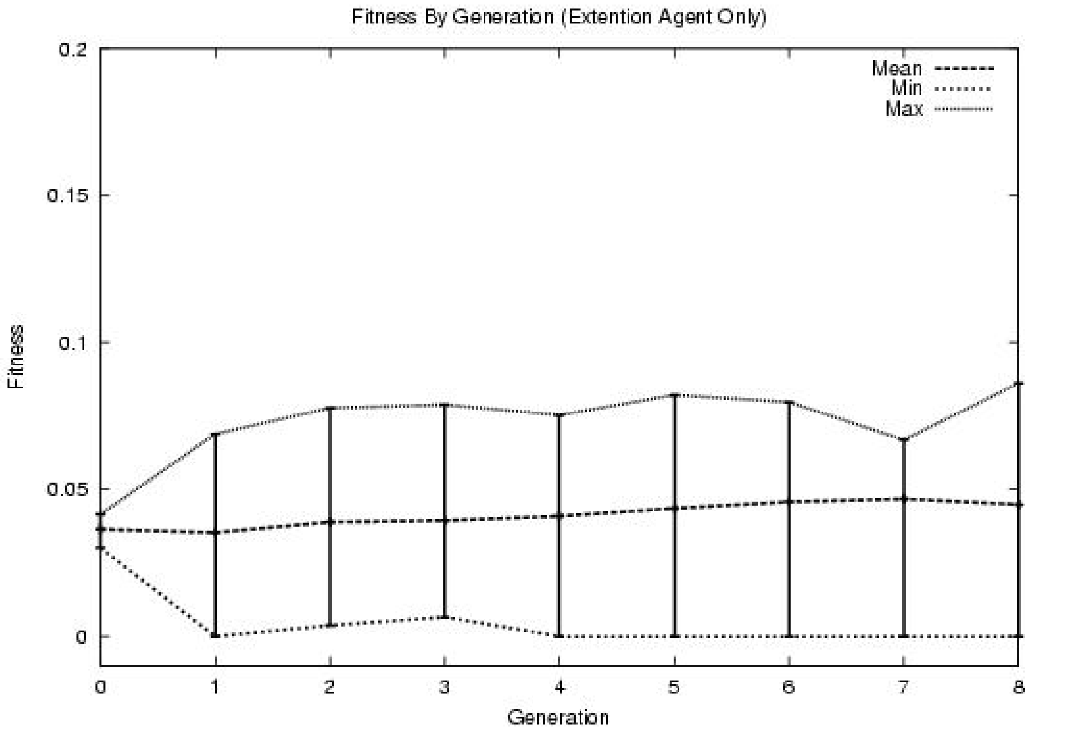
Results of Single Agents, Capturer

Generation	Max	Min	Mean	Std. Dev.	Sumfitness
0	0.0777	0.0777	0.0777	7.85e-09	0.777
1	0.0777	0.0777	0.0777	2.95e-05	0.777
2	0.0777	0.0777	0.0777	2.95e-05	0.777
3	0.0777	0.0777	0.0777	0.00181	0.777
4	0.0777	0.0777	0.0777	0.00181	0.777
5	0.0777	0.0777	0.0777	0.0142	0.777
6	0.0777	0.0777	0.0777	0.0142	0.777
7	0.0777	0.0777	0.0777	0.0397	0.777
8	0.0777	0.0777	0.0777	0.0397	0.777



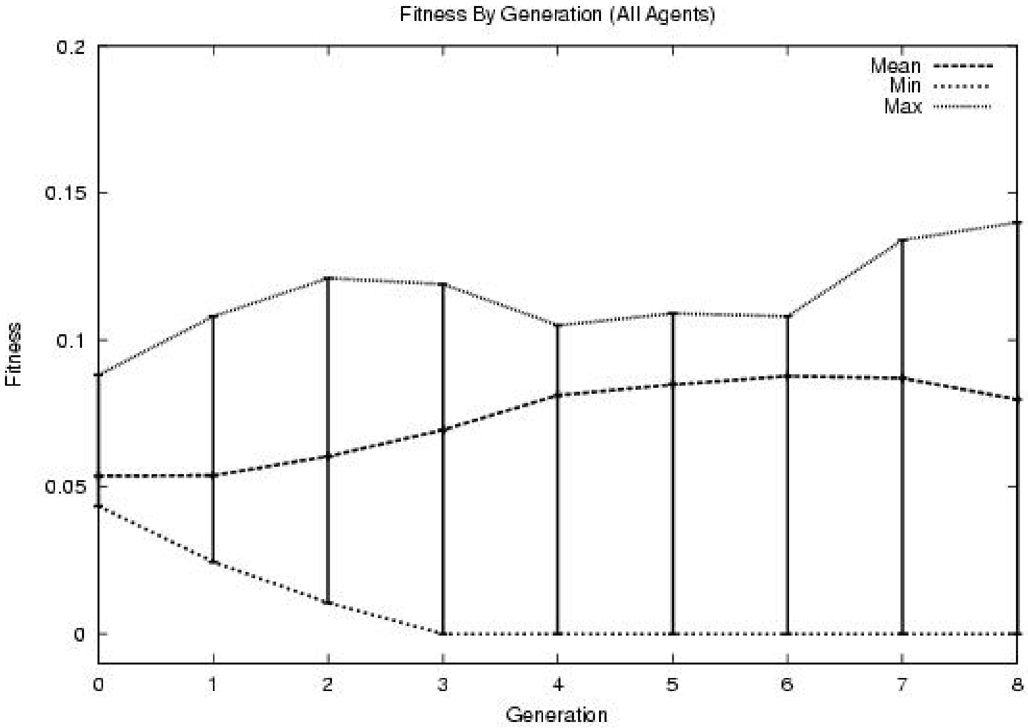
Results of Single Agent Experiment: Extender

Generation	Max	Min	Mean	Std. Dev.	Sumfitness
0	0.0415	0.0301	0.0365	0.00428	0.365
1	0.0689	0	0.0354	0.0319	0.354
2	0.0777	0.0037	0.0389	0.0347	0.389
3	0.0788	0.00656	0.0394	0.0641	0.394
4	0.0753	2.76e-10	0.0409	0.0667	0.409
5	0.0821	8.1e-09	0.0436	0.0899	0.436
6	0.0797	5.57e-09	0.0458	0.0893	0.458
7	0.0669	0	0.0468	0.102	0.468
8	0.0861	0	0.0449	0.105	0.449

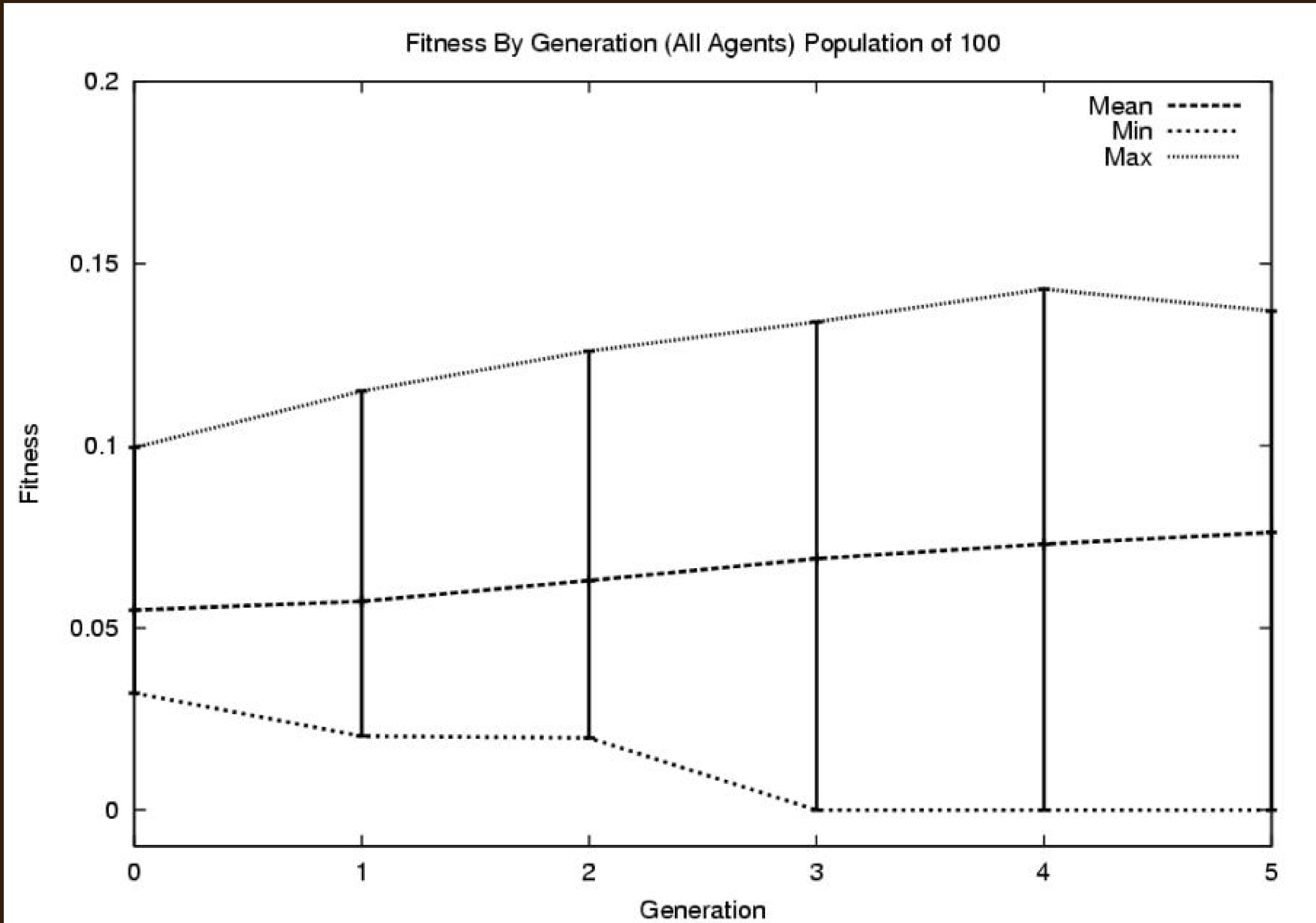


Results of Multiagent Experiment

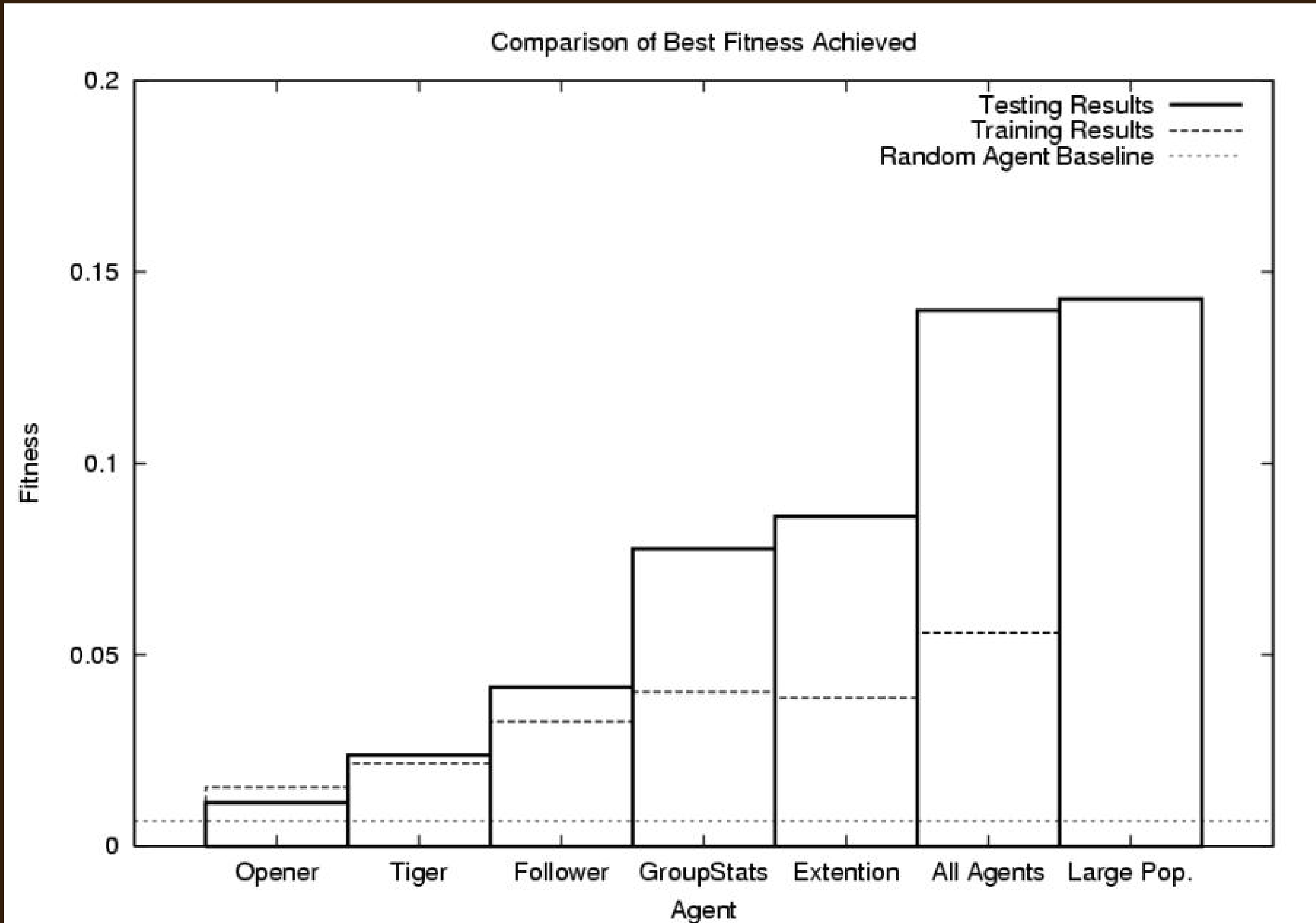
Generation	Max	Min	Mean	Std. Dev.	Sumfitness
0	0.0881	0.0435	0.0537	0.0138	0.537
1	0.108	0.0245	0.0539	0.0454	0.539
2	0.121	0.0106	0.0604	0.0575	0.604
3	0.119	3.49e-10	0.0694	0.0865	0.694
4	0.105	3.15e-09	0.0812	0.0867	0.812
5	0.109	2.82e-09	0.0848	0.103	0.848
6	0.108	2.15e-09	0.0877	0.103	0.877
7	0.134	0	0.087	0.113	0.87
8	0.14	0	0.0798	0.118	0.798



Results of Multiagent Experiment: Large Population



Comparison Plot



Results Summary (Multiagent, Population 100)

- 0.143 Highest fitness

Results Summary (Multiagent, Population 100)

- 0.143 Highest fitness
- 0.076 Highest mean fitness

Results Summary (Multiagent, Population 100)

- 0.143 Highest fitness
- 0.076 Highest mean fitness
- Student’s T-test
  - T-test, 100 population confidence: 3.89E-21
  - T-test, 10 population confidence: 5.04E-4

Contributions

- Unique approach to go

Contributions

- Unique approach to go
- Probabilistic methods for go

Contributions

- Unique approach to go
- Probabilistic methods for go
- Multiagent paradigm

Contributions

- Unique approach to go
- Probabilistic methods for go
- Multiagent paradigm
- Scalability

Limitations

- Board Size

Limitations

- Board Size
- Number of Agents

Limitations

- Board Size
- Number of Agents
- Time to run genetic algorithms
  - Training sets
  - Populations
  - Larger summation networks
  - Generations

Limitations

- Board Size
- Number of Agents
- Time to run genetic algorithms
  - Training sets
  - Populations
  - Larger summation networks
  - Generations
- Programmer’s knowledge of go

Future

- Larger population size

Future

- Larger population size
- Larger board size

Future

- Larger population size
- Larger board size
- More agents

Future

- Larger population size
- Larger board size
- More agents
- Agents of higher complexity

Future

- Larger population size
- Larger board size
- More agents
- Agents of higher complexity
- Larger summation network

Questions

- **Thank You**

Questions

- Thank You
- Thread Pools

Questions

- Thank You
- Thread Pools
- Search

Questions

- Thank You
- Thread Pools
- Search
- Minimax

Questions

- Thank You
- Thread Pools
- Search
- Minimax
- Dr. Arvin Agah

Questions

- **Thank You**
- **Thread Pools**
- **Search**
- **Minimax**
- **Dr. Arvin Agah**
- **Committee**

- Thank You
- Thread Pools
- Search
- Minimax
- Dr. Arvin Agah
- Committee
- Texts
  - Genetic Algorithms in Search Optimization, and Machine Learning
  - Numerical Recipes in C: The Art of Scientific Computing