How to Win the Game

Opponent's Strategy	**Best Strategy Choice(s)**
Binomial	Geometric, Pascal
Equilikely	Geometric, Binomial, Pascal
Geometric	(Geometric)
Pascal	(Pascal)

Different-Distribution Trials

- Higher reward makes it easier to distinguish the better strategy.

- Increasing the constant reward from 0 to 5 makes the game last longer. Increasing the percentage extends the game very slightly.

- Increasing the constant reward from 0 to 5 (in some cases) changes the “winning” strategy and (always) increases the difference between the two.

- Raising the percentage from 25 to 50 increases the magnitude of the better strategy.

- Increasing the rewards does not make as big of an impact on the number of fights or natural deaths as in the same-distribution trials.

Same-Distribution Trials

- Higher reward makes it harder to distinguish the "winner."

- Higher reward makes the game last longer.

- Increasing the constant reward from 0 to 5 causes the number of fights to increase and the number of natural deaths to decrease.

- In the case with both strategies as uniformly distributed strength and a constant reward of 0, there more fights occur than natural deaths. (The other same-distribution trials with constant reward of 0 consistently show more natural deaths than fights.)

Results

Generalities:

- Most interesting behavior occurs during the first 100 moves. Good performance in the beginning of the simulation dictates the "winning" strategy.

- Higher reward results in more fights.

- Higher reward decreases the number of natural deaths due to starvation.

Extensions to the Model

- Real-valued attributes and states
- More teams playing at one time
- "Rebirth" of defeated opponents
- Penalty for fighting
- Decreased desire to fight as agent gets poorer (weaker)
- Implement a more realistic fight for equally strong agents
- Integration of moves

Flaws in the Model

- Fight Status
- Fighting occurs when the reward is 0
- Constant reward for movement

Reward Experimentation

Two basic types of rewards were used:

- **Constant** (0 and 5)

The constant 0 is the "control," used to simulate a lack of incentive for fighting. Unfortunately, this lack of reward did not inhibit fighting.

Constants are used to give the agent an incentive to move. This reward type is added to the moving agent's strength regardless of whether it fights.

- **Percentage of Occupant's Strength** (25% and 50%)

Percentages are used to give incentive to fight the opponent. This type of reward is added to the moving agent's strength only if it fights.

The Agent Moves

- The agent selects the last position seen with maximum reward and goes there, regardless of vulnerability and distance.
- If the site was occupied, the former occupant is “killed” and permanently removed from play. Fight status for the new site is turned “on,” signifying a fight has occurred and disabling another fight until both teams have finished their turn.
- The reward is added to the agent’s strength. The metabolism rate is subtracted from the agent’s strength.

Movement Rules

- Look out as far as vision permits in the four principal lattice directions.

- Throw out all cells occupied by agents of the same team.

- Throw out all cells occupied by stronger opponents.

- The reward of each remaining cell is computed.

- If no "safe" move exists, the agent does not move. The metabolism rate is subtracted from the agent's strength.

Strength Experimentation

Four different strategies are available:

- $Equilikely(1, 20)$
- $Geometric(0.75)$
- $Pascal(5, 0.25)$
- $Binomial(20, 0.3)$

Ten combinations of two strategies were used to compare effects of strength distribution on the number of fights and natural deaths in the simulated model.

Individual Agents

Each agent is randomly placed in the landscape, where no unoccupied cell has higher probability of being chosen than another.

The **metabolism** rate for each agent is created as an $Equilikely(1, 4)$ random variate.

The **vision** attribute for each agent is created as an $Equilikely(1, 6)$ random variate.

Rather than use an **initial strength** of 10 for each agent, strategy inputs allow for a relatively even or skewed distribution of strength over agents in a team.

Building the Agent Teams

500 agents are created in each team and are labeled with their team number.

The agents have a "team manager" which orchestrates their movement and stores team information.

Strength distribution strategies are given on input to the simulation. The teams can have the same or different strategies.

The Agents

1000 agents interact in the sugarscape environment.

- Genetic Attributes
    - Metabolism
    - Vision
- States
    - Cell Location
    - Resource Strength (Wealth)
- Team Number

Building the Landscape

For each (x, y):

- initial resource level is set to the capacity $c(x, y)$;
- growback rate $\alpha(x, y)$ is set to 1;
- occupancy states are set to "empty."

- States
  - Resource Level
  - Occupancy
  - Team Number of Occupant
  - Strength of Occupant
  - Fight Status

The Landscape

- Two-dimensional grid of cells 100 x 100
- Attributes
  - Capacity

$$c(x, y) = \lfloor C_1(x, y) + C_2(x, y) + 0.5 \rfloor$$

$$C_1(x, y) = 4exp\left(-\left(\frac{x - X/4}{\sigma_x}\right)^2 - \left(\frac{y - Y/4}{\sigma_y}\right)^2\right)$$

$$C_2(x, y) = 4exp\left(-\left(\frac{x - 3X/4}{\sigma_x}\right)^2 - \left(\frac{y - 3Y/4}{\sigma_y}\right)^2\right)$$

$$\sigma_x = X/3 \text{ and } \sigma_y = Y/3$$

  - Resource Grow-Back Rate

Research Focus

- Incorporate combat into the quantized model used in class.

- Simulate the sugarscape as a “game” using several different strategies for strength distribution and fighting rewards.

- Find the winning strategies.

Outline

- Model
    - Experiments
    - Flaws
    - Extensions
- Results and Conclusion

Growing Artificial Societies: Combat on the Sugarscape

Rachel Seigfried