Generalized game of life

2003. 05. 22

20025059 YongDuk Kim
0. Content

- Introduction
- Original game of life
- Create generalized game of life Using GA
- Result
- Conclusion
1. Introduction

- **Goal**
  - Implement original game of life system
  - Make this system more general

- This is achieved by using Genetic Algorithm.
2. Original Game of Life

- **Rule**
  - If the number of surrounding cells is less than 2 or greater than 3, the current cell dies.
  - If the number of living cells is exactly 2, or if the number of living cells is 3, maintain status.
  - If the current cell is dead, but has three living cells surrounding it, it will come to life.

- **Implementation**
  - 40 X 40 cells
  - Each bound is connected
  - Using C++
3. Create Generalized game of life using GA

- GA is used to make generalized game of life.

Procedure GA

Begin

\[ t \leftarrow 0 \]

Initialize P(t)

Evaluate P(t)

While (not termination-condition) do

Begin

\[ t \leftarrow t+1 \]

Select P(t) from P(t-1)

Alter P(t)

Evaluate P(t)

End

end
3. Create Generalized game of life using GA

- **Components of GA**
  - A genetic representation
  - A way to create an initial population
  - Selection method
  - An evaluation function
  - Genetic operators
3. Create Generalized game of life using GA

- A genetic representation

- A way to create initial population
  - Randomly
  - Set initial state of the grid

- Current cell is alive and has “8” neighbors.
3. Create Generalized game of life using GA

- **Selection method**
  - Roulette wheel method
    - Probability of selection
      \[ p_i = \frac{\text{eval}(v_i)}{\sum_{i=1}^{\text{pop size}} \text{eval}(v_i)} \]

- **Evaluation function**
  - Best solution has maximum fitness.
  - My interest world
    - Active
    - Not crowded
3. Create Generalized game of life using GA

How?

- **Active**
  - Look at 2 generations in the Game of life, then check how many cells are different.
  - Let’s call this fitness1.

- **Not crowded**
  - Find a deviation from my desired number of cells.
  - Then invert this deviation, because we want to maximize fitness.
  - Let’s call this fitness2.