PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL

ESCOLA POLITÉCNICA

ENGENHARIA DE SOFTWARE

DISCIPLINA DE INTELIGÊNCIA ARTIFICIAL – PROF. SÍLVIA MARIA W. MORAES

Trabalho 1 da Disciplina – Algoritmo Genético

Gabriel Ferreira Kurtz

gabriel.kurtz@acad.pucrs.br

1. Introdução

A proposta do primeiro trabalho é implementar um algoritmo genético capaz de localizar a saída em uma matriz quadrada de tamanho n, onde a entrada é sempre na posição (0,0) e a saída pode ser em qualquer lugar, definida no arquivo de entrada e desconhecida pelo algoritmo. Há espaços na matriz que podem ser caminhados e outros são muros.

2. Execução e Configurações

O trabalho foi realizado em Python. A execução recebe como argumento o nome do arquivo de entrada. Exemplo:

$ python caminhar.py labirinto1_10.txt

As demais configurações devem ser realizadas no código alterando as constantes globais, sendo as configurações do programa:

- UNICODE: ativa visual com setas Unicode quando possível (ex: VSCode);
- LOG_COMPLETO: alterna entre print completo ou simplificado;
- GERACOES_ENTRE_LOG_SIMPLES: quantas gerações entre cada print no log simples.

Enquanto outras variáveis globais ajustam o algoritmo genético:

- GERACOES: total de gerações a simular;
- FATOR_MAXIMO_PASSOS: ajuda a definir número de passos em um Cromossomo por geração quando não encontra a saída, em relação ao tamanho da matriz. Ex: matriz de largura 10, fator 2 => máximo de 100 passos (10^2);
- ELITISMO: quantos Cromossomos são selecionados por elitismo para compor a população intermediária;
- TORNEIO: número de torneios por geração (cada torneio gera 2 Cromossomos para a população intermediária);
- MUTACAO_MINIMA: menor taxa de mutação que o algoritmo pode adotar;
- MUTACAO_MAXIMA: maior taxa de mutação possível.

3. Cromossomo

O algoritmo implementa a classe Cromossomo. Sua principal informação (o cromossomo de fato) é o atributo Cromossomo.direcoes, uma matriz de duas dimensões, do mesmo tamanho do labirinto, onde cada casa define uma direção de movimento, ou seja, guarda a ultima direção que o algoritmo andou quando passou naquela casa de modo a formar um caminho.

A cada geração procura-se aprimorar estas direções para formar um caminho mais eficiente, primeiramente encontrando a saída e depois diminuindo o número de passos para chegar até a saída (encurtando o caminho).

4. Aptidão

A fórmula de aptidão antes de encontrar a saída é a distância euclidiana entre a entrada (0,0) e onde o Cromossomo parou, sendo que quanto maior melhor.

Após encontrar a saída, a heurística inicia com um valor fixo maior que todas as possibilidades da distância euclidiana, de modo que todos os Cromossomos que encontrarem a saída são mais aptos que os que não encontraram. A ordenação entre os que já encontraram é por quem chegou à saída em menos passos.

5. População Intermediária

É formada com alguns Cromossomos de elitismo (os X mais aptos) e outros de Torneio, em número definido conforme explicado nas configurações.

Os filhos do torneio são cruzados com uma máscara aleatória, ou seja, para cada posição da matriz um dos filhos recebe o gene do pai e o outro da mãe, onde quem recebe cada gene de quem é definido com 50% de chance.

6. Mutação

Optei por realizar a mutação de uma forma um pouco diferente, para ajudar a exploração do labirinto. Sempre que o programa anda no labirinto, há uma chance do próximo movimento ser aleatório (desde que válido). Assim ele ajuda a encontrar a saída no início e também a melhorar depois.

A mutação inicial é de 10% e aumenta em 5% a cada geração enquanto nenhum Cromossomo acha a saída, até o máximo configurado. Nas gerações em que algum cromossomo achou a saída, a mutação reduz imediatamente ao mínimo configurado, para manter um bom caminho ao longo das gerações mas ainda ser capaz de inovar um pouco.

7. Simulação e Melhor Cromossomo

A simulação ocorre pelo número de gerações que foi definido, sempre seguindo a ordem do algoritmo genético. Na exibição simples, é mostrado o melhor Cromossomo global e o melhor atual a cada X gerações (configuração). No completo, são exibidos todos cromossomos em todas gerações.

O cromossomo mais apto é gravado globalmente e exibido no final. Procurei incluir o mínimo de fatores não-aleatórios que pudessem facilitar a solução, deixando a solução ser proveniente do algoritmo genético ao máximo.