

T1 - Algoritmo Genético

Daniela Amaral e Vinicius Lima

A nossa Resolução de Labirinto de 12 por 12 utilizando Algoritmo Genético foi desenvolvida em Java 8. Operadores genéticos utilizados:

- Torneio;
- Cruzamento uniponto com ponto de corte randômico;
- Para a mutação, conforme o percentual de mutação parametrizado, o método recebe o trajeto do cromossomo, isto é, a sequência de direções, escolhe um índice aleatório deste trajeto, e altera o seu valor para uma direção, também aleatória;

Na função de aptidão: Como neste problema a saída do labirinto é conhecida. Nós utilizamos a distância de manhattan entre a última posição atingida pelo cromossomo mais a soma das penalidades que ele sofreu enquanto percorria o labirinto. As nossas penalidades foram: 20 pontos por bater na parede. 20 pontos para repetir uma posição e 50 pontos por ir para uma posição inválida, isto é, sair do labirinto.

Alguns pontos relevantes sobre a nossa implementação:

- Na passagem de uma geração para a próxima, antes de ser feito o cruzamento, a nossa população é ordenada pelo score de aptidão e nós selecionamos a metade mais apta da população. Esta metade é duplicada e passada para que seja feito o cruzamento. Então antes de fazermos o cruzamento, nós descartamos a metade menos apta da população.
- O ponto de parada do nosso algoritmo é encontrar a solução ótima, isto é, encontrar a saída do labirinto. Para garantir que isto ocorra, nós implementamos um mecanismo que reseta completamente a população a cada 150 mil gerações, pois entendemos que neste número de gerações, todos os cromossomos da população já estão iguais e dificilmente vai encontrar a saída. Pois em nenhum momento nós controlamos na nossa solução, a semelhança entre os cromossomos. Tanto o torneio quanto o cruzamento podem ocorrer entre 2 cromossomos iguais. Mas mantendo a taxa de mutação alta, ainda conseguimos garantir que o algoritmo funcione e encontre a saída.
- Sobre a movimentação no labirinto, nós implementamos de forma ao agente somente se movimentar SE a próxima posição for válida, isto é, não ser nem parede, nem fora do labirinto, para estes dois casos, nós apenas contamos a penalidade e mantemos o agente na mesma posição.

Configurações utilizadas:

- Tamanho da população: 21
- Quantidade de movimentos: 45
- Chance de mutação: 85%
- Prints a cada 1000 gerações

Todas as configurações acima mais o nome do arquivo de entrada são passados como argumento.

O arquivo de saída dos resultados é "resultados.txt" localizado na raiz do projeto, o arquivo de entrada do labirinto precisa estar dentro da pasta "files", e possui estrutura igual ao arquivo "lab.txt".

Para rodar o algoritmo genético e em sequência o A*, rodar a classe Main do arquivo Main.

Link do vídeo de apresentação, só é possível acessar o vídeo pelo link: <https://youtu.be/EAZ0SH4rStQ>