Resolução de Labirintos com o uso de Algoritmo Genético

Raul Carneiro , Emanuel Felipe

01/04/2020

1 Introdução

Labirintos são construcões, criadas com o intuito de dificultar a passagem de uma pessoa dentro da mesma. Não se sabe ao certo sua origem, mas sabe-se que o labirinto mais antigo a ser mencionado na história foi o Labirinto de Creta, originado da cultura grega, onde o mesmo foi construido no intuito de alojar o Minotauro, uma das figuras mais conhecidas da mitologia grega. Atualmente, tem-se usado labirintos para confecção de jogos, onde se tem um ponto de partida e o jogador deve encontrar a saída sem que encoste nas paredes da construção, como no famoso "LABIRINTO DO EXORCISTA", onde os jogadores devem chegar ao final de cada fase sem encostar nas paredes. Levando em conta a dificuldade de tais labirintos, resolvemos utilizar o Algoritmo Genético para verificar o minimo de gerações possiveis para encontrar a saida do labirinto, verificando o tempo levado para chegar a saída do mesmo e analisando o número de fracassos e sucessos ao longo do experimento.

2 Objetivo Geral

Como o algoritmo genético é um modelo computacional baseado na evolução, onde a solução encontrada é uma potencial solução (podendo ou não ser a melhor forma de achar a solução para o problema especifico), podemos encontrar, nesse caso, um potencial caminho para chegar ao final do labirinto, onde o algoritmo irá ultilizar de suas diversas gerações para achar tal caminho. Entre essas gerações, o algoritmo pegará o individuo que teve o melhor percurso, ou seja, foi o mais próximo de chegar ao final, e cruzará com os outros indviduos. Com isso, ao longo do processo, as próximas gerações "aprendem" como chegar no final do percurso. Um algoritimo que faria um melhor trabalho na busca pelo final do labirinto é o algoritmo

A*,já que este é um algoritmo de busca, e, diferente do algoritmo descrito anteriormente, é capaz de encontrar a melhor solução para o problema proposto (no caso o final do labirinto). O algoritmo A*, tambem chamado de algoritmo estrela, deve partir de um nodo inicial e chegar ate o nodo final, e para fazer isso ele pega todos os seus vizinhos do nodo, aonde ele se encontra, e aplica uma função heurística. Essa função retorna um número que indica a distância até o nodo final, e com isso, o vizinho que tiver a menor distância se torna o nodo atual. Ele continua usando este método até encontrar o melhor caminho para o nodo final. Ao longo do artigo, nós também faremos uma comparação entre os dois algoritmos, mostrando a velocidade e a precisão para encontrar o fim, lembrando que o algoritmo genético não é um algoritmo de busca, e desde agora já se torna óbvio o fato dele ser mais devagar para encontrar o fim do labirinto, no entanto, a forma como o esse algoritmo se comporta no decorrer desse desafio é algo interessante para reforçar o aprendizado.

3 Objetivo Específico

Como mencionado no tópico anterior, nosso objetivo geral para esse problema proposta seria encontrar uma geração que conseguiu percorrer o labirinto por completo, ou seja, chegar até a saída. Para que essa solução seja encontrada, devemos implementar no código dois fatores: a criação de um labirinto pré-definido ou aleatório, e o cruzamento entre as gerações até que o labirinto seja concluído. Para resolver o labirinto iremos usar tanto o algoritmo genético, quanto o algoritmo estrela. Com o primeiro algoritmo, iremos gerar vários indviduos, ver como irão se comportar e se acharam o caminho até o final. Caso não encontrem, iremos cruzar o indivíduo que melhor se saiu com os individuos da geração anterior até que alguma geração encontre, com êxito, o caminho correto, já o algoritmo A*

4 Referencial Teórico

4.1 Algoritmo genético

Para a resolução dos objetivos específicos, é necessário ter conhecimento sobre o Algoritmo Genético e o A*. Nesse caso, o Algoritmo Genético apresenta certos conceitos que, durante a implementação do código, contribuem com a busca pelo resultado visionado. Entre esses conceitos, temos o Crossover, traduzido no português como "Recombinação".Na Biologia, a Recombinação é um processo genético marcado pela troca de um material genético durante a Meiose, na

qual ocorre uma divisão celular que corta os genes ao meio, gerando cromátides-irmãs. Após esse processo, ocorre o cruzamento entre duas cromátides não-irmãs, ou seja, de dois cromossomos diferentes. Esse processo é nomeado Crossing-over, na qual também é presente no processo do Algoritmo Genético. Como mencionado anteriormente, o Crossing-over é usado para cruzar duas cromátides não-irmãs, que geram um novo cromossomo. No Algoritmo Genético, esse processo apresenta a mesma ideia, porém, ele cruza dois indivíduos, que apresentaram resultados sobre certa situação, gerando em um novo indivíduo que apresenta as mesmas características dos dois indivíduos cruzados, e assim, o ciclo se repete. Outro conceito presente no AG seria a "Seleção". Na Biologia, o conceito de Seleçao segue a ideia de um processo visionado por um biólogo chamado Charles Darwin, nomeada de Seleção Natural. A Seleção Natural trata-se de uma teoria composta na ideia de que certos indivíduos apresentam novas características baseadas no ambiente em que vivem, ou seja, com o intuito de sobreviver e reproduzir em um certo local, esses indivíduos passam por mutações, também conhecidas como evoluções, que alteram certo aspecto fisico do corpo. Nesse caso, como explicado no livro " A Origem das Espécies", escrita pelo próprio biólogo, diversas aves de uma ilha apresentavam bicos de formatos diferentes, cada um com uma finalidade. Por exemplo, aves que apresentavam bicos pequenos eram capazes de coletar alimentos de lugares de pouco acesso, enquanto aves com bicos grandes tinham uma facilidade maior em coletar e quebrar alimentos que apresentavam mais dureza na casca. Com isso, como explicado anteriormente, esse fenômeno da Seleção mostra-se presente em AG durante o começo do processo, especificamente durante a escolha dos indivíduos que serão cruzados para gerar um novo indivíduo. Em comparação com o Algoritmo Genético, o A* é considerado a melhor escolha quando se fala de labirintos na programação, portanto, é de grande importânâcia entender como esse algoritmo funciona.

4.2 Algoritmo estrela

Primeiramente, o A* não apresenta tantas relações com a própria Biologia, no entanto, existem dois conceitos de grande importância no A*: a Busca em Largura(Breadth First Search) e o Algoritmo de Dijkistra. A Busca em Largura trata-se de um algoritmo de busca, com o uso de grafos, na qual o algoritmo atravessa cada grafo com o intuito de encontrar o alvo desejado. Os grafos são estruturados como uma árvore, apresentando uma raiz e um topo. Com isso, a busca se inicia a partir do vértice raíz, explorando os outros vértéices ao longo do percurso, até que, no fim, o alvo é encontrado. Já o Algortimo de Dijikistra tem como principal função a busca por um caminho mais curto dentro de um percurso. Nesse caso, um grafo é criado contendo

todos os pontos de interesse, assim como um caminho entre todas elas. O algoritmo calcula o melhor caminho possível, respeitando os pontos específicados pelo código do programa. No fim, o uso do A* apresenta uma facilidade maior ao encontrar a solução encontrada dentro de um labirinto, porém, a criação de um indivíduo que apresenta a melhor solução durante a travessia apresenta o Algoritmo Genético como seu destaque.

4.3 Linguagem C

Para o desenvolvimento do nosso algoritmo genético, faremos o uso da linguagem C. Essa linguagem teve sua primeira versão criada em 1972, por Dennis Ritchie, como um dos softwares a serem distribuidos com o sistema operacional UNIX do computador PDP-11, porém, essa linguagem mostrou-se presente uma década antes. A linguagem teve seu surgimento iniciado em 1960 com a linguagem ALGOL 60. Nesse caso, o ALGOL era uma linguagem de alto nível, que permitia que o programador trabalhasse "longe da máquina" sem se preocupar com os aspectos de cada comando ou dado que era armazenado ou processado, sendo que ela foi inventada para substituir o FORTRAN. Em 1970, Ken Thompson implementou um compilador que seria uma versão "lite" da linguagem CPL, que teve o nome de linguagem B, no entando, tanto a linguagem B quanto a CPL se mostraram limitadas, sendo usadas apenas para resolver algumas classes de problemas. Com isso,devido à esses e outros problemas, Dennis Ritchie foi encarregado de projetar uma nova linguagem, que seria a sucessora da linguagem B: a linguagem C. A linguagem projetada por Ritchie apresentava como foco o mantimento do contato com a máquina e, também, o abrimento de portas para o desenvolvimentos de programas em áreas diversas, como por exemplo, nas áreas de engenharia, comercial e ciêntifica. Por muito tempo, a sintaxe utilizada na liguagem C foi a mesma implementada no UNIX versão 5.0, apresentando como principal documento o livro "The C Programming Language" que é considerada a bíblia dessa linguagem. Com esse conhecimento em mente, a equipe tinha duas opções para a construção do algoritmo, sendo elas a linguagem C, descrita acima, e a linguagem Python. Mesmo com as diversas vantagens que a linguagem Python proporciona, a equipe optou por desenvolver em C, pois ao longo do curso, essa linguagem foi bastante utilizada, contribuindo com o desenvolvimento do algortimo, devido a familiarização da mesma.

5 Bibliografia

5.1 encurtador.com.br/oBNU7