

Universidade Federal de Uberlândia
Faculdade de Computação

Problema do Robô Coletor de Latas

Lorena da Silva Elias, Victor Hugo Eustáquio Lopes

Computação Evolutiva

Prof. Paulo Henrique Ribeiro Gabriel

Resumo

Neste relatório será apresentada a implementação de um algoritmo evolutivo que fornece a menor sequência de movimentos que o robô deve executar para coletar todas as latas disponibilizadas na malha. Avaliamos o algoritmo implementado com três diferentes tamanhos de malha: 10×10 , 20×20 e 50×50 . Em cada execução, foi atribuído um conjunto de latas aleatoriamente, considerando que 20% das células são ocupadas por latas. Os resultados e considerações são apresentados ao final.

Introdução

Inicialmente, o robô está na posição (0,0) da malha. O problema é resolvido ao encontrar o menor número de passos que o robô deve fazer para coletar todas as latas disponíveis na malha se deslocando apenas uma célula por vez no eixo x ou no eixo y.

Método

O algoritmo genético implementado possui estrutura geracional. A técnica de reprodução implementada foi a de troca total da população a cada geração.

Olhando para a posição atual, temos as seguintes possibilidades: ir pra cima, direita, esquerda, baixo, ficar na mesma posição. Cada uma dessas posições pode ser que seja: uma parede, tenha uma lata ou não tenha uma lata, totalizando 243 de cenários possíveis. Sendo assim, os cromossomos foram modelados como uma lista de 243 genes. Cada gene é um número entre 0 e 6, que representa uma ação de movimentação (0 = cima, 1 = direita, 2 = esquerda, 3 = baixo, 4 = aleatório, 5 = ficar parado e 6 = pegar lata). Na população inicial, esses números são preenchidos aleatoriamente. Para todas as outras escolhas aleatórias é usado um gerador de números pseudo-aleatórios.

```
[ 0, 6, 4, 1, 4, 6, 6, 1, 3, 6, 0, 3, 6, 4, 5, 5, 3, 1, 6, 1, 2, 1, 3, 6, 6, 0, 4, 6, 4, 4, 0, 0, 2, 0, 4, 6,
6, 3, 0, 1, 3, 5, 4, 1, 5, 2, 0, 6, 3, 5, 5, 4, 2, 6, 1, 1, 5, 2, 1, 4, 0, 1, 2, 4, 5, 5, 0, 4, 2, 2, 2, 2,
6, 0, 1, 6, 4, 0, 2, 0, 4, 2, 5, 5, 0, 5, 6, 3, 1, 2, 1, 0, 6, 4, 2, 0, 2, 6, 5, 1, 5, 0, 4, 3, 3, 0, 4, 6,
0, 4, 0, 5, 5, 5, 3, 2, 6, 4, 0, 1, 6, 1, 1, 3, 2, 1, 2, 2, 0, 2, 1, 1, 1, 0, 5, 6, 3, 1, 1, 1, 3, 5, 5, 5,
0, 0, 5, 5, 0, 5, 6, 4, 0, 1, 2, 5, 3, 4, 5, 0, 0, 1, 6, 0, 6, 6, 6, 6, 4, 1, 3, 3, 3, 4, 3, 5, 3, 0, 6, 0,
5, 3, 4, 4, 0, 6, 4, 2, 4, 3, 2, 5, 5, 1, 3, 0, 4, 5, 5, 5, 5, 2, 1, 1, 2, 0, 5, 5, 1, 3, 3, 2, 0, 2, 5, 6,
2, 1, 5, 5, 3, 1, 3, 1, 5, 0, 4, 6, 6, 5, 0, 1, 6, 2, 2, 3, 0, 3, 1, 4, 1, 6, 1 ]
```

Figura 1. Exemplo de representação do cromossomo

O fitness é determinado pelo quão bem o robô se sai em 50 sessões de limpeza diferentes. Uma sessão de limpeza consiste em colocar o robô no início (0, 0) e jogar as latas ao acaso na malha. O robô anda 80 passos em cada sessão. O fitness resultante de cada sessão é o número de pontos de recompensa que o robô acumula menos o total de penalidades que ele recebe. O fitness total é sua pontuação média nas 50 sessões de limpeza diferentes, cada uma com uma configuração diferente de latas.

A escolha dos pais que gerarão novos filhos para a próxima geração considera o fitness dos indivíduos. Quanto maior o fitness, maior a chance de ele ser escolhida como pai.

Na geração de novos filhos, o operador de crossover é aplicado aos cromossomos dos pais cortando-os ao meio e recombinaando cada metade com a metade do outro pai, gerando assim dois filhos. Na geração desse novo filho, o cromossomo deste passa por mutação com 5% de probabilidade de modificação em cada um dos seus genes.

Os parâmetros de entrada do algoritmo, foram estabelecidos como mostra a tabela a seguir.

POPULATION_SIZE	100
NUM_GENERATIONS	500
STEPS	80
CLEANING_SESSIONS	50
MATRIX_LEN	10, 20, 30
PERCENTAGE_CANS	0.2
MUTATION	0.05
NUM_GENES	243
REWARD	10
PICKUP_PENALTY	1
WALL_PENALTY	2

Pseudo-código

1. Inicializar os parâmetros
2. Inicializar a população
3. Para cada geração
 - a. Calcular o fitness de cada indivíduo.
 - b. Encontrar o indivíduo com melhor fitness (gBest)
 - c. Gerar os filhos para a nova população da próxima geração
 - i. Aplicar o operador de crossover
 - ii. Aplicar o operador de mutação
4. Repetir a partir de 3. enquanto não exceder o número de gerações.

Avaliação

Para avaliar o algoritmo genético implementado, cada tamanho de malha: 10x10, 20x20 e 30x30 foi executado dez vezes. De cada execução foi guardado a melhor sequência de movimentos, a pior e a média de acordo com o resultado da função objetivo. Também anotamos o número de latas, já que a função objetivo nos dá somente um panorama geral para avaliação levando em consideração todo o caminho do robô.

Resultados

A tabela abaixo mostra o fitness resultante de 30 execuções para cada tamanho de malha.

	10X10	20X20	30X30
Melhor	2.88	3.6	5.4
Pior	1.23	1.12	1.8
Média	1.436	1.345	3.566
Tempo de Execução	1571s	1628s	1452s

Referências

MITCHELL, M. An Introduction to Genetic Algorithms. Mit Press. Massachusetts, 1997.