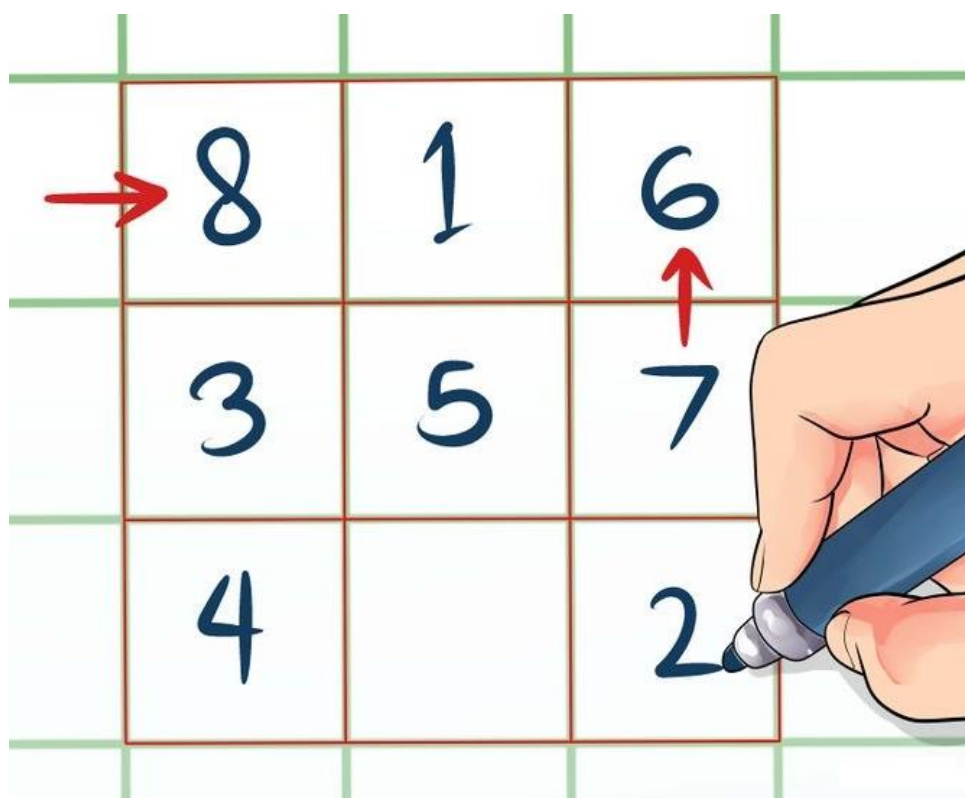


Relatório

Quadrado Mágico

2º Trabalho Prático



Grupo 05:

Alexandra Brito nº50075

Cyrill Brito nº52875

Índice

Introdução.....	3
Algoritmo evolutivo	4
Avaliação	4
Seleção	4
Crossover	5
Mutação	6
Unicidade na população.....	6
Reset	7
Otimização	7
Conclusão	8
Referencias.....	9

Introdução

Este relatório tem como objetivo principal esclarecer e informar todos os métodos e técnicas utilizadas para resolver o 2º trabalho pratico da disciplina de inteligência artificial, “*quadrado magico*”.

O quadrado mágico é um jogo que consiste numa tabela quadrada de números em progressão aritmética em que a soma de cada coluna, de cada linha e das duas diagonais são iguais.

2	7	6	→15
9	5	1	→15
4	3	8	→15
↙15	↓15	↓15	↘15

Figura 1 - Exemplo de um quadrado magico resolvido

Para descobrir qual o valor da soma de cada linha, coluna, e diagonal têm quando o quadrado está resolvido é utilizada a formula: $n * (n^2 + 1) / 2$, (valor target) onde n é o numero de linhas/colunas no quadrado.

Algoritmo evolutivo

No início o nosso algoritmo cria uma população aleatória de cubos mágicos e esta população é avaliada. Todas as gerações seguintes serão criadas a partir da população da geração anterior. Para a criação das novas gerações começa-se por selecionar dois cubos pais e com o crossover cria-se um novo cubo, o cubo filho. Após o crossover o cubo tem uma baixa probabilidade de sofrer uma mutação.

Estes processos (seleção, crossover e mutação) são repetidos até obtermos uma população do tamanho desejado.

Além destes procedimentos habituais para os algoritmos genéticos, foram aplicados os seguintes regras:

- A população de cada geração não permite que existam cubos com a mesma configuração.
- É verificado se a população está a evoluir, se esta não o estiver o algoritmo destrói a população e cria uma nova aleatória.

Avaliação

Para conseguir avaliar cada cubo mágico criamos uma função que calcula o custo de cada cubo. O custo representa o quanto longe está o cubo de uma solução, ou seja, quando o custo é zero o cubo está correto, e quanto maior o valor do custo mais longe de uma solução está. Este custo é calculado da seguinte forma:

1. Calcular o valor target: objetivo para a soma de cada linha, coluna ou diagonal de um cubo resolvido.
2. Calcular as somas de cada linha, coluna e diagonal.
3. Calcular o erro de cada linha, coluna e diagonal: fazemos a diferença absoluta do valor target e os valores calculados anteriormente, no ponto 2.
4. O custo é a soma de todos os erros calculados, no ponto 3.

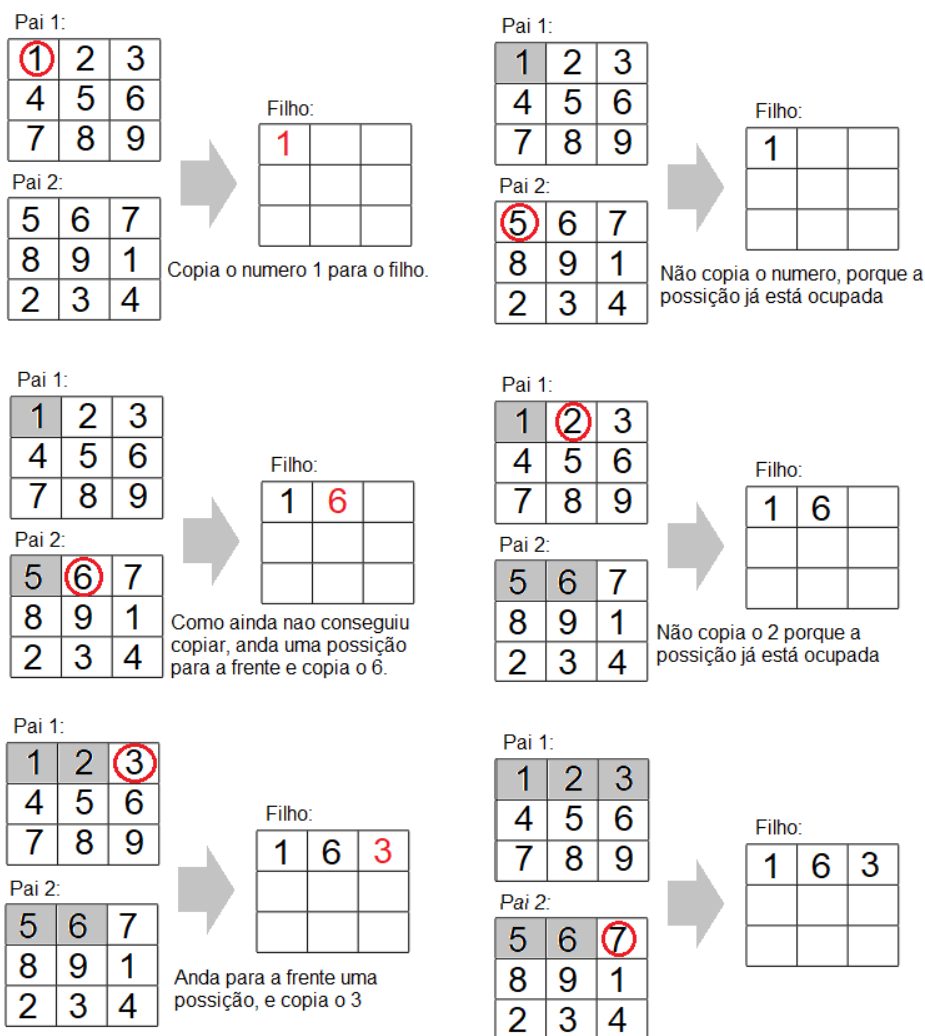
Seleção

A seleção dos pais que utilizamos é bastante simples. Primeiro começamos por ordenar os cubos da população pelos seus custos, obtendo então os melhores cubos no início da lista. De seguida é escolhido um cubo aleatoriamente dos 10% melhores cubos.

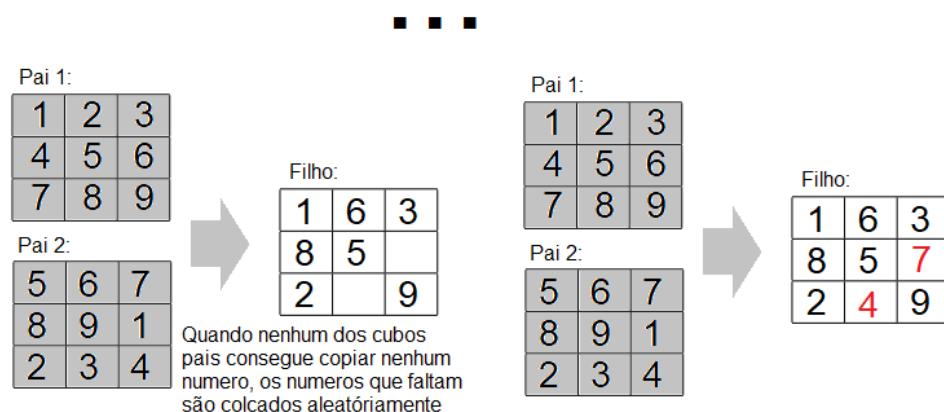
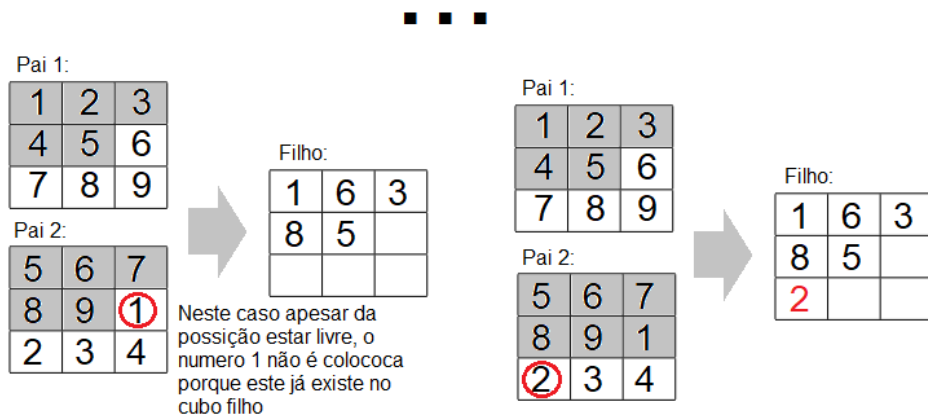
Crossover

O crossover tem como objetivo de a partir de dois cubos criar um cubo filho, onde se encontram semelhanças dos dois cubos pais. Tendo este objetivo em mente criamos o nosso próprio crossover que se apropria ao problema. O crossover que criamos é efetuado a partir dos seguintes princípios:

- Copiar um número do cubo pai, alternado entre os dois pais, para o cubo filho.
- Quando se copia um numero de um cubo pai para o cubo filho, a posição deste numero permanece a mesma.
- Caso o numero que se pretende copiar já exista no cubo filho ou a posição já esteja ocupada, tentasse copiar o próximo número do mesmo cubo pai.
- Caso já não existam mais números no cubo pai que possam ser copiados passasse para o outro pai.
- Quando nenhum dos cubos pais tenham números que possam ser copiados, os números que faltam no cubo filho são colocados aleatoriamente.



■ ■ ■



Mutação

As mutações permitem adicionar diversidade na população. Cada cubo, após ser criado pelo crossover, tem uma probabilidade de 5% de ser sujeito a uma mutação. Esta mutação consiste na troca de dois números aleatórios do cubo.

Unicidade na população

Devido ao crossover utilizado e à seleção dos melhores cubos, é normal que a população, ao longo das gerações, fique com cubos semelhantes ou iguais. Isto gera um problema em que a população, devido à pouca diversidade, não consegue evoluir. Para resolver este problema sempre que existe um cubo repetido é-lhe aplicada uma mutação, criando assim um cubo semelhante, mas com uma pequena alteração.

Reset

Apesar de fazermos mutações e de garantirmos a unicidade, existem casos em que a população encontra-se num estado quase estático, em que os cubos não conseguem evoluir. Para que a população não fique neste estado, foi criado um contador que conta o numero de gerações em que a população não evoluiu. Se este contador chegar a 500, é feito um reset á população, ou seja, criasse uma nova população totalmente aleatória.

Otimização

Após toda a estrutura do algoritmo estar feita corremos alguns testes em que variávamos os seguintes valores, com o objetivo de obter melhores resultados:

- Tamanho da população (melhor valor: 900)
- Percentagem de mutação (melhor valor: 0.05)
- Percentagem de pais disponíveis para o crossover (melhor valor: 0.10)
- Numero de gerações ate fazer reset (melhor valor: 500)

Conclusão

Este trabalho foi uma grande oportunidade de aprendizagem, quando começamos a resolver o problema pouco ou nada sabíamos sobre algoritmos genéticos. Após batalharmos e investigarmos sobre este tipo de algoritmos, com objetivo de resolver este problema, obtemos um conhecimento amplo sobre o assunto.

Em relação ao concurso realizado na aula, foi uma grande motivação para não deixar o programa simplesmente a funcionar, mas sim tentar sempre melhorar e otimizar.

Referencias

1. https://en.wikipedia.org/wiki/Magic_square, visitado pela ultima vez a 26 de abril de 2017.
2. <http://pt.wikihow.com/Resolver-um-Quadrado-M%C3%A1gico>, visitado pela ultima vez a 26 de abril de 2017.