**ALGORITMOS GENÉTICOS**

**GUSTAVO DI DOMÊNICO**

O trabalho consiste em resolver o problema de um viajante (denominado Fanático) que deseja assistir a maior quantidade de jogos possível através da implementação de um algoritmo genético. Utilizando a tabela de jogos da copa do mundo, o programa deve mostrar o melhor caminho entre as cidades sedes que possibilite o menor investimento do viajante, ou seja, a menor rota possível entre todas as cidades.

A solução foi desenvolvida o framework GAF (http://johnnewcombe.net/gaf) para a plataforma .NET que consiste em um núcleo de um algoritmo genético onde pode-se definer as características de execução:

1. **Composição do cromossomo:** cada cromossomo é representado por um número de 1 a 4 que representa qual dos jogos o viajante irá assistir.
2. **População inicial:** fora gerado randomicamente uma população de 1000 indivíduos.
3. **Elitismo:** apenas o melhor indivíduo avança para a próxima etapa.
4. **Mutação:** o parâmetro de mutação foi calibrado para dois cenários: 1% e 5%**.**
5. **Crossover:** o parâmetro de crossover foi calibrado para dois cenários: 80% e 90%.
6. **Função de fitness:** a função de fitness é inversamente proporcional a distância total que se encontra no cromossomo. Quanto maior a distância, menor será o resultado da função de fitness.

Em ambos os cenários a solução ideal, segundo o algoritmo, é uma distância de 13.866 km, com a única diferença que no primeiro cenário ela convergiu na **décima** geração em media, enquanto que no segundo cenário a conversão ocorreu por volta da **décima quinta** geração. Segundo o resultado, a melhor rota para ser visitada dentre todas as possibilidades é:

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Cidade** | **Jogo** | **Distância Acumulada** |
| **13/6** | Salvador | Espanha x Holanda | 0 |
| **14/6** | Belo Horizonte | Colômbia x Grécia | 964 |
| **15/6** | Rio de Janeiro | Argentina x Bósnia | 1398 |
| **16/6** | Curitiba | Irã x Nigéria | 2073 |
| **17/6** | Belo Horizonte | Bélgica x Argélia | 3077 |
| **18/6** | Rio de Janeiro | Espanha x Chile | 3511 |
| **19/6** | São Paulo | Uruguai x Inglaterra | 3868 |
| **20/6** | Curitiba | Honduras x Equador | 4206 |
| **21/6** | Belo Horizonte | Argentina x Irã | 5210 |
| **22/6** | Rio de Janeiro | Bélgica x Rússia | 5644 |
| **23/6** | São Paulo | Holanda x Chile | 6001 |
| **24/6** | Belo Horizonte | Costa Rica x Inglaterra | 6490 |
| **25/6** | Rio de Janeiro | Equador x França | 6924 |
| **26/6** | São Paulo | Coréia do Sul x Bélgica | 7281 |
| **28/6** | Belo Horizonte | To be determined | 7770 |
| **29/6** | Recife | 9831 |
| **30/7** | Brasília | 11488 |
| **1/7** | São Paulo | 12361 |
| **4/7** | Rio de Janeiro | 12718 |
| **5/7** | Brasília | 13866 |

É evidente na tabela anterior que estamos considerando que estamos partindo Salvador, mas podemos considerer a partida do primeiro jogo, do dia 12/6; basta adicionar a distância de São Paulo a Salvador. Finalmente o algoritmo não está considerando os últimos dias de jogos, pois apenas existirá um jogo por dia e logicamente o viajante tem condições de assistir todos eles. Novamente apenas é necessário adicionar as distâncias para as cidades dos jogos finais.

Na tabela abaixo temos o resumo dos jogos por seleção que foram vistos, bem como a seleção brasileira evidenciada. Adicionalmente podemos perceber que a seleção mais assistida foi a Bélgica, então podemos presumir que o viajante (Fanático) seja de origem belga.

|  |  |
| --- | --- |
| **Seleção** | **Jogos** |
| Brasil | 1 |
| Croácia | 1 |
| Espanha | 2 |
| Holanda | 2 |
| Colômbia | 1 |
| Grécia | 1 |
| Argentina | 2 |
| Bósnia | 1 |
| Irã | 2 |
| Nigéria | 1 |
| Bélgica | 3 |
| Argélia | 1 |
| Chile | 2 |
| Uruguai | 1 |
| Inglaterra | 2 |
| Honduras | 1 |
| Equador | 2 |
| Rússia | 1 |
| Costa Rica | 1 |
| França | 1 |
| Coréia do Sul | 1 |

Levando em conta que a melhor solução foi encontrada logo nas primeiras gerações, eu acredito que o emprego do algoritmo genético para solucionar o problema foi excelente. Se analisarmos o problema, percebemos que existe uma semelhança ao problema NP-completo do caixeiro viajante e é sabido que para obter a melhor solução para essa classe de problema, teríamos que testar todas as combinações possíveis. Outra abordagem seria utilizar um algoritmo guloso, A\* por exemplo, que juntamente com uma heurística poderia dar uma solução aproximada da melhor possível.