Trabalho de Grafos

Raul Soares

Samuel Morais

Introdução:

O trabalho desenvolvido teve como objetivo aplicar o conhecimento teórico obtido na sala de aula no desenvolvimento de um programa que resolve um problema específico e suas instâncias. O projeto foi criado em C++, e usa apenas a Biblioteca Padrão C++. O programa funciona corretamente e consegue valores próximos aos valores de referência disponibilizados no enunciado do trabalho, porém não foi adequado a todas as normas requiridas.

Formulação:

A formulação se deu primeiro ao analisar o problema e as instâncias do problema. Concluimos que se tratava de um problema TSP métrico euclidiano, e portanto selecionamos o algoritmo de aproximação Twice-Around e o algoritmo de busca local (melhoria) 2-opt para solucionar o problema. Em especial, o algoritmo 2-opt foi feito de tal forma que ele fosse estocástico, usando o motor Mersenne Twister para gerar os valores randômicos.

Essa não foi a primeira abordagem ao problema, mas visto que a abordagem usando o algoritmo de Christofides não estava dando frutos, optamos por trocar de algoritmo.

Descrição da Solução:

Conforme foi dito, utilizamos os algoritmos Twice-Around e 2-opt para solucionar o problema. De forma simples, o programa é chamado juntamente com três parâmetros, o nome do arquivo de entrada, a quantidade de tentativas que o algoritmo irá realizar tentando otimizar a solução encontrada e o nome do arquivo de saída. Em seguida inicia o cronômetro e faz a extração de dados do arquivo de entrada, que retorna os dados limpos na variável “coordenadas”. Após, a função chama o algoritmo twice-around, passando “coordenadas” como parêmetro. O algoritmo twice-around executa seus passos de gerar a AGM, construir o grafo euleriano, executar o algoritmo de Hierholzer e limpar o grafo resultante, tendo por fim o ciclo hamiltoniano.

O ciclo hamiltoniano é retornado a função main, e é impresso na tela o caminho encontrado (rota inicial) e a distância máxima inicial. Logo depois, é chamado o algoritmo 2-opt, que faz a melhoria utilizando uma semente randômica para selecionar as arestas que irão ser removidas e as que serão adicionadas. Após uma comparação, o algoritmo verifica se houve melhora, e se for positivo, realiza a troca em definitivo. Ele retorna para a função main e imprime na tela a rota otimizada e a distância máxima otimizada, além de escrever no arquivo “rota.txt” a solução encontrada. Após a execução do algoritmo, também é impresso na tela o tempo de execução do programa.

Todas as operações em que são necessárias saber a distância entre dois pontos no plano, a função calcular\_distância é chamada para executar a fôrmula de distância entre dois pontos. As distâncias máximas são processadas usando a função calcular\_distância\_máxima, que percorre o ciclo hamiltoniando registrando a maior distância entre dois pontos encontrada.

Resultados Obtidos:

Cada instância foi testada 5 vezes devido sua natureza estocástica, portanto apresentamos uma rota otimizada em um arquivo .txt separado, do mesmo nome usado para apontar qual instância estamos falando, a distância máxima inicial e a média da distância máxima otimizada. Na saída padrão sempre é impresso os valores citados, mais a rota inicial e o tempo de execução. O parâmetro de número de randomizações foi de 1000000000.

Instância 1 : N/A (Houve erro ao fazer as adaptações para usar TSP métrico geográfico)

Instância 2:

Tempo de execução(milissegundos): (74689 + 78060 + 74895 + 74801 + 74741)/5 = 75437.2

Distância inicial: 2425.94

Distância minimizada: (1592.94 + 1406.2 + 1452.34 + 1579.33 + 1579.33)5 = 1522.028

Desvio SI/SF: 0,37

Desvio SF/SO: 15,31

Instância 3:

Tempo de execução(milissegundos): (79284 + 79453 + 79420 + 79349 + 79321)/5 = 79365.4

Distância inicial: 1868.67

Distância minimizada: (1546.96 + 1546.96 +1546.96+ 1546.96 + 1546.96)5 = 1546.96

Desvio SI/SF: 17,21

Desvio SF/SO: 4,58

Instância 4:

Tempo de execução(milissegundos): (86099 + 85622 + 85866 + 85951 + 85383)/5 = 85784.20

Distância inicial: 3218.99

Distância minimizada: (1133.29 + 1206.95 + 1206.95 + 1133.29 + 1133.29)5 = 1162.754

Desvio SI/SF: 63,87

Desvio SF/SO: 2,55

Instância 5:

Tempo de execução(milissegundos): (78386 + 78170 + 78153 + 78335 + 78032)/5 = 78215.2

Distância inicial: 1004.56

Distância minimizada: ( 672.725 + 672.725 + 672.725 + 672.725 + 672.725)5 = 672.725

Desvio SI/SF: 33,03

Desvio SF/SO: 18,83

Instância 6:

Tempo de execução(milissegundos): (79969 + 79988 + 79318 + 79557 + 80072)/5 = 79780.8

Distância inicial: 1061

Distância minimizada: (552.081 + 792.654 + 503.469+ 475.626+ 552.081)5 = 575.1822

Desvio SI/SF: 45,78

Desvio SF/SO: 25,06

Instância 7:

Tempo de execução(milissegundos): (73799 + 73738 + 74330 + 73529 + 73502)/5 = 73779.6

Distância inicial: 473.768

Distância minimizada: (178.844 + 168.689 + 183.096 + 178.844 + 181.108)5 = 178.1162

Desvio SI/SF: 62,40

Desvio SF/SO: -22,95

Instância 8:

Tempo de execução(milissegundos): (72708 + 72907 +73091 + 72685 + 73201)/5 = 72918.4

Distância inicial: 961.769

Distância minimizada: ( 249.804+ 471.861 + 356.329 + 253.955 + 253.955)5 = 317.1808

Desvio SI/SF: 67,02

Desvio SF/SO: 16,13

Instância 9:

Tempo de execução(milissegundos): (70799 + 70810 + 70763 + 70751+ 70657)/5 = 70756

Distância inicial: 111.162

Distância minimizada: (39.0512 + 42.1545 + 40.1123 + 38.0132 + 33.6155)5 = 38.58934

Desvio SI/SF: 65,29

Desvio SF/SO: -34,78

Instância 10:

Tempo de execução(milissegundos): (78917 + 79618 + 79316 + 80742 + 80501)/5 = 79818.8

Distância inicial: 973.519

Distância minimizada: (250.17 + 289.606 + 274.751 + 240.972 + 289.606)5 = 269.021

Desvio SI/SF: 72,36

Desvio SF/SO: 11,90

Configuração do computador:

AMD Ryzen 5 3600 6-Core Processor 3.59 GHz

16,0 GB RAM

Asus Tuf Gaming X-570

Conclusão

Após executar e analisar o programa, percebemos que alguns casos tem um limite inferior que o algoritmo alcança e não consegue ultrapassar, mesmo que tenha uma solução ótima menor. Isso pode ser visto quando começa repetir o mesmo resultado. Também notamos que o tempo de execução varia muito pouco, independente dos tamanhos dos grafos.

Os resultados flutuaram em volta do valor de referência, e acreditamos que a diferença entra as máquinas podem ter sido um grande influenciador pela discrepância dos resultados. Concluimos então que o trabalho foi bem sucedido em grande parte de seu desenvolvimento e execução.

Bibliografia

Grafos – Goldbarg & Goldbarg

Notas de aula

Gemini