OTIMIZAÇÃO DA FUNÇÃO OBJETIVO PARA MEDIR A DISSIMILARIDADE ENTRE DADOS OBSERVADOS EM POÇOS E RESULTADOS SIMULADOS POR MODELOS ESTRATIGRAFICOS FORWARD - SFM[1]

Antonio Guilhermo François Soares De Marco Grapiglia[2], Daniel Fabian Bettú[3]

[1] Vinculado ao projeto "Calibração de modelos Forward de reservatórios com dados de poços."
[2] Acadêmico do Curso de Engenharia de Petróleo – CESFI – Bolsista PROIP – antonioguilhermo38@gmail.com
[3] Orientador, Departamento de Engenharia de Petróleo– CESFI – daniel.bettu@udesc.br

Na última etapa da pesquisa, foi desenvolvida uma metodologia que codifica sequências de fácies sedimentares em sequências de letras, que, com base em teorias relacionadas a correção ortográfica as compara, obtendo então um número que mede a dissimilaridade entre quaisquer sequências de simulação do software Dionisios e do poço real quantitativamente.

A próxima logica evolução é então desenvolver um método, em Python, que possa achar a combinação de agrupamentos que resulte na menor distância entre as duas sequências, o método de otimização estudado primeiramente é o método de Dijkstra, que tem como base a teoria de grafos. A base teórica será definida a seguir:

Um grafo é uma estrutura definida por vértices e arestas, um exemplo de fácil entendimento é a aplicação em gps, onde cada vértice seria um ponto de partida, chegada e pontos de referência; e as arestas os caminhos que os conectam. Grafos podem ser direcionados, em que somente um dos sentidos da aresta é permitido; ou não direcionados, em que ambos os sentidos são possíveis. Dijkstra é um método que encontra o menor caminho entre dois vértices em um grafo.

Além disso deve-se definir o conceito de horizontes estratigráficos, em uma sequência de fácies, ao invés de definir cada vértice como as fácies em si, cada vértice será o ponto de topo e base de cada fácies. Ou seja, para uma sequência com 5 fácies, haverá 6 horizontes, um no topo da primeira fácies, um na base da última fácies e os outros na transição entre a base e topo das fácies que se tocam.

Para aplicar Dijkstra nessa situação então deve-se primeiro definir de alguma forma a arquitetura do grafo que será trabalhado, que é representado na figura 1.

Neste caso, 5 camadas de simulação e 200 do poço, cada coluna representa um horizonte da simulação, portanto 6 colunas; cada linha representa um horizonte, portanto 201 horizontes para 200 camadas do poço.

Cada vértice representa um horizonte, as arestas o valor de dissimilaridade entre os dois trechos, então, após a definição do grafo, é aplicado Dijkstra entre o vértice inicial e o final, S e Z, respectivamente, e o menor valor de dissimilaridade é então obtido.

Além do estudo dessa função, o estudo de funções de otimização baseadas em gradiente foi feito utilizando: Gradiente Conjugado, Steepest Descent e Quasi Newton BFGS. Estes foram estudados teoricamente e aplicados a função de benchmark Himmelblau.

Esses métodos se baseiam centralmente no fato de que a direção contraria ao gradiente, o gradiente negativo, é a direção em cada ponto de uma função em que seu valor diminui mais rapidamente, chegando mais rápido a mínimos; além disso, o tamanho de cada passo deve ser

definido, em todos os métodos é utilizado Line Search com esse objetivo.

Com base nos testes, foi observado que Steepest Descent em geral depende muito do chute inicial em comparação com os outros métodos; Quasi Newton e Gradiente Conjugado tem desempenhos similares, porém tem diferenças que podem ser significativas na aplicação, Quasi Newton necessita de uma aproximação do Hessiano da função, em contraponto, Gradiente Conjugado não necessita do Hessiano nem de sua aproximação, já que utiliza uma combinação linear da memória dos gradientes já calculados e dos gradiente atuais, sem necessitar de Hessiano ou de sua aproximação. Vale ressaltar que quaisquer conclusões de performance dos métodos feitas com base em testes feitos em funções simples como Himmelblau não são conclusivas para aplicações especificas como esta, para que conclusões definitivas sejam chegadas é preciso testá-los na aplicação em si, portanto, os testes feitos aqui são ilustrativos da capacidade de cada função, além de oferecer um modelo para a final aplicação, em que a função deverá ser definida e o código de cada método modificado para acomodar as restrições que devem ser feitas.

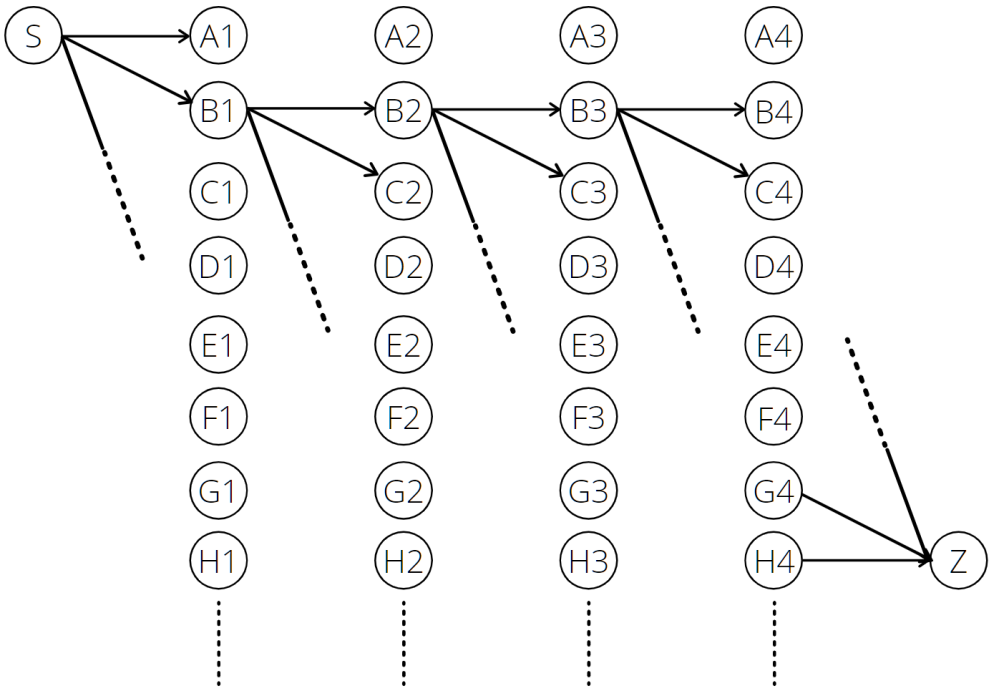

Figura 1. *Arquitetura do grafo*

Palavras-chave: Otimização Global, Modelo Estratigráfico Inverso, Dijkstra.