

Relatorio: Trabalho 1

Paralelização do problema do caixeiro viajante usando OMP

Thiago da Costa P. Senff

October 25, 2023

Contents

1 Algoritmo Sequencial

O Algoritmo sequencial é bastante simples, ele resolve o problema do caixeiro viajante na força bruta usando recursão para montar uma árvore onde cada folha possui um dos possíveis caminhos e por fim verifica qual é o menor. A implementação do algoritmo tem 3 detalhes importantes, primeiro ela faz cortes em caminhos que serão piores que o melhor caminho que ele já achou anteriormente. O algoritmo também utiliza uma heurística que dá preferência de escolha para a cidade mais próxima quando adicionando uma nova cidade no caminho. Por último vale dizer que o algoritmo faz uma implementação bastante duvidosa quando verificando quais cidades já estão no caminho, ele faz a varredura do caminho inteiro verificando a existência da cidade isso poderia ser resolvido com um hash, um array com uma posição para cada cidade contendo 0 caso a cidade não esteja no caminho e 1 caso a cidade já esteja no caminho, isso tornaria a verificação $O(1)$ e tornaria o algoritmo muito mais eficiente. Essa modificação não foi implementada por conta de limitações de testes, pois a entrada paralela ficaria muito rápida e poderia atrapalhar os testes com tempos muito pequenos.

2 Estrategia de Parelização

A estratégia de Parelização utilizada foi de percorrer a árvore em ramos separados paralelamente, utilizando o comando `for` do `omp` na primeira iteração recursiva, assim cada thread cuida de árvores de tamanho $n-1$ paralelamente, essa divisão não é feita na mesma função com a função inicial. Uma dificuldade que essa implementação enfrenta é manter o tamanho do caminho mínimo para fazer os cortes, sendo necessário um uso de `critical` para o momento onde a distância mínima é verificada e atualizada. Foi considerado paralelizar as árvores em níveis inferiores utilizando `tasks` mas testes iniciais mostraram que em geral o tempo na minha máquina se manteve similar com alguns casos com variações de 0 a 6 segundos para melhor e para pior, caso hajam mais núcleos de processamento disponíveis essa abordagem poderia se tornar viável.

3 Metodologia de Testes

Para o teste foi feito um teste sobre 3 entradas diferentes, para o código paralelo o código foi rodado usando desde 1 processador até 4, o máximo da máquina, cada teste foi executado 20 vezes e estaremos olhando para a média deles, foi garantido também que todos os testes tivessem uma duração maior

que 10 segundos para diminuir possíveis influências de processos externos. Para os testes, foi inicialmente pensado em fazer uso do pior e melhor caso do algoritmo, essa ideia acabou sendo descartada, o teste de pior caso aumenta o tempo muito rapidamente então para que o teste mais rápido tivesse no mínimo os 10 segundos de tempo de execução o terceiro teste acabaria levando uma quantidade irrazoável. Testar usando o melhor caso por outro lado é possível porém cria uma grande vantagem para a execução sequencial, que ainda se demonstrou mais lenta mas demonstrando um speedup inferior que todos os outros testes iniciais feitos. Foi também considerado, durante a parte de planejamento, remover a poda de subárvores com respostas piores que o ótimo atual o que faria os algoritmos tivessem tempo de execução somente baseado no tamanho da entrada. Essa ideia porém é uma grande desvantagem para o algoritmo sequencial, que além de perder uma de suas poucas otimizações faria com que não fosse necessário usar critical na versão paralela do algoritmo, uma das poucas fontes de possíveis problemas que ele pode ter, além de tornar o algoritmo menos eficiente de forma geral. No final a abordagem tomada foi de criar testes que tendem ao pior caso do algoritmo mas que ainda são mais fáceis do que o verdadeiro pior caso, para alcançar isso simplesmente garantimos que o algoritmo irá escolher uma cidade errada no início da árvore devido a heurística de cidades mais próximas forçando ele a explorar combinações de caminhos errados.

3.1 Especificações da máquina

4 Análise Sequencial

Rodando o algo

5 Resultados

6 Análise Paralelo

7 Conclusão