

Relatório Lab 01 - Pickup Delivery Problem

Branch and Bound

Marina Miranda Aranha RA174233

Algoritmo

O algoritmo proposto segue os seguintes passos, em alto nível:

- Inicia a formação da árvore de ramificações, percorrendo todas as possibilidades de percursos a partir do vértice inicial (source) passando por todos os vértices uma única vez e terminando no vértice de destino final (target). À medida que encontra soluções melhores que a anterior, armazena a solução e seu valor.
- Realiza as seguintes estratégias de corte:
 - ◆ Por inviabilidade: poda os ramos em que possui um vértice delivery aparecendo antes do seu respectivo pickup
 - ◆ Por não ser promissor: dado um caminho viável usa seu valor final como limitante para as próximas iterações, assim ao explorar um novo vértice, se sua soma intermediária for maior que a do limitante, poda este ramo. Caso o algoritmo chegue ao final da ramificação e a soma seja menor do que a atual do limitante, então o valor é atualizado.

Foram implementadas algumas estratégias para criar um limite superior e podar mais ramos:

1. Calcula uma solução inicial adicionando todos os vértices pickup e depois todos os vértices delivery
2. Constrói um caminho inicial com uma estratégia gulosa de buscar sempre pelos ramos de menor custo a cada iteração, respeitando a regra de pickup-delivery

Resultados Obtidos

Para todas as instâncias, foram feitos 3 testes diferentes: sem a utilização dos limites superiores descritos em 1 e 2, com o limite superior 1 e com o limite superior 2.

Instância de 12 nós:

Para todos os testes feitos, o algoritmo teve uma boa performance, executando em menos de 1s. Não houve melhora de tempo significativa entre os três testes.

Resultado escolhendo o menor bound entre estratégias 1 e 2:

```
Limitante = 6408.54  
Menor custo: 3604.98  
Caminho percorrido :  
0 -> 10 -> 4 -> 11 -> 5 -> 2 -> 8 -> 9 -> 6 -> 3 -> 7 -> 1  
Time taken: 0.00043s
```

Instância de 22 nós:

Para esta instância, o algoritmo não apresentou boa performance, e ultrapassou o limite de tempo estabelecido, apresentando o seguinte resultado:

```
Minimum cost : 4289.956543  
Path Taken : 0 6 7 18 10 14 2 11 19 4 16 12 17 8 20 21 15 3 5 13 9 1  
Time taken: 2387.31585s ~ 40min
```

Se olharmos para as duas primeiras instâncias do problema, podemos observar que os valores estão distribuídos em um intervalo mais ou menos uniforme, sem grandes variações entre si. Dessa forma podemos calcular um limitante com a seguinte expressão, que é calculado para cada nó:

```
naive_bound = nNodes * curr_weight / curr_level;
```

Testando com este limitante, o algoritmo apresentou uma melhora considerável:

```
Limitante = 8660.36  
Menor custo: 4289.96  
Caminho percorrido:  
0 -> 6 -> 7 -> 18 -> 10 -> 14 -> 2 -> 11 -> 19 -> 4 -> 16 -> 12 -> 17 -> 8  
-> 20 -> 21 -> 15 -> 3 -> 5 -> 13 -> 9 -> 1  
Time taken: 422.14059s ~ 7min
```