TP de Grafos: Compartilhamento de viagens

Alef Miranda

6 de junho de 2017

1 Problema

O problema de modo geral consiste em assinalar motoristas e passageiros a partir de um grafo de viagens compartilháveis. A descrição completa do problema não é o foco desse documento. O problema, comumente conhecido como *Capacitated Slugging Problem* é *NP-Hard* como demonstrado em Ma and Wolfson [2013].

2 Solução

Por se tratar de um problema computacionalmente difícil, escolheu-se por resolvê-lo utilizando o paradigma de *tentativa e erro*. A ideia básica é construir todas as soluções viáveis e retornar aquela com a maior recompensa. O algoritmo está escrito abaixo utilizando pseudo-código:

```
Data: Um grafo de compartilhamento de viagens
Result: O compartilhamento de recompensa máxima
// improve(solution, edges, graph)
// Solução inicial é vazia e todas as arestas são consideradas
// solution = (value: 0.0, edges: ∅) // edge = graph.edges
best = solution
for edge in edges do
    if edge can be added to solution then
        update graph with remaining seats
        better =
         (value : best.value + edge.value, edges : solution.edges ∪ edge)
        compute remaining edges
        improved = improve better solution with remaining edges
        if improved is better than best solution then
            best = improved
        end
    end
end
return best
```

3 Complexidade

Como todas as combinações possíveis de arestas são geradas, a função *improve* pode ser chamada até $e!$ (fatorial) vezes no pior caso, isto é,o número de arestas cortadas para cada aresta adicionada à solução ser igual a 1 (a própria aresta adicionada). É possível melhorar a complexidade de tempo dado que as solulções intermediárias compartilham arestas, mas isso não foi implementado. A figura 1 apresenta o tempo de execução do algoritmo para diferentes tamanhos de grafos. Para esses experimentos usou-se grafos acíclicos densos $O(v^2)$ para o número de arestas. O número de vértices desse experimento varia de 10 A 500 em intervalos de 50 vértices (10, 50, 100...).

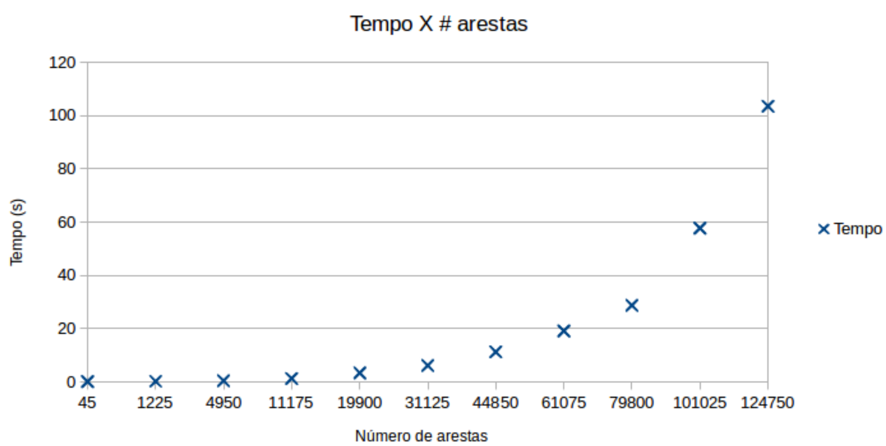

Figura 1: Experimentos – Pelo gráfico pode-se perceber uma curvatura exponencial para o tempo de execução do algoritmo. Deve-se lembrar que os vértices aumentos de forma uniforme e não as arestas (que aumentam de forma quadrática para esses experimentos).

A complexidade de espaço está relacionada ao tamanho da maior solução, pois irá representar a profundidade das chamadas recursivas. Como em cada uma das chamadas temos um novo conjunto de arestas construído, a complexidade de espaço é $O(e^2)$, de forma mais específica $O(e(e-1)/2)$.

Referências

Shuo Ma and Ouri Wolfson. Analysis and evaluation of the slugging form of ridesharing. In *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 64–73. ACM, 2013.