

Universidade Federal da Paraíba

Centro de Informática

Análise e Projeto de Algoritmo

Professor: Bruno Bruck

Aluno: Ednaldo Martins da Silva

Projeto Final da Disciplina: Problema de Otimização Combinatória

Problema de Otimização Combinatória: 1. Caixeiro Viajante

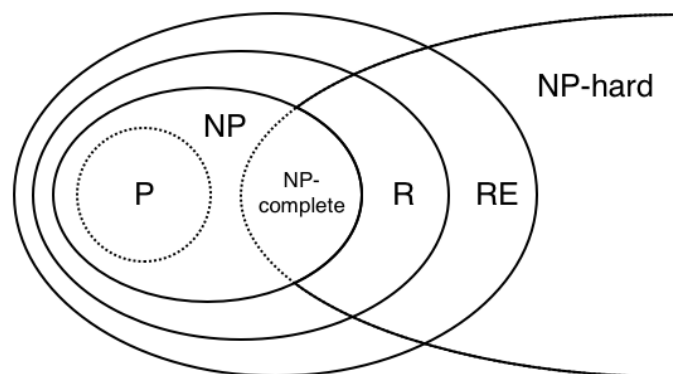
Introdução

O problema do caixeiro viajante é um problema de otimização combinatória. Para solucionar este problema devemos encontrar a menor rota possível passando por várias cidades, uma única vez cada, não importando a ordem em que são visitadas, e retornar ao ponto de origem. Para este caso temos um problema da Classe NP-Difícil.

Classe NP

NP ou **Non-Deterministic Polynomial time** é a classe das linguagens que têm verificadores de tempo polinomial. Enquanto as linguagens da Classe P podem ser decididas rapidamente, as linguagens da Classe NP podem ser verificadas rapidamente.

Para solucionar alguns problemas podemos evitar a força bruta para obter uma solução de tempo polinomial. Todavia isso nem sempre é possível, e isto pode ocorrer porque não exista uma solução de Classe P para o problema, ou porque ainda não descobrimos uma maneira possível para executar uma solução de Classe P para o problema.



Fonte: <https://cs.stackexchange.com/questions/90659>.

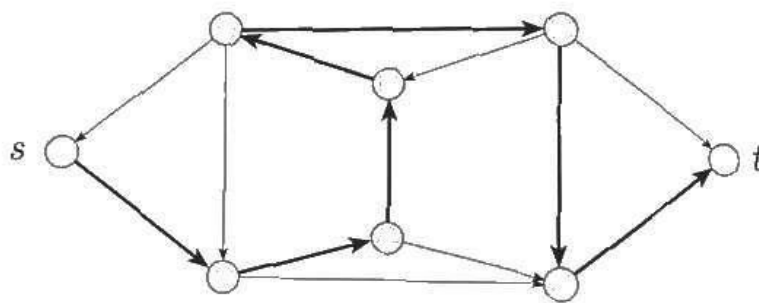
Também existem problemas da Classe NP-Completo (ou NP-Completeness), que são problemas mais difíceis da Classe NP, e que provavelmente não fazem parte da Classe P, e caso encontre uma prova para tal, que mostre que faça parte, então estaríamos afirmando que qualquer problema NP-Completo que estão interligados poderiam ser solucionados rapidamente ou em tempo polinomial. Para entendermos esse fato podemos fazer analogia a um problema qualquer que possui um grau de dificuldade X e outro de X^2 , portanto se solucionarmos o problema de dificuldade X^2 então encontraríamos a solução para o problema de dificuldade X , pois estão interligados.

Redutibilidade em Tempo Polinomial

Para solucionar esses problemas é utilizado um método chamado de **Redutibilidade em Tempo Polinomial**. De acordo com Sipser (2007, p. 288) quando o problema A se reduz ao problema B, uma solução para B pode ser usada para resolver A. Agora definimos uma versão da redutibilidade que leva em consideração a eficiência da computação. Quando o problema A é eficientemente redutível ao problema B, uma solução eficiente para B pode ser utilizada para resolver A eficientemente.

Caminho Hamiltoniano

Um **caminho hamiltoniano** em um grafo direcionado G é um caminho direcionado que passa por cada nó exatamente uma vez. Podemos dizer que se um grafo direcionado contém um caminho hamiltoniano, ou seja, um possível caminho que passe por todos os nós ou vértices até chegar ao destino, então este grafo possui um caminho hamiltoniano.



Fonte: Sipser, página 280.

Vemos na imagem acima um grafo direcionado onde o caminho hamiltoniano vai de 's' para 't'.

O problema do caminho hamiltoniano é NP-Completo, portanto, é improvável encontrar uma solução que resolva esse problema em tempo polinomial. Um problema similar ao do caminho hamiltoniano é o problema do **caixeiro viajante**.

Caixeiro Viajante

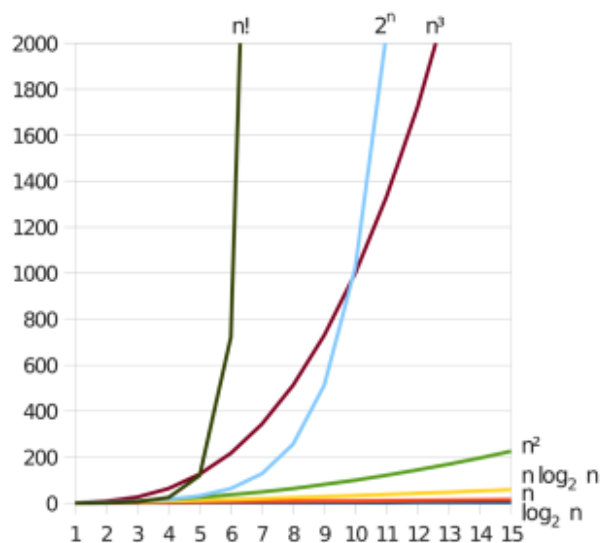
Como visto no início, o problema do caixeiro viajante tem como objetivo encontrar a rota que faça uma viagem percorrendo um percurso mínimo, indo do ponto de partida, passando por todas as cidades uma única vez, e retornando.

Vamos tomar como exemplo um grafo com 4 cidades, daí temos o grafo $G=\{1,2,3,4\}$, onde o conjunto de valores contido no grafo representam os vértices, ou seja, as cidades para o nosso exemplo. Digamos que esse é um grafo completo, onde todas as cidades estão conectadas entre si. Vejamos então os possíveis caminhos partindo da cidade 1:

12341, 12431, 13421, 13241, 14321, 14231.

Note que encontramos 6 possíveis caminhos. Partindo da cidade 1 temos que tomar a decisão entre 3 cidades possíveis a seguir. Ao chegar na segunda cidade temos que decidir entre 2 cidades seguintes, pois uma delas já se foi e não devemos voltar nela e também não devemos voltar na origem enquanto não percorremos todas as cidades restantes. Após tomar a segunda decisão restará apenas uma cidade antes de voltar para origem. Veja, decidimos entre 3 cidades, depois 2 (duas), e então por 1 (uma). Notemos que esse caso parece uma equação resolvida com fatorial, onde temos N cidades e que sabemos quantos caminhos são possíveis tomar partindo da equação $R=(N-1)!$, onde R é o número de rotas encontradas e N o número de cidades.

Temos aqui um problema de complexidade de Algoritmo que torna a solução impraticável quando precisamos de um N grande, principalmente levando em conta que o tempo para solução desse problema se torna muito pior cada vez que esse N aumenta um pouco.



Se levarmos em consideração situações reais temos muitos problemas ao calcular a melhor rota quando trabalhamos com muitas cidades. Suponhamos que temos 8 cidades, então temos $R=(8-1)! \Rightarrow R=5040$ rotas possíveis, todavia se tivermos uma situação com 21 cidades, então teríamos $R=(21-1)! \Rightarrow R=2432902008176640000$ de rotas possíveis. Isso é inviável. Isso acontece porque esse é um método de força bruta. Até o momento ainda não foi possível encontrar um método ou técnica que realize esse trabalho com complexidade em tempo polinomial, e como visto durante a leitura, talvez nem exista.

REFERÊNCIAS

SIPSER, Michael. Introdução a Teoria da Computação. **Cengage Learning**, p. 271-311, 2007.

https://www.ime.usp.br/~pf/analise_de_algoritmos/aulas/NPcompleto.html

<http://www.mat.ufrgs.br/~portosil/caixeiro.html>

<https://cs.stackexchange.com/questions/90659>

https://homepages.dcc.ufmg.br/~loureiro/alg/071/paa_ProbNP_1pp.pdf

https://pt.wikipedia.org/wiki/Caminho_hamiltoniano

[https://pt.wikipedia.org/wiki/NP_\(complexidade\)](https://pt.wikipedia.org/wiki/NP_(complexidade))