UNIP
UNIVERSIDADE PAULISTA

Ciência da Computação e Sistemas de Informação

PESQUISA OPERACIONAL (PO)

OTIMIZAÇÃO

Prof. Arthur

arthur.battaglia@docente.unip.br

2021/2

O PRINCÍPIO DE OTIMIZAÇÃO EM REDES

Esse princípio estabelece que, se o roteador J estiver no caminho ótimo entre o roteador I e o roteador K, o caminho ótimo de J até K também estará na mesma rota.

Como consequência, o conjunto de rotas ótimas de todas as origens para um determinado destino forma uma árvore com raiz no destino.

Uma árvore como essa é chamada de "árvore de escoamento".

Na figura a seguir a unidade métrica de distância é o número de hops.

Fonte: Redes de Computadores – Andrew S. Tanenbaum – 4a edição

O PRINCÍPIO DE OTIMIZAÇÃO EM REDES

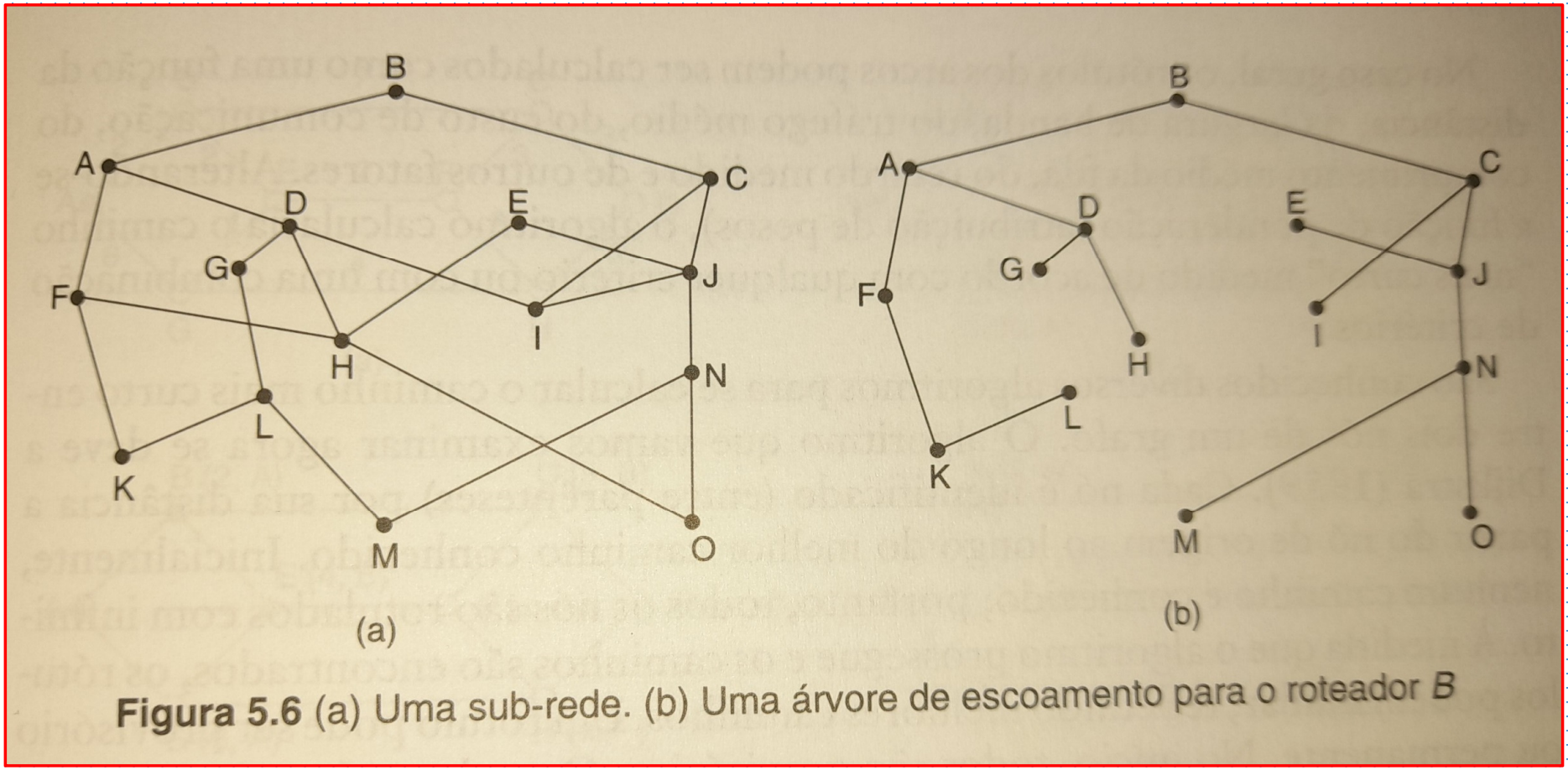

Figura 5.6 (a) Uma sub-rede. (b) Uma árvore de escoamento para o roteador *B*

Fonte: Redes de Computadores – Andrew S. Tanenbaum – 4a edição

O PRINCÍPIO DE OTIMIZAÇÃO EM REDES

A árvore de escoamento não contém loops. Isso significa que cada pacote de bits será entregue dentro de um número finito e limitado de hops.

Na prática, enlaces e roteadores podem sair do ar e voltar à atividade durante a operação, por isso diferentes roteadores podem tomar decisões diferentes sobre a topologia atual.

O princípio da otimização, e a árvore de escoamento, permitem que se faça um comparativo para detectar quais outros algoritmos de roteamento podem ser medidos.

Fonte: Redes de Computadores – Andrew S. Tanenbaum – 4a edição

ROTEAMENTO PELO CAMINHO MAIS CURTO

A ideia é criar um grafo da sub-rede, com cada nó representando um roteador e cada aresta indicando uma linha de comunicação (enlace).

Para escolher uma rota entre determinado par de roteadores, o algoritmo encontra o caminho mais curto entre eles no grafo. O caminho mais curto pode ser, por exemplo, o mais rápido.

No algoritmo de Dijkstra cada nó é identificado (entre parênteses) por sua distância a partir do nó de origem ao longo do melhor caminho conhecido.

A rota entre o roteador de origem A até o roteador de destino D pode ser determinada através da pesquisa operacional.

Fonte: Redes de Computadores – Andrew S. Tanenbaum – 4a edição

ROTEAMENTO PELO CAMINHO MAIS CURTO

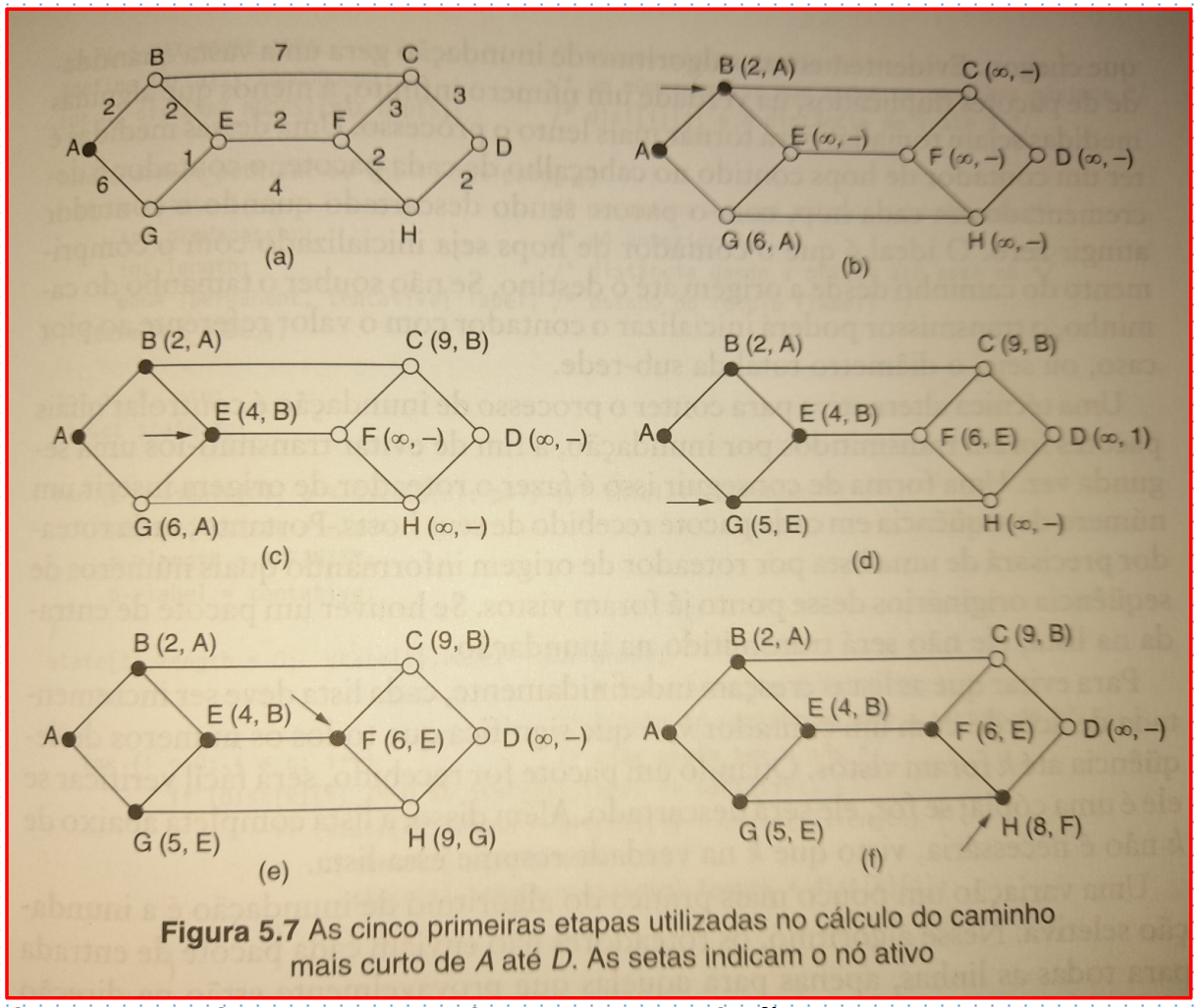

Figura 5.7 As cinco primeiras etapas utilizadas no cálculo do caminho mais curto de *A* até *D*. As setas indicam o nó ativo

Fonte: Redes de Computadores – Andrew S. Tanenbaum – 4a edição

O PROBLEMA DA MOCHILA

É um problema de maximização.

Descreve exemplos que envolvem um contêiner (a 'mochila') com uma capacidade fixa e um número de itens como conteúdos.

Problemas da mochila são um aplicativo típico de programação inteira (IP). Em problemas da mochila, há um contêiner (a 'mochila') com uma capacidade fixa (um número inteiro) e um número de itens. Cada item tem um peso associado (um número inteiro) e um valor associado (outro número inteiro). O problema consiste em preencher a mochila sem exceder a sua capacidade, enquanto maximiza o valor geral de seus conteúdos.

Fonte: https://www.ibm.com/docs/pt-br/icos/12.9.0?topic=programming-knapsack-problems

O PROBLEMA DA MOCHILA

Suponha dado um conjunto de objetos, cada um com um certo peso e um certo valor. Quais dos objetos devo colocar na minha mochila para que o valor total seja o maior possível? Minha mochila tem capacidade para 15 kg apenas.

Este problema é um dos mais conhecidos na literatura e facilmente encontrado em situações reais, como investir um capital em várias aplicações por exemplo. Ou, no caso dos paramédicos, como carregar a mochila de primeiros socorros com os medicamentos e equipamentos mais necessários e na maior quantidade possível.

Fonte: https://www.ime.usp.br/~pf/analise_de_algoritmos/aulas/mochila-bool.html

O PROBLEMA DA MOCHILA

Há três soluções clássicas para o problema de encontrar um conjunto de itens nos quais a soma de seus pesos não exceda um valor dado e que a soma da suas utilidades seja a maior possível.

- O primeiro algoritmo é através da força bruta.
- O segundo algoritmo utiliza a técnica de programação dinâmica.
- O terceiro é um algoritmo de aproximação que utiliza a técnica de algoritmos gulosos.

PROGRAMAÇÃO DINÂMICA

Ao contrário da programação linear, não existe um padrão matemático para formulação do "problema de programação dinâmica". Em vez disso, a programação dinâmica é um tipo geral de abordagem à solução de problemas, e as equações específicas usadas devem ser desenvolvidas para se adequarem a cada situação.

ALGORITMOS GULOSOS

De forma geral, os algoritmos relacionados com otimização lidam com uma sequência de passos, sendo que em cada passo há um conjunto de escolhas/opções. Uma estratégia para resolver problemas de otimização são os algoritmos gulosos, os quais escolhem a opção que parece ser a melhor no momento (escolha ótima), e esperam que desta forma consiga-se chegar a uma solução ótima global.

Embora nem sempre seja possível chegar a uma solução ótima a partir da utilização de algoritmos gulosos, eles são eficientes em uma ampla variedade de problemas.

Fonte: https://www.ic.unicamp.br/~rocha/msc/complex/algoritmosGulososFinal.pdf

ALGORITMOS GULOSOS

Os algoritmos gulosos tomam decisões com base apenas na informação disponível, sem se preocupar com os efeitos futuros de tais decisões, isto é, eles nunca reconsideram as decisões tomadas, independentemente das consequências. Não há, portanto, necessidade de avaliar as alternativas e nem de empregar procedimentos elaborados permitindo que decisões anteriores sejam desfeitas.

Fonte: https://www.ic.unicamp.br/~rocha/msc/complex/algoritmosGulososFinal.pdf

O PROBLEMA DO CAIXEIRO VIAJANTE

É um exemplo de minimização.

O problema do caixeiro viajante consiste em descobrir a rota que torna mínima a viagem total. Exemplificando o caso n = 4: se tivermos quatro cidades A, B, C e D, uma rota que o caixeiro deve considerar poderia ser: saia de A e daí vá para B, dessa vá para C, e daí vá para D e então volte a A.

Entre os métodos de roteirização mais utilizados, está o Problema do Caixeiro Viajante (PCV), que consiste em traçar uma rota que retorne o menor percurso, partindo de um ponto, visitando todos os outros somente uma vez e retornando ao ponto de origem (Ballou, 2006).

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

O PROBLEMA DO CAIXEIRO VIAJANTE

O problema do caixeiro é um clássico exemplo de problema de otimização combinatória.

A primeira coisa que podemos pensar para resolver esse tipo de problema é reduzi-lo a um problema de enumeração: achamos todas as rotas possíveis e, usando um computador, calculamos o comprimento de cada uma delas e então vemos qual a menor (é claro que se acharmos todas as rotas estaremos contando-as, daí podermos dizer que estamos reduzindo o problema de otimização a um de enumeração).

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

UNIP
UNIVERSIDADE PAULISTA

Ciência da Computação e Sistemas de Informação

PESQUISA OPERACIONAL (PO)

OTIMIZAÇÃO

Prof. Arthur

arthur.battaglia@docente.unip.br

2021/2