

CAPÍTULO II

II.1) PROBLEMA DE CAMINHO MÍNIMO (Introdução)

Faremos inicialmente uma abordagem de problemas de caminho mínimo, necessários, como veremos, em vários algoritmos de fluxo de custo mínimo. Os algoritmos utilizados nos problemas de caminho mínimo são, em geral, muito eficientes quando aplicados a redes acíclicas e sem arcos de comprimento (ou custo) negativo. No caso mais geral o problema se torna bem mais difícil de resolver em "tempo satisfatório", dizemos neste caso que o problema é NP-Completo!

Em geral, os algoritmos de caminho mínimo se baseiam em rótulos (associados a cada nó) que expressam distâncias. Se o rótulo de um determinado nó é infinito devemos ainda encontrar um caminho ligando a fonte a esse nó. Se o rótulo é finito ele representa a distância através de um caminho qualquer.

Classificamos os algoritmos de caminho mínimo em dois grupos: algoritmos de atualização de rótulos (label setting) e algoritmos de correção de rótulos (label correcting). As duas abordagens são iterativas. Na atualização de rótulos designamos um nó como permanente (ótimo) a cada iteração. Na correção de rótulos todos os rótulos são temporários até a última iteração quando se tornam permanentes (ótimos) simultaneamente. Os algoritmos de atualização de rótulos são utilizados apenas em redes acíclicas e sem arcos de custo negativo.

Estudaremos o caso mais geral, os algoritmos de correção de rótulos onde se admite a presença de ciclos de custo negativo. Mostraremos como obter modificações no algoritmo de correção de rótulos de forma a detectar e identificar a presença destes ciclos (caso existam).

II.2) FORMULAÇÃO DO PROBLEMA:

Seja $G=(N,A)$ uma rede direcionada com arcos de comprimento c_{ij} associados a cada par $(i,j) \in A$ e $s \in N$ o nó fonte da rede.

Definimos também $C = \max\{c_{ij} : (i,j) \in A\}$ para fins de análise de complexidade como será visto posteriormente.

No problema de caminho mínimo abordado aqui buscamos o menor caminho da fonte s aos demais nós da rede. Alternativamente este problema pode ser formulado como um problema de programação linear onde desejamos enviar uma unidade de fluxo pelo caminho mais barato (c_{ij} denotando custos) aos demais nós da rede ($N-\{s\}$). Formalmente temos:

$$\min \sum_{(i,j) \in A} c_{ij} x_{ij}$$

$$s.a \quad \sum_{\{j:(i,j) \in A\}} x_{ij} - \sum_{\{j:(i,j) \in A\}} x_{ji} = \begin{cases} n-1, & \text{para } i = s \\ -1 & \text{para todo } i \in N - \{s\} \end{cases}$$

onde: $x_{ij} \geq 0, \quad \forall (i,j) \in A$

Em nosso estudo do problema de caminho mínimo faremos algumas "restrições":

i) *Todos os arcos são de comprimento inteiro positivo.*

Algoritmos cuja complexidade depende de C (p.ex: algoritmos de correção de rótulos) assumem tal condição em função do modelo teórico de máquina (em nosso caso máquina RAM). A Random Access Machine (RAM) exige que todos os dados envolvidos no problema sejam inteiros. Pode-se mostrar que a utilização da RAM é equivalente à utilização de outros modelos teóricos de computador (vide [3]).

Note que sempre podemos transformar números racionais em números inteiros fazendo a multiplicação por um inteiro escolhido convenientemente. Desta forma esta hipótese não se torna realmente restritiva.

ii) *A rede contém um caminho direcionado de s a todos os demais nós ($N-\{s\}$) da rede.*

Esta hipótese pode ser satisfeita considerando-se um arco fictício (s,i) com custo suficientemente grande para cada nó i não conectado a s por um caminho direcionado.

iii) *A rede não contém ciclo de custo negativo.*

Na presença de ciclos negativos o custo pode ser reduzido indefinidamente (verificação trivial). Neste caso, precisamos impor certas condições para se evitar a repetição de nós no caminho mínimo. A adição destas "simples" condições trazem sérias implicações computacionais. Como discutido anteriormente este problema é NP-Completo.

iv) *A rede é direcionada.*

Se a rede não é direcionada e todos os arcos de comprimento não-negativo podemos transformá-la em direcionada. Essa transformação não pode ser feita na presença de arcos de comprimento negativo já que cada arco de comprimento negativo produziria um ciclo negativo. Uma transformação mais elaborada é realizada para contornar essa situação (vide [Ahuja; seção 2.4]).

Note que se desejamos enviar apenas uma unidade de fluxo de s a t temos a seguinte formulação:

$$\begin{aligned} \min \quad & \sum_{(i,j) \in A} c_{ij} x_{ij} \\ \text{s.a} \quad & \sum_{\{j:(i,j) \in A\}} x_{ij} - \sum_{\{j:(i,j) \in A\}} x_{ji} = \begin{cases} 1, & \text{para } i = s \\ -1 & \text{para } i = t \\ 0 & \text{para } i \in N - \{s, t\} \end{cases} \\ \text{onde:} \quad & x_{ij} \geq 0, \quad \forall (i, j) \in A \end{aligned}$$

Como a matriz A é unimodular e temos 1 e -1 em cada uma de suas colunas podemos escrever a formulação dual como:

$$\begin{aligned} \max \quad & d(t) - d(s) \\ \text{s.a} \quad & d(j) - d(i) \leq c_{ij} \quad \forall (i, j) \in A \end{aligned}$$

onde d(i) e d(j) representam as distâncias de **i** e **j** respectivamente até **s** (Ahuja, seção 9.4). Note que obtemos a distância mínima entre **s** e **t** resolvendo-se um problema de maximização!!

II.3) CONDIÇÕES DE OTIMALIDADE:

Abaixo, descrevemos as condições de otimalidade do problema de caminho mínimo. Como veremos, além de servir como critério de parada de um determinado algoritmo, nos sugere o desenvolvimento e criação de novos algoritmos.

Teorema II.1: (Condições de Otimalidade do Problema de Caminho Mínimo)

Seja $G=(N,A)$ um grafo direcionado. Para cada nó $j \in N$, seja $d(j)$ o comprimento de algum caminho direcionado da fonte **s** ao nó **j** e c_{ij} o comprimento de um arco $(i,j) \in A$. Desta forma os números $d(j)$ representam caminhos mínimos se e somente se eles satisfazem as seguintes condições de otimalidade:

$$d(j) \leq d(i) + c_{ij}, \quad \forall (i, j) \in A$$

Demonstração: Mostraremos inicialmente que se os rótulos $d(j)$ p/ cada $j \in N$, definem caminhos mínimos a partir da fonte **s** eles devem satisfazer a desigualdade acima.

Caso tenhamos algum nó $j \in N$ tal que $d(j) > d(i) + c_{ij}$ podemos obter um caminho mais curto passando pelo nó i , basta fazer $d'(j) = d(i) + c_{ij}$. Logo, $d(j)$ não representa um caminho mínimo a partir de s , ou seja, obtemos $d'(j) < d(j)$. Chegamos portanto a uma contradição.

Reciprocamente, seja $d(j)$ um rótulo qualquer satisfazendo:

$$d(j) \leq d(i) + c_{ij}, \quad \forall (i, j) \in A \quad (i)$$

Mostraremos que cada um desses rótulos $d(j)$ definem um caminho mínimo de s até j . Seja $s = i_1 - i_2 - \dots - i_k = j$ um caminho direcionado P de s até j . Da desigualdade (i) temos:

$$\begin{aligned} d(j) &= d(i_k) \leq d(i_{k-1}) + c_{i_{k-1}i_k}, \\ d(i_{k-1}) &\leq d(i_{k-2}) + c_{i_{k-2}i_{k-1}}, \\ &\dots\dots\dots \\ d(i_2) &\leq d(i_1) + c_{i_1i_2} = c_{i_1i_2} \end{aligned}$$

Observe na última desigualdade que: $d(i_1) = d(s) = 0$. Fazendo a substituição sucessiva dessas inequações obtemos:

$$d(j) \leq c_{i_{k-1}i_k} + c_{i_{k-2}i_{k-1}} + \dots + c_{i_1i_2} = \sum_{(i,j) \in P} c_{ij}$$

Note agora que $d(j)$ é um limite inferior para qualquer caminho direcionado da fonte s até j , sendo portanto um caminho mínimo. •

Observação: Veja que estas condições de otimalidade equivalem à obtenção de uma solução dual viável para o dual do problema de caminho mínimo.

Baseado nas condições de otimalidade expostas acima já temos a base necessária para formular um algoritmo geral para obtenção do caminho mínimo:

II.4) ALGORITMOS DE CORREÇÃO DE RÓTULOS: (Label Correcting)

Assumiremos inicialmente que nossa rede não admite ciclos de custo negativo. Posteriormente, quando trabalharmos com problemas de custo mínimo (no próximo capítulo) estaremos interessados em casos mais gerais e a determinação desses ciclos de custos negativos será fundamental em alguns algoritmos.

Os algoritmos de correção de rótulos mantêm um conjunto de rótulos a cada etapa. O rótulo $d(j)$ é infinito quando não descobrimos ainda um caminho direcionado de s até j . Para cada nó j , definimos um índice $\text{pred}(j)$ que denota o nó anterior a j no caminho correspondente ao comprimento $d(j)$. Isto nos ajudará determinar explicitamente o caminho mínimo ao final das iterações.

O algoritmo genérico de correção de rótulos procura atualizar sucessivamente os rótulos até que as condições de otimalidade sejam satisfeitas.

Algoritmo: (Correção de Rótulos)

Início

$d(s) := 0$;

$\text{pred}(s) := 0$;

$d(j) := \infty$ para cada $j \in N - \{s\}$;

Enquanto algum arco (i,j) satisfaz $d(j) > d(i) + c_{ij}$ **faça**

$d(j) := d(i) + c_{ij}$;

$\text{pred}(j) := i$;

fim enq;

fim.

Exemplo: Abaixo apresentamos um exemplo de um grafo com 5 nós e 6 arestas. os rótulos são representados ao lado de cada nó:

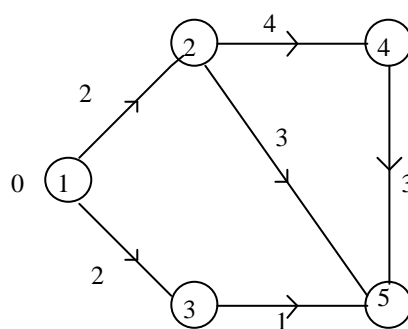


Figura II.1: Algoritmo de Correção de Rótulos

Analisando os arcos (i,j) na sequência: $(1,3)$; $(1,2)$; $(2,4)$; $(4,5)$; $(2,5)$ e $(3,5)$ obtemos a seguinte configuração final:

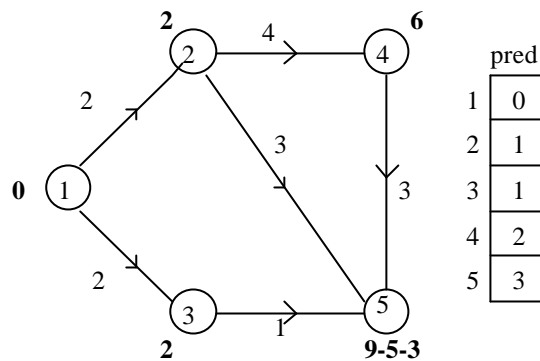


Figura II.2: Algoritmo de Correção de Rótulos

Note que para cada arco (i,j) (na ordem descrita acima) fizemos a atualização de $d(j)$ se $d(j) > d(i) + c_{ij}$. Ao lado do nó 5, por exemplo representamos todas as alterações do rótulo $d(j)$.

Observe que o vetor de predecessores nos permite obter explicitamente o caminho mínimo da fonte (nó 1) aos demais nós. No caso acima: 1-3-5.

O algoritmo genérico para correção de rótulos não especifica nenhum método para seleção dos arcos a serem analisados. Uma estratégia é "varrer" uma lista arbitrária de arcos sequencialmente para cada nó e verificar os arcos que violam as condições de otimalidade. Dessa forma teríamos $O(m)$ passos a cada iteração. É possível reduzir este limite para uma média de $O(n/m)$ passos por iteração. Em Ahuja (seção 5.3) é descrito um algoritmo de correção de rótulos modificado cuja complexidade final é $O(nmC)$, sendo portanto pseudo-polinomial. Outras implementações mais sofisticadas podem ainda ser utilizadas.

II.4.1) Análise de Complexidade:

Observe que no algoritmo genérico de correção de rótulos cada atualização de $d(j)$ pode ser considerada como operação elementar. Desejamos então calcular o número de atualizações desses rótulos no pior caso (número máximo).

Note que cada caminho de s a j tem no máximo $n-1$ vértices. Como cada arco tem o comprimento limitado por C , cada rótulo $d(j)$ será limitado superiormente por $nC-C$. Da mesma forma se supomos a existência de arcos de comprimento negativo (mas ainda s/ ciclos de custo negativo) temos que $-nC+C$ será um limite inferior para um rótulo qualquer $d(j)$ onde $C = \max\{|c_{ij}| : (i,j) \in A\}$. Desta forma, se tivermos o acréscimo de apenas uma unidade de comprimento a cada iteração teríamos $2nC-2C$ atualizações de $d(j)$ no pior caso. Como podemos fazer tais atualizações em todos os vértices distintos de s , a complexidade local será dada por:

$2n^2 - 4nC + 2C$ (multiplicamos o número de atualizações de $d(j)$ por $n-1$), logo a complexidade assintótica será $O(n^2C)$.

Note que se todos os arcos forem de comprimento positivo teremos também $O(n^2C)$ iterações (verificação análoga). Logo, o algoritmo acima terá complexidade assintótica pseudo-polinomial, observe que não temos um polinômio definido no tamanho do problema (observe que poderíamos fazer $C = 2^n$!).

II.5) GRAFOS COM CICLOS DE CUSTO NEGATIVO:

Assumimos anteriormente que a rede $G=(N,A)$ não continha ciclos de custo negativo e descrevemos um algoritmo que resolve o problema de caminho mínimo. Vejamos agora como fazer alterações neste algoritmo de forma a identificar a presença dos ciclos de custo negativo (caso existam). Para tanto, consideremos inicialmente a seguinte definição:

Definição II.1: Definimos o *custo reduzido* \bar{c}_{ij} de um arco (i,j) com respeito aos rótulos $d(\cdot)$ como:
 $\bar{c}_{ij} = c_{ij} + d(i) - d(j)$. •

Vejamos então as modificações exigidas pelo algoritmo genérico de correção de rótulos. Se a rede $G=(N,A)$ contém um ciclo de custo negativo, o conjunto de rótulos $d(j)$ não irá satisfazer as condições de otimalidade. Note que para um ciclo qualquer direcionado $w \in W$ (onde w é o conjunto de todos os ciclos em G) temos:

$$\sum_{(i,j) \in W} \bar{c}_{ij} = \sum_{(i,j) \in W} c_{ij}$$

onde $\bar{c}_{ij} = c_{ij} + d(i) - d(j)$ (verificação trivial). Observe que $\sum_{(i,j) \in W} c_{ij} < 0$, desta forma teríamos algum $\bar{c}_{ij} \in w$ negativo e consequentemente $d(j) > d(i) + c_{ij}$ contrariando o teorema II.1 (condições de otimalidade).

Desta forma o algoritmo genérico de correção de rótulos irá reduzir indefinitivamente o valor dos rótulos $d(j)$ do ciclo w e nunca terminará. Note entretanto que $-nC$ é um limite inferior para os rótulos $d(j)$ caso a rede $G=(N,A)$ não contenha ciclos de custo negativo. Consequentemente, se encontrarmos algum nó k cujo rótulo seja inferior a $-nC$ teremos encontrado um ciclo negativo. A obtenção explícita deste ciclo pode ser feita através do vetor de predecessores iniciando-se no vértice k .

Podemos obter, evidentemente, formas alternativas mais eficientes para a detecção de ciclos com custo negativo. Algoritmos baseados na ideia de *grafo predecessor* ou *algoritmos de correção de rótulos FIFO* (first-in first-out) são analisados em (Ahuja, seção 5.5).