
CAPÍTULO 2

Grafos e Redes

1. Introdução

Um **grafo** é uma representação visual de um determinado conjunto de dados e da ligação existente entre alguns dos elementos desse conjunto. Desta forma, em muitos dos problemas que nos surgem, a forma mais simples de o descrever é representá-lo em forma de grafo, uma vez que a representação visual que apresenta trará vantagens na construção de um modelo matemático com vista à resolução do problema. Por esta razão é que muitas áreas do conhecimento (matemática, telecomunicações, engenharia civil, gestão, etc.) recorrem à teoria de grafos para resolver muitos problemas.

Uma vez que as redes surgem em numerosos contextos e em variadas formas, muitos domínios têm contribuído com ideias importantes, para a evolução do estudo de problemas de fluxo em redes, já que a forma mais simples de modelar matematicamente muitos problemas inerentes a esses domínios é utilizar redes. Apesar desta diversidade ter produzido inúmeros benefícios, como a introdução de perspectivas muito ricas e variadas, também trouxe alguns custos : por exemplo, a literatura da teoria de grafos e de redes não possui uniformização, fazendo com que vários autores adotem uma grande variedade de convenções e notação. Desta forma, podem-se formular problemas de fluxo em redes de diferentes formas padrão e utilizar muitos grupos alternativos de definições e terminologia. As definições e terminologia aqui adoptadas são, na sua maioria, as que se encontram em Ahuja et al. [1].

2. Conceitos fundamentais de grafos

Um **grafo** é uma estrutura constituída por dois conjuntos finitos, um de vértices (nós ou nodos) e o outro de arestas (arcos ou ramos), e pode ser representado por $\mathcal{G} = (\mathbf{N}, \mathbf{A})$, em que \mathbf{N} e \mathbf{A} são os conjuntos de vértices e arestas, respectivamente, com $\mathbf{A} \subseteq \mathbf{N} \times \mathbf{N}$.

Cada aresta é representada por um par (i, j) , com $i \neq j$ e $i, j \in \mathbf{N}$, em que i é a sua *cauda* e j a sua *cabeça*; diz-se que (i, j) *sai* do vértice i e *chega* ao vértice j . Uma aresta diz-se **dirigida (orientada)** se for representada um par ordenado de vértices distintos e **não dirigida (não orientada)** se for representada por um par não ordenado de vértices distintos — uma aresta que liga os vértices i e j representa-se por (i, j) ou por (j, i) . Um arco dirigido (i, j) pode ser visto como uma rua de um só sentido, que permite fluxo apenas de i para j e um arco não dirigido (i, j) pode ser visto como uma rua de dois sentidos, que permite fluxo em ambas as direcções (de i para j e de j para i).

A aresta (i, j) é *incidente* nos vértices i e j . A aresta (i, j) é uma *aresta de saída* do vértice i e uma *aresta de chegada* do vértice j . O vértice j diz-se **adjacente** ao vértice i se (i, j) for uma aresta. Duas **arestas** são **adjacentes** se forem ambas incidentes relativamente ao mesmo vértice. Um **vértice** é de **ordem** k se tiver k arestas a ele adjacente.

A *lista de adjacência de arestas* de um vértice i , $A(i)$, é o conjunto de arestas que saem de i ; isto é, $A(i) = \{ (i, j) \in \mathbf{A} : j \in \mathbf{N} \}$. A *lista de adjacência de vértices* de um vértice i , $V(i)$, é o conjunto de vértices adjacentes a i ; isto é, $V(i) = \{ j \in \mathbf{N} : (i, j) \in \mathbf{A} \}$. Tanto $|A(i)|$ como $|V(i)|$ correspondem à quantidade de arcos que saem do nó i .

Existem 3 tipos de grafos :

- **dirigido (orientado)** — todas as arestas são dirigidas,
- **não dirigido (não orientado)** — todas as arestas são não dirigidas e,
- **misto** — algumas arestas são dirigidas e outras são não dirigidas.

Um **grafo** diz-se **completo** se entre quaisquer dois vértices existir uma aresta dirigida (grafos dirigidos) ou não dirigida (grafos não dirigidos).

A **densidade** de um **grafo** é a razão entre a quantidade de arestas do grafo e a quantidade de arestas do grafo completo com o mesma quantidade de vértices.

Um grafo $\mathcal{G}' = (\mathbf{N}', \mathbf{A}')$ é um subgrafo de $\mathcal{G} = (\mathbf{N}, \mathbf{A})$ se $\mathbf{N}' \subseteq \mathbf{N}$ e $\mathbf{A}' \subseteq \mathbf{A}$.

Considerem-se dois vértices, s e t , de um grafo $G = (N, A)$. Um **trajecto** de s para t é um sucessão de vértices e arestas, $[s = n_1, (n_1, n_2), n_2, \dots, (n_{k-1}, n_k), n_k = t]$, tal que :

$$\text{a) } n_i \in N, \forall i \in \{1, \dots, k\},$$

$$\text{b) } (n_i, n_{i+1}) \in A, \forall i \in \{1, \dots, k-1\}.$$

Alternativamente, um trajecto pode ser representado apenas pela sucessão de vértices ou de arestas. A notação que vai ser seguida é a sucessão de vértices; ou seja, $[s = n_1, n_2, \dots, n_k = t]$.

Um **caminho** de s para t é um trajecto de s para t sem vértices repetidos. Um **ciclo** é um trajecto de s para s . Um **ciclo simples** é um caminho de s para s . Um **circuito (ciclo dirigido)** é um ciclo formado por arestas dirigidas. Um grafo dirigido sem ciclos diz-se **acíclico**. Um grafo diz-se **ligado (conexo)** se existir um trajecto entre quaisquer dois vértices.

Uma **árvore** é um grafo com um e um só caminho entre quaisquer dois vértices (ou seja, é um grafo conexo sem ciclos). Uma **árvore geradora (abrangente ou total)** de um grafo G é uma árvore formada por arestas de G e que contenha todos os vértices de G .

3. Conceitos fundamentais de redes

Uma **rede** é um grafo cujos vértices e/ou arestas têm associado valores numéricos (custos, capacidades e/ou oferta e procura). A terminologia utilizada em redes é **nós (nodos)** e **arcos**, em vez de vértices e arestas, respectivamente.

Uma rede pode ser representada por $G = (N, A, C)$, em que (N, A) é um grafo e C é o conjunto dos valores numéricos associados aos arcos (*comprimentos* dos arcos); ou seja, ao arco (i, j) está associado o valor c_{ij} (o *comprimento* do arco (i, j) é de c_{ij}). De uma maneira geral, os conceitos utilizados para grafos são extensíveis a redes.

Considere-se um caminho p de s para t na rede G . O *comprimento* do caminho p , $c(p)$, é a soma dos *comprimentos* dos arcos que pertencem àquele caminho; ou seja,

$$c(p) = \sum_{(i,j) \in p} c_{ij}.$$

O conjunto de todos os caminhos de s para t numa rede G identifica-se por P . Ao caminho de menor *comprimento* dá-se o nome de **caminho mais curto**.

Define-se **árvore mínima** (**árvore dos caminhos mais curtos**) com raiz em s , como a árvore que contém todos os nós da rede G acessíveis a partir de s , em que para cada nó r o único caminho de s para r é o caminho mais curto entre aqueles nós.

A **árvore geradora de custo mínimo** é a árvore geradora de G , em que é mínimo o somatório dos custos associados às respectivas arestas.

4. Representação computacional de redes

A eficiência computacional de um algoritmo para resolver problemas de otimização em redes não depende apenas das suas características intrínsecas, mas também, e muito, das estruturas de dados utilizadas para representar a rede no computador (formas de armazenar e manipular os dados associados à rede) e para armazenar os resultados intermédios necessários ao algoritmo. Geralmente, para representar uma rede no computador são necessários dois tipos de informação :

- a topologia da rede (estrutura dos nós e dos arcos),
- os dados (custos, capacidades e oferta/procura) associados aos nós e aos arcos.

Normalmente, o esquema utilizado para armazenar a topologia da rede irá sugerir, naturalmente, uma forma de armazenar a informação associada aos arcos e aos nós.

As representações mais comuns de uma rede são as seguintes : matriz de adjacência, matriz de incidência, listas de adjacência e vectores simulando listas múltiplas.

4.1. Matriz de adjacência

Dada uma rede dirigida $G = (N, A, C)$ com n nós e m arcos, a matriz de adjacência, ou mais exactamente a matriz de adjacência nó-nó, armazena a rede G numa matriz $M = \{ m_{ij} \}$ quadrada de ordem n , em que cada linha e cada coluna corresponde a um nó. Cada elemento m_{ij} da matriz assume um dos seguintes valores : 1 (se $(i, j) \in A$) ou 0 (se $(i, j) \notin A$).

Por outro lado, para cada parâmetro associado aos arcos da rede (custo, comprimento, capacidade, etc.) terá que existir uma matriz do mesmo tipo e para cada parâmetro associado aos nós terá que existir um vector de n elementos.

Nas redes não dirigidas, pode considerar-se que cada arco (i, j) corresponde a dois arcos dirigidos : (i, j) e (j, i) . Neste caso a matriz de adjacência é simétrica.

Nos problemas em que a densidade da respectiva rede é muito baixa (matriz com grande quantidade de elementos nulos), como acontece na maioria dos problemas de circulação, este tipo de estrutura é bastante ineficiente, tanto relativamente ao tempo de pesquisa, como ao espaço de memória necessário ao armazenamento dos dados. Isto porque a matriz de adjacência permite armazenar n^2 arcos e a rede apenas tem m (ou $2m$ caso seja uma rede dirigida).

Apesar disso, a simplicidade desta representação permite a sua utilização para implementar, muito facilmente, os algoritmos de redes. Isto porque pode-se :

- determinar os parâmetros associado a qualquer arco (i, j) , tomando simplesmente o elemento (i, j) das respectivas matrizes;
- obter facilmente os arcos que saem do nó i , examinando a linha i : se o j -ésimo elemento dessa linha tem o valor um (1) então (i, j) é um arco da rede;
- obter os arcos que chegam ao nó j , examinando a coluna j : se o i -ésimo elemento dessa coluna tem o valor um (1) então (i, j) é um arco da rede.

4.2. Matriz de incidência

Dada uma rede dirigida $G = (N, A, C)$ com n nós e m arcos, a matriz de incidência, ou mais exactamente a matriz de incidência nó-arco, armazena a rede G numa matriz constituída por n linhas e m colunas, em que cada linha está associada a um nó e cada coluna a um arco da rede. Cada elemento (i, a) assume um dos seguintes valores : 1 (se o arco a sai do nó i), -1 (se o arco a chega ao nó i) ou 0 (restantes casos).

Nas redes não dirigidas, os valores negativos (-1) são substituídos por positivos ($+1$), uma vez que todo o nó é considerado como *cauda* e como *cabeça* de um arco.

Também esta estrutura se torna bastante ineficiente, tanto relativamente ao tempo de pesquisa, como ao espaço de memória necessário para o armazenamento dos dados, uma vez que a quantidade de elementos reservados para a rede é $n.m$ e esta tem apenas m elementos para serem armazenados (ou $2m$ se for uma rede não dirigida). Em ambos os casos a matriz de incidência apenas tem $2m$ valores diferentes de zero (dos $n.m$ existentes) : cada coluna tem exactamente um $(+1)$ e um (-1) — rede dirigida — ou dois $(+1)$ — rede não dirigida.

Além disso, a quantidade de $(+1)$ numa linha é igual à quantidade de arcos que saem do nó correspondente, qualquer que seja o tipo de rede. Da mesma forma, em redes dirigidas, a quantidade de (-1) numa linha é igual à quantidade de arcos que chegam ao nó

correspondente; conseqüentemente, a soma de (+1) e de (-1), numa linha, corresponde ao grau do nó que lhe está associado.

4.3. Listas de adjacência de arestas

Dada uma rede dirigida $G = (N, A, C)$ com n nós e m arcos, este tipo de representação armazena a lista de adjacência de arestas de cada nó numa simples lista ligada (cada elemento da lista de adjacência é uma lista ligada). Uma lista ligada é um grupo de células, cada uma contendo um ou mais campos. A lista de adjacência do nó i , $A(i)$, será uma lista ligada com $|A(i)|$ células, em que cada célula corresponde a um arco (i, j) da rede.

A célula correspondente ao arco (i, j) terá tantos campos quanto a quantidade de informação que é necessário guardar. Um dos campos guardará o nó j ; depois existirá um conjunto de campos, onde serão guardados, por exemplo, o custo, a capacidade, o comprimento, etc., do arco; por último, terá que existir um outro campo, designado por *ligação* ("link"), que guardará um ponteiro para a próxima célula desta lista ligada (que corresponde a um outro arco que sai do nó i). A última célula da lista ligada conterá no campo *ligação*, por convenção, o valor zero.

Em redes não dirigidas, a informação de cada arco é armazenada em duas células diferentes, uma vez que cada arco pertence a duas listas de adjacência: o arco (i, j) pertence às listas de adjacência do nó i e do nó j — $(i, j) \in A(i)$ e $(i, j) \in A(j)$, pois $(i, j) = (j, i)$.

Uma vez que é necessário armazenar e aceder a n listas ligadas (uma para cada nó), também é necessário uma tabela de ponteiros que apontem para a primeira célula de cada lista ligada. Para tal, define-se uma tabela de dimensão n , designada por *primeiro* ("first"), cujo elemento *primeiro* (i) guarda um ponteiro para a primeira célula da lista de adjacência do nó i , $A(i)$; se $A(i) = \emptyset$ então faz-se *primeiro* (i) = 0.

4.4. Vectores simulando listas múltiplas

Dada uma rede dirigida $G = (N, A, C)$ com n nós e m arcos, nas representações que utilizam vectores para simular listas múltiplas a numeração dos arcos segue uma determinada ordem : primeiro são numerados os arcos que saem do nó 1, depois os arcos que saem do nó 2, etc.. Os arcos que saem do mesmo nó, podem ser numerados por qualquer ordem.

As representações mais simples apenas necessitam de dois vectores de dimensão m :

- **Ant**, em que $\text{Ant}(a) = \text{nó cauda do arco } a$,
- **Suc**, em que $\text{Suc}(a) = \text{nó cabeça do arco } a$.

Para além destes dois vectores, cada um dos parâmetros associados aos arcos da rede necessita de um outro vector, de dimensão m , para armazenar os valores desses parâmetros.

Em redes não dirigidas, tal como acontece para a representação com listas de adjacência de arestas (ver secção 4.3 deste capítulo), a informação associada a cada arco é armazenada duas vezes.

No entanto, estas estruturas ainda podem ser melhoradas através das bem conhecidas representações denominadas por "*forward star form*" e "*reverse star form*".

A "*forward star form*" é uma representação muito semelhante à representação com listas de adjacência de arestas, uma vez que também esta guarda a lista de adjacência de cada nó. No entanto, em vez de listas ligadas são utilizados vectores para guardar a informação associada aos arcos da rede. Assim, determinam-se de forma eficiente, todos os arcos que saem de qualquer nó. Os vectores utilizados são os seguintes :

- **Point**, de dimensão $n+1$, que se define da seguinte forma :

$$\text{Point}(1) = 1$$

$$\text{Point}(i+1) = \text{Point}(i) + \text{quantidade de arcos que saem de } i \quad (1 \leq i \leq n)$$

$$(\text{Point}(n+1) = 1 + m)$$

- **Suc**, em que $\text{Suc}(a) = \text{nó cabeça do arco } a$.

Se $\text{Point}(i) = \text{Point}(i+1)$ então não saem arcos do nó i .

A "*reverse star form*" permite determinar de forma eficiente todos os arcos que chegam a qualquer nó. Os vectores utilizados são os seguintes :

- **RPoint**, de dimensão $n+1$, que se define da seguinte forma :

$$\text{RPoint}(1) = 1$$

$$\text{RPoint}(i+1) = \text{RPoint}(i) + \text{quantidade de arcos que chegam a } i \quad (1 \leq i \leq n)$$

$$(\text{RPoint}(n+1) = 1 + m)$$

- **Ant**, em que $\text{Ant}(a) = \text{nó cauda do arco } a$.

Se $\text{RPoint}(i) = \text{RPoint}(i+1)$ então não chegam arcos ao nó i .

4.5. Comparação entre as várias representações

Como se pode concluir, as representações que utilizam matrizes são muito ineficientes, tornando-se mais adequado a utilização de listas, uma vez que é apenas necessário espaço para representar a informação associada aos arcos que existem, não havendo espaço desperdiçado.

Comparando a representação que utiliza listas de adjacência de arestas com a representação que utiliza vectores para simular listas múltiplas, pode-se concluir que a última necessita de menos espaço para armazenar os dados da rede do que a primeira. No entanto, a primeira é mais eficiente quando se pretende adicionar ou remover arcos (ou nós) à rede.