Umas coisitas de grafos e uma introdução informal à complexidade computacional

J. Orestes Cerdeira

Algoritmia e Programação
Mestrado em Matemática Aplicada às Ciências Biológicas

Inst. Superior de Agronomia, Abril 2001.

1. Introdução

Apresentam-se definições e resultados básicos da teoria dos grafos e complexidade computacional.

Em 2 introduzem-se conceitos e propriedades elementares da teoria dos grafos, com ênfase para a terminologia e metodologia decorrentes da teoria dos emparelhamentos. A teoria dos emparelhamentos ocupa um lugar de relevo na teoria dos grafos. É pois frequente os livros sobre grafos reservarem partes substanciais à exposição desta teoria. No entanto, como referências gerais da teoria dos emparelhamentos citam-se [La], [GM], [NW], [LP] e [Pu]. As três primeiras referências prendem-se principalmente com as questões algorítmicas e as duas últimas recaiem fundamentalmente sobre os aspectos teóricos. Uma interessante perspectiva histórica do desenvolvimento desta teoria está incluída em [LP].

Em 3 apresenta-se uma breve introdução à teoria da complexidade computacional, focando principalmente os aspectos ligados à dificuldadede resolução de problemas. Por simplicidade optou-se por utilizar certos conceitos cujos significados poderão ser eventualmente considerados um tanto imprecisos, em lugar do formalismo subjacente à teoria da computação necessário para atribuir a esses conceitos definições matemáticas rigorosas. Para definições precisas desses conceitos citam-se [AHU] e [GJ], e para exposições gerais mais detalhadas sobre a teoria da complexidade, para além das referências já apontadas, citam-se também [PS] e [JP1].

2. Grafos

Um *grafo (não orientado)* é um par $G = (V, E)$, em que V é um conjunto finito e não vazio e E uma colecção de subconjuntos de cardinalidade 2 de V. Os elementos de V chamam-se *vértices* e os elementos de E as *arestas* do grafo. Utiliza-se n para denotar a cardinalidade de V. Note-se que com esta definição de grafo impede-se a ocorrência de *lacetes*, i.e., arestas do tipo $[v, v]$, mas permite-se a repetição de elementos em E. No entanto, não se dizendo nada em contrário, poder-se-á considerar que tal não acontece e portanto falar-se de E como sendo o conjunto das arestas do grafo. Quando houver necessidade de considerar arestas repetidas, explicita-se este facto dizendo que G é um *multigrafo*, embora por comodidade se continue a referir a E como sendo o conjunto das arestas do grafo. Se $e = [v, u]$ é uma aresta do grafo G, diz-se que e *cobre* ou é *incidente* em v e u, que v e u são os vértices *extremos* de e, e que v e u são *adjacentes*. Dado um vértice v, chama-se *grau de incidência* de v, e representa-se por $d_v(E)$, ao número de arestas incidentes em v. Um *caminho* no grafo G é uma sequência de arestas do tipo

$$P = ([v_{j_1}, v_{j_2}], [v_{j_2}, v_{j_3}], ..., [v_{j_{t-1}}, v_{j_t}]).$$

Os vértices v_{j_1} e v_{j_t} são os vértices *extremos* do caminho P. Diz-se que P *liga* v_{j_1} e v_{j_t}. Se todas as arestas do caminho são distintas, diz-se que o caminho é *simples*. Se cada vértice não ocorre mais do que uma vez, o caminho diz-se *elementar*. O *comprimento* de um caminho é o número de ocorrências de arestas

no caminho. Dados dois vértices v, u incluídos no caminho P, a *distância* de v a u, relativamente a P, é o comprimento do subcaminho de P que liga v a u. Um *ciclo* é um caminho elementar com uma aresta incidente nos vértices extremos do caminho. O *comprimento* de um ciclo é o número de arestas do ciclo. Um grafo sem ciclos diz-se *acíclico*. Um ciclo é *hamiltoniano* se incluir todos os vértices do grafo. Um grafo é *hamiltoniano* se contiver um ciclo hamiltoniano. Um grafo $G = (V, E)$ diz-se *completo* se todo o par de vértices for adjacente. Num grafo completo $|E| = n(n-1)/2$. Um grafo diz-se *bipartido* se existir uma partição de V nos conjuntos V_1, V_2, de tal modo que todas as arestas de G têm um vértice extremo em V_1 e outro em V_2. Os conjuntos V_1, V_2 chamam-se as *classes de bipartição* de G. O resultado que em seguida se enuncia é uma caracterização bem conhecida de grafos bipartidos (a demonstração pode ser vista por exemplo em [Ha] ou [Ch1]).

proposição 1 Um grafo é bipartido sse não contem ciclos de comprimento ímpar.

Se G e G' são dois grafos, e os vértices e arestas de G' são também vértices e arestas de G, então diz-se que G' é um *subgrafo* de G. Dado um subconjunto V' de vértices do grafo G, representa-se por $E(V')$ o conjunto de todas as arestas de G que têm ambos os vértices extremos em V'. O subgrafo de G, $(V', E(V'))$, chama-se *subgrafo induzido* por V'. Se E' é um subconjunto de arestas do grafo G, $V(E')$ denota o conjunto dos vértices cobertos pelas arestas de E'. Um subgrafo de G, cujo conjunto de arestas é E', diz-se subgrafo de *suporte* de G, se $V(E') = V$. Um conjunto A diz-se *maximal* relativamente a uma certa propriedade, se a propriedade for válida para A, mas não for válida para todo o conjunto diferente de A que contenha A. Um grafo é *conexo* se existir um caminho a ligar qualquer par de vértices. Uma *componente conexa* de um grafo é um subgrafo conexo maximal. Uma *floresta* é um grafo acíclico. Uma *árvore* é uma floresta conexa. Facilmente se verifica (ver por exemplo [Ha]) que

proposição 2 Dado um grafo G, as seguintes afirmações são equivalentes:

- **a)** G é uma árvore.
- **b)** Existe um único caminho a ligar qualquer par de vértices de G.
- **c)** G é conexo e tem $n-1$ arestas.
- **d)** G é acíclico e tem $n-1$ arestas.
- **e)** G é acíclico e se se ampliar o conjunto de arestas de G com uma aresta que ligue qualquer par de vértices não adjacentes em G, o grafo resultante contem um único ciclo.

Dado um grafo G, uma floresta que seja subgrafo de G é uma *floresta de* G. Uma árvore que seja subgrafo de G diz-se uma *árvore de* G. Por vezes chama-se a um subconjunto T de arestas de G uma floresta ou uma árvore de G. Quer-se com isto dizer que (V, T) é uma floresta de G, ou que $(V(T), T)$ é uma árvore de G. Assim, T diz-se uma *k-floresta* de G se é uma floresta de G com exactamente

k arestas. T é uma *árvore de suporte* de G se T é uma árvore de G e $V = V(T)$. Uma *cobertura* dos vértices do grafo G é um conjunto T de arestas de G tal que $V = V(T)$. Note-se que uma cobertura dos vértices de um grafo por uma árvore, é uma árvore de suporte do grafo e que uma cobertura por um ciclo é um ciclo hamiltoniano. Um *emparelhamento* de um grafo G é um conjunto de arestas de G tais que cada vértice não é coberto por mais do que uma aresta. Um emparelhamento de G é *máximo* se for de máxima cardinalidade. O *número de emparelhamento* de G, usualmente representado por $\nu(G)$, é a cardinalidade de um emparelhamento máximo de G. Um emparelhamento é *perfeito* se cobre a totalidade dos vértices de G. Uma cobertura dos vértices de G diz-se *mínima* se for a cobertura de cardinalidade mínima. O *número de cobertura* de G, denotado por $\rho(G)$, é a cardinalidade de uma cobertura mínima de G. Os números de emparelhamento e de cobertura de um grafo estão relacionados pela seguinte expressão, a que Lováz e Plummer [LP] chamam uma das identidades de Gallai:

proposição 3 Se G não tem vértices isolados (i.e., vértices com grau de incidência 0), então $\nu(G) + \rho(G) = n$.

Dado um emparelhamento M de G, um *caminho alternado* relativamente a M é um caminho elementar do grafo G, em que as arestas pertencem alternadamente a $E - M$ e M. Um caminho alternado diz-se de *aumento* se ligar vértices não cobertos por M. Note-se que se AP é um caminho alternado de aumento relativamente ao emparelhamento M, $M \oplus AP$, em que $\oplus$ denota a diferença simétrica de conjuntos, é um emparelhamento de G de cardinalidade $|M| + 1$. O teorema que se enuncia em seguida, provado independentemente por Berge [Be] e Norman e Rabin [NR], é uma caracterização de emparelhamento máximo de um grafo.

teorema 1 Um emparelhamento M de G é máximo sse não existirem em G caminhos alternados de aumento relativamente a M.

Este resultado foi transformado por Edmonds [Ed1] num algoritmo polinomial para a determinação do emparelhamento máximo de um grafo. A característica essencial do algoritmo é a de contrair determinados conjuntos de vértices de cardinalidade ímpar. Mais precisamente, o algoritmo explora o seguinte resultado [Ed1]:

proposição 4 Seja M um emparelhamento do grafo G e C um ciclo de cardinalidade $2p + 1$, tal que $|M \cap C| = p$ e $|V(M) \cap V(C)| = 2p$. Seja G' o grafo resultante de G pela contracção de $V(C)$ (i.e., da substituição de $V(C)$ por um único vértice). $M - (M \cap C)$ é um emparelhamento máximo de G' sse M é um emparelhamento máximo de G.

A maioria dos algoritmos mais eficientes para a determinação do emparelhamento máximo, como por exemplo os algoritmos de Lawler [La], Gabow [Ga] e Even e Kariv [EK], utililizam o resultado anterior. O algoritmo de Edmonds [Ed1] é de complexidade $O(n^4)$ e os de Lawler [La] e Gabow [Ga] são ambos de complexidade $O(n^3)$. O algoritmo de Even e Kariv [EK], que generaliza

ao caso de grafos não bipartidos o algoritmo de Hopcroft e Karp [HK], tem complexidade $O(n^{2.5})$.

Considere-se agora um vector c, de componentes reais, definido sobre o conjunto das arestas do grafo G, i.e., uma função $c : E \to \mathbb{R}$. Este vector, que poderá ser utilizado com diferentes interpretações, é designado genericamente por *vector de custos*. A soma dos custos dos elementos de um subconjunto T qualquer de arestas de G, é o *custo* de T e representa-se por $c(T)$. Com um tal vector c definido sobre as arestas de G, podem-se agora estabelecer problemas de optimização combinatória mais gerais do que os problemas respeitantes a subconjuntos de arestas de G. Suponha-se por exemplo que se pretende determinar o emparelhamento máximo de um grafo. Facilmente se verifica que este é um caso particular do problema de optimização que consiste em determinar o emparelhamento de custo máximo de um grafo, com custos definidos sobre asarestas. De facto, dado um grafo G, construa-se um grafo completo G', com o mesmo conjunto de vértices de G, e atribua-se a cada aresta deste novo grafo o custo 1 ou 0, consoante a aresta esteja ou não em G. É óbvio que o emparelhamento máximo de G é o conjunto de arestas de G que ocorrem no emparelhamento de custo máximo de G'. Exemplos de problemas clássicos de optimização combinatória formulados em termos de grafos com custos definidos sobre as arestas são: a determinação da árvore de suporte, do emparelhamento perfeito e do ciclo hamiltoniano de custos mínimos. O primeiro destes problemas é um dos mais fáceis de optimização combinatória e existem algoritmos de resolução de complexidade $O(n^2)$ [Pr], [Di]. Para o segundo problema, que é bastante mais envolvente que o anterior, foi Edmonds [Ed2] quem primeiro estabeleceu um algoritmo polinomial. Utilizando a teoria da dualidade em programação linear e o seguinte poliedro

$$\begin{aligned}\mathcal{P}(M) = \{x \in \mathbb{R}^{|E|} : \\ \sum_{e \in E_v} x_e = 1 \qquad & \forall v \in V \\ \sum_{e \in E(S)} x_e \leq \frac{|S|-1}{2} \qquad & \forall S \subseteq V,\ |S| \geq 3 \text{ ímpar} \\ x_e \geq 0 \qquad & \forall e \in E\},\end{aligned}$$

em que E_v representa o conjunto das arestas incidentes em v, que obviamente inclui os vectores 0-1 de incidência associados aos emparelhamentos perfeitos do grafo $G = (V, E)$, Edmonds [Ed2] apresentou um algoritmo de complexidade $O(n^4)$ para o emparelhamento perfeito de custo mínimo, que prova que

teorema 2 Os vértices do poliedro $\mathcal{P}(M)$ são exactamente os vectores 0-1 de incidência associados aos emparelhamentos perfeitos do grafo G.

Posteriormente foram desenvolvidos algoritmos mais eficientes para a determinação do emparelhamento perfeito de custo mínimo. Por exemplo os algoritmos de Lawler [La] e de Cunningham e Marsh [CM], de complexidade $O(n^3)$, e de Ball e Derigs [BD], de complexidade $O(n|E|logn)$. Para o último daqueles

três problemas combinatórios, usualmente designado por problema do caixeiro viajante, não são conhecidos algoritmos polinomiais e o facto de se tratar de um problema *NP-hard* é normalmente assumido como uma constatação da impossibilidade de tais algoritmos se virem a estabelecer. Mas é precisamente este tipo de questões que se vai considerar de seguida.

3. Complexidade Computacional

Seja π uma propriedade definida sobre um conjunto O_π de objectos matemáticos de representação finita e suponha-se que os elementos de O_π estão representados, de forma *sucinta*, como sequências de símbolos extraídos de um certo alfabeto finito Σ com mais do que um símbolo, por exemplo o alfabeto $\{0,1\}$. Os elementos de O_π chamam-se as *ocorrências* de π. Dizer que as ocorrências estão representadas de forma sucinta significa que se utilizou um esquema de representação para O_π, tal que a representação de cada ocorrência não inclui "excessiva" informação irrelevante. Uma vez as ocorrências assim representadas, tem-se definida uma função $|\ |_\Sigma : O_\pi \to \mathbb{Z}^+$, que a cada ocorrência o faz corresponder o número $|o|_\Sigma$ de ocorrências de símbolos do alfabeto Σ na representação de o.

Uma função $|\ | : O_\pi \to \mathbb{Z}^+$ chama-se *comprimento* ou *tamanho* se estiver *polinomialmente relacionada* com a função $|\ |_\Sigma$, i.e., se existirem dois polinómios p e p' tais que, $\forall o \in O_\pi$, $|o| \leq p(|o|_\Sigma)$ e $|o|_\Sigma \leq p'(|o|)$.

Suponha-se por exemplo que as ocorrências são grafos. Um grafo $G = (V, E)$ pode ser representado de forma sucinta, por exemplo a partir da matriz de adjacência, por $|V|^2 + |V|$ ocorrências de símbolos do alfabeto $\{0, 1, \&\}$, em que se utiliza o símbolo & para assinalar o fim de cada linha da matriz. Assim pode-se dizer que o tamanho de um grafo é $|V|$, o número de vértices. Dado que, não havendo vértices isolados, $|V|/2 \leq |E| \leq |V|(|V| - 1)/2$, pode-se igualmente definir tamanho de um grafo como sendo $|E|$, o número de arestas do grafo. Se as ocorrências fossem números inteiros positivos, poder-se-ia dizer que o comprimento de uma ocorrência n é $logn$, pois numa representação sucinta de um inteiro n, ocorrem $\lceil logn \rceil$ símbolos do alfabeto $\{0, 1\}$, em que $\lceil x \rceil$ denota o menor inteiro que é maior ou igual do que x.

Uma propriedade π define uma partição do conjunto O_π nos conjuntos SIM_π e $\mathrm{N\tilde{A}O}_\pi$. SIM_π é o conjunto das ocorrências *afirmativas* de π, i.e., dos elementos de O_π para os quais a propriedade π é válida e $\mathrm{N\tilde{A}O}_\pi = O_\pi - \mathrm{SIM}_\pi$ é o conjunto das ocorrências *negativas* de π. O problema associado a π, representado por P_π, é o problema de decidir sobre π, i.e., dado uma ocorrência qualquer $o \in O_\pi$, decidir correctamente se $o \in \mathrm{SIM}_\pi$ ou $o \in \mathrm{N\tilde{A}O}_\pi$. Aqui consideram-se apenas propriedades π para as quais existem algoritmos que resolvem P_π.

Seja π uma propriedade e $\mathcal{A}$ um algoritmo para P_π. Para cada ocorrência $o \in O_\pi$, $\mathcal{A}$ responde correctamente "sim" ou "não", consoante $o \in \mathrm{SIM}_\pi$ ou $o \in \mathrm{N\tilde{A}O}_\pi$, ao fim de $N_\mathcal{A}(o)$ *operações elementares*. É usual assumir que cada operação elementar é realizada numa unidade de tempo. Assim diz-se que, com a ocorrência o, $\mathcal{A}$ responde "sim" ou "não" em *tempo* $N_\mathcal{A}(o)$, exprimindo o facto de $\mathcal{A}$ realizar $N_\mathcal{A}(o)$ operações elementares até emitir uma resposta. Define-se

complexidade de um algoritmo $\mathcal{A}$ como sendo uma função $\mathcal{C}_{\mathcal{A}} : \mathbb{Z}^+ \rightarrow \mathbb{Z}^+$, em que $\mathcal{C}_{\mathcal{A}}(n)$ é o número de operações que $\mathcal{A}$ realiza com a "pior" ocorrência de comprimento n, i.e., $\mathcal{C}_{\mathcal{A}}(n) = max\{N_{\mathcal{A}}(o) : o \in O_\pi \text{ e } |o| = n\}$. Dadas duas funções $f, g : \mathbb{Z}^+ \rightarrow \mathbb{R}^+$, $f(n)$ diz-se $O(g(n))$ e escreve-se $f(n) = O(g(n))$, se existir uma constante $\lambda > 0$ tal que, para valores de n a partir de uma certa ordem, $f(n) \leq \lambda\ g(n)$. Um algoritmo $\mathcal{A}$ é *polinomial* se existir um polinómio p tal que $\mathcal{C}_{\mathcal{A}}(n) = O(p(n))$. Um algoritmo não polinomial diz-se *exponencial*. Chama-se P ao conjunto dos problemas de decidir sobre propriedades, para os quais existem algoritmos polinomiais de resolução. Edmonds [Ed1] propôs a utilização do atributo "bom" para os algoritmos polinomiais, o que de alguma forma vem ao encontro do conceito de valor que do ponto de vista prático se faz relativamente à aplicabilidade dos algoritmos. De facto, a utilização de algoritmos exponenciais, que mais não são do que versões de métodos de pesquisa exaustiva, é normalmente limitada a ocorrências de comprimentos reduzidos. Para certas ocorrências, mesmo de comprimentos razoáveis, a utilização destes algoritmos é totalmente impraticável. Daí ser costume chamar intratáveis os problemas que apenas admitem algoritmos exponenciais. Por outro lado, é comum, ao estabelecer um algoritmo polinomial para a resolução de um problema, dizer-se que o problema fica "bem resolvido". Se é certo ser duvidosa a utilização prática de algoritmos, que apesar de polinomiais, tenham complexidades do tipo, digamos n^{100}, constata-se que, de uma forma geral, os graus dos polinómios que majoram as complexidades dos algoritmos polinomiais estabelecidos para a resolução de problemas relevantes não são superiores a 5 ou 6, sendo praticáveis para a maior parte das ocorrências com que normalmente se depara em situações reais. Utilizando esta terminologia, pode-se pois definir P como sendo o conjunto dos problemas de decidir sobre propriedades, que admitem "bons" algoritmos.

Uma propriedade π diz-se *NP*, se existir um polinómio p e um algoritmo $\mathcal{A}$ tal que para toda a ocorrência $o \in O_\pi$, $o \in \text{SIM}_\pi$ sse existir um objecto matemático $c(o)$ (o *certificado*) tal que com o *input* $(o, c(o))$ o algoritmo $\mathcal{A}$ responde "sim" em tempo não superior a $p(|o|)$. Note-se que desta definição não decorre, pelo menos directamente, a existência de algoritmos polinomiais para os problemas de decidir sobre propriedades *NP*. A definição apenas estabelece que toda a ocorrência afirmativa pode ser certificada por um algoritmo polinomial, no comprimento da ocorrência, desde que uma determinada informação adicional (o certificado da ocorrência), cuja existência é assegurada, seja fornecida. Note-se que, uma vez que $p(|o|)$ é um majorante do número de operações que o algoritmo realiza até se certificar de que a ocorrência o é de facto afirmativa, obviamente o número de símbolos utilizados na representação do certificado $c(o)$ não poderá ser maior do que $p(|o|)$. Daí ser usual dizer-se que as propriedades *NP* são sucintamente certificáveis. Muitas propriedades interessantes são propriedades *NP*. Por exemplo ser hamiltoniano ou ter um emparelhamento perfeito são propriedades *NP*, definidas sobre o conjunto dos grafos. Dado um grafo $G = (V, E)$ hamiltoniano (ou que contenha um emparelhamento perfeito), um ciclo hamiltoniano C (um emparelhamento perfeito M) do grafo constitui um certificado sucinto do facto de G ser hamiltoniano (conter um emparelhamento perfeito). É possível

estabelecer um algoritmo que, a partir de G e C (M), responda "sim" após se certificar de que C (M) é de facto um ciclo hamiltoniano (um emparelhamento perfeito) de G. Um tal algoritmo poderá ser obviamente definido de forma a que não mais do que $O(|E|)$ operações sejam realizadas até emitir a resposta "sim". Dado que o tamanho do grafo G é $|E|$, com G e C (M) o algoritmo responde "sim" em tempo polinomial no tamanho da ocorrência.

Há propriedades *NP* que negadas são também propriedades *NP*. Por exemplo a não existência de um emparelhamento perfeito é também uma propriedade *NP*. De facto, o teorema de Tutte [Tu1,Tu2] ao estabelecer que um grafo $G = (V, E)$ tem um emparelhamento perfeito sse $\mathcal{C}_i(G - S) \leq |S|$, $\forall S \subset V$, em que $\mathcal{C}_i(G - S)$ denota o número de componentes conexas ímpares (com número ímpar de vértices) do subgrafo de G induzido por $V - S$, mostra que a não existência de emparelhamento perfeito é também uma propriedade *NP*. Se um grafo $G = (V, E)$ não contem um emparelhamento perfeito, então vai existir um subconjunto de vértices S, tal que $\mathcal{C}_i(G-S) > |S|$. Pode-se portanto estabelecer um algoritmo que, a partir de G e S, responda "sim" após se certificar de que S é de facto um subconjunto dos vértices de G e de que $\mathcal{C}_i(G - S) > |S|$. Um tal algoritmo poderá obviamente ser definido de forma a que a resposta "sim" seja emitida ao fim de não mais do que $O(|E|)$ operações terem sido realizadas.

Uma propriedade *NP* cuja negação é também propriedade *NP* diz-se *bem caracterizada*. Um teorema estabelecendo a equivalência de uma propriedade *NP* com a negação de outra propriedade *NP* diz-se uma *boa caracterização*. O teorema de Tutte é pois uma boa caracterização da existência de emparelhamentos perfeitos. Uma boa caracterização é genericamente enunciada do seguinte modo: $o \in O_\pi$ verifica π sse $o' \in O_{\pi'}$ não verifica π', sendo π e π' ambas propriedades *NP*.

Note-se que, dado que π é propriedade *NP*, toda a ocorrência afirmativa de π é sucintamente certificável. Por outro lado, uma vez que qualquer ocorrência negativa o de π corresponde a uma ocorrência afirmativa o' de π' e, como π' é propriedade *NP*, é portanto também sucintamente certificável.

Há no entanto um grande número de propriedades *NP* que têm resistido ao estabelecimento de boas caracterizações. Há mesmo um consenso generalizado sobre a existência de propriedades *NP* que não admitem boas caracterizações, i.e., que quando negadas deixam de ser *NP*. Por exemplo não é de forma alguma evidente que, dado um grafo G não hamiltoniano, exista um certificado tal que, com G e esse certificado, se possa vir a estabelecer um algoritmo que, em tempo polinomial no tamanho de G, se certifique de que G não é hamiltoniano. De facto, é amplamente conjecturado que não ser hamiltoniano não é propriedade *NP*. Chama-se *co-NP* às propriedadesque resultam da negação de propriedades *NP*. Utilizando*NP* e *co-NP* para denotar os conjuntos de problemas de decidir sobre propriedades *NP* e *co-NP*, respectivamente, a conjectura estabelece pois que *NP*$\neq$*co-NP*.

Há alguns resultados óbvios que relacionam *P*, *NP* e *co-NP*. Por exemplo, uma forma polinomial de decidir correctamente "sim" ou "não" relativamente a uma propriedade π, é simultaneamente uma forma polinomial de certificar, para toda a ocorrência afirmativa de π, que de facto é afirmativa e, para toda a

ocorrência negativa, de que de facto é negativa. Por outras palavras,

proposição 5 $P \subseteq NP \cap co\text{-}NP$.

Por outro lado, tendo em conta que ao decidir polinomialmente sobre uma propriedade NP está-se também a decidir polinomialmente sobre a negação da propriedade, tem-se que

proposição 6 Se $NP \neq co\text{-}NP$, então $P \neq NP$,

i.e., é pelo menos tão seguro trabalhar sobre a conjectura $P \neq NP$, do que sobre $NP \neq co\text{-}NP$.

Admitindo que a conjectura $P \neq NP$ é válida, em NP figuram simultaneamente problemas intrinsecamente "difíceis", os que não admitem algoritmos polinomiais, e problemas "fáceis", os que pertencem a P e para os quais existem "bons" algoritmos. Pode-se pensar estabelecer em NP uma hierarquia relativamente à dificuldade de resolução dos problemas. Por outras palavras, dados dois problemas P_π e $P_{\pi'}$ de NP, atribuir significado a expressões do tipo: "P_π é tão "difícil" quanto $P_{\pi'}$", ou "$P_{\pi'}$ é pelo menos tão "difícil" quanto P_π", ou ainda "P_π é mais "fácil" do que P_π". Neste sentido, torna-se útil a noção de transformação polinomial entre problemas.

Sejam π e π' duas propriedades e P_π, $P_{\pi'}$ os respectivos problemas de decidir sobre elas. Diz-se que P_π é *polinomialmente transformável* em $P_{\pi'}$, e escreve-se $P_\pi \propto P_{\pi'}$, se existir uma função $f : O_\pi \to O_{\pi'}$ tal que

a) $\forall o \in O_\pi$, $f(o)$ obtem-se em tempo polinomial no comprimento de o;

b) $\forall o \in O_\pi$, $o \in \mathrm{SIM}_\pi$ sse $f(o) \in \mathrm{SIM}_{\pi'}$.

Note-se que se $P_\pi \propto P_{\pi'}$, então qualquer algoritmo para $P_{\pi'}$ poderá ser utilizado, com adicionalmente um número polinomial de operações (o tempo de converter o em $f(o)$) no tamanho da ocorrência o de π, para resolver o problema P_π. Traduz-se este facto dizendo que $P_{\pi'}$ é pelo menos tão "difícil" quanto P_π.

Considerem-se, por exemplo, as propriedades existência de emparelhamento perfeito em grafos (EEP) e realização de expressões booleanas na forma conjuntiva normal (REB). Uma expressão booleana na forma conjuntiva normal é um agrupamento de cláusulas ligadas pelo operador lógico conjunção. Uma cláusula é uma expressão que envolve variáveis booleanas e os operadores disjunção e negação, sendo desnecessário o uso de parênteses. A expressão diz-se realizável se existir uma afectação de valores verdade-falso sobre as variáveis que a torne verdadeira.

Pode-se facilmente verificar que $P_{EEP} \propto P_{REB}$. Dado um grafo qualquer $G = (V, E)$, associe-se a cada aresta $[v, u]$ de G a variável booleana $x_{[v,u]}$ e interprete-se o valor verdade afecto à variável $x_{[v,u]}$ como sendo a escolha da aresta $[v, u]$. Para cada vértice v de G definam-se as seguintes cláusulas:

$$C_0 = (x_{[v,v_1]} \vee x_{[v,v_2]} \vee ... \vee x_{[v,v_t]});$$

$$(\overline{x}_{[v,v_i]} \vee \overline{x}_{[v,v_j]}), \text{com } i := 1, ..., t-1, \text{ e } j := i+1, ..., t,$$

em que $v_1, ..., v_t$ são os vértices adjacentes a v, $\vee$ representa o operador disjunção e $\overline{x}$ a negação de x.

A cláusula C_0 assume o valor verdade sse pelo menos uma das arestas incidentes em v for escolhida. A expressão resultante de ligar pela conjunção as $t(t-1)/2$ cláusulas restantes, será realizável sse não for escolhida mais do que uma aresta incidente em v. Assim, a conjunção desta expressão com a cláusula C_0 é realizável sse se escolher uma e uma só aresta incidente em v. Note-se que a construção desta expressão poderá ser realizada em tempo $O(t^2)$, que é obviamente menor ou igual do que $O(|E|^2)$. Portanto, procedendo de forma análoga relativamente a cada um dos vértices de G e ligando as expressões resultantes pelo operador conjunção obtem-se, em tempo $O(|V||E|^2)$, uma ocorrência de REB que será afirmativa sse o grafo G contiver um emparelhamento perfeito. Pode-se pois concluir que decidir se uma dada expressão booleana é realizável é pelo menos tão "difícil" quanto decidir da existência de emparelhamentos perfeitos em grafos.

Já o recíproco deste resultado parece não ser válido, i.e., $P_{REB} \not\propto P_{EEP}$ (se $P \neq NP$). Dois problemas P_π e $P_{\pi'}$ são *polinomialmente equivalentes* se $P_\pi \propto P_{\pi'}$ e $P_{\pi'} \propto P_\pi$. Neste caso diz-se que P_π é tão "difícil" quanto $P_{\pi'}$. A relação $\propto$ vai fragmentar NP em classes de problemas igualmente "difíceis". Coloca-se assim a questão de saber se existe uma majoração para o grau de "dificuldade" de resolução dos problemas em NP. Por outras palavras, definindo *NP-complete* como sendo a classe dos problemas mais "difíceis" de NP, i.e., *NP-complete* $= \{P_\pi \in NP : \forall P_{\pi'} \in NP,\ P_{\pi'} \propto P_\pi\}$, será que *NP-complete* $\neq \emptyset$? A resposta pela afirmativa a esta questão foi dada por Cook [Co] ao estabelecer que

teorema 3 O problema de decidir sobre a realização de expressões boleanas na forma conjuntiva normal é *NP-complete*.

O teorema 3 permite concluir que, se π é uma propriedade NP e P_{REB} (ou qualquer outro problema *NP-complete*) $\propto P_\pi$, então P_π é também um problema *NP-complete*. Assim, ao pretender provar que um dado problema P_π associado a uma propriedade NP é *NP-complete*, bastará seleccionar um problema *NP-complete* "adequado", e mostrar que não é mais "difícil" do que P_π. Procedendo deste modo, tem-se hoje classificados de *NP-complete* várias centenas de problemas relevantes de áreas diversas como álgebra, teoria dos números, lógica, teoria da representação, bases de dados, programação matemática e muitas outras. O livro de Garey e Johnson [GJ] contem uma lista de cerca de 300 problemas conhecidos que se sabe serem *NP-complete*. Esta lista de problemas continua a ser regularmente ampliada na secção *The NP-Completeness Column: an Ongoing Guide* de D. S. Johnson, incluída em *Journal of Algorithms*.

Das demonstrações que têm sido apresentadas para a inclusão de novos problemas na classe *NP-complete*, algumas são razoavelmente simples:

exemplo 1 ([Ka]): P_{ESI} é o problema de decidir da existência de soluções inteiras de um sistema de desigualdades $Ax \geq b$, em que A é uma matriz $m \times n$ de elementos inteiros e b um vector com m componentes inteiras.

ESI é uma propriedade *NP* (ver por exemplo [PS]). Vai-se mostrar que $P_{REB} \propto P_{ESI}$. Seja $E = C_1 \wedge C_2 \wedge ... \wedge C_m$ uma expressão booleana qualquer na forma conjuntiva normal, em que $C_i, i = 1, ..., m$ são as m cláusulas e $\wedge$ denota o operador conjunção. Para cada cláusula C_i, construa-se uma desigualdade do tipo $\geq$, cujo membro esquerdo é obtido ao substituir em C_i as disjunções pelo operador $+$ e as variáveis x que ocorrem negadas por $1 - x$, e coloque-se o valor 1 no membro direito. Após se ter assim procedido relativamente a cada uma das m cláusulas, obtem-se um sistema de desigualdades lineares $Cx \geq b$, que constitui uma ocorrência de ESI e que admite solução 0-1 sse a expressão for realizável. Bastará pois incluir no sistema $Cx \geq b$, para cada uma das n variáveis, as inequações $x_j \geq 0$ e $-x_j \geq -1, j = 1, ..., n$, obrigando as soluções inteiras a serem 0-1. Obviamente a construção deste sistema de desigualdades pode ser obtido em tempo polinomial no tamanho da expressão dada. Pode-se pois concluir que P_{ESI} é *NP-complete*.

Outras demonstrações são mais complicadas:

exemplo 2 ([GCV]): P_{ECC} é o problema de decidir, dado um grafo $G = (V, E)$ e dois vértices v e u de G, da existência em G de um caminho elementar T ligando v e u tal que o grafo $G' = (V, E - T)$ seja conexo.

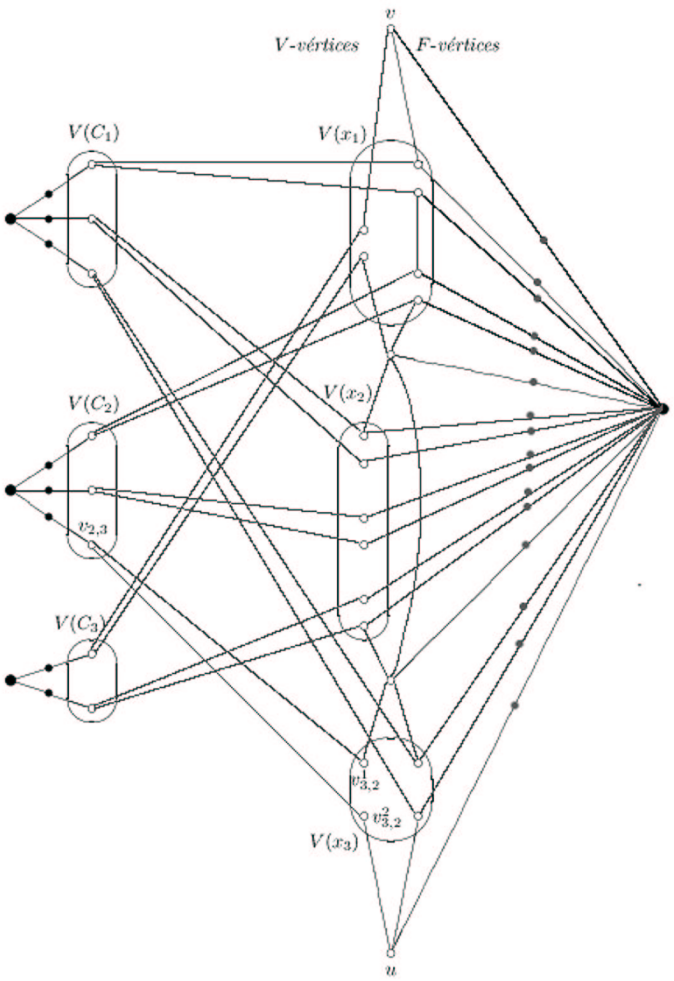
É óbvio que P_{ECC} é problema *NP*. Como no exemplo anterior, mostra-se aqui também que $P_{REB} \propto P_{ECC}$. Seja então E uma ocorrência qualquer de REB, constituída pelas m cláusulas $C_i, i = 1, ..., m$, e $X = \{x_1, x_2, ..., x_n\}$ o conjunto das variáveis intervenientes. A partir de E vai-se construir um grafo $G = (V, E)$ com uma determinada estrutura. Veja-se o exemplo da figura 1, onde se define o grafo correspondente à expressão booleana

$$E = (x_1 \vee \overline{x}_2 \vee x_3) \wedge (x_1 \vee \overline{x}_2 \vee \overline{x}_3) \wedge (\overline{x}_1 \vee \overline{x}_2),$$

em que o conjunto de variáveis é $X = \{x_1, x_2, x_3\}$. Espera-se que o exemplo seja suficientemente elucidativo para que da descrição que se segue fique clara a forma de construir, para qualquer expressão dada E, o grafo correspondente. O conjunto dos vértices de G vai incluir dois vértices que se designam por v e u (que na figura são representados como sendo vértices do tipo $\circ$).

A cada cláusula $C_i, i = 1, ..., m$ associe-se o conjunto de vértices $V(C_i)$ (os vértices $\circ$ que aparecem no lado esquerdo da figura 1), definido do seguinte modo: $v_{i,j} (j = 1, ..., n) \in V(C_i)$ sse a variável x_j ocorre na cláusula C_i. Associe-se também a cada variável $x_j, j = 1, ..., n$ um conjunto de pares de vértices $V(x_j)$ (os vértices $\circ$ no centro da figura 1). O par de vértices $v^1_{j,i}, v^2_{j,i} (i = 1, ..., m) \in V(x_j)$ sse a variável x_j figura na cláusula C_i. Os vértices $v^1_{j,i}$ e $v^2_{j,i}$ vão ser ambos adjacentes ao vértice $v_{i,j} \in V(C_i)$ e chamam-se *V-vértices* se na cláusula C_i a variável x_j ocorrer negada. Caso contrário, chamam-se *F-vértices*. (Na figura 1 os *V-vértices* aparecem do lado esquerdo de u e v, e os *F-vértices* do lado direito). Disponham-se os conjuntos $V(x_j), j = 1, ..., n$ em série, como na figura 1, colocando um vértice extra entre cada dois desses conjuntos. Complete-se agora, de acordo com a figura 1, as ligações entre os vértices já considerados (i.e., os vértices $\circ$ da figura). Note-se que no grafo

assim obtido, qualquer caminho elementar a ligar v e u atravessa cada um dos conjuntos $V(x_j)$ utilizando ou todos os *V-vértices*, ou então todos os *F-vértices*.



Grafo correspondente a
$(x_1 \vee \overline{x}_2 \vee x_3) \wedge (x_1 \vee \overline{x}_2 \vee \overline{x}_3) \wedge (\overline{x}_1 \vee \overline{x}_2)$

fig.1

Tem-se assim definida uma correspondência biunívoca entre as afectações de valores verdade-falso sobre o conjunto de variáveis X e os caminhos elementares que ligam os vértices v e u. Dada uma afectação $t : X \to \{V, F\}$ de valores verdade-falso sobre X, o caminho que lhe corresponde inclui os $t(x_j)$-*vértices* de

$V(x_j), j = 1, ..., n$. Note-se que uma afectação de valores verdade-falso realiza a expressão E sse o caminho que lhe corresponde não inclui a totalidade dos vértices de cada um dos conjuntos $V(C_i)$.

Vai completar-se agora a construção do grafo G de forma a que, dado um caminho T a ligar v e u, ao eliminar T de G o grafo resultante se mantenha conexo sse T for um desses caminhos. Para isso, bastará ampliar o grafo já construído com determinados vértices e arestas como decorre do exemplo da figura 1, em que estes novos vértices são representados por $\bullet$. Com o grafo G assim construído, os caminhos de v para u que quando eliminados deixam o grafo desconexo, ou correspondem a afectações de valores verdade-falso sobre X que não realizam E, ou então incluem pelo menos um dos vértices $\bullet$. Portanto, os caminhos que quando eliminados de G mantêm o grafo conexo correspondem às afectações que realizam a expressão. Note-se, finalmente, que a construção do grafo G pode ser realizada em tempo $O(\sum_{j=1}^{n} N(x_j))$, em que $N(x_j)$ representa o número de ocorrências em E da variável x_j, e portanto polinomialmente no tamanho de E, podendo assim concluir-se que P_{ECC} é *NP-complete*.

Há no entanto outras demonstrações bem mais simples:

exemplo 3 (a demonstração original é de Cook [Co] e utiliza o problema de decidir da existência de *cliques* com uma dada cardinalidade): P_{ESGI} é o problema de decidir, dados dois grafos G e G', se existe um subgrafo de G isomorfo a G', i.e., idêntico à excepção da designação dos vértices. Obviamente que $ESGI$ é propriedade *NP*.

É fácil mostrar, a partir do problema de decidir se um grafo é hamiltoniano P_{GH}, que é sabido ser *NP-complete* [Ka], que $P_{GH} \propto P_{ESGI}$. De facto, P_{GH} é o caso particular do problema P_{ESGI} quando se restringe G' a ser um ciclo hamiltoniano com tantos vértices quantos G. Consequentemente P_{ESGI} é pelo menos tão "difícil" quanto P_{GH} e portanto também *NP-complete*.

De forma análoga se poderia provar que o problema do caixeiro viajante (PCV) dado um grafo $G = (V, E, c)$, em que c um vector de custos definido sobre as arestas de G, determinar o ciclo hamiltoniano de custo mínimo de G) é pelo menos tão "difícil" quanto P_{GH}, o problema de decidir se um grafo é hamiltoniano. De facto, bastará dado um grafo qualquer G, construir um grafo completo G', com o mesmo conjunto de vértices do que G, e atribuir a cada aresta deste novo grafo o custo 0 ou 1, consoante a aresta pertença ou não a G. O grafo G é hamiltoniano sse a solução do PCV respeitante a G' for um ciclo de custo 0.

Note-se no entanto que PCV não é um problema de decidir sobre uma propriedade. A resposta é um ciclo hamiltoniano e não um "sim" ou "não". Por outras palavras, PCV não é um problema *NP*. Problemas como este, que são pelo menos tão "difíceis" quanto os problemas *NP-complete*, independentemente de serem ou não problemas de decidir sobre propriedades, são designados *NP-hard*. Como o PCV, são *NP-hard* muitos relevantes *problemas de optimização combinatória*, i.e., problemas em que cada ocorrência é a representação de um conjunto S de objectos combinatórios, as *soluções admissíveis*, e de uma *função*

de custo $c : S \rightarrow \mathbb{R}$ e em que se pretende obter resposta à questão: qual a solução s^* que max ou $min\{c(s) : s \in S\}$? Ao ser-se confrontado com um problema desta natureza, e atendendo a que se a conjectura $P \neq NP$ for válida não existe forma polinomial de o resolver, torna-se pois importante estabelecer formas "razoáveis" para obter respostas à questão formulada pelo problema. Por forma "razoável" entenda-se um compromisso entre o tempo requerido para a determinação da resposta e qualidade da resposta obtida.

Suponha-se que para um problema *NP-hard* de optimização combinatória POC se dispõe de um algoritmo polinomial $\mathcal{A}$ que, para toda a ocorrência o (cuja a função de custo c é positiva), produz como resposta uma solução admissível $s_{\mathcal{A}}(o)$ que verifica a seguinte desigualdade:

$$\frac{|c(s_{\mathcal{A}}(o)) - c(s^*(o))|}{c(s^*(o))} \leq \alpha,$$

em que $s^*(o)$ é a solução pretendida.

O algoritmo $\mathcal{A}$ é pois uma forma "razoável" de resolver o problema e chama-se algoritmo α-*aproximativo* [PS] para o POC. O estabelecimento de algoritmos α-*aproximativos* para problemas *NP-hard* de optimização combinatória é importante, não só do ponto de vista teórico, mas também prático, sendo esses algoritmos tanto mais interessantes quanto menor as suas complexidades e os valores de α correspondentes. No entanto, se $P \neq NP$, nem sempre existem algoritmos polinomiais α-*aproximativos* para os problemas *NP-hard* de optimização combinatória. Sabe-se por exemplo que

teorema 4 ([SG]) Se $P \neq NP$, qualquer que seja $\alpha > 0$, não existe algoritmo polinomial α-*aproximativo* para o problema do caixeiro viajante.

Este resultado é pois uma indicação de que, se $P \neq NP$, não será pelo estabelecimento de algoritmos α-*aproximativos* que se virá a obter uma forma "razoável" de abordar o PCV. No entanto, este resultado pessimista é invalidado quando se restringe as ocorrências do PCV à classe relevante de ocorrências em que os vectores de custo verificam a desigualdade triangular. Um vector não negativo de custos definido sobre as arestas de um grafo completo verifica a *desigualdade triangular* se, para todo o par de vértices v, u, o caminho de custo mínimo que liga v e u for a aresta $[v, u]$. De facto, sendo válida a desigualdade triangular existem vários algoritmos polinomiais α-*aproximativos* para o PCV (ver por exemplo [RSL], [PS] e [JP2]). O algoritmo de Christofides [Ch2] é um exemplo de algoritmo $\frac{1}{2}$-*aproximativo* de complexidade $O(n^3)$, sendo n é o número de vértices do grafo, para o PCV restrito a vectores de custo que satisfazem a desigualdade triangular. Assim, apesar de resultados pessimistas como o teorema 4 respeitantes à existência de algoritmos polinomiais α-*aproximativos* para problemas de optimização combinatória, é por vezes possível que, ao limitarem-se as ocorrências dos problemas a classes eventualmente relevantes, tais algoritmos venham a existir.

Referências :

[AHU] A. V. AHO, J. E. HOPCROFT e J. D. ULLMAN, *The Design and Analysis of Computer Algorithms*, Addison-Wesley, Reading, Mass., 1974.

[BD] M. O. BALL e U. DERIGS, "An Analysis of Alternate Strategies for Implementing Matching Algorithms", *Networks*, 13, 1983, 517-549.

[Be] C. BERGE, "Two Theorems in Graph Theory", *Proceedings of the National Academy of Science U.S.A.*, 43, 1957, 842-844.

[Ch1] N. CHRISTOFIDES, *Graph Theory: An Algorithmic Approach*, Academic Press, London, 1975.

[Ch2] N. CHRISTOFIDES, "Worst-Case Analysis of a New Heuristic for the Travelling Salesman Problem", *Tecnical Report*, Graduate School of Industrial Administration, Carnegie-Mellon University, 1976.

[CM] W. H. CUNNINGHAM e A. B. MARSH III, "A Primal Algorithm for Optimum Matching", *Mathematical Programming Study*, 8, 1978, 50-72.

[Co] S. A. COOK, "The Complexity of Theorem-Proving Procedures", *Proccedings of the 3rd Annual ACM Symposium on Theory of Computing Machinery*, (A.C.M., New York) 1971, 151-158.

[Di] E. W. DIJKSTRA, "A Note on Two Problems in Connection with Graphs", *Numerische Mathematik*, 1, 1959, 269-271.

[Ed1] J. EDMONDS, "Paths, Trees and Flowers", *Canadian Journal of Mathematics*, 17, 1965, 449-467.

[Ed2] J. EDMONDS, "Maximum Matching and a Polyhedron with (0,1) Vertices", *Journal of Research of the National Bureau of Standards*, 69B, 1965, 125-130.

[EK] S. EVEN e O. KARIV, "An $O(n^{5/2})$ Algorithm for Maximum Matching in General Graphs", *Proceedings of the 16th Annual Symposium on Foundations of Computer Science*, IEEE Computer Society Press,New York, 1975, 100-112.

[Ga] H. N. GABOW, "An Efficient Implementation of Edmonds' Algorithm for Maximum Matching on Graphs", *Journal of the Association for Computing Machinery*, 23, 1976, 221-234.

[GJ] M. R. GAREY e D. S. JOHNSON, *Computers and Intractability: A Guide to the Theory of $\mathcal{NP}$-Completeness*, Freeman, San Francisco, 1979.

[GCV] F. GÖBEL, J. O. CERDEIRA e H. J. VELDMAN, "Label-Connected Graphs and the Gossip Problem", *Discrete Mathematics*, 87, 1991, 29-40.

[GM] M. GONDRAN e M. MINOUX, *Graphs and Algorithms*, Wiley-Interscience, Chichester, 1984.

[Ha] F. HARARY, *Graph Theory*, Addison-Wesley, Reading, Mass., 1969.

[HK] J. E. HOPCROFT e R. M. KARP, "An $n^{5/2}$ Algorithm for Maximum Matchings in Bipartite Graphs", *SIAM Journal on Computing*, 2, 1973, 225-231.

[JP1] D. S. JOHNSON e C. H. PAPADIMITRIOU, "Computational Complexity", in *The Traveling Salesman Problem: A Guide Tour of Combinatorial Optimization*, Eds.: E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, D. B. Shmoys, John Wiley & Sons, New York, 1985, 37-85.

[JP2] D. S. JOHNSON e C. H. PAPADIMITRIOU, "Performance Guarantees for Heuristics", in *The Traveling Salesman Problem: A Guide Tour of Combinatorial Optimization*, Eds.: E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, D. B. Shmoys, John Wiley & Sons, New York, 1985, 145-180.

[Ka] R. M. KARP, "Reducibility among Combinatorial Problems", in *Complexity of Computer Computations*, Eds: R. E. Miller e J. W.Thatcher, Plenum Press, New York, 1972, 85-103.

[La] E. L. LAWLER, *Combinatorial Optimization: Networks and Matroids*, Holt, Rinehart and Winston, New York, 1976.

[LP] L. LOVÁZ e M. D. PLUMMER, *Matching Theory*, Annals of Discrete Mathematics, 29, North-Holland, Amsterdam, 1986.

[NR] R. Z. NORMAN e M. D. RABIN, "An Algorithm for the Minimum Cover of a Graph", *Poceedings of the American Mathematical Society*, 10, 1959, 315-319.

[NW] G. L. NEMHAUSER e L. A. WOLSEY, *Integer and Combinatorial Optimization*, Wiley-Interscience, New York, 1988.

[Pr] R. C. PRIM, "Shortest Connection Networks and Some Generalizations", *Bell System Technological Journal*, 36, 1957, 1389-1401.

[PS] C. H. PAPADIMITRIOU e K. STIEGLITZ, *Combinatorial Optimization: Algorithms and Complexity*, Prentice-Hall, Englewood Cliffs, New York, 1982.

[Pu] W. R. PULLEYBLANK, "Matchings and Extensions", in *Handbook of Combinatorics*, Eds.: R. Graham, M. Grötschel, L.Lóvaz, North-Holland, 1995, 179-232.

[RSL] D. J. ROSENKRANTZ, R. E. STEARNS e P. M. LEWIS, "An Analysis of Several Heuristics for the Traveling Salesman Problem", *SIAM Journal on Computing*, 6, 1977, 563-581.

[SG] S. SAHNI e T. GONZALEZ, "P-Complete Approximation Problems", *Journal of the Association for Computing Machinery*, 23, 1976, 555-565.

[Tu1] W. T. TUTTE, "The Factorization of Linear Graphs", *The Journal of the London Mathematical Society*, 22, 1947, 107-111.

[Tu2] W. T. TUTTE, "The Factors of Graphs", *Canadian Journal of Mathematics*, 4, 1952, 314-328.