Caminhos mínimos em grafos mistos

Orlando Lee e Y. Wakabayashi [1]

Abril 1995

Abstract

O problema de encontrar caminhos mínimos em um grafo (orientado ou não-orientado) tem sido largamente tratado na literatura. Um fato bastante conhecido é que esse problema torna-se intratável quando o grafo possui circuitos negativos. Isso justifica a importância de se considerar o problema de detectar circuitos negativos e encontrar caminhos mínimos em sua ausência. Uma grande variedade de algoritmos polinomiais foi desenvolvida para resolver esses problemas em suas versões orientadas ([Be58], [Fo56], [Fl62], [Wa62], [Ye70]). No caso não-orientado, Tobin ([To73], [To75]) mostrou como reduzir cada um desses problemas a um problema de emparelhamento de peso máximo. Aparentemente, até 1986 nenhuma atenção foi dada ao caso mais geral em que o grafo de entrada é misto (contém tanto arcos quanto arestas), quando Arkin [Ar86] demonstrou que ambos os problemas são NP-completos. Em vista de tais resultados, seria interessante descobrir classes especiais de grafos mistos para as quais esses problemas podem ser resolvidos em tempo polinomial. Neste artigo descrevemos um algoritmo polinomial para encontrar caminhos mínimos em grafos mistos acíclicos. Além disso, mencionamos casos particulares desse(s) problema(s) que continuam em aberto.

1 Introdução

Grafos mistos podem ser vistos como uma generalização do conceito de grafos orientados e não-orientados, no sentido de que podem conter tanto arcos quanto arestas (sem orientação). Aqui estamos interessados no seguinte problema: *dado um grafo misto $M = (V, E, A)$ com uma função custo $c : (E \cup A) \rightarrow \mathbf{R}$ definida sobre o conjunto dos arcos e arestas de M, encontrar um caminho mínimo de s a t, onde $s, t \in V$.*

Quando a função c assume apenas valores não-negativos, o problema pode ser resolvido em tempo polinomial pelo algoritmo de Dijkstra [Di59]. É um fato bem conhecido que o problema torna-se mais difícil quando c pode assumir valores negativos. No caso especial em que existem circuitos negativos no grafo, o problema é NP-difícil, como pode ser mostrado através de uma simples redução do problema do circuito hamiltoniano. Assim, é interessante investigarmos o problema de detectar circuitos negativos e o de encontrar caminhos mínimos em sua ausência.

Esses dois problemas podem ser resolvidos em tempo polinomial por vários algoritmos no caso especial em que M é um grafo orientado ([Be58], [Fo56], [Fl62], [Wa62], [Ye70]; em [La76] pode-se encontrar uma discussão detalhada sobre tais algoritmos).

Em grafos não-orientados, tais problemas são um pouco mais difíceis. A redução trivial ao caso orientado, através da substituição de uma aresta por um par de arcos reversos, não funciona uma vez que pode introduzir um circuito negativo formado por dois novos arcos. Apesar disso, ambos os problemas podem ser resolvidos em tempo polinomial através de uma redução ao problema de encontrar um emparelhamento (de peso) máximo em um grafo ([To73], [To75], [La76]).

No caso mais geral, Arkin [Ar86] mostrou que o problema de detectar circuitos negativos

[1] Instituto de Matemática e Estatística — Universidade de São Paulo — Caixa Postal 66281 — Ag. Cidade São Paulo — 05389-970 — São Paulo-SP — Brasil (e-mail: lee@ime.usp.br, yw@ime.usp.br).

em grafos mistos e encontrar caminhos mínimos em grafos mistos *sem circuitos negativos* são NP-completos. Descreveremos neste trabalho um algoritmo polinomial para encontrar caminhos mínimos em grafos acíclicos e faremos uma análise de sua complexidade. Mencionaremos ainda alguns casos particulares desse(s) problema(s) que continuam em aberto.

2 Notações e apresentação do problema

Um **grafo misto** é uma terna $M = (V, E, A)$ onde V é um conjunto finito, E é um conjunto finito de pares não-ordenados de elementos de V, e A é um conjunto finito de pares ordenados de elementos de V. Dizemos então que V é o conjunto dos **vértices** de M, E é o conjunto das **arestas** de M e A é o conjunto dos **arcos** de M. Quando $A = \emptyset$ dizemos que M é um **grafo não-orientado** e o indicamos simplesmente por $M = (V, E)$. Quando $E = \emptyset$ dizemos que M é um **grafo orientado** e o indicamos simplesmente por $M = (V, A)$.

Denotamos um elemento de E por $\{u, v\}$ ou por uv. Denotamos um elemento de A por (u, v). Algumas vezes, para explicitar que u e v são os vértices que definem uma aresta e (resp. um arco a) utilizaremos a notação $e = uv$ (resp. $a = (u, v)$). Denotamos um elemento genérico de $E \cup A$ por $l = [u, v]$ (ou $[u, v]$) indicando que l ou é uma aresta $l = uv$ de E ou é um arco $l = (u, v)$ de A.

Um caminho (resp. circuito) em M é uma seqüência alternante de vértices e arcos/arestas de M da forma $C = (v_1, l_1, v_2, l_2, v_3, \ldots, v_{n-1}, l_{n-1}, v_n)$ onde $l_i = [v_i, v_{i+1}]$ ou $l_i = [v_{i+1}, v_i]$ para $1 \leq i \leq n-1$, e $v_i \neq v_j$ para $1 \leq i < j \leq n$ (resp. $1 \leq i < j \leq n-1$ e $v_1 = v_n$). Dizemos que C é **viável** se $l_i = [v_i, v_{i+1}]$ para $1 \leq i \leq n-1$. Dizemos que um caminho (resp. circuito) é **orientado** (resp. **não-orientado**) se ele não contém nenhuma aresta (resp. nenhum arco). Quando não houver ambigüidade, denotaremos um caminho (resp. circuito) apenas por sua seqüência de vértices, por exemplo, $C = (v_1, \ldots, v_n)$. Dizemos que um grafo misto M é **acíclico** se não existem circuitos viáveis em M.

O **comprimento** de um caminho (resp. circuito) C, denotado por $|C|$, é definido como o número de arcos e arestas de C. Para uma dada função custo $c : (E \cup A) \to \mathbf{R}$, definimos o **custo** de um caminho (resp. circuito) C, denotando-o por $c(C)$, como a soma dos custos de cada arco e de cada aresta do caminho (resp. circuito) C. Um caminho (resp. circuito) **mínimo** é um caminho (resp. circuito) de custo mínimo. Um **circuito negativo** é um circuito de custo negativo.

Neste trabalho, sempre que nos referirmos a caminhos ou circuitos estes serão viáveis. Algumas vezes, por questão de ênfase, mencionaremos isso explicitamente. O(s) problema(s) que aqui nos interessam podem ser formulados da seguinte forma: *dado um grafo misto $M = (V, E, A)$, vértices $s, t \in V$ e uma função custo $c : (E \cup A) \to \mathbf{R}$, (a) decidir se existe em M um circuito viável negativo; e (b) caso não exista, encontrar um caminho viável mínimo de s a t.*

Como mencionamos na introdução, Arkin [Ar86] mostrou que esses problemas são NP-completos. Em vista disso, é interessante considerar o problema de encontrar classes de grafos mistos para as quais esses problemas podem ser resolvidos em tempo polinomial. Neste trabalho consideramos o caso em que o grafo misto é acíclico e descrevemos um algoritmo polinomial para encontrar caminhos mínimos em tais grafos.

Em vez de tratarmos do problema de encontrar um caminho mínimo de s a t, vamos tratar do problema mais geral de encontrar, para cada vértice $v \in V - \{s\}$, um caminho mínimo de s a v.

Assim, podemos supor sem perda de generalidade que s é uma **fonte**, ou seja, é um vértice no qual incidem apenas arcos que saem dele (todo arco entrando em s pode ser removido e toda aresta sv incidindo em s pode ser substituída por um arco (s, v)).

Nas considerações a seguir, vamos supor que $M = (V, E, A)$ é um grafo misto acíclico, s é uma fonte de M, e que c é uma função custo definida sobre os arcos e as arestas de M.

É imediato que $F = (V, E)$ é uma floresta, isto é, um grafo não-orientado acíclico. Daqui em diante iremos nos referir a cada componente dessa floresta (que por sua vez é um subgrafo de M) como uma **m-árvore**. Note que uma m-árvore pode consistir de um único vértice.

O seguinte lema descreve uma importante propriedade que será utilizada no algoritmo que descreveremos. Observe que o número N mencionado no lema é o número de vértices do grafo obtido a partir de M contraindo cada uma de suas m-árvores a um vértice.

Lema 2.1 *Seja $M = (V, E, A)$ um grafo misto acíclico, s uma fonte de M, e $N = |V| - |E|$. Então existe uma rotulação dos vértices de M, $r : V \to \{1, \ldots, N\}$, tal que para todo $u, v \in V$:*

1. $r(s) = 1$,
2. *Se $(u, v) \in A$ então $r(u) < r(v)$,*
3. *u e v pertencem a uma mesma m-árvore de M se e somente se $r(u) = r(v)$.*

Não é difícil ver que uma rotulação que satisfaz as condições acima pode ser obtida em tempo linear contraindo as m-árvores e obtendo uma ordenação topológica do grafo orientado resultante.

Antes de descrevermos o algoritmo vamos estabelecer algumas notações. Para cada vértice v de M, seja

$$\sigma(v) := \text{ custo de um caminho viável mínimo de } s \text{ a } v.$$

Quando não existir um caminho viável de s a v escreveremos $\sigma(v) = +\infty$. Para cada vértice v de M seja

$$\alpha(v) := \min\{\sigma(u) + c(a) : a = (u, v) \in A\}.$$

Note que $\alpha(v)$ é o custo de um caminho viável mínimo de s a v que termina em um arco. Se $T = (VT, ET)$ é uma m-árvore de M e $x, y \in VT$, então denotamos por T_{xy} o caminho em T que vai de x a y.

O seguinte lema descreve uma relação de recorrência entre os $\sigma(v)$ que é fundamental para demonstrar a corretude do algoritmo que descreveremos. Note que ele não é válido para grafos mistos em geral.

Lema 2.2 *Seja $T = (VT, ET)$ uma m-árvore de um grafo acíclico M e $v \in VT$. Então*

$$\sigma(v) = \min \begin{cases} \alpha(v) \\ \alpha(u) + c(T_{uv}), & u \in VT - \{v\}. \end{cases}$$

Em particular, se $(\{v\}, \emptyset)$ é uma m-árvore então $\sigma(v) = \alpha(v)$.

A seguir descrevemos o algoritmo. Ele recebe um grafo misto acíclico M e uma fonte s de M devolvendo para cada vértice v de M o custo de um caminho mínimo de s a v. Na descrição do algoritmo omitimos o processo de construção dos caminhos mínimos e descrevemos apenas o cálculo dos custos. É relativamente simples alterar o algoritmo para obter tais caminhos.

Algoritmo CamMin

Entrada: um grafo misto acíclico $M = (V, E, A)$,
uma função custo $c : E \cup A \to \mathbf{R}$ e
uma fonte s de M.
Saída: um vetor $l : V \to \mathbf{R} \cup \{+\infty\}$ tal que $l(v) = \sigma(v)$ para todo $v \in V$.

(* Inicialização *)
$N := |V| - |E|$;
Seja $r : V \to \{1, \ldots, N\}$ uma rotulação satisfazendo o Lema 2.1 e tal que $r(s) = 1$;
$l(s) := 0$;
Para cada $v \in V - \{s\}$ faça
    $l(v) := +\infty$;
    $L(v) := +\infty$;
$k := 2$;

(* Passo Principal *)
Enquanto $k \leq N$ faça   (* Invariante: Se $r(v) < k$ então $l(v) = \sigma(v)$ *)
    Para cada $v \in V$ tal que $r(v) = k$ faça
        $L(v) := \min\{l(u) + c(a) : a = (u, v) \in A\}$;
    Para cada $v \in V$ tal que $r(v) = k$ faça
        Seja $T = (VT, ET)$ a m-árvore de M que contém v;
        $l(v) := \min\left(\{L(v)\} \cup \{L(u) + c(T_{uv}) : u \in VT - \{v\}\} \right)$;
    $k := k + 1$;

Devolva l;

O algoritmo atua em níveis calculando inicialmente os custos dos caminhos mínimos para os vértices de rótulo igual a 2 e armazenando tais valores no vetor l; depois para os vértices de rótulo igual a 3, e assim sucessivamente para os vértices de rótulo $4, 5, \ldots, N$. Pode-se provar que ao final do algoritmo, $l(v) = \sigma(v)$ para todo $v \in V$. Não é difícil verificar que o mesmo tem complexidade $O(|V|^2)$. É interessante notar que o algoritmo acima é uma generalização do conhecido algoritmo de caminhos mínimos para grafos orientados acíclicos (veja [La76]).

Cabe aqui observar que o algoritmo que descrevemos também pode ser utilizado para outra classe de grafos mistos. Trata-se do caso em que $M = (V, E, A)$ é um grafo misto em que nenhum arco pertence a um circuito viável de M. Chamaremos um tal grafo de grafo **arco-acíclico**.

Nesse caso, os componentes de $G = (V, E)$ podem conter circuitos, mas o grafo orientado D, obtido a partir de M pela contração dos componentes de G, é acíclico (observe que é possível decidir em tempo polinomial se um grafo misto é arco-acíclico). Para descobrir se M possui um circuito negativo basta descobrir se um dos componentes de G possui um circuito negativo, o que pode ser feito em tempo polinomial, como mencionamos na introdução. Não é difícil alterar o algoritmo acima para encontrar caminhos mínimos em grafos arco-acíclicos. Neste caso, também devemos ter uma subrotina para calcular custos de caminhos mínimos em grafos não-orientados.

Em seu artigo, Arkin [Ar86] menciona dois casos particulares do problema de detectar circuitos negativos e do problema de encontrar caminhos mínimos em sua ausência que podem ser resolvidos em tempo polinomial. O primeiro caso é aquele em que todas as arestas do grafo misto de entrada têm custo não negativo; nesse caso, é fácil ver que ambos os problemas podem ser reduzidos às suas versões orientadas. O segundo caso é aquele em que o grafo de entrada tem um número *fixo* k de arestas; nesse caso, como cada aresta tem duas possíveis orientações, existem 2^k possíveis orientações do grafo misto de entrada. Assim, os problemas originais podem ser reduzidos aos problemas de detectar circuitos orientados negativos e encontrar caminhos orientados mínimos em cada um desses 2^k grafos orientados. O caso especial em que o grafo misto de entrada tem um número fixo de arcos continua em aberto.

Não encontramos na literatura nenhum trabalho descrevendo outros casos particulares desses problemas para os quais se conhecem algoritmos polinomiais para resolvê-los. Seria interessante descobrir se para outras classes de grafos mistos, além das que mencionamos acima, o problema de detectar circuitos negativos e o problema de encontrar um caminho viável mínimo de s a t são fáceis ou continuam NP-difíceis. Em particular, os dois problemas estão em aberto no caso de grafos mistos planares.

Referências

[Ar86] E.M. Arkin, Complexity of Cycle and Path Problems in Graphs, Tese de Doutorado, Stanford University, 1986.

[Be58] R.E. Bellman, On a Routing Problem, *Quart. Appl. Math.*, **16** (1958), 87–90.

[Di59] E.W. Dijkstra, A Note on Two Problems in Connexion with Graphs, *Numerische Mathematik*, **1** (1959), 269-271.

[Fl62] R.W. Floyd, Algorithm 97: Shortest Path, *Communications of ACM*, **5** (1962), 345.

[Fo56] L.R. Ford Jr. , Network Flow Theory, The Rand Corp., P-923, 1956.

[La76] E.L. Lawler, *Combinatorial Optimization: Networks and Matroids*, Holt, Rinehart & Winston, N. York, 1976.

[To73] R.L. Tobin, Finding a Minimal Weight Path in an Undirected Network, em: 44th National ORSA Meeting, San Diego, California (1973).

[To75] R.L. Tobin, Minimal Complete Matchings and Negative Cycles, *Networks*, **5** (1975), 371–387.

[Wa62] S. Warshall, A Theorem on Boolean Matrices, *Journal of ACM*, **9** (1962), 11–12.

[Ye70] J.Y. Yen, An Algorithm for Finding Shortest Routes from All Sources Nodes to a Given Destination in a General Network, *Quart. Appl. Math.*, **27** (1970), 526–530.