A NOTE OF THE COMPUTATION OF THE k-CLOSURE OF A GRAPH

Jayme Luiz Szwarcfiter

NCE 0286

January, 1986

Universidade Federal do Rio de Janeiro
Núcleo de Computação Eletrônica
Caixa Postal 2324
20001 - Rio de Janeiro,RJ
BRASIL

A NOTE ON THE COMPUTATION OF THE k-CLOSURE OF A GRAPH

Jayme Luiz Szwarcfiter

ABSTRACT

Bondy and Chvátal introduced the concept of k-closure of a graph and described an algorithm which constructs it in $0(n^4)$ steps. In this note is presented a method having complexity $0(n^3)$.

RESUMO

Bondy e Chvátal apresentaram o conceito de k-fechamento de um grafo e descreveram um algoritmo que o constrói em $0(n^4)$ passos. Nessa nota apresenta-se um método cuja complexidade é $0(n^3)$.

1. INTRODUCTION

G denotes a simple undirected graph, $|V(G)| = n \geq 3$ and $|E(G)| = m$. The degree of $v \in V(G)$ is written $d_G(v)$ and $\overline{G}$ is its complement, that is, $V(G) = V(\overline{G})$ and $e \in E(G)$ iff $e \notin E(\overline{G})$. The k-closure $c_k(G)$ of G has been introduced by Bondy and Chvátal [2], who described some of its different applications. One of them provides a sufficient condition for hamiltonicity, stating that if $c_n(G)$ is complete then G is hamiltonian. The latter is strictly better than some previous conditions based on vertex degrees, the Dirac's descendants [3-4, 6-8]. That is, whenever G satisfies any of the conditions [3-4, 6-8] then $c_n(G)$ is complete. Ainouche and Christofides [1] described a different closure $c'(G)$ which also guarantees a hamilton cycle whenever it is complete and such that $c'(G)$ is complete whenever $c_n(G)$ is so. However, the problem of finding $c'(G)$ is NP-hard and therefore as hard as solving a general hamilton cycle problem. In contrast, $c_k(G)$ can be computed in polynomial time by applying algorithm [2] of complexity $O(m_c n^2)$, $m_c = |E(c_k(G))|$. In the present note we describe an implementation of this algorithm which requires $O(m_c n)$ time. This also reduces the overall complexity for finding a hamilton cycle in a graph whose n-closure is complete, because the remaining steps involved in the production of the cycle requires no more than $O(m_c n)$ time [2].

Let k be an integer, $0 \leq k \leq 2n-3$. Define $c_k(G)$ recursively as follows. If G is complete or $d_G(v) + d_G(w) < k$ for any non-edge $(v,w) \in E(\overline{G})$ then $c_k(G) := G$. Otherwise $c_k(G) := c_k(G+(v,w))$, for some $(v,w) \in E(\overline{G})$ such that $d_G(v) + d_G(w) \geq k$.

2. THE ALGORITHM

In order to compute the k-closure of a graph G we define the k-defficiency of a non-edge $e = (v,w) \in E(\overline{G})$ as the value

$$f_k(e) := \max\{0,\ k - d_G(v) - d_G(w)\}.$$

The algorithm can then be described as follows.

In the initial step, let G be a graph and k an integer, $0 \le k \le 2n-3$. Compute the degree $d_G(v)$ and the k-defficiency $f_k(e)$ of each $v \in V(G)$ and $e \in E(\overline{G})$, respectively. Let S be the set of non-edges $e \in E(\overline{G})$ satisfying $f_k(e) = 0$.

In the general step, if $S=\phi$ the algorithm terminates ($c_k(G) := G$). Otherwise, choose $(v,w) \in S$ and for each non-edge $e \in E(\overline{G})-S$ incident to either v or w, decrease $f_k(e)$ by one and if the value of $f_k(e)$ dropped to zero then include e in S. Next, remove (v,w) from S, but include it in G. Finally, repeat the general step.

Except for the last one, each computation of the general step of the above algorithm adds a new edge to the closure of G and requires $0(n)$ time for completion. Therefore the general step is executed p+1 times, $p=m_c-m$, that is, the complexity of the algorithm is $0(m_c n)$. At the beginning of each of these p+1 computations, the set S contains exactly the pairs (v,w), $v,w \in V(G)$, such that the sum of the current degrees of v and w in G is at least k and (v,w) has not yet been included in G. The correctness of the method then follows by induction.

3. CLOSURE AND TOPOLOGICAL SORTING

Given a digraph D the problem of topological sorting consists of arranging its vertices into a sequence in which v_i precedes v_j whenever v_i reaches v_j. The algorithm of Knuth [5] constructs such a sequence by initially defining a set S' consisting of those vertices v_i having indegree $d'(v_i)$ zero. Then iteratively choose $v_i \in S'$, remove it from S' and add it to the output sequence. Next, for each vertex v_j such that $(v_i,v_j) \in E(D)$, decrease $d'(v_j)$ by one and if $d'(v_j)$ dropped to zero include v_j in S'. The process terminates when $S'=\phi$.

Therefore there is a duality between the closure of an undirected graph G and topological sorting of D. That is, replace non-edges $e \in E(\overline{G})$ by vertices $v \in V(D)$ and defficiencies f(e) by indegrees d'(v). Then the ordering in which the non-edges are included in G corresponds to that in which the vertices of D appear in the topological sorting arrangement. Consequently, if L(G) denotes the line graph of G it follows

Theorem: Let G be an undirected graph. Then $c_k(G)$ is complete iff there is an acyclic orientation of $L(\overline{G})$ in which the indegree of each of its vertices $e \in E(L(\overline{G}))$ is at least $f_k(e)$.

REFERENCES

[1] A. Ainouche and N. Christofides, Strong Sufficient Conditions for the Existence of Hamiltonian Circuits in Undirected Graphs, J. Combinatorial Theory B 31 (1981) 339-343.

[2] J.A. Bondy and V. Chvátal, A Method in Graph Theory, Discrete Math. 15 (1976) 111-135.

[3] V. Chvátal, On Hamilton's Ideals, J. Combinatorial Theory B 12 (1972) 163-168.

[4] G.A. Dirac, Some Theorems on Abstract Graphs, Proc. London Math. Soc. 2 (1952) 69-81.

[5] D.E. Knuth, The Art of Computer Programming 1 (Addison-Wesley, Reading, 1968 - second ed. 1973).

[6] M. Las Vergnas, Sur une Proprieté des Arbres Maximaux dans un Graphe, C.R. Acad. Sci. Paris Ser A-B 272 (1971) 1297-1300.

[7] O. Ore, Note on Hamiltonian Circuits, Amer. Math. Monthly 67 (1960) 55.

[8] L. Pósa, A Theorem Concerning Hamiltonian Lines, Publ. Math. Inst. Hungar. Acad. Sci. 7 (1962) 225-226.