POLYNOMIAL COMPLETE CONSECUTIVE INFORMATION RETRIEVAL PROBLEMS

Lawrence T. Kou

TR 74-193

(Revised copy)

Department of Computer Science
Cornell University
Ithaca, New York 14850

POLYNOMIAL COMPLETE CONSECUTIVE INFORMATION RETRIEVAL PROBLEMS

Lawrence T. Kou

INTRODUCTION

The consecutive retrieval property of a file organization is the following. A set of queries Q is said to have consecutive retrieval property with respect to a set of records R if there exists an organization of the record set (without duplication of any record) such that for every $q_i \in Q$, all relevant records can be stored in consecutive storage locations. In linear storage systems (e.g. tape, surface of drum, cylinder of a disk pack), if the query set Q has consecutive retrieval property with respect to the record set R, then to store the pertinent records in consecutive storage locations will provide a file organization with minimum storage space and minimum retrieval time.Let the query set Q be $\{q_1, q_2, \ldots, q_m\}$ and the record set R be $\{r_1, r_2, \ldots, r_n\}$. The relationship between Q and R is conveniently represented by an $n \times m$ 0-1 matrix B. The $(i,j)^{th}$ entry of B is 1 iff record r_i is pertinent to query q_j. This matrix is called the Record-Query incidence matrix.

$$B = \begin{array}{c|ccccc|} & q_1 & q_2 & q_3 & \cdots & q_m \\ \hline r_1 & 1 & 1 & 0 & & 1 \\ r_2 & 0 & 1 & 0 & \cdots & 1 \\ r_3 & 1 & 0 & 1 & & 1 \\ \vdots & & \vdots & & & \vdots \\ r_n & 1 & 0 & 1 & & 0 \end{array}$$

It should be clear that Q has the consecutive retrieval property with respect to R iff there exists a permutation of the rows of B such that the 1's in each column appear in consecutive positions. To find such a permutation, if it exists, was first solved by Fulkerson and Gross in their study of incidence matrix and interval graphs [4]. A different solution was given by Eswaran in his study of consecutive information retrieval [3]. If B is n×m and m is bounded by a polynomial of n, algorithms that have time bound $O(p(n))$ for some polynomial p can be found in [3,4].

However, not all pairs of Q and R have consecutive retrieval property [3,5,6,7]. As a matter of fact, in most practical cases, the consecutive retrieval property is not substantiated. Hence, in general or in practical, either duplication of records is allowed so that pertinent records corresponding to any query are always stored consecutively or, storing the pertinent records corresponding to a query in several blocks of consecutive storage locations is necessary so that each record is stored only once. The former gives rise to a problem of minimizing storage space (minimizing duplication of records) subjected to minimal retrieval time and the latter gives rise to a problem of minimizing retrieval time (minimizing blocks of consecutive storage) subjected to minimal storage space. These two problems can be stated formally as follows:

(A) Problem of minimizing duplications of records

Given an nxm incidence matrix B, let $Q_j = \{r_i \mid b_{ij}=1\}$ for

$1 \leq j \leq m$. Find the minimum length sequence x in the alphabet $R = \{r_1, r_2, \ldots, r_n\}$ such that the elements of Q_j appear consecutively in x, for $j = 1, 2, \ldots, m$.

(B) Problem of minimizing blocks of consecutive storage of relevant records

Given an nxm incidence matrix B, find a permutation of B such that the total number of blocks of consecutive 1's in the columns of B is minimized.

It is shown in this paper that both of these problems are polynomial complete. Loosely speaking, it implies that if one can find an efficient algorithm to solve one of these two problems then many known difficult problems (e.g. Hamiltonian circuit problem, job scheduling problem, travelling salesman problem, to name a few) would all have efficient algorithms to solve them, an unlikely event.

COST GRAPH OF INCIDENCE MATRIX

The cost graph referred here is simply a complete digraph (all selfloops are ignored in this paper) with nonnegative integer cost associated with each edge in the graph. The cost graph associated with an incidence matrix is defined as follows. Given an nxm incidence matrix B, the cost graph G of B is a 3-tuple (V,E,f) such that $V = \{1, 2, \ldots, n\}$ is the set of vertices in the graph. (Vertex i corresponds to row i in B.) $E = \{[i,j] \mid i \neq j \text{ and } i,j \in V\}$ is the set of edges in the graph. $f: E \rightarrow I$, where I is the set of nonnegative integers, is the cost function and for all $[i,j] \in E$, $f([i,j]) = \sum_{s=1}^{m} b_{is} * b_{js}$

where b_{ij} is the $(i,j)^{th}$ entry in B, * is a binary operation defined by 0*0=0, 0*1=0, 1*0=1 and 1*1=0.

Example 1.

Given

$$B = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

the cost graph G is shown in Fig. 1.

For any incidence matrix, there is a unique cost graph associated with it. However, not every cost graph has a corresponding incidence matrix. A simple exercise will show that the cost graph in Fig. 2 has no corresponding incidence matrix. Given a cost graph G, if there exists an incidence matrix B whose associated cost graph is G, then G is said to be 0-1 matrix realizable. For a cost graph $G = (V,E,f)$ if $i,j \in V$ and $i \neq j$ imply $f([i,j]) = f([j,i])$, then G is said to have symmetrical costs. The following two theorems concern certain classes of cost graphs that are 0-1 matrix realizable.

Theorem 1. Let $G_n = (V,E,f)$ be a cost graph with n vertices, $n \geq 3$, and with symmetrical costs. If only edges [1,2] and [2,1] have cost (n-1) individually while every other edge has cost (n-2), then there exists an $n \times m_n$ incidence matrix B_n such that

(i) B_n realizes G_n;

(ii) $m_n = \frac{n(n-1)}{2} + 1$;

(iii) each row of B_n contains (n-1) 1's.

Proof. The proof is given by induction on n.

For n = 3, let

$$B = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

Now assume the theorem holds for n = k. Then, for n = k+1, consider

$$B_{k+1} = \left(\begin{array}{c|c} B_k & I_k \\ \hline 0\ 0\ \ldots\ 0\ 0 & 111\ \ldots\ 1 \end{array}\right)$$

where I_k is a k×k identity matrix.

Part (i):

$$f([1,2]) = \sum_{s=1}^{m_{k+1}} b_{1s} * b_{2s} \quad \text{(by definition)}$$

$$= \sum_{s=1}^{m_k} b_{1s} * b_{2s} + \sum_{s=m_k+1}^{m_{k+1}} b_{1s} * b_{2s}$$

$$= (k-1) + 1$$ (By induction hypothesis that (i) is true for n = k and by property of identity matrix)

$$= k$$

Similarly, f([2,1]) = k.

For $1 \leq i \leq k$, $1 \leq j \leq k$, $i \neq j$, and $[i,j] \neq [1,2]$ or $[2,1]$,

$$f([i,j]) = \sum_{s=1}^{m_{k+1}} b_{is} * b_{js} \quad \text{(by definition)}$$

$$= \sum_{s=1}^{m_k} b_{is} * b_{js} + \sum_{s=m_k+1}^{m_{k+1}} b_{is} * b_{js}$$

$= (k-2) + 1$ (by induction hypothesis that (i) is true for $n = k$ and by property of identity matrix)

$= k-1$

Furthermore, for all $i = 1,2,\ldots,k$,

$$f([k+1,i]) = \sum_{s=1}^{m_{k+1}} b_{(k+1)s} * b_{is} \quad \text{(by definition)}$$

$$= \sum_{s=1}^{m_k} b_{(k+1)s} * b_{is} + \sum_{s=m_k+1}^{m_{k+1}} b_{(k+1)s} * b_{is}$$

$$= 0 + (k-1) \quad \text{(by the construction of row } k+1\text{)}$$

$$= k-1$$

$$f([i,k+1]) = \sum_{s=1}^{m_{k+1}} b_{is} * b_{(k+1)s} \quad \text{(by definition)}$$

$$= \sum_{s=1}^{m_k} b_{is} * b_{(k+1)s} + \sum_{s=m_k+1}^{m_{k+1}} b_{is} + b_{(k+1)s}$$

$= (k-1) + 0$ (by induction hypothesis that (iii) is true for $n = k$ and by construction of row $k+1$)

$$= k-1$$

Hence, for $n = k+1$. B_n realize G_n.

Part (ii):

$m_{k+1} = m_k + k$

$= \frac{k(k-1)}{2} + 1 + k$ (by induction hypothesis that (ii) is true for $n = k$)

$$= \frac{(k+1)k}{2} + 1$$

Hence, for $n = k+1$, $m_n = \frac{n(n-1)}{2} + 1$.

Part (iii):

In B_{k+1} , for $1 \leq i \leq k$, row i contains $(k-1)$ 1's in the first m_k entries due to B_k and contains one 1 in the last k entries due to I_k. So for $1 \leq i \leq k$, row i of B_{k+1} contains k 1's. Row $(k+1)$ contains k 1's by construction. Hence, for $n = k+1$, each row of B_n contains $(n-1)$ 1's.

The proof by induction is thus completed.

Remark: If the cost graph is such that the only two edges that have individual cost $(n-1)$ are $[i,j]$ and $[j,i]$ instead of $[1,2]$ and $[2,1]$, simply interchanging row 1 and row 2 respectively with row i and row j will give a realization of the corresponding new cost graph.

Theorem 2. Let $G = (V,E,f)$ be a cost graph with n vertices, $n \geq 3$, and with symmetrical costs. Let u be a positive integer, $1 \leq u \leq \frac{n(n-1)}{2}$. If $S = \{[i_1,j_1], [j_1,i_1], [i_2,j_2], [j_2,i_2],\ldots, [i_u,j_u], [j_u,i_u]\}$ is a set of 2u edges which have cost $u(n-2) + 1$ each while every other edge in G has cost $u(n-2)$, then there exists an $n \times m$ incidence matrix B such that

(i) B realizes G;

(ii) $m = u(\frac{n(n-1)}{2} + 1)$;

(iii) each row in B contains $u(n-1)$ 1's.

Proof. Let G_k, $1 \leq k \leq u$, be a cost graph with n vertices and with symmetrical costs such that only edges $[i_k,j_k]$ and $[j_k,i_k]$

have cost (n-1) each while every other edge has cost (n-2). By Theorem 1, G_k is 0-1 matrix realizable. Let B_k be the incidence matrix constructed for G_k as in Theorem 1. Now consider

$$B = (B_1 \,\vdots\, B_2 \,\vdots\, \cdots \,\vdots\, B_u)$$

The corresponding cost graph of B is obviously the superposition of cost graphs $G_1, G_2, \ldots, G_u$ (since the cost functions are additive). The theorem follows immediately from the construction of B and Theorem 1.

INCIDENCE MATRIX AND THE HAMILTONIAN PATHS IN THE CORRESPONDING COST GRAPH

Let B be an nxm incidence matrix and G= (V,E,f) be the corresponding cost graph. A Hamiltonian path in G is a simple path in G that includes every vertex exactly once. A Hamiltonian path in G can be specified by a sequence of n vertices, $(i_1, i_2, \ldots$ $\ldots, i_n)$, where the $i_1, i_2, \ldots, i_n$ are all distinct. The cost of a Hamiltonian path in G is the sum over the costs of the edges on the path. The following Lemmas give the relationship between the cost of a Hamiltonian path in G and the total number of consecutive 1's in the columns of B.

Lemma 1. Let B be an nxm incidence matrix and G = (V,E,f) be the corresponding cost graph. Then the cost of the Hamiltonian path (1,2,.....,n) is k if and only if the total number of blocks of consecutive 1's in the columns of B is k+c, where c is the number of 1's in the n^{th} row of B.

Proof. Let N be the total number of blocks of consecutive 1's in the columns of B and N_i be the total number of blocks of consecutive 1's that end at row i of B. Obviously,

$$N = N_1 + N_2 + \ldots + N_n .$$

By the definition of the associated cost graph, it should be clear that, for $1 \leq i < n$, $N_i = k_i$ in B iff $f([i,i+1]) = k_i$ in G. On the other hand, $N_n = c$.

Hence,

$$\begin{aligned} &\text{the cost of the Hamiltonian path } (1,2,\ldots.,n) \\ &= f([1,2]) + f([2,3]) +\ldots+ f([n-1,n]) \\ &= N_1 + N_2 +\ldots.+ N_{n-1} \\ &= N - c \end{aligned}$$

The proof is thus completed.

Lemma 2. Let B be an nxm incidence matrix and G = (V,E,f) be the corresponding cost graph. Then, G has a Hamiltonian path of cost k if and only if there exists an nxn permutation matrix P such that the total number of blocks of consecutive 1's in the columns of PB is k+c, where c is the number of 1's in the n^{th} row of PB.

Proof. Since each Hamiltonian path $(i_1,i_2,\ldots.,i_n)$ in G has a one to one correspondence with a permutation of rows in B, the proof of this Lemma is immediate from Lemma 1.

POLYNOMIAL COMPLETENESS OF GENERAL CONSECUTIVE RETRIEVAL PROBLEMS

Let NP be the class of languages that can be accepted by a nondeterministic polynomial time bounded Turing machine. A language L_1 is polynomially reducible to a language L_2 (written as $L_1 \propto L_2$) iff there exists a deterministic polynomial time bounded Turing machine which will convert each string x in the alphabet of L_1 into a string y in the alphabet of L_2 such that $x \in L_1$ iff $y \in L_2$. A language L is polynomially complete iff L is in NP and every language in NP is polynomially reducible to L. A problem that requires a yes or no answer can be considered as a language such that a string x is in the language iff an instance of the problem that has a yes answer is encoded into the string x. A yes or no problem P_1 is said to be polynomially reducible to a yes or no problem P_2 iff the corresponding languages L_1, L_2, respectively, are such that $L_1 \propto L_2$. A yes or no problem is polynomially complete iff the corresponding language is polynomial complete. The reader is referred to [1,2,8] for the discussions of polynomial complete problems, the polynomial reducibility and the encoding of problems onto Turing tapes.

In the following, several yes or no problems are introduced first and all óf them are to be shown as polynomial complete problems.

Problem 1.

Given: an undirected graph $G = (V,E)$ (without loss of generality it is assumed that $|V| = |\{1,2,...,n\}| = n \geq 3$ and G is not a complete graph).

Question: Is there a Hamiltonian path in G?

Problem 2.

Given: a cost graph $G = (V,E,f)$ and a positive integer u such that

(i) $V = \{1,2,\ldots,n\}$ and $n \geq 3$;

(ii) $1 \leq u \leq \frac{n(n-1)}{2}$

(iii) there exists a set S of $2u$ edges in G,

$S = \{[i_1,j_1], [j_1,i_1], [i_2,j_2], [j_2,i_2],\ldots,[i_u,j_u], [j_u,i_u]\}$ such that $[p,q] \in S \Rightarrow f([p,q]) = u(n-2) + 1$ and $[p,q] \in E$, $[p,q] \notin S \Rightarrow f([p,q]) = u(n-2)$.

Question: Is there a Hamiltonian path in G such that its cost is $u(n-1)(n-2)$?

Problem 3.

Given: an $n\times m$ incidence matrix B and a non-negative integer k

Question: Let $\#(X)$ denote the total number of blocks of consecutive 1's in the columns of an incidence matrix X. Does there exist an $n\times n$ permutation matrix P such that $\#(PB) = k$?

Problem 4.

Given: a finite set $R = \{r_1,r_2,\ldots,r_p\}$, a family of subsets F, $F = \{Q_i \mid 1\leq i\leq q,\ Q_i\subseteq R\}$ and a non-negative integer k

Question: Does there exist a string x in the alphabet R such that the length of x equals to k and for $j = 1,2,\ldots,q$ the elements of Q_j appear consecutively in x?

Problems of whether a Hamiltonian path exists in an undirected or a directed graph have been shown to be polynomial complete in [8]. Although the original problems were concerning the Hamiltonian circuit instead of Hamiltonian path, almost identical proofs as those shown in [8] can be constructed to show that the Hamiltonian path problem is polynomial complete. In the following, Problems 2, 3, 4 are all shown to be polynomial complete.

Theorem 3. Problem 2 is polynomial complete.

Proof. The language L corresponding to problem 2 is certainly in NP. A polynomial time bounded nondeterministic Turing machine can be constructed such that it will guess a correct Hamiltonian path and then check if the cost of the path is equal to $u(n-1)(n-2)$. It remains to show that every language in NP is polynomially reducible to L. Since Problem 1 is polynomial complete, it is sufficient to show that Problem 1 $\propto$ Problem 2.

Let the undirected graph $G = (V, E)$ be an instance for Problem 1. A polynomial time bounded deterministic Turing machine can be constructed to do the following:

(i) set $u = \frac{n(n-1)}{2} - |E|$;

(ii) construct a cost graph $G_1 = (V_1, E_1, f)$ such that $V_1 = V$ and for $i \neq j$, if the undirected pair $\{i,j\} \notin E$, then set $f([i,j]) = f([j,i]) = u(n-2) + 1$ and if $\{i,j\} \in E$, then set $f([i,j]) = f([j,i]) = u(n-2)$.

G_1 is an instance of Problem 2. Furthermore, by the construction of G_1, G has a Hamiltonian path $(i_1, i_2, \ldots, i_n)$ if and only if the cost of the path in G_1 is $u(n-1)(n-2)$. The proof is thus completed.

Theorem 4. Problem 3 is polynomial complete.

Proof. The language L corresponding to Problem 3 is certainly in NP. A polynomial time bounded nondeterministic Turing machine can be constructed to guess a correct permutation matrix P and then check if $\#(PB) = k$. Given an instance of Problem 2, by Theorem 2, a polynomial time bounded deterministic Turing machine can be constructed to set the value of k equal to $u(n-1)^2$ and assign an nxm incidence B such that

(i) B realizes G;

(ii) $m = u(\frac{n(n-1)}{2} + 1)$;

(iii) each row in B contains u(n-1) 1's.

This is an instance of Problem 3. Furthermore, by the construction of B and Lemma 2, there exists an nxn permutation matrix P such that $\#(PB) = u(n-1)(n-2) + u(n-1) = u(n-1)^2$ if and only if the cost graph G has a Hamiltonian path with cost equal to u(n-1)(n-2). Therefore, Problem 2 $\propto$ Problem 3. The proof is thus completed.

Theorem 5. Problem 4 is polynomial complete.

Proof. It is easy to see that the language L corresponding to Problem 4 is in NP. In the following, it is going to show that Problem 1 $\propto$ Problem 4.

Let the undirected graph G = {V,E} be an instance of Problem 1. A polynomial time bounded deterministic Turing machine can be constructed to do the following:

(i) set R = E ;

(ii) set $F = \{Q_1,Q_2,\ldots\ldots,Q_n\}$ where $Q_i = \{\{i,j\} | \{i,j\} \in E\}$ for $i = 1,2,\ldots\ldots,n$;

(iii) set $k = 1 - n + \sum_{i=1}^{n} |Q_i|$.

This is an instance of Problem 4. Notice that, for $i \neq j$ and $Q_i, Q_j \varepsilon F$, $Q_i \cap Q_j = \{i,j\}$ if and only if $\{i,j\} \varepsilon E$. Therefore, there exists a Hamiltonian path in G if and only if there exists a string x such that the length of x equals k and for i = 1,2,.. ....,n the elements of Q_i appear consecutively in x. Hence, Problem 1 $\propto$ Problem 4. The proof is thus completed.

Remark: In Theorem 4, if $\#(PB) = k = u(n-1)^2$, then for any nxn permutation matrix P', $\#(PB) \leq \#(P'B)$. Also, in Theorem 5, if the length of x equals to $k = 1 - n + \sum_{i=1}^{n} |Q_i|$, then x is the minimum length string in the alphabet R such that for i = 1,2,.. ....,n elements of Q_i appear consecutively in the string.

CONCLUSION

The general problems concerning about consecutive information retrieval have been shown to be polynomial complete. In view of this negative results and the increasing need for file organization techniques, good heuristic approaches for the problems seem to be necessary and acceptable.

ACKNOWLEDGMENT

The author is obliged to Professor John E. Hopcroft for reading the manuscript of this paper and for many valuable discussions.

REFERENCES

1. Aho, A.V., J.E. Hopcroft and J.D. Ullman, The Design and Analysis of Computer Algorithms, to appear, 1974.

2. Cook, S.A., "The Complexity of Thoerem Proving Procedures," Proceedings 3rd ACM Conference on Theory of Computing, May, 1970.

3. Eswaran, Kapali P., "Consecutive Retrieval Information System," Ph.D. thesis, Electrical Engineering & Computer Science Department, University of California, Berkeley, May, 1973.

4. Fulkerson, D.R. and D.A. Gross, "Incidence Matrices and Interval Graphs," Pacific Journal of Mathematics, Vol. 15, No. 3, 1965.

5. Ghosh, Sakti, P., "File Organization: The Consecutive Retrieval Property," CACM, September 1972, Vol. 15, No. 9.

6. --------, "Consecutive Storage of Relevant Records with Redundancy," RJ 933, IBM Research Report, 1972.

7. --------, "On the Theory of Consecutive Storage of Relevant Records," Information Science 6, 1973.

8. Karp, R.M.,"Reducibility among Combinatorial Problems," in R. Miller and J. Thatcher (eds.), Complexity of Computer Computations, Plenum, 1972.

9. Sahni, S., "Some Related Problems from Network Flows, Game Theory and Integer Programming," Proceedings 13th Annual Symposium on Switching and Automata Theory, October, 1972.

10. Sethi, R., "Complete Register Allocation Problems," 5th Annual ACM Symposium on Theory of Computing, May, 1973.

11. Ullman, J.D., "Polynomial Complete Scheduling Problems," ACM 4th Symposium on Operating System Principles, October, 1973.

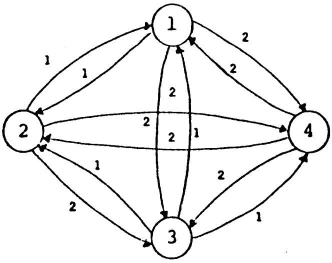

Fig. 1. Cost graph for B in Example 1.

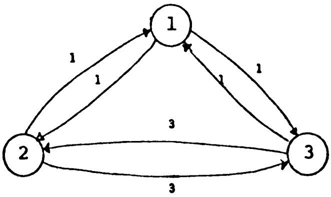

Fig. 2. A cost graph corresponding to no incidence matrix.