On the Coding of the Paths in a Graph and Its Applications

ISAO SHIRAKAWA*, TADAO KASAMI** AND HIROSHI OZAKI*

1. *Introduction*

This paper considers the coding of the paths in a given graph, which can be applied to the design of an examining machine of commutation tickets in a railroad network. In Japan one can use a ticket at any station on the stipulated route. Therefore it is required to code each stipulated route so that the coding may simplify the checking machine's decision scheme of whether a station is on the stipulated route of a given commutation ticket.

2. *Expression of the Paths*

Given a connected graph G with n nodes and m edges, we shall denote the set of the chords i.e. the branches in a cotree C_T with respect to a certain tree T, by $\{c_i;\ i=1, 2, \ldots\ldots, \mu\}$, where $\mu=m-n+1$. Denote by L_i, the fundamental circuit uniquely defined c_i and branches of T, the set $\{L_i;\ i=1, 2, \ldots\ldots, \mu\}$ forming a fundamental system of circuits of G.[1] Let $P_{\alpha\beta}$ represent an arbitrary path in G which has v_α and v_β as its end nodes, and $P_{\alpha\beta}{}^T$ the tree path between v_α and v_β. If $P_{\alpha\beta}$ contains chords of C_T, then we denote the set of those chords by $\{c_{ik}\}$, and define subgraph $L(P_{\alpha\beta})$ of G as

$$L(P_{\alpha\beta})=\sum_k \oplus L_{ik},^{***} \tag{1}$$

where each L_{ik} corresponds to c_{ik}, and $\oplus$ indicates the elementwise exclusive sum. If $P_{\alpha\beta}$ contains no chord, then $L(P_{\alpha\beta})$ is empty. Thus subgraph $L(P_{\alpha\beta})$ is obviously either a circuit, an edge disjoint union of circuits, or an empty graph, and we can set

$$p_{\alpha\beta}=p_{\alpha\beta}{}^T \oplus L(P_{\alpha\beta}). \tag{2}$$

Therefore the problem of determining whether or not $e \epsilon p_{\alpha\beta}$ for a given edge e in G can be reduced to that of deciding whether either $e \epsilon P_{\alpha\beta}{}^T$ or $e \epsilon L(P_{\alpha\beta})$, but not both, while the problem of checking whether $v \epsilon p_{\alpha\beta}$ for a given node v can be done by looking into whether each edge incident to v is in $P_{\alpha\beta}$ or not; consequently we have only to handle edges incident to v in the decision problem of $v \epsilon P_{\alpha\beta}$.

Therefore, we consider a coding method of the paths $\{p_{\alpha\beta}\}$ of G by which we

---

This paper first appeared in Japanese in Joho Shori (the Journal of the Information Processing Society of Japan), Vol. 6, No. 2 (1965), pp. 81-88.

* Dept. of Electronics, Faculty of Engineering, University of Osaka, Miyakojima, Osaka.

** Dept. of Control Engineering, Faculty of Engineering, Science University of Osaka, Toyonaka.

*** $\sum_k \oplus a_k = a_1 \oplus a_2 \oplus \ldots\ldots$

can efficiently decide whether $e\epsilon P_{\alpha\beta}{}^T$ and whether $e\epsilon L(P_{\alpha\beta})$. Let e_α and e_β denote the end branches of $P_{\alpha\beta}{}^T$, where e_α and e_β are incident to v_α and v_β, respectively, and then $T_{\alpha\beta}{}^T$ can be identified by e_α and e_β, while $L(P_{\alpha\beta})$ by the chords $\{c_{ik}\}$ which $P_{\alpha\beta}$ contains. We assign a distinct number $N(e_i)$ to each branch e_i, and then $P_{\alpha\beta}{}^T$ can be identified by $N(e_\alpha)$ and $N(e_\beta)$; whereas to know which chords of C_T a $P_{\alpha\beta}$ passes we define $\{A_i;\ i=1, 2, \ldots\ldots, \mu\}$ such that

$$A_i=\begin{cases}1 \text{ if } P_{\alpha\beta} \text{ passes chord } c_i,\\ 0 \text{ otherwise.}\end{cases} \tag{3}$$

Thus $P_{\alpha\beta}$ can be expressed by $N(e_\alpha)$, $N(e_\beta)$, and $\{A_i\}$. To code each $P_{\alpha\beta}$ we need $2s+\mu$ bits, if we use s bits for the binary expression of each $N(e_i)$.

Under this coding method, to decide whether $e\epsilon L(P_{\alpha\beta})$ for a given edge e, we have only to look into whether or not

$$\sum_i A_i=1 \mod 2, \tag{4}$$

where $\{A_i\}$ corresponds to the set $\{L_i;\ e\epsilon L_i\}$ of the fundamental circuits. The number of elements in $\{L_i\}$ for e is generally not very large; hence it is simple to decide whether $e\epsilon L(P_{\alpha\beta})$. The main problem of our coding method is, therefore, to find a way of assigning number $N(e_1)$ to each branch e_i, which simplifies our decision scheme for $e\epsilon P_{\alpha\beta}{}^T$ with a given edge e.

3. *Assignment of the Branches and the Decision Scheme*

We consider an assigning rule by which we can decide whether $e\epsilon P_{\alpha\beta}{}^T$ for a given e only by the comparison of $N(e)$, $N(e_\alpha)$ and $N(e_\beta)$.

(1) Pick up as long a path* as possible in graph G, and choose a tree T such that T contains the path thus picked up and that each chord with respect to T is incident to a *fork* in G, i.e. to a node of degree greater than two in G.

(2) Call the path picked up in (1) the *trunk* of T, and depict the tree in an arborescent graph.

(3) By a *bough* we mean a tree path P such that either of the end nodes of P is not of degree 2 in G and P contains no fork in G. Give to each branch on the trunk an arrowhead orienting downwards and to each branch not on the trunk an arrowhead orienting away from the trunk.

(4) Each bough is symbolized as follows:

(4. 1) b_1 denotes the lower end bough of the trunk.

(4. 2) Let $\{b_k;\ k=1, 2, \ldots\ldots, M\}$ denote the set of boughs already symbolized, and $T-\{b_k\}$ the complement of $\{b_k\}$ in T. If there is a path in $T-\{b_k\}$ which issues out of the fork in T at which b_M starts, then the lower end bough in the leftmost one of such paths containing an end node of T must be symbolized by b_{M+1}.

(4. 3) Otherwise, the bought not yet symbolized which has a node in common with b_M is symbolized by b_{M+1}.

---

* By the length of a path we mean the number of edges on the path.

(5) Assign number $N(e_i)$ to each branch e_i as follows:

(5.1) Assign zero to the lower end branch in b_1.

(5.2) If branch e_{ki} in b_k has been assigned number $N(e_{ki})$, then assign $N(e_{kj})=N(e_{ki})+1$ to the branch e_{kj} in b_k immediately above e_{ki}.

(5.3) If the upper end branch e_p of b_k has been assigned number $N(e_p)$, then assign number $N(e_q)=N(e_p)+1$ to the lower end branch e_q of b_{k+1}.

All the branches having been assigned numbers in this way, we can decide whether $e\epsilon P_{\alpha\beta}{}^T$ for a given e as follows. In what follows we set $N(e_\alpha)\leq N(e_\beta)$ for each $P_{\alpha\beta}{}^T$.

(1) For each branch e on the trunk:

$e\epsilon P_{\alpha\beta}{}^T$ if and only if $N(e_\alpha)\leq N(e)\leq N(e_\beta)$.

(2) For each branch e not on the trunk:

Let the *posterity* of an element x of T (x: a node, a branch, or a bough) denote the subgraph of T which consists of the paths orienting away from and including x; while by the *ancestory* of an element x we mean the subgraph of T which consists of the paths orienting toward and including x. If $\varphi(e)$ represents the smallest of the branch numbers in the posterity of branch e, then $e\epsilon P_{\alpha\beta}{}^T$ if and only if either of the following two conditions is satisfied.

$$\text{(a)}\quad N(e_\alpha)<\varphi(e),\ \text{and}\ \varphi(e)\leq N(e_\beta)\leq N(e). \tag{5}$$

$$\text{(b)}\quad \varphi(e)\leq N(e_\alpha)\leq N(e),\ \text{and}\ N(e)\leq N(e_\beta) \tag{6}$$

Note that the decision procedure whether $e\epsilon P_{\alpha\beta}{}^T$ is simpler in the case of (1) than that of (2). That is why we have picked as long a path as possible for the trunk of tree T.

We have seen that whether or not $e\epsilon P_{\alpha\beta}{}^T$ can be decided only by the comparison of branch numbers. From the practical viewpoint, it is desirable to make comparisons serially; in other words to make comparisons serially by scanning bitwise the binary expression of $N(e_\alpha)$ and $N(e_\beta)$ of a given $P_{\alpha\beta}$ only *once*. For testing (5) and (6) serially, some elaborate assigment of $N(e_i)$ for each e_i is required. For this purpose we must consider an assignment method with the prefix property that each bough b_k has a certain prefix in its binary expression. Then it is possible to decide whether $N(e_\alpha)<\varphi(e)$, $\varphi(e)\leq N(e_\beta)\leq N(e)$, or $\varphi(e)\leq N(e_\alpha)\leq N(e)$ in (5) and (6) only by comparing the prefix of $N(e)$ and the corresponding bits of $N(e_\alpha)$ or $N(e_\beta)$. Here we omit the details of the assignment method. The interested readers are asked to refer to the authors' paper (2).

4. *Decision Process for Nodes*

A node v in G is said to lie on a given path $P_{\alpha\beta}$, if and only if at least one of the edges incident to v is on the path. Thus the problem of deciding whether $v\epsilon P_{\alpha\beta}$ or not can be reduced to that of $e\epsilon P_{\alpha\beta}$, as stated before. We denote the sets of chords, if any, and branches incident to node v_j by $\{c_{jk};\ k=1, 2, \ldots\ldots, \rho_v\}$ and $\{e_{jl};\ l=1, 2, \ldots\ldots, \eta_v\}$, respectively. If, for given $P_{\alpha\beta}$, any of $\{L_{jk}\}$ corresponding to $\{c_{jk}\}$ is contained in $\{L_i;\ A_i=1\}$, then $v\epsilon P_{\alpha\beta}$. Otherwise, we must test whether $e_{jl}\epsilon L(P_{\alpha\beta})$ for each l, in which case the steps in the decision procedure

of $e_{jl}\epsilon P_{\beta\alpha}{}^{T}$ can be simplified reduced, since $e_{jl}\epsilon P_{\alpha\beta}{}^{T}$ if either one of the three conditions is satisfied:

(i) e_α is in the posterity of e_{jl}, but e_β is not;

(ii) e_β is in the posterity of e_{jl}, but e_α is not;

(iii) e_α and e_β are both in the posterity of e_{jl}, and $N(e_{jl})=N(e_\beta)$.

If e_{jl} is not on the trunk, then to test whether e_α or e_β is in the posterity of e_{jl} in (i), (ii), or (iii) above, it is sufficient to compare prefix $p(e_{jl})$ of e_{jl} and the corresponding bits of $N(e_\alpha)$ or $N(e_\beta)$.

5. *Applications to Commutation Tickets*

The decision scheme in graph G stated so far would not very efficiently be applied to railroad network R, since to decide whether a node v of degree two is on a given path we should have to test the two edges incident to v. In such a case it is more desirable to transform R into another graph G_R such that each node of degree two in R corresponds to an edge in G_R with the same incidence relation, while each node of degree one in R corresponds to an edge in G_R one of the end nodes of which is of degree one, and each node of degree more than two in R corresponds to a node of the same degree in G_R with the same incidence relation.

Apply the assignment procedure to G_R as stated before, and then the paths $\{P_{\alpha\beta}\}$ are coded with $2s+\mu$ bits. At the station in R corresponding to a branch e in G_R we must equip the examining machine which has the decision scheme for $e\epsilon P_{\alpha\beta}$, while at the station in R corresponding to a fork v in G_R the machine with the decision scheme for $v\epsilon P_{\alpha\beta}$ must be set up.

We have applied our method to the network of Kinki Nippon Railroad Company, one of the greatest private railroad companies in Japan, which has some 280 stations and 3 fundamental circuits, and in which we can enumerate some 170,000 paths, and so we need 18 bits only for coding each path without any consideration of the decision scheme. In our method we need 21 bits for the coding, with $s=9$ and $\mu=3$, and even if we code the paths without prefix properties we need the same 21 bits. Eventually, it follows that by introducing only three redundant bits in the coding we can much reduce the number of steps necessary for the decision scheme.

6. *Application to the Routing Problem of Transportation Network*

By a *transportation network* N we mean a linear graph whose nodes correspond to stations and edges to routes along which commodities are transported. In the case that a certain commodity is required to be transported through a prescribed path so that it can be carried along the route. To this routing problem in transportation networks we can apply our coding method.

Construct graph G_N corresponding to N with the same rule in the previous section, and apply the assignment procedure to G_N. The function of the routing machine at each fork station is to decide from which edge of those incident to the

station one has to carry a commodity away. Hence, apart from the decision scheme of commutation tickets, in the routing procedure the fact that a commodity is to be routed at fork v in G_N means that v is on its prescribed route $P_{\alpha\beta}$, that is, the fact that $P_{\alpha\beta}$ covers one or two edges incident to v. If the machine is to be equipped with the memory storing the information about the edge from which each commodity is carried in the fork v, then the routing scheme at v is to be reduced to singling out the other edge incident to v which is on $P_{\alpha\beta}$. If no such edge exists, stop the commodity at v. Thus the routing scheme in this case can be implemented by using the decision process for $v\epsilon P_{\alpha\beta}$ in last section. On the other hand, if the machine is to be required to route each commodity without the information about the edge from which it is carried in, then we can select the edge on $P_{\alpha\beta}$ by using steps in the decision scheme for $v\epsilon P_{\alpha\beta}$. If there is only one such edge, then stop the commodity at v. If there are two, then we single out the one from which it is to be carried away. For this purpose, preassign a direction to each chord, and define the functions $\{B_i;\ =1, 2, \ldots\ldots, \mu\}$ and X in addition to $N(e_\alpha)$, $N(e_\beta)$, and $\{A_i\}$ such that

$$B_i=\begin{cases}1 & \text{if } P_{\alpha\beta} \text{ passes } c_i \text{ in the direction of } c_i,\\ 0 & \text{otherwise}\end{cases}$$

$$X=\begin{cases}1 & \text{if } N(e_\alpha)<N(e_\beta),\\ 0 & \text{otherwise}\end{cases}$$

If the two edges are proved to be on $P_{\alpha\beta}{}^T$, then denote them by e_i and $e_j(N(e_i)>N(e_j))$, and route the commodity to e_i if $X=1$ or otherwise to e_j. If the two edges are not both on $P_{\alpha\beta}{}^T$, at least one edge of the two is a chord. Denote such a chord by c_k, and route the commodity to c_k if $A_k f_v(B_k)=1$, or to the other edge if $A_k f_v(B_k)=0$, where

$$f_v(B_i)=\begin{cases}B_i & \text{if } c_i \text{ is directed away from } v,\\ \tilde{B}_i & \text{otherwise.}*\end{cases}$$

Thus the routing procedure is carried out by using some steps in the decision scheme in the railroad network.

Acknowledgement

The authors wish to thank Dr. M. Nakai, Mr. K. Inoue and Mr. H. Oda, Kinki-Sharyo Co., Ltd., for their supports and instructive advices.

References

[1] SESHU, S. AND M.B. REED, Linear Graphs and Electrical Networks, Addison-Wesley Pub. Co., Inc., Reading, Mass. (1961).

[2] SHIRAKAWA, I., T. KASAMI, H. OZAKI, K. INOUE AND H. ODA, Considerations on the Coding of the Paths and its Applications, Rept. of TGCT, Inst. Elec. Commn. Engrs of Japan (1965) (in Japanese).

* $\tilde{B}_i$ means the negation of B_i, i.e. $1-B_i$.