

# 双环网络的 $[+h]$ 边优先寻径策略

方木云<sup>1),2)</sup> 屈玉贵<sup>1)</sup> 赵保华<sup>1)</sup>

<sup>1)</sup>(中国科学技术大学计算机科学系 合肥 230027)

<sup>2)</sup>(安徽工业大学计算机学院 安徽 马鞍山 243002)

**摘 要** 提出一种先走 $[+h]$ 边、当走 $[+h]$ 边不利时才走 $[+1]$ 边的 $[+h]$ 边优先寻径策略;得出 $[+h]$ 边优先最短路径和双环网络的“竹筏”(一种新 $L$ 形瓦)型空间解;“竹筏”中节点之间的 $[+h]$ 边优先最短路径存在递推关系;由节点的 $[+h]$ 边优先最短路径推出双环网络的直径公式;利用 VB6.0 和 SQL Server 2000 仿真了 $[+h]$ 边优先寻径策略;作者曾提出的 $[+1][+h]$ 双边寻径策略是固定路径,寻找节点,而 $[+h]$ 边优先寻径策略是固定节点,寻找路径;传统 $L$ 形瓦难以构造但易求其等价双环网络的直径,而新 $L$ 形瓦易构造但难以求其等价双环网络的直径;指出了陈忠学文中的几个错误。

**关键词** 双环网络; $[+h]$ 边优先寻径; $[+h]$ 边优先最短路径;“竹筏”; $L$ 形瓦  
**中图法分类号** TP302

## $[+h]$ -Link Prior Routing Strategy for Double-Loop Network

FANG Mu-Yun<sup>1),2)</sup> QU Yu-Gui<sup>1)</sup> ZHAO Bao-Hua<sup>1)</sup>

<sup>1)</sup>(Department of Computer Science, University of Science and Technology of China, Hefei 230027)

<sup>2)</sup>(Department of Computer Science, Anhui University of Technology, Ma'anshan, Anhui 243002)

**Abstract** A  $[+h]$ -link prior routing strategy, that  $[+h]$ -link is chosen prior until it is not favourable, then  $[+1]$ -link is chosen, is proposed. The shortest  $[+h]$ -link prior path and the bamboo raft (a new kind of  $L$ -shaped Tile) for a double-loop network are obtained. There is inferring relationships among nodes in the bamboo raft. The diameter of a double-loop network can be derived from the shortest  $[+h]$ -link prior path. VB6.0 and SQL Server2000 are selected to simulate the  $[+h]$ -link prior routing strategy. The  $[+1][+h]$  routing strategy proposed by the authors is fixing path then seeking nodes, while the  $[+h]$ -link prior routing strategy is fixing nodes then seeking path. The traditional  $L$ -shaped Tile is difficult to be constructed but is easy to calculate the diameter of its equivalent double-loop network, while the new  $L$ -shaped Tile is easy to be constructed but is difficult to calculate the diameter of its equivalent double-loop network. Some mistakes in the paper published by Chen in 2001 are pointed out.

**Keywords** double-loop network;  $[+h]$ -link prior routing; the shortest  $[+h]$ -link prior path; bamboo raft;  $L$ -shaped tile

## 1 引 言

双环网络是计算机互连网络或通信系统的一类

重要拓扑结构,广泛应用于计算机局域网和各种平行处理结构,其图论模型是指这样一个有向图  $G(N;1,h)$ : 它的每个顶点记为  $0,1,2,\dots,N-1$ ,并从每个顶点  $i$  发出两条有向边  $i \rightarrow i+1(\bmod N)$  和

$i \rightarrow i+h(\bmod N)$ , 分别记为 $[+1]$ 边,  $[+h]$ 边, 其中  $h$  是自然数, 且  $1 < h < N$ . 关于双环网络的研究成果已经得到很多<sup>[1-10]</sup>, 本文提出一种寻径方法:  $[+h]$ 边优先寻径策略, 得出一些有意义的结果和一个直观的图形, 指出文献[1]中的几个错误.

2 双环网络  $G(N;1,h)$  的寻径策略

由于双环网络的对称性, 仅需考虑从节点 0 到其它节点的最短路径. 从节点 0 发出两条有向边  $[+1]$  和  $[+h]$  去访问其它节点, 有五种访问方式: (1) 只走  $[+1]$  边; (2) 只走  $[+h]$  边; (3) 优先走  $[+1]$  边, 后走  $[+h]$  边; (4) 优先走  $[+h]$  边, 后走  $[+1]$  边; (5)  $[+1]$  和  $[+h]$  边同时走. 对于某些特殊节点来说, 前 3 种方法可以用, 但是对于普遍意义的节点来说, 后两种方法才能更快地找到最短路径. 作者在文献[9]中已提出  $[+1][+h]$  双边寻径策略. 因此, 本文提出  $[+h]$  边优先寻径策略.

**引理 1**<sup>[1]</sup>. 在双环网络中从节点 0 出发, 经过  $x$  条  $[+1]$  边,  $y$  条  $[+h]$  边达到节点  $v$  的充分必要条件是  $v = x + yh(\bmod N)$ .

证明. 见文献[1].

**定义 1.** 若存在整数  $x$ , 使得  $x[+1]$  是 0 到  $v(0 < v < N)$  的路径, 则称  $x[+1]$  是 0 到  $v$  的单一  $[+1]$  边路径; 设  $x'[+1]$  是 0 到  $v$  的单一  $[+1]$  边路径, 若不存在比  $x'$  更小的  $x$ , 使  $x[+1]$  也是 0 到  $v$  的单一  $[+1]$  边路径, 则称  $x'[+1]$  是 0 到  $v$  的单一  $[+1]$  边最短路径; 若存在整数  $y$ , 使得  $y[+h]$  是 0 到  $v(0 < v < N)$  的路径, 则称  $y[+h]$  是 0 到  $v$  的单一  $[+h]$  边路径; 设  $y'[+h]$  是 0 到  $v$  的单一  $[+h]$  边路径, 若不存在比  $y'$  更小的  $y$ , 使  $y[+h]$  也是 0 到  $v$  的单一  $[+h]$  边路径, 则称  $y'[+h]$  是 0 到  $v$  的单一  $[+h]$  边最短路径.

对于单位步长的双环网络, 两种路径的存在性有如下结论<sup>[1]</sup>.

**结论 1.** 0 到  $v(0 < v < N)$  的单一  $[+1]$  边路径和单一  $[+1]$  边最短路径均存在. 当  $\gcd(N, h) = 1$ , 0 到  $v(0 < v < N)$  的单一  $[+h]$  边路径和单一  $[+h]$  边最短路径均存在; 当  $\gcd(N, h) = m \neq 1$ , 若  $v = 0(\bmod m)$ , 0 到  $v(0 < v < N)$  的单一  $[+h]$  边路径和单一  $[+h]$  边最短路径均存在, 若  $v \neq 0(\bmod m)$ , 0 到  $v(0 < v < N)$  的单一  $[+h]$  边路径和单一  $[+h]$  边最短路径均不存在.

现在要研究的问题是寻找 0 到  $v$  的最短路径,

设  $x^* + y^* = \min\{x + y \mid x[+1] + y[+h], x \geq 0, y \geq 0\}$  是 0 到  $v$  的最短路径, 一般情况下,  $x^*$  和  $y^*$  不唯一, 即便  $x^*$  和  $y^*$  唯一, 但最短路径表现形式也有  $C_{x^*+y^*}^{x^*}$  种. 如何求得这个最短路径长度? 一种思路是: 去寻找某一条特殊的最短路径; 另一种思路是: 同时找出所有最短路径. 作者在文献[9]中已讨论过同时找出所有最短路径的方法, 因此, 下面只介绍寻找某种特殊的最短路径的方法.

3  $[+h]$  边优先的最短路径寻径策略

不失一般性, 先研究 0 到任意节点  $v$  的最短距离. 为了最快达到  $v$  点, 显然优先走  $[+h]$  边, 当走  $[+h]$  边不再有利时, 走  $[+1]$  边, 这种策略称为  $[+h]$  边优先的最短路径寻径策略.

**定义 2.** 设从节点 0 到节点  $v$  共有  $t$  条最短路径:  $x_v^{(i)}[+1] + y_v^{(i)}[+h] (i = 1, 2, \dots, t)$ , 若  $y_v^{(j)} = \max\{y_v^{(i)}, i = 1, 2, \dots, t\}$ , 且所有  $[+h]$  边依次在前, 所有  $[+1]$  边依次在后, 则称  $x_v^{(j)}[+1] + y_v^{(j)}[+h]$  为从节点 0 到节点  $v$  的  $[+h]$  边优先最短路径.

文献[1]中定义了  $[+1]$  边优先最短路径, 并称该路径存在且唯一, “唯一”这个结论是错误的, 因为定义没有强调  $[+1]$  边和  $[+h]$  边的次序, 所以仍有  $C_{x_v+y_v}^{x_v}$  条路径, 只能说  $x_v, y_v$  的值唯一, 严格来讲, 文献[1]中的  $[+1]$  边优先最短路径应改称为:  $[+1]$  边最大最短路径.

从实际情况来看,  $[+h]$  边优先最短路径的概念更有意义, 体现更快的接近节点  $v$  的寻径策略. 从定义可知  $[+h]$  边优先最短路径必为最短路径 (是一条特殊的最短路径), 反之不成立. 并且, 从节点 0 到任意节点  $v$  的  $[+h]$  边优先最短路径存在, 不仅  $x_v, y_v$  的值唯一, 路径形式也唯一: 所有  $[+h]$  边依次在前, 所有  $[+1]$  边依次在后.

**引理 2.** 设  $v = kh + r, 0 \leq r < h$ , 则  $x[+1] + y[+h]$  是  $G(N;1,h)$  中从节点 0 到节点  $r$  的  $[+h]$  边优先最短路径的充分必要条件是:  $x[+1] + (y+k)[+h]$  是节点 0 到节点  $v$  的  $[+h]$  边优先最短路径.

证明. 必要性.

(1) 是路径, 因为  $x[+1] + y[+h]$  是  $G(N;1,h)$  中从 0 到  $r$  的  $[+h]$  边优先最短路径, 因而  $r = x + yh(\bmod N)$  (由引理 1 得), 代入  $v = kh + r = kh + x + yh(\bmod N) = x + (y+k)h(\bmod N)$ , 所以  $x[+1] + (y+k)[+h]$  是 0 到  $v$  的路径;

(2) 是最短路径, 用反证法, 设另有  $x_1[+1] + (y_1 + k)[+h]$  是 0 到  $v$  的最短路径, 则  $x_1 + (y_1 + k) < x + (y + k) \Rightarrow x_1 + y_1 < x + y$ , 由引理 1 知  $x_1[+1] + y_1[+h]$  是 0 到  $r$  的路径, 所以  $x[+1] + y[+h]$  不是 0 到  $r$  的最短路径, 这与已知条件矛盾, 因而  $x[+1] + (y + k)[+h]$  是 0 到  $v$  的最短路径;

(3) 是  $[+h]$  边优先最短路径, 用反证法, 设另有  $x_1[+1] + (y_1 + k)[+h]$  是 0 到  $v$  的  $[+h]$  边优先最短路径, 则  $x_1 + (y_1 + k) = x + (y + k)$  并且  $y_1 + k > y + k \Rightarrow y_1 > y$ , 由引理 1 和第 2 步证明知  $x_1[+1] + y_1[+h]$  是 0 到  $r$  的最短路径, 所以  $x[+1] + y[+h]$  不是 0 到  $r$  的  $[+h]$  边优先最短路径, 这与已知条件矛盾, 因而  $x[+1] + (y + k)[+h]$  是 0 到  $v$  的  $[+h]$  边优先最短路径。

充分性。

(1) 是路径, 因为  $x[+1] + (y + k)[+h]$  是  $G(N; 1, h)$  中从 0 到  $v$  的  $[+h]$  边优先最短路径, 因而  $v = x + (y + k)h \pmod{N}$  (由引理 1 得), 代入  $v = kh + r \Rightarrow r = v - kh = x + yh \pmod{N}$ , 所以  $x[+1] + y[+h]$  是 0 到  $r$  的路径;

(2) 是最短路径, 用反证法, 设另有  $x_1[+1] + y_1[+h]$  是 0 到  $r$  的最短路径, 则  $x_1 + y_1 < x + y \Rightarrow x_1 + (y_1 + k) < x + (y + k)$ , 由引理 1 知  $x_1[+1] + (y_1 + k)[+h]$  是 0 到  $v$  的路径, 所以  $x[+1] + (y + k)[+h]$  不是 0 到  $v$  的最短路径, 这与已知条件矛盾, 因而  $x[+1] + y[+h]$  是 0 到  $r$  的最短路径;

(3) 是  $[+h]$  边优先最短路径, 用反证法, 设另有  $x_1[+1] + y_1[+h]$  是 0 到  $r$  的  $[+h]$  边优先最短路径, 则  $x_1 + y_1 = x + y \wedge y_1 > y \Rightarrow y_1 + k > y + k$ , 由引理 1 和第 2 步证明知  $x_1[+1] + (y_1 + k)[+h]$  是 0 到  $v$  的  $[+h]$  边优先最短路径, 所以  $x[+1] + (y + k)[+h]$  不是 0 到  $v$  的  $[+h]$  边优先最短路径, 这与已知条件矛盾, 因而  $x[+1] + y[+h]$  是 0 到  $r$  的  $[+h]$  边优先最短路径。证毕。

文献[1]中为了证明其引理 3, 提出了引理 2, 其引理 2 的结论和证明都是错误的 (见本文第 7 节), 其引理 3 的结论正确, 证明过程有误, 因为它用到其引理 2。

由定义 2 和引理 2 可知, 从节点 0 到任意节点  $v$  的最短路径都转化为求 0 到节点  $r (= v - kh)$  的  $[+h]$  边优先最短路径问题。

**引理 3.** 设  $x[+1] + y[+h]$  是 0 到  $i$  的  $[+h]$  边优先最短路径, 且  $x \neq 0$ , 则  $(x - 1)[+1] + y[+h]$  是 0 到  $i - 1$  的  $[+h]$  边优先最短路径。

证明。

(1) 是路径, 因为  $x[+1] + y[+h]$  是 0 到  $i$  的  $[+h]$  边优先最短路径, 因而  $i = x + yh \pmod{N}$  (由引理 1 得), 又有  $i \neq 0, x \neq 0$ , 则  $i - 1 = x - 1 + yh \pmod{N}$ , 所以  $(x - 1)[+1] + y[+h]$  是 0 到  $i - 1$  的路径;

(2) 是最短路径, 用反证法, 设另有  $(x_1 - 1)[+1] + y_1[+h]$  是 0 到  $i - 1$  的最短路径, 则  $x_1 - 1 + y_1 < x - 1 + y \Rightarrow x_1 + y_1 < x + y$ , 由引理 1 知  $x_1[+1] + y_1[+h]$  是 0 到  $i$  的路径, 所以  $x[+1] + y[+h]$  不是 0 到  $i$  的最短路径, 这与已知条件矛盾, 因而  $(x - 1)[+1] + y[+h]$  是 0 到  $i$  的最短路径;

(3) 是  $[+h]$  边优先最短路径, 用反证法, 设另有  $(x_1 - 1)[+1] + y_1[+h]$  是 0 到  $i - 1$  的  $[+h]$  边优先最短路径, 则  $(x_1 - 1) + y_1 = (x - 1) + y$  并且  $y_1 > y$ , 由引理 1 和第 2 步证明知  $x_1[+1] + y_1[+h]$  是 0 到  $i$  的最短路径, 所以  $x[+1] + y[+h]$  不是 0 到  $i$  的  $[+h]$  边优先最短路径, 这与已知条件矛盾, 因而  $(x - 1)[+1] + y[+h]$  是 0 到  $i$  的  $[+h]$  边优先最短路径。

由引理 3 可知, 当  $i$  不是单一  $[+h]$  边优先最短路径节点时, 它与  $i - 1$  的  $[+h]$  边优先最短路径之间存在递推关系。

证毕。

**定理 1.** 设  $x_i[+1] + y_i[+h]$  是 0 到  $i$  的  $[+h]$  边优先最短路径, 则  $0 \rightarrow i + 1$  的  $[+h]$  边优先最短路径  $x_{i+1}[+1] + y_{i+1}[+h]$  满足:

(1) 当  $x_i + y_i + 1 \leq s_{i+1}$  或  $s_{i+1}$  不存在时,  $x_{i+1} = x_i + 1, y_{i+1} = y_i$ ;

(2) 当  $x_i + y_i + 1 > s_{i+1}$  时,  $x_{i+1} = 0, y_{i+1} = s_{i+1}$ 。

其中  $s_{i+1}$  是 0 到  $i + 1$  的单一  $[+h]$  边优先最短路径的长度。

这个定理的证明完全同文献[1]。

**定义 3.** 称以下节点为节点 0 所对应的“非平常节点”: 0 到它们的  $[+h]$  边优先最短路径正好就是 0 到它们的单一  $[+h]$  边最短路径。

“非平常节点”正是  $[+h]$  边优先寻径的结果, 有了“非平常节点”的定义后, 与文献[1]一样可以求出 0 所对应的在  $0 < i < h$  范围内的“非平常节点”。因为当  $0 < i < h$  时,  $0 \rightarrow i$  的  $[+h]$  边优先最短路径的长度不会超过  $i$ 。计算  $i_t = t \times h \pmod{N}$  ( $0 < t < h$ ), 若  $0 < i_t < h$  且  $t \leq i_t$ , 则记录  $(t, i_t)$ , 这些  $i_t$  就是可能的“非平常节点”。假设已经求出  $0 < i < h$  范围内所有可能的“非平常节点”:  $i_1 < i_2 < \dots < i_u$ , 共有  $u$  个以及 0 到它们的单一  $[+h]$  边优先最短路径的长度是

$s_{i_1}, s_{i_2}, \dots, s_{i_u}$ , 可以得出如下两个定理.

**定理 2.** 确定“非平常节点”.

(1)  $i_1$  是“非平常节点”, 0 到  $i_1$  的  $[+h]$  边优先最短路径是  $s_{i_1} [+h]$ ;

(2) 若  $s_{i_1} + (i_2 - i_1) < s_{i_2}$ , 则  $i_2$  不是“非平常节点”, 且 0 到  $i_2$  的  $[+h]$  边优先最短路径是  $(i_2 - i_1) [+1] + s_{i_1} [+h]$ ; 否则  $i_2$  是“非平常节点”, 0 到  $i_2$  的  $[+h]$  边优先最短路径是  $s_{i_2} [+h]$ ;

(3) 对于  $3 \leq j \leq u$ , 设 0 到  $i_{j-1}$  的  $[+h]$  边优先最短路径是  $x_{i_{j-1}} [+1] + y_{i_{j-1}} [+h]$ , 若  $(i_j - i_{j-1}) + x_{i_{j-1}} + y_{i_{j-1}} < s_{i_j}$ , 则  $i_j$  不是“非平常节点”, 且 0 到  $i_j$  的  $[+h]$  边优先最短路径是  $(i_j - i_{j-1} + x_{i_{j-1}}) [+1] + y_{i_{j-1}} [+h]$ ; 否则  $i_j$  是“非平常节点”.

**定理 3.** 确定  $0 \rightarrow h$  内任意点的  $[+h]$  边优先最短路径: 设  $0 < i_1 < i_2 < \dots < i_w < h$  是节点 0 所对应的在 0 到  $h$  之内的所有的  $w$  个“非平常节点”, 0 到它们的  $[+h]$  边优先最短路径分别是  $x_{i_1} [+1] + y_{i_1} [+h]$ ,  $x_{i_2} [+1] + y_{i_2} [+h]$ ,  $\dots$ ,  $x_{i_w} [+1] + y_{i_w} [+h]$ , 则对任意节点  $j (i_j < j < i_{j+1})$ ,  $i_j$  和  $i_{j+1}$  是“非平常节点”, 0 到  $j$  的  $[+h]$  边优先最短路径为  $(j - i_j + x_{i_j}) [+1] + y_{i_j} [+h]$ ; 若  $j < i_1$ , 则 0 到  $j$  的  $[+h]$  边优先最短路径为  $j [+1]$ .

利用  $[+h]$  边优先的最短路径寻径策略, 将双环网络的节点分成  $p (= \lceil N/h \rceil) (N = ph + q)$  个长度为  $h$  的等长区间和一个长度为  $q$  的区间, 设  $0 < i_1 < i_2 < \dots < i_r < \dots < i_w < h$  是  $0 \rightarrow h$  区间内的所有的  $w$  个“非平常节点” ( $i_r$  是  $r$  前面的第一个“非平常节点”), 0 到它们的  $[+h]$  边优先最短路径分别是  $s_{i_1}, s_{i_2}, \dots, s_{i_r}, \dots, s_{i_w}$ , 这  $w+1$  个区间的长度分别是  $l_{i_0}, l_{i_1}, l_{i_2}, \dots, l_{i_r}, \dots, l_{i_u}, L_h := \max\{(s_{i_j} + l_{i_j}) | 0 \leq j \leq w\}$ ,  $L_r := \max\{(s_{i_j} + l_{i_j}) | 0 \leq j < r\}$ , 注:  $s_{i_0} = 0$ .

**推论 1.** 在  $G(N; 1, h)$  中, 节点 0 到任意节点  $v (0 < v < N)$  的最短路径长度  $l = k + s_{i_r} + r - \sum_{j=0}^{r-1} l_{i_j}$ , 其中  $v = kh + r (0 \leq r < h)$ ,  $i_r$  是  $r$  前面的第一个“非平常节点”. 双环网络的直径  $d = \max\{p-1 + L_h, p + L_r\}$ .

$k$  称为直接模长;  $s_{i_r}$  称为回转模长,  $r - \sum_{j=0}^{r-1} l_{i_j}$  称为余步,  $k$  和  $s_{i_r}$  都是优先走单一  $[+h]$  边得来的,  $r - \sum_{j=0}^{r-1} l_{i_j}$  是后走单一  $[+1]$  边得来的. 这就是优先走  $[+h]$  边寻径的思想.

**推论 2.** 在  $G(N; 1, h)$  中, 当  $N \geq h \times (h-1)$

时,  $0 \rightarrow h$  内非平常节点肯定不存在, 则 0 到任意节点  $v (0 < v < N)$  的最短路径长度  $l = k + r$ , 其中  $v = kh + r (0 \leq r < h)$ . 双环网络的直径  $d = \max\{p-1 + h-1, p+q\}$ , 其中  $N = ph + q$ .

**推论 3.** 在  $G(N; 1, h)$  中, 当  $d > p + h - 2$  时, 则在  $0 \rightarrow h$  内至少存在一个“非平常节点”.

证明. 假设  $0 \rightarrow h$  内不存在“非平常节点”, 则  $d = p + h - 2$ , 则与  $d > p + h - 2$  与矛盾. 所以当  $d > p + h - 2$  时, 则在  $0 \rightarrow h$  内至少存在一个“非平常节点”.

$[+h]$  边优先寻径可以构造一个直观模型——“竹筏”  $FG(1, h)$ , 将  $p (= \lceil N/h \rceil)$  个长度为  $h$  的等长区间和一个长度为  $q$  的区间称为  $p+1$  根竹子, “非平常节点”是其中的“竹节”, 平常节点分布在竹节之间.

“竹筏”构造如下:

(1) 将  $p+1$  根竹子  $F_i (0 \leq i \leq p)$  向右依次摆好;

(2) 将竹子  $F_i (0 \leq i \leq p)$  之间对应节点  $v_{0j}, v_{1j}, \dots, v_{pj} (0 \leq j \leq h-1)$  用一根绳索连起来, 最后一根竹子的  $0 \leq j < q$ ;

当  $q=0$  时, 称为矩形“竹筏”; 当  $q \neq 0$  时, 称为  $L$  形“竹筏”. 显然“竹筏”是一种新的  $L$  形瓦, 与传统的  $L$  形瓦形式一样, 只是节点摆放的次序不一样.

由“竹筏”的定义, 可以得到一些性质:

(1)  $th, th+1, \dots, (t+1)h-1$  这  $h$  个节点必顺序摆在第  $t$  根竹子上, 其中  $0 \leq t \leq p-1$ , 并且  $ph, ph+1, \dots, ph+q-1$  顺序摆在第  $p$  根竹子上.

(2) 设第  $i (0 \leq i \leq p-1)$  根竹子上点  $j (0 \leq j \leq h-1)$  的  $[+h]$  边优先最短路径是  $x_{ij} [+1] + y_{ij} [+h]$ , 则第  $i+1$  根竹子上点  $j$  的  $[+h]$  边优先最短路径是  $x_{ij} [+1] + (y_{ij} + 1) [+h]$ .

(3) 设第  $i (0 \leq i \leq p)$  根竹子上点  $j$  的  $[+h]$  边优先最短路径是  $x_{ij} [+1] + y_{ij} [+h]$ , 则第  $i$  根竹子上点  $j+1$  的  $[+h]$  边优先最短路径是  $(x_{ij} + 1) [+1] + y_{ij} [+h]$ , 其中, 点  $j$  和  $j+1$  位于同一个“竹节”之内.

$[+h]$  边优先寻径的算法同文献[1].

## 4 $[+h]$ 边优先寻径策略的实现

用 Visual Basic 6.0 和 SQL Server 2000 编程来实现  $[+h]$  边优先寻径策略的算法, 以文献[1]的  $DL(39, 17)$  (注: 表示  $G(39; 1, 17)$ , 以下相同) 为例,

列出最短路径的一种空间解——“竹筏”，见图 1。

G(39,17)的“竹筏”解			
G(39, 17)的“竹筏”解			
	竹1	竹2	竹3
	16 4[+1]+3[+h]	33 4[+1]+4[+h]	
	15 3[+1]+3[+h]	32 3[+1]+4[+h]	
	14 2[+1]+3[+h]	31 2[+1]+4[+h]	
	13 1[+1]+3[+h]	30 1[+1]+4[+h]	
竹节点	12 3[+h]	29 4[+h]	
	11 4[+1]+5[+h]	28 4[+1]+6[+h]	
	10 3[+1]+5[+h]	27 3[+1]+6[+h]	
	9 2[+1]+5[+h]	26 2[+1]+6[+h]	
	8 1[+1]+5[+h]	25 1[+1]+6[+h]	
竹节点	7 5[+h]	24 6[+h]	
	6 6[+1]	23 6[+1]+1[+h]	
	5 5[+1]	22 5[+1]+1[+h]	
	4 4[+1]	21 4[+1]+1[+h]	38 4[+1]+2[+h]
	3 3[+1]	20 3[+1]+1[+h]	37 3[+1]+2[+h]
	2 2[+1]	19 2[+1]+1[+h]	36 2[+1]+2[+h]
	1 1[+1]	18 1[+1]+1[+h]	35 1[+1]+2[+h]
	0 0[+1]	17 0[+1]+1[+h]	34 0[+1]+2[+h]

图 1 DL(39,17)的空间解“竹筏”(一种新 L 形瓦)

5 与[+1][+h]双边寻径策略的比较

作者在文献[9]中提出了[+1][+h]双边寻径策略,得到一种空间解——“螺旋环”,下面将两者进行比较。

[+h]边优先寻径策略适合求 0 到某一个节点  $v$  的一条特殊最短路径—[+h]边优先最短路径和该路径长度;其思路是:固定节点,寻找路径;其策略是:优先走[+h]边;其关键是求出  $0 \rightarrow h$  内的“非平常节点”;其它节点的[+h]边优先最短路径和长度可以由递推关系式求出。

[+1][+h]双边寻径策略适合求 0 到所有节点的所有最短路径和它们的长度;其思路是:固定路径,寻找节点;其策略是:同时走[+1][+h]边;其关键是记住已访问节点集和刚刚访问节点集;所有节点的最短路径和长度可以直接得出。

[+h]边优先寻径策略的思想只适合单位步长的双环网络,不适合非单位步长的双环网络和双环以上的网络;它利用了[+1]边的特殊性。[+1][+h]双边寻径策略的思想不仅适合单位步长的双环网络,还适合非单位步长的双环网络和双环以上的网络。

[+h]边优先寻径策略得到一个直观模型——“竹筏”:由  $p$  根长度为  $h$  的竹子和一根长度为  $q$  的竹子组成,“竹节”代表“非平常节点”,平常节点分布在竹节之间。“竹筏”特点是:竹子长度、竹子上节点、节点数和节点位置固定;但 0 到节点的距离需要计算;竹子之间对应节点和“竹节”中的节点[+h]边优先最短路径存在递推关系。这个特点是由其“固定节点,寻找路径”的策略决定的。

[+1][+h]双边寻径策略得到一个直观模型——“螺旋环”:由  $d+1$  个环组成,“螺旋环”特点是:环个数、环直径大小、环上节点和个数需要求解;但 0 到环的距离固定;同环节点到 0 的距离相等,不同环

之间节点距离存在简单关系—等差为 1;所有最短路径都在“螺旋环”中出现;双环网络直径就是顶环数。这个特点是由其“固定路径,寻找节点”的策略决定的。

从算法复杂度来看:[+h]边优先寻径策略需要复杂比较、计算和排序,耗时间,但存储节点少,少占空间;[+1][+h]双边寻径策略存储节点和节点状态多,多占空间,但计算简单,少耗时间。

“竹筏”可以看成是静态的空间解,“螺旋环”体现动态的空间解。

从实现角度看,借助数据库 SQL Server 来解决存储,[+1][+h]双边寻径策略更容易实现。

6 与传统 L 形瓦的比较

从节点[+h]边优先最短路径值的角度来说,[+h]边优先寻径策略得到的直观模型是“竹筏”;但从节点摆放位置的角度来说,[+h]边优先寻径策略得到的直观模型是一种新的 L 形瓦,其参数值不等于相对应的传统 L 形瓦的参数值。传统 L 形瓦构造相对困难,需按规则构造,但计算双环网络的直径相对容易<sup>[10]</sup>。新的 L 形瓦构造相对容易,由  $h$  去分割  $N$ ,然后顺序摆放就行,但计算双环网络的直径相对困难(见图 1)。从节点值的角度来说,传统 L 形瓦是均匀的,其值是连续变化的;新的 L 形瓦是非均匀的,其值跳变的。但从节点位置的角度来说,传统 L 形瓦是非均匀的,其顺序是跳变的;新的 L 形瓦是均匀的,其顺序是连续变化的。

简单来说,新的 L 形瓦易构造但难以求其直径;传统的 L 形瓦难以构造但易求其直径;两者都有其均匀和非均匀的一面。

7 文献[1]中的错误

(1)文献[1]中的[+1]边优先最短路径定义不严密,只能是[+1]边最大最短路径。另外,[+1]边优先最短路径没有明确意义。本文的[+h]边优先最短路径有明确意义:优先走[+h]边,当走[+h]边不利时,才走[+1]边。

定义 4. 将文献[1]中[+1]边优先最短路径定义中加上:所有[+1]边依次在前,所有[+h]边依次在后。则称以下节点为节点 0 所对应的“奇异点”:0 到它们的单一[+1]边最短路径=0 到它们的单一[+h]边最短路径=0 到它们的[+h]边优先最短

径 $=0$ 到它们的 $[+1]$ 边优先最短路径。

则 $[+1]$ 边优先最短路径和 $[+h]$ 边优先最短路径的关系是:当节点不是“奇异点”时,两者的 $[+1]$ 边和 $[+h]$ 边边数分别相等,路径形式对称;当节点是“奇异点”时,前者是单一 $[+1]$ 边最短路径,后者是单一 $[+h]$ 边最短路径。“奇异点”是“非平常节点”的特殊情况。

(2)文献[1]中的引理 2 结论和证明都是错误的,下面看其引理 2 的叙述:

设 $v = kh + r, 0 \leq r < h, x [+1] + y [+h]$ 是 $DL(N, h)$ 中从节点 0 到节点  $r$  的 $[+1]$ 边优先最短路径,则 $x \leq r, y \geq k$ 。

结论 $x \leq r$ 是正确的,证明过程也正确;但结论 $y \geq k$ 是错误的,证明过程中有一个不等式 $x + kh < x + (k-1)h$ 是无法理解的,除非 $h < 0$ ,而这个不等式是推出结论的关键步,所以结论存在问题。下面就看文献[1]的一个例子:

$DL(39, 17)$ ,取 $v = 38$ ,则 $k = 2, r = 4$ 。 $0 \rightarrow 4$ 的 $[+1]$ 边优先最短路径文献[1]的例 1 已经求出来: $4 [+1] + 0 [+17]$ ,即 $y = 0$ ,显然, $y < k$ 。因此结论 $y \geq k$ 是错误的。

(3)文献[1]中的引理 3 结论正确,证明过程是错误的。

原文中:“由引理 2 知 $y_1 - k \geq 0$ ”,这个 $x_1, y_1$ 是 0 到  $v$  的一条路径,而不是 0 到  $r$  的一条路径。

## 8 总 结

文献[1]提出了 $[+1]$ 边优先最短路径,本文从 $[+h]$ 边优先寻径策略出发,对应定义了 $[+h]$ 边优先最短路径,得出不少类似结论;定义了该策略的直观模型——“竹筏”,是一种新的节点空间解形式,不同于传统 $L$ 形瓦和文献[9]中的“螺旋环”。

## 参 考 文 献

[1] Chen Zhong-Xue, Jin Fan. On the  $[+1]$ -link prior shortest

path and optimal routing for double-loop networks. Journal of Computer Research & Development, 2001, 38(7): 788-792(in Chinese)

(陈忠学,靳蕃. 双环网络 $[+1]$ 边优先最短路径及其寻径策略. 计算机研究与发展, 2001, 38(7): 788-792)

[2] Feng Fei-Ling, Jin Lin-Gang. Characteristics analysis and routing control for a class of double-loop networks. Chinese Journal of Computers, 1994, 17(11): 859-865(in Chinese)

(冯斐玲,金林钢. 一类双环网的特征分析及寻径控制. 计算机学报, 1994, 17(11): 859-865)

[3] Raghavendra C S, Gerla M, Avizienis A. Reliable loop topologies for local computer networks. IEEE Transactions on Computers, 1985, C234(1): 46-54

[4] Guan D J. An optimal message routing algorithm for double-loop networks. Information Processing Letters, 1998, 65: 255-260

[5] Wong C K, Copperhsmith D. A combinatorial problem related to multimodule memory organizations. Journal of Association for Computing Machinery, 1974, 21: 392-402

[6] Liu Huan-Ping, Zhu Yan-Gong, Yang Yi-Xian. An algorithm for finding the shortest path in double-loop networks. Journal of Electronics, 1999, 21(2): 202-205(in Chinese)

(刘焕平,朱延功,杨义先. 双环网 $D(N, h)$ 的最短路径选择算法. 电子科学学刊, 1999, 21(2): 202-205)

[7] Xu Jun-Ming, Liu Qi. A type of infinite families of 4-optimal double loop networks. Science in China(Series A), 2003, 33(1): 71-74(in Chinese)

(徐俊明,刘琦. 一类 4 紧优双环网无限族. 中国科学(A 辑), 2003, 33(1): 71-74)

[8] Li Xiao-Ming, Fang Bin-Xing. A fault tolerant routing algorithm for optimal dual-ring networks. Chinese Journal of Computers, 1990, 13(7): 549-552(in Chinese)

(李晓明,方滨兴. 一种适于最佳双环网的容错路由算法. 计算机学报, 1990, 13(7): 549-552)

[9] Fang Mu-Yun, Zhao Bao-Hua. Method to calculate the diameter of undirected double-loop networks  $G(N; \pm 1, \pm s)$ . Journal on Communications, 2007, 28(2): 124-129(in Chinese)

(方木云,赵保华. 一种新的无向双环网络 $G(N; \pm 1, \pm s)$ 直径求解方法. 通信学报, 2007, 28(2): 124-129)

[10] Li Qiao, Xu Jun-Ming, Zhang Zhong-Liang. The infinite families of optimal double loop networks. Science in China(Series A), 1993, 23(9): 979-992(in Chinese)

(李乔,徐俊明,张忠良. 最优双环网络的无限族. 中国科学(A 辑), 1993, 23(9): 979-992)



**FANG Mu-Yun**, born in 1968, Ph.D. candidate, associate professor. His research interests include software testing, software reliability.

**QU Yu-Gui**, born in 1946, professor. Her research interests include software environment and communication protocol.

**ZHAO Bao-Hua**, born in 1947, professor, Ph.D. supervisor. His research interests include software environment and communication protocol.

Background

Double loop networks have been widely used in the topological structure of computer interconnection networks and communication systems. Many researches focus on the following two problems: (1) constructing optimal double-loop networks  $G(N;1,h)$  and  $G(N;r,h)$ , and (2) optimal routing algorithms for  $G(N;1,h)$  and  $G(N;r,h)$ .

In the past five years, the authors have simulated the  $L$ -shaped Tile for  $G(N;1,h)$  and  $G(N;r,h)$ . The authors have proposed  $[+1][+h]$  routing strategy for directed double loop networks and undirected double loop networks.

This paper discusses a  $[+h]$ -link prior routing strategy, that  $[+h]$ -link is chosen prior until it is not favourable, then  $[+1]$ -link is chosen. The shortest  $[+h]$ -link prior path and the bamboo raft (a new kind of  $L$ -shaped Tile) for a double-loop network are obtained. There is inferring relationships among nodes in the bamboo raft. The diameter of a double-

loop network can be derived from the shortest  $[+h]$ -link prior path. VB6.0 and SQL Server2000 are selected to simulate the  $[+h]$ -link prior routing strategy.

Chen Zhong-Xue discussed similar work like  $[+1]$ -link prior routing strategy. Compared to  $[+1]$ -link prior routing strategy,  $[+h]$ -link prior routing strategy is well understood and induces a new spatial model called bamboo raft (a new kind of  $L$ -shaped Tile).

The  $[+1][+h]$  routing strategy proposed by the authors is fixing path then seeking nodes, while the  $[+h]$ -link prior routing strategy is fixing nodes then seeking path. The traditional  $L$ -shaped Tile is difficult to be constructed but is easy to calculate the diameter of its equivalent double-loop network, while the new  $L$ -shaped Tile is easy to be constructed but is difficult to calculate the diameter of its equivalent double-loop network.