

星图上基于循环置换的任意两点之间的最短路径算法

王 静, 郭大昌

(广东工业大学 应用数学学院, 广东 广州 510006)

摘要: 针对路由选择对网络性能起重要作用, 提出了星图上任意两点之间的最短路径算法. 运用群论的循环置换的性质证明了两点之间的距离公式, 给出了两点之间所有最短路径个数的一般代数表达式.

关键词: 星图; 最短路径; 循环置换; 距离

中图分类号: O152.1

文献标识码: A

文章编号: 1007-7162(2010)03-0035-06

在众多的网络拓扑结构中, 星图是最典型的一类网络拓扑结构. 它的拓扑结构特点和超立方体在一定程度上是很相似的, 比如: 正规性、对称性、可嵌入性等. 除此之外, 它还有其自身的优点: 直径短、结点的度和结点存储空间小等, 为此深受研究者的欢迎. 关于星图和超立方体二者拓扑结构特点的比较, 具体见参考文献[1].

由于路由选择对星图网络拓扑结构的性能起着重要的作用, 一些学者在这方面做了大量的工作. 文献[2]给出了两条简单的路由规则, 反复运用这两条规则, 任意两个结点就可以找到它们之间的最短路径. Chen Chi-Chang、Chen Jianer 给出了两点之间的并行路径算法^[3]. Yeh Sheng-I 等人已研究出了一种无错误路由规则^[4], 将从源结点到目的结点的置换写成如下形式的循环置换的乘积: $(X \cdots XA)_{k_1} (X \cdots X)_{k_2} \cdots (X \cdots X)_{k_c}$, 其中 $k_1 \geq 0, k_2 \geq k_3 \geq \cdots \geq k_c, A$ 表示目的结点的第 1 维, X 表示目的结点的其它任一维, 并将该形式称为一个模式, 然后找出该模式所有的优先模式, 并找出每个优先模式对应的优先邻结点, 再把每个优先邻结点作为新的源结点, 如此这样做下去, 把从源结点到目的结点所经过的所有中间结点找到, 那么它们之间的最短路径也就找到了.

由于星图的高度对称性, 任意两点之间的路由路径都可以映射为一个特定的结点和特殊结点 $I = 123 \cdots n$ 之间的路由路径. 本文运用群论^[5]的循环置换^[5]的性质将源结点到目的结点之间的置换化为一系列的对换^[5]的乘积. 只要从源结点出发从左到右依次乘以这些对换就可以到达目的点, 它们之间的

路径即为最短路径. 相比无错误路由规则, 本文给出的方法比较直接, 而且省去了找优先邻结点所需要的大量工作. 本文同时也证明了两结点之间的距离公式.

当两点之间有大量的数据需要同时传送时, 为了使它们安全、快速、有效地传送到目的结点, 寻找两点之间的平行路径也尤为重要. 鉴于此, 本文给出了任意两点之间所有最短路径个数的一般代数表达式. 它对寻找并行路径有很大帮助, 只需从所有最短路径中提取出内点不交的路径即为并行路径.

1 预备知识

定义 1 星图(维星图)满足:

(1) 维星图的顶点集对应集合上所有的置换, 即

$$V = \{u_1 u_2 \cdots u_n \mid u_i \in \{1, 2, \cdots, n\} (i = 1, 2, \cdots, n), \text{且当 } i \neq j \text{ 时}, u_i \neq u_j\};$$

(2) 维星图的边集

$$E = \{(u, v) \mid u, v \in V, \text{且存在 } i > 1, u_1 = v_i, u_i = v_1, \text{对任何 } j \neq i, u_j = v_j\}.$$

实际上, 星图 S_n 顶点集里的每个元素都是从自然排列 $123 \cdots n$ 到 $\{1, 2, \cdots, n\}$ 上其它排列的一个映射, 这里 $123 \cdots n$ 是原像, 但通常都是省掉原像 $123 \cdots n$, 只写其映射的排列, 也即像. 例如, 在 S_3 中, $\{1, 2, 3\}$ 上所有映射构成的集合为

$\left\{ \begin{pmatrix} 123 \\ 123 \end{pmatrix}, \begin{pmatrix} 123 \\ 231 \end{pmatrix}, \begin{pmatrix} 123 \\ 312 \end{pmatrix}, \begin{pmatrix} 123 \\ 132 \end{pmatrix}, \begin{pmatrix} 123 \\ 321 \end{pmatrix}, \begin{pmatrix} 123 \\ 213 \end{pmatrix} \right\}$, 但顶点集通常都只写成其像的集合: $\{123, 231, 312, 132, 321, 213\}$. 另外, n 维星图 S_n 的顶点集构成了一个

群^[5],该群为 n 次对称群,其阶为 $n!$,故 n 维星图 S_n 的顶点个数为 $n!$.

在 S_n 中,两个顶点 u, v 相邻,当且仅当这两个顶点所对应的置换仅在第1维和另外一维上的数字不同,即其中一个置换右乘以 $T = \{(1, 2), (1, 3), \dots, (1, n)\}$ 中某一个对换后就可得到另一个置换. 为方便起见,用 $\sigma(i) (i = 2, 3, \dots, n)$ 来记 $(1, i)$,它表示将结点第1维上的数字和第 n 维的数字进行交换. 这里规定,在 n 维星图上,结点上最左端的维是第1维,最右端的维为第 n 维,也就是满足低维在左,高维在右. 例如在 S_3 中, $123[\sigma(2)] = 213$,即结点123通过交换第1维和第2维得到213,从而结点123和点213相邻. 因此,星图的每个点通过作用 $\sigma(i)$ 连接到另外的顶点上. 故星图 S_n 是群 V 和生成集 T 所对应的Cayley图^[6].

一个顶点的度数是指这个顶点所关联的边数. 在星图 S_n 中,两个点有边相连,当且仅当这两个顶点标号中的 n 维数字里只有第1维的数字和另外一维上的数字不同,剩余的 $n-2$ 维上的数字都相同,有 $n-1$ 种情况,因此可知任何一个顶点都关联着 $n-1$ 条边,即 n 维星图 S_n 每个顶点的度数都是 $n-1$.

图1和图2给出了二阶星图 S_2 和三阶星图 S_3 .



图1 二阶星图 S_2

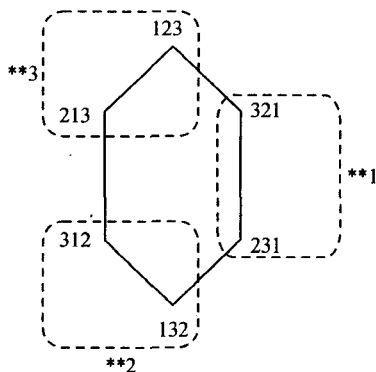


图2 三阶星图 S_3

定义2 路由^[4]:路由是指在互联网络中将消息由源结点发送到目的结点所经过的一系列中间结点.

定义3 距离^[4]:在星图 S_n 中两结点 u 和 v 的距离是两结点间的最短路由路径长度.

定义4 置换:集合 A 上的置换是指从 A 到 A 上的一个双射.

例如从 A 到 A 上的一个双射为: $1 \rightarrow 4, 2 \rightarrow 2, 3 \rightarrow 5, 4 \rightarrow 3, 5 \rightarrow 1$,那么可以将这个置换写为

$$\psi = \begin{pmatrix} 12345 \\ 42531 \end{pmatrix}.$$

定义5 循环:集合 A 上的置换称为长为 n 的循环,如果存在 $a_1, a_2, \dots, a_n \in A$,使得 $a_1 \Psi = a_2, a_2 \Psi = a_3, \dots, a_{n-1} \Psi = a_n, a_n \Psi = a_1$,并且对于 $x \in A$,当 $x \Psi = x$ 时, $x \notin \{a_1, a_2, \dots, a_n\}$. 此时,记 $\Psi = (a_1, a_2, \dots, a_n)$.

定义6 不相连^[5]:两个循环如果它们中没有公共的元素称为不相连.

定义7 对换:长为2的循环称为一个对换.

定理1 ^[5]有限集合上的每一个置换都可以写成一些不相连的循环的乘积.

定理2 ^[5]至少有两个元素的有限集合上的每一个置换都可以写成一些对换的乘积.

推论1 ^[5]对于任意的循环置换 $(a_1, a_2, \dots, a_n), a_i \in \{1, 2, \dots, n\} (i = 1, 2, \dots, n)$,当 $i \neq j$ 时, $a_i \neq a_j$,有 $(a_1, a_2, \dots, a_n) = (a_1, a_2)(a_1, a_3) \cdots (a_1, a_n)$.

2 星图中任意两点间的距离

在互联网中路由设计应该使路由路径尽可能地短,因为它对应的是通信耗费的时间. 在寻找任意两结点间的路径中,人们习惯上将路由用从源结点到目的结点的置换来表示,而且这个置换是用一些不相连的循环置换的乘积来表示的,并且每个循环置换的长度是大于等于2的.

关于星图中任意两个结点间的距离,在文献[2]中已经给出了人们认可的计算公式. 本文运用群论的循环置换的性质,给出了其证明过程.

性质1 恒等置换可以表示为偶数个任意对换的乘积. 例如:

$I = (a, b)(a, b), \forall a, b \in \{1, 2, \dots, n\}$,这里用符号 I 表示任意一个恒等置换.

定理3 对于 $\forall a, b \in \{1, 2, \dots, n\}, (a, b) = (1, a)(1, b)(1, a)$.

证明 给对换 (a, b) 两边同时乘以恒等置换 $(1, a)(1, a)$ 是不影响结果的:

$(a, b) = [(1, a)(1, a)](a, b)[(1, a)(1, a)]$
 $= (1, a)[(1, a)(a, b)(1, a)] = (1, a)(1, b)(1, a)$. 证毕.

同理也可以两边同时乘以 $(1, b)(1, b)$ 得到:
 $(a, b) = (1, b)(1, a)(1, b)$, 以上两种表示形式是等价的.

推论1、性质1及定理3在下面的证明中反复用到.

定理4 假设从源结点 u 到目的结点 v 的置换为 Ψ , A 表示目的结点第1维上的数字, X 表示目的结点其它任何一维上的数字, c 为长度至少为2的循环置换的个数 ($c \geq 1$), m 表示所有循环置换中数字个数的和, 即源结点与目的结点错位数字的个数, 显然 $m \leq n$. 源结点 u 到目的结点 v 间的距离用 $d(\Psi)$ 表示, 则

$$d(\Psi) = c + m - \begin{cases} 0, & \text{如果源结点第1维的数字也为 } A; \\ 2, & \text{否则.} \end{cases}$$

证明 由于星图的高度对称性, 寻找任意两结点之间的最短路径就可以映射为特定的结点到特殊结点 $I = 123 \cdots n$ 之间的路径. 下面分两种情况进行证明.

(1) 定理4证明之第一种情况. 源结点的第1维数字和目的结点的第1维数字相同, 也即同为1时:

假设源结点的置换为 $P = x_1, x_2, x_3 \cdots x_n, x_1 = 1, x_i \in \{2, 3, \cdots, n\} (i = 2, 3, \cdots, n)$, 且当 $i \neq j$ 时, $x_i \neq x_j$, 目的结点为恒等置换 $I = 123 \cdots n$. $C_1, C_2, \cdots, C_c$ 为 c 个不相连的循环置换, 用 $|C_i|$ 表示循环置换 $C_i (i = 2, \cdots, c)$ 的长度, 且 $|C_i| \geq 2$. m 为 c 个循环置换中数字个数的总和, 且满足 $m < n$. 再设 $2 < k_1 < k_2 < \cdots < k_c \leq n$, 不失一般性, 这里假设 $C_1 = (2, 3, \cdots, k_1)$, $C_2 = (k_1 + 1, k_1 + 2, \cdots, k_2)$, $\cdots$, $C_c = (k_{c-1} + 1, k_{c-1} + 2, \cdots, k_c)$, 并且 $k_c = m + 1$, 才能保证这些循环长度的总和为 m . 可得 $C_1 = (2, 3, \cdots, k_1) = (2, 3)(2, 4) \cdots (2, k_1) = (1, 2)(1, 3) \underbrace{(1, 2)(1, 2)}_{(1, 2)} (1, 4) \underbrace{(1, 2)(1, 2)}_{(1, 2)} (1, 5) (1, 2) \cdots (1, 2) (1, k_1 - 1) \underbrace{(1, 2)(1, 2)}_{(1, 2)} (1, k_1) (1, 2) = (1, 2)(1, 3)(1, 4)(1, 5) \cdots (1, k_1 - 1)(1, k_1)(1, 2)$.

同理可以得到

$$C_2 = (1, k_1 + 1)(1, k_1 + 2)(1, k_1 + 3) \cdots (1, k_2)(1, k_1 + 1), \cdots,$$

$$C_c = (1, k_{c-1} + 1)(1, k_{c-1} + 2)(1, k_{c-1} + 3) \cdots (1, k_c - 1)(1, k_c)(1, k_{c-1} + 1).$$

由上面可知循环 C_i 是由 k_i 个连续对换来完成的, 这些对换分别通过交换第1维和第2, 3, $\cdots$, k_i 维及第2维上的数字得到. 对换的个数是循环 C_i 的长度加1, 即 $k_i = |C_i| + 1$, C_2 由 $k_2 - k_1 + 1 = |C_2| + 1$ 个对换来完成, 分别通过交换第1维和第 $k_1 + 1, k_1 + 2, \cdots, k_2$ 维及 $k_1 + 1$ 维上的数字得到, $\cdots$, C_c 是由 $k_c - k_{c-1} + 1$ 个对换完成的. 因此, 从源结点 p 到目的结点 I 的置换可以表示为

$$\Psi = \begin{pmatrix} I \\ p \end{pmatrix} = \begin{pmatrix} 123 \cdots n \\ x_1 x_2 x_3 \cdots x_n \end{pmatrix} = C_1 C_2 \cdots C_c = (2, 3, \cdots,$$

$$k_1)(k_1 + 1, k_1 + 2, \cdots, k_2) \cdots (k_{c-1} + 1, k_{c-1} + 2, \cdots, k_c) = [(1, 2)(1, 3)(1, 4)(1, 5) \cdots (1, k_1 - 1)(1, k_1)(1, 2)][(1, k_1 + 1)(1, k_1 + 2) \cdots (1, k_2)(1, k_1 + 1)] \cdots [(1, k_{c-1} + 1)(1, k_{c-1} + 2)(1, k_{c-1} + 3) \cdots (1, k_c - 1)(1, k_c)(1, k_{c-1} + 1)] = [\sigma(2)\sigma(3)\sigma(4) \cdots \sigma(k_1 - 1)\sigma(k_1)\sigma(2)][\sigma(k_1 + 1)\sigma(k_1 + 2) \cdots \sigma(k_2)\sigma(k_1 + 1)] \cdots [\sigma(k_{c-1} + 1)\sigma(k_{c-1} + 2)\sigma(k_{c-1} + 3) \cdots \sigma(k_c - 1)\sigma(k_c)\sigma(k_{c-1} + 1)],$$

故从 p 到 I 的置换可以用一系列的对换 $\sigma(i) (i = 2, 3, \cdots, n)$ 的乘积来表示, 也即从源结点出发, 从左到右依次乘以这些对换, 就可以到达目的点. 并且这些对换的个数为所有循环置换的长度之和加 c , 即

$$d(\psi) = (|C_1| + 1) + (|C_2| + 1) + \cdots + (|C_c| + 1) = k_1 + (k_2 - k_1 + 1) + (k_3 - k_2 + 1) + \cdots + (k_c - k_{c-1} + 1) = k_c + c - 1 = (m + 1) + c - 1 = c + m. \text{ 证毕.}$$

其实上述的证明过程同时也给出了寻找两个结点之间最短路径的方法, 就是首先要将源结点到目的结点的置换化为一组对换 $\sigma(i) (i = 2, 3, \cdots, n)$ 的乘积, 然后从源结点出发从左到右依次地乘以这些对换, 最终到达目的结点.

由上面的证明过程可以得到: $\forall a_1, a_2, \cdots, a_m \in \{2, 3, \cdots, n\} (m \leq n)$, 当 $i \neq j$ 时, $a_i \neq a_j$, 循环置换 $(a_1, a_2, a_3, \cdots, a_m) = (1, a_1)(1, a_2)(1, a_3) \cdots (1, a_{m-1})(1, a_m)(1, a_1)$.

性质2 任意一个长度为 n 的循环置换有下列 n 种等价表示方法:

$$(a_1, a_2, \cdots, a_n) = (a_2, a_3, \cdots, a_n, a_1) = (a_3, a_4, a_5, \cdots, a_1, a_2) = \cdots = (a_n, a_1, a_2, \cdots, a_{n-2}, a_{n-1}), \forall a_1, a_2, \cdots, a_n \in \{1, 2, \cdots, n\}.$$

定理5^[5] 如果 $\lambda = (i_1, i_2, \cdots, i_k)$ 和 $\tau = (j_1, j_2, \cdots, j_l)$ 为集合 A 上两个不相连的循环置换, 那么 $\lambda\tau = \tau\lambda$.

定理6 当源结点和目的结点第1维上的数字

相同时,它们之间有 $N = c! \prod_{i=1}^c |C_i|$ 条最短路径. 这里 c 和 C_i 同上所示.

证明 该证明过程的前提假设以及符号表示均同上述证明. 由性质2知下列 $m(m \leq n)$ 种表示形式是等价的.

$$(a_1, a_2, a_3, \dots, a_m) = (1, a_1)(1, a_2)(1, a_3) \cdots (1, a_{m-1})(1, a_m)(1, a_1),$$

$$(a_2, a_3, \dots, a_m, a_1) = (1, a_2)(1, a_3)(1, a_4) \cdots (1, a_m)(1, a_1)(1, a_2),$$

$$(a_3, a_4, a_5, \dots, a_1, a_2) = (1, a_3)(1, a_4)(1, a_5) \cdots (1, a_1)(1, a_2)(1, a_3),$$

⋮

$$(a_m, a_1, a_2, \dots, a_{m-2}, a_{m-1}) = (1, a_m)(1, a_1)(1, a_2) \cdots (1, a_{m-2})(1, a_{m-1})(1, a_m).$$

因而,用对换的乘积表示从源结点到目的结点之间置换的形式不是唯一的. c 个循环分别可以表示为 $|C_1|, |C_2|, \dots, |C_c|$ 种不同序列对换的乘积, C_i ($i=1, 2, \dots, c$) 采用其中任何一种形式时,都对应着一条最短路径,故 $C_1 C_2 \cdots C_c$ 对应的所有最短路径总

共有 $|C_1| \times |C_2| \times \cdots \times |C_c| = \prod_{i=1}^c |C_i|$ 条.

再由定理5知:任意个循环的乘积构成的置换,各循环之间的排列顺序是可以互换的,由 c 个循环的乘积构成的置换 $C_1 C_2 \cdots C_c$ 有 $c!$ 种等价的排列形式,而每一种形式对应的最短路径都有 $\prod_{i=1}^c |C_i|$ 条,故所有的最短路径总共有 N 条, $N = c! \prod_{i=1}^c |C_i|$. 证毕.

(2) 定理4证明之第二种情况. 当源结点和目的结点第1维上的数字不同时:

这里同样假设源结点的置换为 $P = x_1 x_2 x_3 \cdots x_n$, $x_i \in \{1, 2, \dots, n\}$ ($i=1, 2, 3, \dots, n$), 且当 $i \neq j$ 时, $x_i \neq x_j$, 其中 $x_i \neq 1$. 目的结点为恒等置换 $I = 123 \cdots n$, $C_1, C_2, \dots, C_c$, 为 c 个不相交的循环置换, $|C_i|$ 表示循环置换 C_i ($i=1, 2, \dots, c$) 的长度, 且 $|C_i| \geq 2$. m 为 c 个循环置换中数字个数的总和, $m \leq n$. 再设 $2 < k_1 < k_2 < \cdots < k_c$, 不失一般性, 假设 $k_c < x_1 \leq n$, $C_1 = (1, x_1, 2, 3, \dots, k_1)$, $C_2 = (k_1 + 1, k_1 + 2, \dots, k_2)$, $\dots$, $C_c = (k_{c-1} + 1, k_{c-1} + 2, \dots, k_c)$. 这里 $k_c = m - 1$, 才能保证这些循环长度的总和为 m . 从源结点到目的结点的置换 Ψ 为

$$\Psi = \begin{pmatrix} I \\ p \end{pmatrix} = \begin{pmatrix} 123 \cdots n \\ x_1 x_2 x_3 \cdots x_n \end{pmatrix} = C_1 C_2 C_3 \cdots C_c = (1, x_1, 2, 3, \dots, k_1)(k_1 + 1, k_1 + 2, \dots, k_2) \cdots (k_{c-1} + 1, k_{c-1} + 2, \dots, k_c) = [(1, x_1)(1, 2)(1, 3) \cdots (1, k_1)][(1, k_1 + 1)(1, k_1 + 2)(1, k_1 + 3) \cdots (1, k_2 - 1)(1, k_2)(1, k_1 + 1)] \cdots [(1, k_{c-1} + 1)(1, k_{c-1} + 2)(1, k_{c-1} + 3) \cdots (1, k_c - 1)(1, k_c)(1, k_{c-1} + 1)] = [\sigma(x_1)\sigma(2)\sigma(3) \cdots \sigma(k_1)][\sigma(k_1 + 1)\sigma(k_1 + 2) \cdots \sigma(k_2 - 1)\sigma(k_2)\sigma(k_1 + 1)] \cdots [\sigma(k_{c-1} + 1)\sigma(k_{c-1} + 2)\sigma(k_{c-1} + 3) \cdots \sigma(k_c - 1)\sigma(k_c)]\sigma(k_{c-1} + 1),$$

故只要从源结点出发从左到右依次乘以上面的对换就可以到达目的结点, 也即它们之间的距离为所有对换的个数, 因而

$$d(\Psi) = (|C_1| - 1) + (|C_2| + 1) + \cdots + (|C_c| + 1) = k_1 + (k_2 - k_1 + 1) + (k_3 - k_2 + 1) + \cdots + (k_c - k_{c-1} + 1) = k_c + (c - 1) = c = m - 2. \text{ 证毕.}$$

定理7 当源结点和目的结点第1维不相同时, 它们之间有 $N = c! \prod_{i=2}^c |C_i|$ 条最短路径.

证明 不同于第一种情况的是循环 C_1 只有一种表示形式, 故 $N = c! |C_2| |C_3| \cdots |C_c| = c! \prod_{i=2}^c |C_i|$. 证毕.

下面以例子说明在星图中找任意两点间的最短路径时, 本文的方法比无错路由规则更方便、更直接.

例1 设源结点为 $u = 13254$, 目的结为 $v = 12345$, 则从源结点到目的结点的置换为

$$\Psi = \begin{pmatrix} 12345 \\ 13245 \end{pmatrix} = (23)(45) = (12)(13)(12)(14)(15)(14), c=2, m=4, \text{ 距离为}$$

$d(\Psi) = c + m = 2 + 4 = 6$. 最短路径总共有 $N = c! |C_1| |C_2| = 2! \times 2 \times 2 = 8$ 条, 长度都为6. 它们分别是

$$13254 \xrightarrow{\sigma(2)} 31254 \xrightarrow{\sigma(3)} 21354 \xrightarrow{\sigma(2)} 12354 \xrightarrow{\sigma(4)} 52314 \xrightarrow{\sigma(5)} 42315 \xrightarrow{\sigma(4)} 12345,$$

$$13254 \xrightarrow{\sigma(3)} 23154 \xrightarrow{\sigma(2)} 32154 \xrightarrow{\sigma(3)} 12354 \xrightarrow{\sigma(4)} 52314 \xrightarrow{\sigma(5)} 42315 \xrightarrow{\sigma(4)} 12345,$$

$$13254 \xrightarrow{\sigma(2)} 31254 \xrightarrow{\sigma(3)} 21354 \xrightarrow{\sigma(2)} 12354 \xrightarrow{\sigma(5)} 42351 \xrightarrow{\sigma(4)} 52341 \xrightarrow{\sigma(5)} 12345,$$

$$13254 \xrightarrow{\sigma(3)} 23154 \xrightarrow{\sigma(2)} 32154 \xrightarrow{\sigma(3)} 12354 \xrightarrow{\sigma(5)} 42351$$

$\xrightarrow{\sigma(4)} 52341 \xrightarrow{\sigma(5)} 12345,$
 $13254 \xrightarrow{\sigma(4)} 53214 \xrightarrow{\sigma(5)} 43215 \xrightarrow{\sigma(4)} 13245 \xrightarrow{\sigma(2)} 31245$
 $\xrightarrow{\sigma(3)} 21345 \xrightarrow{\sigma(2)} 12345,$
 $13254 \xrightarrow{\sigma(4)} 53214 \xrightarrow{\sigma(5)} 43215 \xrightarrow{\sigma(4)} 13245 \xrightarrow{\sigma(3)} 23145$
 $\xrightarrow{\sigma(2)} 32145 \xrightarrow{\sigma(3)} 12345,$
 $13254 \xrightarrow{\sigma(5)} 42351 \xrightarrow{\sigma(4)} 53241 \xrightarrow{\sigma(5)} 13245 \xrightarrow{\sigma(3)} 23145$
 $\xrightarrow{\sigma(2)} 32145 \xrightarrow{\sigma(3)} 12345,$
 $13254 \xrightarrow{\sigma(5)} 43251 \xrightarrow{\sigma(4)} 53241 \xrightarrow{\sigma(5)} 13245 \xrightarrow{\sigma(2)} 31245$
 $\xrightarrow{\sigma(3)} 21345 \xrightarrow{\sigma(2)} 12345.$

若用无错误路由规则,首先要找出源结点的所有优先邻结点,这里用 $u^{(i)}$ 来表示结点 u 的第 i 个邻结点,它是通过交换 u 的第 1 维和第 i 维得到的.

$$u^{(2)} = 31254, \left(\begin{smallmatrix} 12345 \\ 31254 \end{smallmatrix} \right) (132) (45), d((132) (45)) = 5.$$

$$u^{(3)} = 23154, \left(\begin{smallmatrix} 12345 \\ 23154 \end{smallmatrix} \right) (123) (45), d((123) (45)) = 5.$$

$$u^{(4)} = 53214, \left(\begin{smallmatrix} 12345 \\ 53214 \end{smallmatrix} \right) (154) (23), d((154) (23)) = 5.$$

$$u^{(5)} = 43251, \left(\begin{smallmatrix} 12345 \\ 43251 \end{smallmatrix} \right) (145) (23), d((145) (23)) = 5.$$

可知 u 的 4 个邻结点距离目的结点都比源结点更近一步,因此都是其优先邻结点,这里只以 $u^{(2)}$ 为例,再找 $u^{(2)}$ 的优先邻结点,寻找方法如上,可以得到 $u^{(2)}$ 的 3 个邻结点都是其优先邻结点,然后以 $(u^{(2)})^{(3)}$ 为新的源结点,再找其优先邻结点,如此这样找下去,直到找到距离目的结点距离为 1 时,它们之间的最短路径就找到了.例如: $13254 \rightarrow 31254 \rightarrow 21354 \rightarrow 12354 \rightarrow 52314 \rightarrow 42315 \rightarrow 12345$.

例 2 设源结点为 $u = 125436$, 目的结点为 $v = 321546$. 首先通过函数 ϕ 将目的结点映射到特殊结点 $I = 123456$ 上,即 $\phi(3) = 1, \phi(2) = 2, \phi(1) = 3, \phi(5) = 4, \phi(4) = 5, \phi(6) = 6$, 则源结点也做相应的变换后就映射到 $u' = 324516$ 上. 因为 $\left(\begin{smallmatrix} 123456 \\ 324516 \end{smallmatrix} \right) = (1345) = (13)(14)(15)$, 因而可以找到 u, v 之间的

一条最短路径为

$$125436 \xrightarrow{\sigma(3)} 521436 \xrightarrow{\sigma(4)} 421536 \xrightarrow{\sigma(5)} 321546.$$

由上例可以看出,本文给出的寻找任意两点之间的最短路径方法直接、方便,不需要做太多的工作.

3 星图中任意两点间的最短路径算法

下面给出星图中任意两点间最短路径寻找的具体过程:

(1) 如果目的结点 $v = v_1 v_2 v_3 \cdots v_n$ 不是恒等置换 $I = 123 \cdots n$, 则做映射 ϕ , 使得 $\phi(v_1) = 1, \phi(v_2) = 2, \phi(v_3) = 3, \cdots, \phi(v_n) = n$, 源结点 $u = u_1 u_2 u_3 \cdots u_n$ 也做相应的映射后得到 $u' = \phi(u_1) \phi(u_2) \phi(u_3) \cdots \phi(u_n)$.

(2) 将 u' 到 I 的置换化为 $\psi = \left(\begin{smallmatrix} I \\ u' \end{smallmatrix} \right) = \left(\begin{smallmatrix} 1 & 2 & 3 & \cdots & n \\ \phi(u_1) \phi(u_2) \phi(u_3) \cdots \phi(u_n) \end{smallmatrix} \right) = C_1 C_2 \cdots C_c$, 其中 C_i ($i = 1, 2, \cdots, c$) 是长度至少为 2 的循环置换.

(3) 将 C_i ($i = 1, 2, \cdots, c$) 用一系列对换 $\sigma(j) j \in \{2, 3, \cdots, n\}$ 的乘积来表示, 即 $C_i = \prod_{j \in \{2, 3, \cdots, n\}} \sigma(j)$.

(4) 从源结点 $u = u_1 u_2 u_3 \cdots u_n$ 开始, 从左到右依次地乘以这些对换得到的所有结点即为从源结点到目的结点最短路径所经过的中间结点, 那么也就找到了它们之间的最短路径.

满足上述过程的算法如下:

Algorithm Shortest_Path()

{Input: sour, dest; // sour 表示源结点, dest 表示目的结点

I, Newsour; // I 表示恒等置换, Newsour 表示一个新的结点

$\psi = \left(\begin{smallmatrix} dest \\ sour \end{smallmatrix} \right);$ // ψ 表示从源结点到目的结点的置换

int $d(\psi)$; // $d(\psi)$ 表示源结点和目的结点间的距离

if ($d(\psi) = 0$) return; // 源结点即为目的点

while ($d(\psi) \neq 0$)

for (int $j = 1; j < n + 1; j++$)

{ if ($((dest, j) \neq (I, j))$) // ($dest, j$)

表示目的结点的第 j 维, (I, j) 表示恒等置换的第 j

维,并判断目的结点是否为恒等置换

$\{(I,j) = \phi((dest,j)); (Newsour,j) = \phi((sour,j));$

$\}\}$ // ϕ 为一映射函数,将目的结点

dest 映射到恒等置换 I 上,源结点 sour 也做相应的变换,映射到新的结点 Newsour 上

$int\ c < \left\lfloor \frac{n}{2} \right\rfloor;$

for (int $k = 1; k < c + 1; k++$)

$\{\omega = \begin{pmatrix} I \\ Newsour \end{pmatrix} = C_1 C_2 \cdots C_c;$

$C_k = func(I, Newsour); \}$ // C_k 是长

度大于等于2的循环,循环里是恒等置换 I 和新结点 Newsour 中的错位数字

for (int $p = 1; p < c + 1; p++$)

$\{C_p = \prod_{l \in \{2,3,\dots,n\}} \sigma(l); \}$ // 将循环 C_p 用

一系列对换的乘积表示

$Path = sour(\prod_{k=1}^c (\prod_{l \in \{2,3,\dots,n\}} \sigma(l)));$

$\}$ // 从源结点 sour 出发依次乘以这些

对换就可以到达目的结点,它们之间的路径 Path 也即为最短路径。

4 结论

本文从群论的角度出发,运用循环置换的性质

将源结点到目的结点之间的置换化为一系列的对换的乘积,只要从源结点出发从左到右依次乘以这些对换就可以到达目的结点,它们之间的路由路径也即为最短路由路径,并给出了最短路由路径算法。相比无错路由规则,本文给出的方法更直接、更方便。同时也证明了两结点之间的距离公式,给出了任意两点之间所有最短路由路径个数的一般代数表达式。

参考文献:

- [1] Day Khaled, Tripathi Anand. A Comparative Study of Topological Properties of Hypercubes and Star Graphs[J]. IEEE Transactions on Parallel and Distributed Systems, 1994, 5(1):31-38.
- [2] Akers S B, Harel D, Krishnamurthy B. The star graph: An attractive alternative to the n-cube[C] // Proc Int Conf. New York: Parallel Processing, 1987:393-400.
- [3] Chen Chi-Chang, Chen Jianer. Vertex-Disjoint Routings in Star Graphs[J]. IEEE Algorithms and Architectures for Parallel Processing, 1995, 1:460-464.
- [4] Yeh Sheng-I, Yang Chang-Biau, Chen Hon-Chan. Fault-Tolerant Routing on the Star Graph with Safety Vectors[J]. IEEE Parallel Architectures, Algorithms and Networks, 2002(5):266-271.
- [5] 唐高华. 近世代数[M]. 北京:清华大学出版社,2008.
- [6] 徐俊明. 组合网络[M]. 北京:科学出版社,2007.

The Shortest Path Algorithm Between Two Arbitrary Nodes Based on Cycle Permutation in the Star Graph

Wang Jing, Guo Da-chang

(Faculty of Applied Mathematics, Guangdong University of Technology, Guangzhou 510006, China)

Abstract: For the routing plays an important role in the performance of the network, the shortest path algorithm between two arbitrary nodes in the star graph is proposed. The distance formula between two points is proved by using the nature of the cycle permutation of the Group Theory, and the general algebraic expression for the number of all the shortest paths between two nodes is provided.

Key words: the star graph; the shortest path; cycle permutation; the distance